```
/*	(Evtl. Namen: Turboid, Chromin, Hydronium, Chromanoid, Lenium, Gilenium, Jelenium, boosta, Menolium)



	Documentation:
	-----------------------------------------------------------------------------------------------------------
	Function name:				Purpose:
	-----------------------------------------------------------------------------------------------------------
```

Syntaxalternativen / -optimierung:

[Object].setByRef(value)	Verändert aus dem Inneren einer Funktion eine übergebene Variable referentiell => Variablenname vor Übergabe hat dann denselben neuen Wert >>Um auch Namen von Nicht-Objekte (z.B. Zahlenwerte & Strings) ändern zu können, vor Übergabe zu Objekt transformieren: nonObj = Object(nonObj); >> Setzen auf 'null' führt in FF zu "undefined" (IE o.k.) >> Setzen auf 0 ist hier nicht äquivalent zum Setzen auf 'false', sondern 'true' >> Typ bleibt Objekt, auch wenn String oder Number übergeben - Vergleichbarkeit mit echten Strings/Numbers in if-Abfragen etc. aber wie normal >> Statt 'if(x)...' sollte es heißen: 'if(isTrue(x))...' oder 'if(exists(x))...'
loop(fnc [, interval [, laps]])	Wie setInterval, jedoch reibungsloser für FF 2 und mit Option für Begrenzung >> Ist kein Intervall angegeben worden, wird automatisch 1 Millisek. genommen >> laps: Wie oft die Schleife maximal durchlaufen werden soll (optional) >> Kann mit loop.exit() angehalten werden
loop.exit()	Bricht loop()-Schleife ab >> Wird im Inneren der Funktion in loop(funktion) eingesetzt
loop.then(fnc)	Für Code, der erst nach Beendigung/Abbruch der loop()-Schleife ausgeführt werden soll, statt gleichzeitig >> Wird direkt nach loop() eingesetzt
loop.turbo(fnc[, acc [,laps]])	Wie loop(), jedoch schneller >> acc: Beschleunigungsfaktor (niedrigste Beschleunigung: 1)
loop.timer.add(fnc [, fncLabel])	Reiht eine Funktion zur Einreihung in eine Schlange von Funktionen, die sich eine 50-ms-Loop teilen (jedes Mal "gleichzeitige" Abarbeitung aller Funktionen) >> wsch. Vorteil: Performancegewinn gegenüber der Vorgehensweise, für jede Funktion eine eigene Loop zu bestimmen >> Gut geeignet für eigene Eventhandler, die sich nicht aus bereits vorhandenen ableiten lassen >> Es sollten keine alerts, zu lange dauernde for-Schleifen oder andere Browserstopper in der jeweiligen Fkt. vorhanden sein.
loop.timer.addSlow(fnc [, fncLabel])	Wie loop.timer.add(), jedoch statt 50ms- mit 1000ms-Abarbeitungsintervall >> Technisch ist es aber dieselbe Schlange wie die von loop.timer.add()
loop.timer.drop(fncLabel/index)	Löscht eine Funktion aus der Schlange per Indexangabe (nach Reihenfolge im Code oder in loop.timer.add() definiertem Label)
delay(fnc, sec)	Bietet im IE bessere CPU-Performance als window.setTimeout() >> sec: Sekundenangabe zwischen 0 und 1; nur eine einzige Stelle nach dem Komma erlaubt (0.1, 0.2, 0.3 etc.) >> Im IE etwas unpräzise (Fehlertoleranz steigt mit höheren Sekundenangaben von ca. 20ms bis ca. 120ms)
tempchange(varName, tempValue, msec)	Ändert den Wert einer globalen (!) Variable für msec Millisekunden vorübergehend >> Jeder erneute Aufruf vor Ablauf der Zeit verlängert die Zeit bis zur Wiederherstellung um msec Sekunden ab dem Zeitpunkt des erneuten Aufrufs >> varName: name der Variable als String >> (Evtl. noch weiterprogrammieren für Anwendbarkeit auf lokale Objekte statt nur globale) >> Für Objektvariablen: [Object].tempchange(varName, tempValue, msec) (auch vom Inneren eines Objects mit 'this.tempchange()' nutzbar)
defer(fnc, msec)	Schiebt die Ausführung einer Funktion um msec auf >> Jeder erneute Aufruf verhindert die Ausführung nochmals, bis msec Sekunden vergangen sind
from(obj, function(key){ ... })	Für reibungslosen Durchlauf von Assoziativen Array und Objekten (Alternative zu: for(var i in myObj){ if(keyOk(i)){ ...}};) >> Verschachtelung mehrerer froms leider problematisch >> Anwendungsweise: from(myObj, function(i){ ... });
keyOk(key)	Für reibungslosen Durchlauf von Assoziativen Array und Objekten >> Überprüft, ob gefundener Schlüssel zur eine Prototyp-Eigenschaft ist >> Sollte immer anstelle der normalen Schreibweise der for-in-Schleife benutzt werden >> Anwendungsweise: for(var i in myObj){ if(keyOk(i)){ ... ... }}
include(scriptName, resumingCode)	Führt Code aus einer anderen Datei aus >> Code sollte in 'resumingCode' fortgestezt werden >> Voraussetzung: HTML-Body-Bereich vorhanden >> Achtung: Variablen sind den beiden Dateien dann gegenseitig bekannt >> Bei erneutem Aufruf mit gleichem 'scriptName' keine neue Einbindung, sondern nur Ausführung des 'resumingCode'
include[scriptName].onready	Feuert, wenn das Skript zu Ende geladen wurde >> Nur sinnvoll, wenn oben 'resumingCode' nicht genutzt werden kann
include.register(fnc1 [, fnc2, ...], file)	Meldet mehrere in externem Skript ausgelagerte Funktionen auf ein Mal so an, dass man sie sofort nutzen kann >> Vorteil: 1. Man muss sich nicht bei jedem Fkt.-Aufruf darum kümmern, ob sie schon geladen wurde; 2. Platzersparnis; 3. Schnelleres Laden am Anfang >> Derzeit nur für externe Funktionen mit max. 20 Parametern >> Externe Datei wird durch die Registrierung noch *nicht* geladen, sondern erst beim späteren Aufruf der ersten von den registrierten Funktionen

>> Falls ausgelagerte Funktion Rückgabewert hat, dann nicht wie normal 'x = fnc();', sondern:

```
fnc();
fnc.resume(function(){
        x = fnc.retVal;
});
```

Funktionen:

thisFunctionName()	Gibt den Funktionsnamen der umgebenden Funktion an >> gut um selbe Debuggingzeile ohne weitere Änderungen in jede Funktion einfügen zu können)
[Function].getHead()	Gibt den Funktionskopf einer Funktion an
[Function].getBody()	Gibt den Funktionskörper einer Funktion an
[Function].getName()	Gibt Funktionsnamen einer Funktion an (von Funktionskörper befreit)
[Function].getParameters()	Gibt die Parameter einer Funktion als String zurück
[Function].clone()	Gibt ein Duplikat der Funktion zurück, so dass der Var.-Name nicht nur ein Zeiger auf ein- und dieselbe Funktion ist
[Function].get(varName)	Gibt Inhalte von Parametern an, die nicht aus den Klammern stammen, sondern übergeben wurden, indem genau *vor* dem Funktionsaufruf geschrieben wurde:

```
function bspFkt(){
        alert(bspFkt.get("kosten"));
}
bspFkt.kosten = 1000;
bspFkt();
```

>> Nach dem Aufruf ist ohne eine erneute Übergabe die Variable nicht mehr abrufbar.

[Function].isEmpty()	Gibt an, ob die Funktion leer ist oder nicht (Kommentare => nicht leer) >> Leere Funktion: function(){}

HTML(-Elemente) & CSS:

document.activeElement	Gibt fokussiertes Element auch im Firefox zurück (sonst nur in IE)
[Element].ondeactivate	Feuert, wenn das Element deaktiviert wird
id(bez)	Kürzt folgendes ab: document.getElementById(bez)
.ondomchange	Registriert, ob das Element oder etwas in seinem Inneren verändert worden ist
.onwidthchange	Registriert, ob das Element sich in der Breite verändert hat (auch ohne CSS-Veränderung, z.B. bei Aufblähung durch zusätzlichen Text)
.onheightchange	Registriert, ob das Element sich in der Höhe verändert hat (auch ohne CSS-Veränderung, z.B. bei Aufblähung durch zusätzliches Kindelement)
.onattrchange	Registriert, ob ein Attribut des Elements verändert wurde (und sei es auch "nur" das style-Attribut) >> FF feuert nur bei echter Veränderung des Wertes, IE auch bei reinem Neusetzen desselben Wertes >> Per '[Ereignis].attrName' lässt sich herausfinden, wie das geänderte Attribut heißt (Ereignis muss übergeben werden!): >> Per '[Ereignis].newValue' lässt sich der neue Wert herausfinden >> Per '[Ereignis].relatedNode' lässt sich der geänderte Attributknoten selbst (statt nur der Name) herausfinden >> Bsp.:

```
elm.onattrchange = function(e){
        alert(e.attrName);
        alert(e.relatedNode);
};
```

.onvaluechange	Registriert bei INPUTs & TEXTAREAs, ob sich der Inhalt beim Tippen verändert (statt nach Hinausklicken wie bei onchange)
.onclickintoframe	Falls id-Objekt ein IFrame ist: Eventhandler, der feuert, wenn in den Inhalt des IFrames geklickt worden ist >> Momentaner Bug: Wenn vorher Button aktiv war, wird im FF2 auf das Ereignis doppelt reagiert (FF3 okay)
.onlocationchange	Falls id-Objekt ein IFrame ist: Eventhandler, der feuert, wenn eine neue Seite im IFrame zu laden begonnen wird
.onloadcomplete	Falls id-Objekt ein IFrame ist: Eventhandler, der feuert, wenn eine neue Seite im IFrame zu Ende geladen worden ist
.onnavigate	Falls id-Objekt ein IFrame ist: Eventhandler, der feuert, wenn eine vom User (!) gewählte neue Seite im IFrame erscheint, auch wenn noch nicht zu Ende geladen
.onafternavigate	Falls id-Objekt ein IFrame ist: Eventhandler, der feuert, wenn eine neue Seite im IFrame vom User (!) gewählt wurde und zu Ende geladen ist >> Gut zur Feststellung, ob auf einen Link geklickt wurde
.onenter	Eventhandler, der feuert, wenn Enter-Taste gedrückt wurde (für Texteingabefelder)
.onuparrow	Eventhandler, der feuert, wenn Pfeil-nach-oben-Taste gedrückt wurde (für Texteingabefelder)
.ondownarrow	Eventhandler, der feuert, wenn Pfeil-nach-unten-Taste gedrückt wurde (für Texteingabefelder)
.onrightarrow	Eventhandler, der feuert, wenn Pfeil-nach-rechts-Taste gedrückt wurde (für Texteingabefelder)
.onleftarrow	Eventhandler, der feuert, wenn Pfeil-nach-links-Taste gedrückt wurde (für Texteingabefelder)
.oncontact	Eventhandler, der feuert, wenn es von einem ebenfalls diesen Handler besitzenden Element grafisch berührt wird. >> In '[Ereignis]'.thisElement und '[Ereignis]'.otherElement sind die beiden "Kollisionspartner" abgelegt
.onslow	Feuert während des Fade-In-/Fade-Out-Prozesses von .show()/.hide(), wenn dieser zu langsam abläuft
.hasShadow	Gibt an, ob das Element einen Schatten hat
.shadow	Enthält das Schatten-Element, falls einer durch .addShadow() hinzugefügt worden ist
.bg	Enthält das Element des synthetischen Hintergrunds, falls einer durch .setBgOpacity() bzw. synthesizeBg() hinzugefügt worden ist
.hide([sec [, lim]])	Lässt das Element optisch verschwinden >> Falls 'sec' angegeben: Fade-Out-Verschwinden innerhalb von 'sec' Sekunden >> Falls 'lim': Opacity-Grenze, welche, beim Verbergen nicht unterschritten werden soll
.show([sec [, lim]])	Zeigt das Element >> Falls 'sec' angegeben: Fade-In-Auftauchen innerhalb von 'sec' Sekunden >> Falls 'lim': Opacity-Grenze, welche, beim Anzeigen nicht überschritten werden soll

>> mit .show(false) lässt sich das langsame Auftauchen anhalten.

.remove() Löscht das Element

.clone(truthValue) Alternative zu .cloneNode() - beugt Problemen durch Klonen geboosteter Elemente vor

>> truthValue: Falls 'true', wird Inhalt des Elements mitkopiert

.empty() Leert das Element von anderen Elementen

.cutOut(first, last) Schneidet eine Kette im Element befindlicher Elemente aus

>> first: erstes auszuschneidendes Element, zulässige Typen: Number oder HTML-Objekt

>> last: letztes auszuschneidendes Element, zulässige Typen: Number oder HTML-Objekt

.insertBefore(element) Vereinfachung der offiziellen insertBefore()-Methode (Elternelement-Erwähnung unnötig, stattdessen einzufügendes Element vorangestellt)

.insertAfter(element) Wie .insertBefore(), hier fürs Einfügen nach einem Element

.add(tagName [, text[, attr1, [attrVal1 [, ...]]]]) Fügt neues Unterelement mit beliegig vielen Attributen hinzu; gibt das entsprechende Element auch zurück

.setInnerHTML(htmCode) Alternative zu innerHTML-Zuweisungen, um Probleme zwischen IE und geboosteten Elementen zu vermeiden (unboostet das eingesetzte automatisch).

.putOver(element) Gibt dem Element die Überlappungshoheit über das übergebene Element (Erhöhung des z-index)

.moveTo(x,y) Setzt Element exakt an die gewünschten Koordinaten

resizeTo(x [, y]) Setzt die Dimensionen des Kastens neu

.wrap() Legt einen DIV-Umschlag, der sich immer mitbewegt, um das Element

>> Derzeit nur für absolut-positionierte Elemente

>> Ab dem Zeitpunkt des Wrappens sind left-, top, width- & height-Stile nur noch über '[diesesElement].wrapper.style.left' usw. korrekt erreichbar

>> Für Textareas heißt die Funktion ".wrapTextarea()" !

.wrapWith(wrappingElement) Hüllt Element mit anderem Element ein

>> Anwendungsbeispiel 1:

```
huelle = add("div", "")
id('meinElement').wrapWith(huelle);
```

>> Anwendungsbeispiel 2:

```
id('meinElement').wrapWith(add("div", ""));
```

.powerFocus() Alternative zu '.focus()', falls letzteres nicht funktioniert oder der Fokus damit zu schnell verloren geht

.hasFocus() Gibt an, ob das Element momentan fokussiert ist.

>> Falls der Fokus durch JS-Code statt den User verliehen wurde, funktioniert diese Funktion zuverlässig nur bei konsequenter Verwendung von:
id().focus() / tg().focus() / cl().focus() / _().focus()

.isIn(obj) Gibt an, ob das Element schon/noch im angegebenen Element eingesetzt ist oder nicht

>> obj: Element oder document-Objekt

.addArrowControl([val]) Für Formulare, damit man mit den Pfeiltasten von Feld zu Feld springen kann

>> 'val': 'false', wenn das Formular nicht zum Submitting bestimmt ist (evtl. onsubmit-Handler werden überschrieben)

.makeHoverToggle([stName, st1, st2]) Lässt Element je nach Mouseover/Mouseout zwischen zwei Zuständen wechseln

>> Ohne Parameter: Sichtbar/Unsichtbar-Effekt

>> stName: Name der CSS-Eigenschaft

>> st1: CSS-Wert 1

>> st2: CSS-Wert 2

.makeDraggable([elm2 [,elm3 [,...]]]) Macht ein Element mausziehbar

>> elm2, elm3 etc.: Weitere Elemente, die "mitgezogen" werden sollen

>> Wenn Element gerade gezogen wird: [Element].isBeingDragged ist wahr

>> Derzeit muss 'position' des zu verschiebenden Elements noch von vorneherein 'absolute' sein.

>> Liegt in dem Element ein weiteres, kann das umgebende Element auch durch Dragging des inneren Elements mitgezogen werden, ohne das das innere eigens angegeben werden muss

>> zum vorigen Punkt: hat das innere Element die positive Eigenschaft '.noDragging', bleiben Versuche, das innere Element zu draggen, für das komplette Konstrukt wirkungslos

>> (Sollte so verbessert werden, dass eventuelle Shadows automatisch mitgezogen werden, ohne eigens angegeben werden zu müssen)

.attachHoverPopUp([htmCode]) Sorgt dafür, dass neben dem Mauszeiger immer ein mitwandernder Kasten mit dem inhalt 'htmCode' angezeigt wird, solange der Zeiger sich über dem Element befindet

>> Mitwanderndes Pop-Up per CSS über Klassennamen ".hoverpopup" erreichbar

>> Gibt Pop-Up als HTML-Element zurück

>> Falls 'htmCode' nicht angegeben, sollte im title-Attribut des Original-HTML-Codes der entsprechende Text stehen.

>> Während des mouseovers wird das title-Attribut geleert und der Inhalt im Attribut 'savedTitle' gesichert

.setBgOpacity(opc) Erhöht/reduziert Lichtundurchlässigkeit des Hintergrunds

>> Vorteil: Weitere Elemente außer Hintergrund in dem Element von Transparenz nicht betroffen

>> opc: Grad der Lichtundurchlässigkeit (0-100)

>> Hintergrund dann einzeln ansprechbar - befindet sich als DIV in [Element].bg

>> Achtung: Weder width noch height des Kastens dürfen "auto" sein - beide müssen explizit angegeben werden!

.addShadow([opc [, ab, cd]]) Fügt einem Kasten einen Schatten hinzu

>> Sollte nur bei absolut-positionierten Kästen angewendet werden

>> opc: Grad der Lichtundurchlässigkeit (0-99)

>> ab: Abstand zwischen den beiden links-oberen Ecken von Kasten & Schatten

>> cd: Abstand zwischen den beiden rechts-unteren Ecken von Kasten & Schatten

>> Schatten dann einzeln ansprechbar - befindet sich als DIV in [Element].shadow

.adoptStyle(styleName, fremdesElem) Lässt bestimmten CSS-Stil immer genau so sein und sich verändern wie der eines anderen Elements.

>> Funktioniert nur bei Stilen im Style-Attribut bzw. solchen, die in Javascript definiert worden sind.

.absorbStyle(styleName, fremdesElem, corr) Wenn fremdem Element eine CSS-Eigenschaft gegeben wird, wird es dieser sofort "entwendet"

>> corr: Funktion für Korrekturen, z.B. falls Absorption Unproportionalitäten nach sich zieht

.getStyle(rule) Gibt Wert einer CSS-Eigenschaft an, auch wenn sie nicht im Tag-Attribut "style", sondern in einem Stylesheet steht

>> Argument 'rule': CSS-Eigenschaft als String, z.B. "position", "left", "top" usw.; Rechtschreibung sowohl Original-CSS als auch JS, z.B. "marginLeft" oder "margin-left"

>> Ist im IE u n d FF auch zu Prozentangaben bei "left", "right" etc. in der Lage

.deleteInlineCSS([styleName]) Löscht die Inline-Regel 'styleName', so dass höchstens noch die entsprechende global definierte Regel gilt

>> Ohne die Angabe von 'styleName' wird der ganze Inline-Code gelöscht

.getX() Gibt absolute x-Position des Elements an, egal wie tief verschachtelt es ist

.getY() Gibt absolute y-Position des Elements an, egal wie tief verschachtelt es ist

.getRightX() Gibt Abstand des rechten Randes zum rechten Bildschirmrand an

.getBottomY() Gibt Abstand des unteren Randes zum unteren Bildschirmrand an

.getScrollX() Gibt an, wieviele Pixel ein overflow:scroll-Element horizontal gescrollt worden ist

.getScrollY() Gibt an, wieviele Pixel ein overflow:scroll-Element vertikal gescrollt worden ist

.getValueWidth() Für Input-Boxen: Gibt Breite an, die der sichtbare (!) Textwert einnimmt

.touches(element) Gibt an, ob dieses Element ein anderes grafisch berührt.

.shows(element) Gibt an, ob sich ein(es der) Kindelement(e) z.B. durch Scrollen optisch im Bereich des Viewports befindet

>> Für overflow:scroll- & overflow:hidden-Elemente geeignet

>> Derzeit nur Berücksichtigung vertikaler Positionen

.getNextElement() Alternative zur nextSibling-Eigenschaft; sucht nur nächstes Tag-Element (statt Knoten allgemein, also keine Textknoten usw.)

>> Für den nächsten Knoten auf der selben, nicht tieferliegenden, Ebene

.getOuterHTML(truthVal) Ähnlich document.all.outerHTML: auch für Firefox

>> truthVal: mus auf 'true' gesetzt werden, wenn der Inhalt des Elements mitgespeichert werden soll

.getAttributeNodes() Gibt Array mit den aktuellen Attributknoten des Elements zurück

>> Attributname per ".nodeName" und Wert per ".nodeValue" ermittelbar

.hasAttributeNode(attr) Gibt an, ob das Element den Attributknoten 'attr' hat.

.replaceText(i, text) Ersetzt in einem Element die Daten des i-ten Textknotens

.scrollToBottom() Lässt scrollbare Elemente (DIVs) ganz nach unten scrollen (falls letztes Element darin ein Tag ist)

.scrollToTop() Lässt scrollbare Elemente (DIVs) ganz nach oben scrollen (falls erstes Element darin ein Tag ist)

.scrollIntoBottomView() Wie [node].scrollIntoView(), jedoch unten statt oben anzeigend

.addUploader(targetDir, buttonTitle) Fügt (z.B. einem DIV) ein funktionsfähiges Upload-Formular hinzu

>> Gibt Formular-Element zurück; darum mgl.: id("xyz").addUploader("verz1","hochladen!").onsubmit = ...

>> Zugriff auf Eingabefeld: z.B. [Upload-Formular-Bezeichner].elements[0].size = ...;

>> Zugriff auf Namen der gewählten Datei incl. Pfad: x = [Upload-Formular-Bezeichner].elements[0].value

.navigate(url) Für IFrames: Empfohlene Alternative zur .src-Zuweisung - Gründe:

>> zwecks onattrchange-Kompatibilität zum IE

>> da .protect() sonst nur unzuverlässig funktioniert

>> da sonst wegen onback- & onforward-Features Probleme beim Navigationsmechanismus auftauchen können

.protect(m) Schützt 1.) IFrames vor Ausbruchsversuchen externer Seiten und 2.) HTML-Elemente (z.B. DIVs) vor Überlagerung durch Flashs externer IFrame-Seiten

>> Bei Iframes:

- Überlässt bei Iframes durch ein Confirm-Fenster dem User die Wahl
- m: Message für den User im Confirm-Fenster bei Ausbruchsversuchen;
- Muss angewendet werden, n a c h d e m das IFrame seinen src-Inhalt bekommen hat!
- Kann einmalig verhindert werden mit disallowProtectorOnce(); wichtig für IE, da dieser bei Javascript-Void-Links den Protector aktiviert
- macht den window.onbeforeunload-Handler unbrauchbar (Ausweichen auf window.onunload!)

>> Bei sonstigen Elementen:

- Klappt derzeit nicht bei Chrome & Safari
- m: Falls mit beliebigem Wert besetzt, kann so auch im Firefox Iframe-Element statt overflow-y Eigenschaft zur Überlagerungsverhinderung genutzt werden

.contentOffset(x, y [, innerW[,innerH]]) Verschiebt den Inhalt eines Elements

>> Gut für IFrames, um Webseiten innerhalb des Ausschnitts hin- und herzubewegen

Achtung: Wenn IFrame-Content vor Verschiebung mit 'location.href' geändert wurde, springt Inhalt im FF auf in 'src'-Attribut definierte URL oder 'about:blank' zurück => Änderungen vor Verschiebung bitte immer über src-Attribut!

>> innerW & innerH, um Content kleiner als Ausschnitt, größer oder "grenzenlos" erscheinen zu lassen (optional)

>> Für x, y, innerW & innerH sowohl numerische Werte als auch Strings mit Prozentangaben möglich

.bindToClipboard(text) Stellt das Element so ein, dass der übergebene Text beim Anklicken immer automatisch in die Zwischenablage gespeichert wird

>> Funktioniert nur auf Servern ("http://...")

>> Setzt Flash voraus

>> Wschl. nur für absolut-positionierte Elemente möglich

.bindToMenu(menu) Lässt beim Überfahren des Ekements mit der Maus ein Menü aufspringen

>> menu: Menü vom Typ ContextMenu

>> Gut für die Konstruktion von Menüleisten (z.B. DIV-Reihe, jedes DIV mit .bindToMenu() behandelt)

.embedVideo(url) Füllt das Element mit Youtube-Video aus (Element sollte absolut-positioniert sein)

>> Element hat dann folgende Funktionen & Handler zur Steuerung des Videos:

.onerror	Feuert, wenn ein Fehler beim Abspielen auftaucht
	>> Lässt sich von Video Fehlercode übergeben:
	100 = Video nicht gefunden
	101 = Video verweigert Abspielen als eingebettetes Element
	150 = 101
.onvideopause	Feuert, wenn Video pausiert wurde (egal ob manuell oder vom Code)
.onvideoplay	Feuert, wenn Video gestartet/fortgesetzt wurde (egal ob manuell oder vom Code)
.onvideoend	Feuert, wenn Video ans Ende angekommen ist (auch bei Loops)
.onvideobuffering	Feuert bei Ladevorgängen (auch solchen, die mitten im Abspielen geschehen)
.onvideostop	Feuert, wenn Video bereit zum erstmaligen manuellen Start (Startknopf mitten im Bild) und beim Stoppen (statt Pausieren)
.playVideo()	Startet Video / setzt es fort
.pauseVideo()	Pausiert Video
.stopVideo()	Stoppt Video
.seektTo(sec)	Springt zur angegebenen Sekunde (falls zuvor pausiert/gestoppt, wird Pause/Stop aufgehoben)
.setPlaybackQuality(q)	Ändert Auflösung des Videos
	>> q: Stringwert ("small" / "medium" / "large" / "hd720" / "default")
	>> nur während des Abspielens, also am besten in 'onvideoplay' einbauen
.getPlayerState()	Gibt Zustand des Videos als Nummer zurück: unstarted (-1), ended (0), playing (1), paused (2), buffering (3), video cued (5)
.getCurrentTime()	Gibt bisher gespielte Zeit in Sekunden zurück
.mute()	Mutes the player.
.unMute()	Unmutes the player.
.isMuted()	Returns true if the player is muted, false if not.
.setVolume(vol)	Sets the volume. Accepts an integer between 0 and 100.
.getVolume()	Returns the volume as an integer
.clearVideo()	Ersetzt das Bild durch eine schwarze Fläche, wenn Video pausiert oder gestoppt (Ladeanimation beim Buffern damit vermeidbar?)

>> Weitere Funktionen & Eventhandler: http://code.google.com/apis/youtube/js_api_reference.html

>> Funktioniert nur von HTTP(S)-Server aus (z.B. localhost)

.tg(tagString [,i]) Gibt das i-te Element innerhalb des Elements anhand des Tagname zurück

>> Verschachtelungen möglich, z.B.: id(...).tg(...).tg(...)

.cl(classString [,i]) Gibt das i-te Element innerhalb des Elements anhand des Class-Namens zurück

>> Verschachtelungen möglich, z.B.: id(...).cl(...).cl(...).tg(...).cl(...).tg(...)

._id(idString) Gibt für den Fall, dass dokumentweit identische IDs existieren, das Element mit nur derjenigen ID zurück, das sich innerhalb dieses Elements befindet

>> Verschachtelungen möglich, z.B.: id(...).id(...).id(...)

>> Achtung Unterstrich

>> Innerhalb eines Elements sollte auf derselben Ebene nur eine einzige ID vorkommen

tg(tagString [,i]) Ähnlich id(idString); Abkürzung für document.getElementsByTagName(tagString)[i]

>> Falls 'i' nicht abgegeben, wird 0-tes Element behandelt/zurückgegeben

>> Falls 'tagString' nicht angegeben, bezieht sich die Fkt. automatisch aufs Body-Element

>> Falls "*" für 'tagString' angegeben: Gibt Array aller aktuellen Elemente des Dokuments zurück.

>> Hat dieselben Properties & Methoden wie id()-Objekt

.css(rules) Stellt CSS-Regel für alle aktuellen und potenziellen Elemente auf, die den Tag-Namen 'tagString' haben

>> Bsp.: tg("div").css("color red; font-size:10px");
>> Könnte u.U. in window.onload eingebaut werden müssen.

.numOfAll — Gibt Anzahl der im Dokument vorhandenen Elemente des betreffenden Tags an
>> Bsp.: alert(tg("div").numofAll);

cl(clString [,i]) — Ähnlich id(idString); Abkürzung für getElementsByClassName(clString)[i]
>> Falls 'i' nicht abgegeben, wird 0-tes Element behandelt/zurückgegeben
>> Falls 'clString' nicht angegeben, bezieht sich die Fkt. automatisch aufs Body-Element
>> Hat dieselben Properties & Methoden wie id()-Objekt

.css(rules) — Stellt CSS-Regel für alle aktuellen und potenziellen Elemente auf, die den Class-Namen 'clString' haben
>> Bsp.: cl("textfeld").css("color red; font-size:10px");
>> Könnte u.U. in window.onload eingebaut werden müssen.

.numOfAll — Gibt Anzahl der im Dokument vorhandenen Elemente der betreffenden Class an
>> Bsp.: alert(cl("blume").numofAll);

_(element) — Fügt einem Element auch ohne id()-/tg()-/cl()-Funktionen deren Methoden hinzu
>> Anwendung z.B. alert(_(elm).getX());
>> Einmalige Anwendung reicht - dem Element stehen fortan alle neuen Methoden immer zur Verfügung
>> Nach Anwendung nicht mehr mit .cloneNode(), sondern nur noch mit .clone() klonen!!
>> Nur Methoden, nicht alle Eventhandler stehen zur Verfügung (derzeit)
>> Auch geeignet zum prüfen, ob eine Variable definiert ist: if(_(xyz)) { ... } [statt if((typeof xyz)=="undefined")]

boostElement(element) — Alternative zur "_()"-Funktion, um ein Element mit id()-/tg()-/cl()-Funktionen auszustatten
>> Eigentlich nur Arbeitsfunktion, aber evtl. nützlich nach innerHTML-Übertragungen, da "_()" dann evtl. versagt

unboost(element) — Gegenteil von boostElement()

layerChange(layerId) — Wechselt einen Teil der HTML-Seite als Schicht, die eine andere ersetzt
>> HTML-Code:
...
...
... (usw.)
>> Oberstes DIV muss Klassennamen "mainLayer" haben
>> Jedes DIV muss eine beliebige ID haben

urlSelectsLayer() — Lässt URL-Parameter mit Muster "layer=layerId" wie "layerChange(layerId)" wirken
>> Voraussetzungen siehe "layerChange(layerId)"

activateCenterVertical( ) — Ermöglicht vertikale Zentrierung des Inhalts von <!-- [centerVertical] //-->...<!-- [/centerVertical] //-->
>> Sollte am Fuße des BODYs eingesetzt werden

printNode(htmNode) — Drucke beliebiges ID-spezifiziertes HTML-Element im ggw. Dokument;
>> Parameter immer als "document.all.<id>", z.B.
>> "printNode(document.all.meinKnoten);"

superblur() — Lässt jegliches momentan fokussierte Element nachhaltig seinen Fokus verlieren
>> Normales blur() oft unzuverlässig
>> Funktioniert nicht in einem IFrame, welches Turboid eingebunden hat, während die umgebende Seite ebenfalls Turboid eingebunden hat

_(expr).isHtmlObj — Gibt zurück, ob die übermittelte Entität ein HTML-Objekt ist

isHtmlAttribute(val) — Gibt zurück, ob ein String ein typischer HTML-Attributname ist
>> Anwendung: if(isHtmlAttribute["xyz"]) ...

makeHtmlTree(arr) — Macht aus ID-basiertem Baumstruktur-Array eine ul-li-Liste (String als Rückgabe)
>> Formatbeispiel des Arrays:

```
var arr = [[],[],[],[],[],[],[],[],[],[],[]];

arr[0]["ID"] = 111;
arr[0]["parentID"] = 0;
arr[0]["content"] = "Hallo";

arr[1]["ID"] = 222;
arr[1]["parentID"] = 111;
arr[1]["content"] = "Leute";

arr[2]["ID"] = 333;
arr[2]["parentID"] = 222;
arr[2]["content"] = "Wie";

arr[3]["ID"] = 444;
arr[3]["parentID"] = 222;
arr[3]["content"] = "heiße";

arr[4]["ID"] = 555;
arr[4]["parentID"] = 111;
arr[4]["content"] = "ich";
```

document.hasDoctype() — Gibt browserübergreifend an, ob das Dokument eine Doctype-Angabe besitzt

CSS.add(rules) — Fügt Stylesheet hinzu
>> Bsp.: CSS.add("div { color:violet; font-size:20px; } .klasseX { background-color:gray; }");

CSS.prepend(rules) — Wie CSS.add(), fügt die neuen jedoch v o r evtl. bereits bestehenden Regeln ein.

CSS.getValue(selector, styleName)	Gibt den Wert der CSS-Eigenschaft eines *Selektors* (nicht nur: Elements) an, auch wenn nur in externem Sheet deklariert
	>> Anwendungsbeispiel:
	x = CSS.getValue(".meineKlasse", "font-weight");
	y = CSS.getValue("#meineId", "font-weight");
	z = CSS.getValue("div", "font-weight");

CSS.rescueFont(s1, styles1 [, s2, styles2 [, ...]])	Korrigiert Font-Regeln, falls der im Sheet festgelegte Font nicht installiert ist, nicht nur mit Ersatzfont, sondern auch -größe & -stil >> styles: String, bestehend aus Angaben der Ersatzeigenschaften, mit Leerzeichen getrennt, z.B. "Calibri 20 bold" (Reihenfolge egal) >> styles (Forts.): bis zu 4 Angaben möglich (Fontname, Fontgröße, Fontstärke, Fontstil) >> in styles-String "notbold" oder "notitalic" statt "normal" >> Anwendungsbeispiel: CSS.rescueFont("#meineId", "Calibri 20 notbold", ".meineKlasse", "Calibri 14 notitalic", "div", "'Courier New' 25 italic");

Strings:

trimString(kette)	Schneidet Leerzeichen vorne und/oder hinten weg [Sollte vll. in 'trim()' umbenannt weden]
replaceString(suchwort, ersatzwort, satz)	Ersetzt gesuchte Zeichenkette beim ersten Vorkommnis >> Gr.-Kl.-Schreibungsempfindlich >> Gibt "satz" unverändert zurück, falls Suchwort nicht gefunden wurde
replaceStrings(suchwort, ersatzwort, satz)	Ersetzt gesuchte Zeichenkette an allen Stellen >> Gr.-Kl.-Schreibungsempfindlich >> Gibt "satz" unverändert zurück, falls Suchwort nicht gefunden wurde >> Achtung, hier kommt 'amsa' statt 'ammmmmsa' raus: replaceStrings("mm","m","ammmmmmmmsa")
replaceWord(suchwort, ersatzwort, satz)	Ersetzt gesuchtes Wort beim ersten Vorkommnis >> Gr.-Kl.-Schreibung egal >> Gibt "satz" unverändert zurück, falls Suchwort nicht gefunden wurde
replaceWords(suchwort, ersatzwort, satz)	Ersetzt gesuchtes Wort an allen Stellen >> Gr.-Kl.-Schreibung egal >> Gibt "satz" unverändert zurück, falls Suchwort nicht gefunden wurde >> Achtung, hier kommt 'amsa' statt 'ammmmmsa' raus: replaceWords("mm","m","ammmmmmmmsa")
replaceAt(pos, ersatzwort, satz)	Überschreibt den String ab pos
shortenSpaces(str)	Macht Mehrfach-Leerzeichen zu Einfach-Leerzeichen
count(gesucht, satz)	Zählt die Vorkommnisse einer Zeichenkette in einer anderen [Sollte vll. in 'count()' umbenannt werden]
tokenize(str, trennz1 [, trenn2, trennz3, ...])	Splittet String anhand beliebig vieler Trennzeichen in Array auf >> Alternative zu [String].split() >> Auch ohne explizite Angaben wird in Anf-Zeichen Eingefasstes als abzuspaltendes Element behandelt, dafür jedoch das Innere davon vor dem Splitting geschützt >> Zum vorigen Punkt: Anf.-Zeichen bleiben nicht erhalten >> Aufeinanderfolgende unterschiedliche Trennzeichen werden wie ein einziges Trennzeichen behandelt >> Gut für Sucheingaben
capitalize(str)	Gibt String mit Grossbuchstaben am Anfang zurück
isCapitalized(str)	Prüft, ob das Wort am Anfang großgeschrieben ist.
toCamelCase(str)	Wandelt z.B. "border-width" in "borderWidth" um.
toCSSName(str)	Wandelt z.B. "borderWidth" in "border-width" um.
randomString(length)	Erzeugt eine Zufalls-Buchstabenkette der gewünschten Länge
hasMailFormat(emailaddress)	Prüft, ob das E-mail-Adressformat gültig ist
filenameFromPath(path)	Gibt den Dateinamen in einem Pfadstring zurück
dirFromPath(path)	Gibt den Verzeichnisteil in einem Pfadstring zurück (d.h. Pfad ohne Dateinamen)
_(value).containsOneOf(v1 [, v2 [, ...]])	Gibt an, ob der Wert einen der übergebenen Strings als Teilstring beinhaltet.

Arrays:

_(val).isArray()	Prüft, ob eine Variable ein echtes Array enthält (gilt nicht für getElementsByTagName()[x] usw.)
[Array].drop(index)	Löscht ein Element aus dem Array (Rückgabe vorhanden, aber Aufnahme nicht unbedingt nötig)
[Array].dropByValue(value)	Löscht alle Elemente, die den übergebenen Wert aufweisen, aus dem Array (Rückgabe vorhanden, aber Aufnahme nicht unbedingt nötig)
[Array].contains(value)	Gibt Wahrheitswert darüber zurück, ob sich ein Wert in einem Array befindet oder nicht >> Schema z.B.: if(meineListe.contains("herbert")) ... >> Nicht für assoziative Arrays (hierfür [Array].getIndexOfKey())

[Array].getIndex(value)	Gibt an, an welchem Index ein Wert in einem Array steht >> Nicht für assoziative Arrays
[Array].getNext()	Ermittelt den Wert des nächsten Array-Elements >> Beginnt mit 0 -tem Element >> Gibt nach Überschreiten des Endes "null" zurück
[Array].setNext()	Setzt einen Punkt, ab dem die getNext()-Methode weitermachen soll
[Array].getPrev()	Wie getNext(), jedoch für das vorige Element >> Gibt nach Überschreiten des Anfangs "null" zurück
[Array].getCurrent()	Gibt aktuelles Element zurück
[Array].getKeyAtPoint(index)	Gibt Schlüssel des Elements eines Assoziativen Arrays per Indexangabe zurück
[Array].getValueAtPoint(index)	Gibt Wert des Elements eines Assoziativen Arrays per Indexangabe zurück
[Array].getIndexOfKey(key)	Gibt den Index eines Schlüssels in einem Assoziativen Array zurück
[Array].avg()	Ermittelt den Durchschnitt der Werte eines Arrays
[Array/Object].getLength()	Gibt die Anzahl der Elemente eines Assoziativen Arrays bzw.
[Array].insertAt(i, val)	Fügt an gewünschter Stelle eines Array ein neues Element ein
getKeyOfPeak(Array/Object)	Gibt Schlüssel der Eigenschaft mit dem höchsten Wert zurück
getKeyOfLowest(Array/Object)	Gibt Schlüssel der Eigenschaft mit dem niedrigsten Wert zurück
valuesort(toBeSorted [, -1])	Gibt Array/Object nach Werten sortiert zurück >> Bei Angabe von -1 wird vom höchsten zum niedrigsten Wert sortiert, sonst umgekehrt
keysort(toBeSorted)	Sortiert Assoziatives Array nach Schlüssel
hasArrayElements(var)	Prüft, ob 'var' ein Array ist und Elemente enthält
ignoreUC	Verhindert, dass [Array].sort() groß- und kleingeschriebene Wörter in zwei verschiedene Gruppen einordnet >> Anwendungsweise: [Array].sort(ignoreUC)

Arithmetik & Zeit:

randomTo(max)	Gibt Zufallszahl kleiner/gleich der angegebenen Grenze zurück
getDigits(value)	Gibt Array mit Ziffern der angegebenen Zahl zurück
countDigits(value)	Gibt Zifferanzahl der übergebenen Zahl zurück
round(value, [numOfDigits])	Alternative zu Math.round - hier mit Begrenzungsmöglichkeit für Nachkommastellen
numIn(str)	Extrahiert aus einem String (z.B. "width:50px") nur die Zahl und gibt sie als Number-Typ zurück
makeEven(num)	Macht eine ungerade Zahl gerade, indem sie nach unten korrigiert wird (Gerade Zahlen werden beibehalten) >> Alternative Notation: _(num).makeEven()
forceDigits(val, num)	Erzwingt die gewünscht Stellenanzahl durch die eventuelle Voranstellung von zusätzlichen Nullen >> Rückgabewert: String mit dem Ergebnis
ignoreLeadingZeros(numAsStr)	Gibt führende Nullen besitzende Zahl korrekt wieder, da parseInt() und numIn() hierbei immer 0 zurückgeben >> Typ des Rückgabewerts: number >> Gut für Weiterverwendung von mit forceDigits() erzeugten Werten
isOdd(n)	Prüft, ob die Zahl ungerade ist >> Alternative Notation: _(num).isOdd()
isInt(n)	Prüft, ob die Zahl eine Ganz- oder eine Kommazahl ist. >> Alternative Notation: _(num).isInt()
_(value).isOneOf(p1 [, p2 [, p3 [...]]])	Prüft, ob die Zahl "value" einem der Parameter entspricht >> Auch für Strings
new Clock()	Richtet Stopuhr-Objekt ein
.start()	Startet/Resettet Objekt-Stopuhr
.stop()	Gibt Objekt-Stopuhrstand zurück (in Millisekunden)
sleep(gewZeit)	Pausiert Programmablauf, Parameter in Millisekunden >> Hält jedoch den Browser auf >> Gut, um sicherzustellen dass zum Unload der Seite eventuelle Daten vollständig an den Server geschickt werden

Fenster & Anzeigefläche:

newWindow(url, x, y, w, h [, fensterName])	Öffnet neues Browserfenster auch bei bestehendem Fenster mit neuen Maßen; auch aus einem neuen Fenster heraus >> Falls ohne Mausevent aufgerufen: Bietet automatisch Returndruck/Buttonklick als Zwischenschritt an >> 'newWindow.onready' feuert, sobald Fenster geöffnet ist
codeToNewWindow(code, x, y, w, h)	Zeigt beliebigen HTML-Code in neuem Fenster an
codeToChildWindow(code, x, y, w, h)	Zeigt beliebigen HTML-Code in neuem Fenster an und übernimmt dabei Stile & Skripte des aktuellen Fensters
maximizeWindow()	Maximiert Abmessungen des aktuellen Fensters
setMinimumWindowInnerSize(x,y)	Verhindert zu kleine Fenster
getViewportWidth([win])	Gibt Breite der Anzeigefläche an (browserübergreifend statt window.innerWidth)
getViewportHeight([win])	Gibt Höhe der Anzeigefläche an (browserübergreifend statt window.innerHeight)
getBrowserHeight()	Liefert die Höhe des gesamten Browserfensters (leider in IE7 und IE8 nur brauchbar, wenn kein zweites Tab offen ist)
document.getX()	Gibt X-Position der linkeren oberen Ecke der Anzeigefläche gemessen am Bildschirm rand an >> in IE nur wenn Mauszeiger sich schon auf Anzeigefläche befunden hat
document.getY()	Gibt Y-Position der linkeren oberen Ecke der Anzeigefläche gemessen am Bildschirm rand an >> in IE nur wenn Mauszeiger sich schon auf Anzeigefläche befunden hat

---

GUI:

---

messageBox(title, htmContent [, x, y, w, h])	Zeigt mausbewegliche Messagebox an >> Anwendung: var mb = messageBox("Mitteilung","..."); >> Gibt MessageBox als Einzelteile enthaltendes Objekt zurück >> Titel, Hauptbereich und Schließknopf lassen sich per CSS manipulieren: Klassennamen '.mbCaption', '.mbCloseButton', '.mbContentArea'
.onclose	Eventhandler, der feuert, wenn auf den Schließen-Knopf geklickt worden ist >> Nur bei Klick, nicht bei Schließen allein durch Programm
.caption	Fenstertitel-Bereich als HTML-Element
.contentArea	Fensterhauptbereich als HTML-Element
.closeButton	Schließknopf als HTML-Element
.resizeTo(w,h)	Verändert Dimensionen
.moveTo(x,y)	Verändert Position
.close()	Schließt Messagebox >> Anwendung z.B.: mb.close();
iWindow(url, title [, x, y, w, h])	Verschiebbarer IFrame im Fensterdesign (Alternative zu newWindow()) >> Gibt messageBox()-Element zurück >> Eingebundene Seite kann Kontrolle über dieses Fenste übernehmen (mit window.close(), window.resizeTo() & window.moveTo() - sofern gleiche Domain)
.onclose	Eventhandler, der feuert, wenn auf den Schließen-Knopf geklickt worden ist >> Nur bei Klick, nicht bei Schließen allein durch Programm
.caption	Fenstertitel-Bereich als HTML-Element
.mainArea	Fensterhauptbereich als HTML-Element
.closeButton	Schließknopf als HTML-Element
.resizeTo(w,h)	Verändert Dimensionen
.moveTo(x,y)	Verändert Position
.close()	Schließt Messagebox >> Anwendung z.B.: mb.close();
message(content [, sec])	Box für einfache Mitteilungen >> content: Mitteilungstext oder HTML-Code >> sec: Bestimmt Anzahl von Sekunden, nach denen das Feld verschwinden soll (optional; ohne Angabe kein Verschwinden) >> Box wird als HTML zurückgegeben >> Eigenschaft [message()].closeButton liefert Schließknopf-Element
.isOpen	>> Gibt an, ob es geöffnet ist (ist schon zu Beginn des Fade-out-Prozesses auf 0)
.onclose	Feuert, wenn das Feld geschlossen worden ist
.close()	Schlie0t das Feld
quickMessage(content [, sec])	Wie message(), verschwindet jedoch nach kurzer Zeit automatisch >> sec: Bestimmt Anzahl von Sekunden, nach denen das Feld verschwinden soll (optional; Verschwinden auch ohne Angabe) >> Box wird zurückgegeben >> Eigenschaft [message()].closeButton liefert Schließknopf-Element
letConfirm(htmContent, yesFnc [, noFnc])	Lässt den User die Durchführung einer Funktion 'yesFnc' vorher bestätigen >> Gibt messageBox-Objekt zurück >> optional wird noFnc ausgelöst, wenn User auf Nein klickt.
Waiting.display()	Zeigt Wartescreen an
Waiting.hide()	Verbirgt Wartescreen
new ProgressBar()	Instantiiert Fortschrittanzeige
.set(x,y,barwidth)	Setzt Fortschrittsanzeige
.fill(seconds)	Füllt Fortschrittsanzeige in seconds Sekunden >> Zur Beschleunigung / Bremsung: seconds verändern, aber vorher .stop() benutzen.
.stop()	Pausiert Fortschrittsanzeige
new ContextMenu()	Gibt ein Kontextmenü zurück, das im Defaultmodus per Maus, Pfeiltasten, Buchstabentasten und Returntaste steuerbar ist >> Arbeitet schlecht mit XHTML-Strict-Modus zusammen >> Nutzungsbsp.:

```
var myMenu = new ContextMenu();
myMenu.addOptions('Crawlers', 'Lexika', 'Page');
myMenu.options['Lexika'].addSubmenu();
myMenu.options['Lexika'].submenu.addOptions('Wikipedia', 'Meyers');
myMenu.options['Lexika'].submenu.options['Meyers'].addSubmenu();
myMenu.options['Lexika'].submenu.options['Meyers'].submenu.addOptions('Physik', 'Bio');
myMenu.options['Crawlers'].onselect = function(){    alert("Sie haben die Suchmaschinen allgemein gewählt."); };
myMenu.options['Lexika'].onselect = function(){ alert("Sie haben Lexika gewählt."); };
myMenu.options['Page'].onselect = function(){ alert("Sie haben haben eine beliebige Seite gewählt."); };
myMenu.options['Lexika'].submenu.options['Meyers'].submenu.options['Bio'].onselect = function(){ alert("Sie haben Bio gewählt."); };
with(myMenu.html.style) { fontFamily = "Arial"; fontSize = "20px"; backgroundColor = "#FFDDFF"; borderStyle = "solid"; borderWidth = "2px"; borderColor = "black"; }
myMenu.options['Lexika'].submenu.html.style.fontSize = "15px";
myMenu.options['Lexika'].submenu.activatedHtml.style.backgroundColor = "gray";
myMenu.activatedHtml.style.backgroundColor = "white";
myMenu.display(200,300);
```

>> Kurzform des obigen Beispielmenüs:

```
var myMenu = new ContextMenu();
with(myMenu){
        addOptions('Crawlers', 'Lexika', 'Page');
        options['Lexika'].addSubmenu().addOptions('Wikipedia', 'Meyers');
        getOption('Lexika|Meyers').addSubmenu().addOptions('Physik', 'Bio');
        // ...
        getOption('Lexika|Meyers|Bio').onselect = function(){ alert("Sie haben Bio gewählt."); };
        // ...
}
```

>> Design auch über CSS-Datei steuerbar:

```
.contextMenu { ... } / * Design der Menütafeln   * /
.sub_contextMenu { ... } / * Design der Submenütafeln; wenn Hauptmenü-Klassenname geändert wird, dann hier automatisch '.sub_neuerName'  * /
.contextMenuOption { ... } / * Design der Optionen   * /
.activatedOption { ... }  /* Design der Option, wenn sie aktiviert ist. * /
.submenu_<parentOption.id>  /* Design eines speziellen Submenüs, z.B. .submenu_322 {  }  * /
```

>> Weitere Existenz des Menüs prüfbar mit: 'if(exists(myMenu))...'

.onmouseused	Eventhandler, der feuert, wenn irgendwo im Haupt- oder in einem Submenü die Maus bewegt wird
.onmouseleftall	Eventhandler, der feuert, wenn Mauszeiger das ganze Menü endgültig verlassen hat
.ondisplay	Eventhandler, der feuert, wenn das Menü geöffnet wird
.onhide	Eventhandler, der feuert, wenn das Menü geschlossen wird
.onaction	Eventhandler, der feuert, wenn irgendwo im Menü oder einem seiner Submenüs eine Option aktiviert/deaktiviert/genutzt oder ein Menü geöffnet/geschlossen wird >> Nur fürs Wurzelmenü >> Bekommt Event-Objekt übergeben: e.src: Quellobjekt der Aktion (entsprechende Option oder (Sub-)Menü) e.type: Art der Aktion ("activate", "deactivate", "select", "display", "hide")
.delay	Zeit, die das Menü als Submenü warten soll, bis es durch ein Mouseover angezeigt wird (in Millisek.)
.hidingDelay	Zeit, die das Menü nach dem Verlassen mit der Maus braucht, bis es automatisch verschwindet >> Default-Wert: 1000 ms >> Zur Verhinderung des automatischen Verschwindens: Sehr große Zahl einsetzen
.dir	Legt fest, ob die Submenüs immer auf der rechten oder auf der linken Seiten geöffnet werden sollen ("right" oder "1" für rechts, "left" oder -1 für links)
.rightAxis	Wenn wahr, ist die x-Koordinate in [ContextMenu].display() ein CSS-"right"-Wert, sonst "left"
.bottomAxis	Wenn wahr, ist die y-Koordinate in [ContextMenu].display() ein CSS-"bottom"-Wert, sonst "top"
.isOpen	Gibt Wahrheitswert darüber zurück, ob dieses Menü momentan geöffnet ist.
.isClosable	Setzt und gibt zurück: Wahrheitswert darüber, ob dieses (evtl. Sub-)Menü schließbar ist oder nicht
.mouseIsUsed	Gibt Wahrheitswert darüber zurück, ob die Maus gerade irgendwo im Haupt- oder in einem Submenü bewegt wird
.mouseIsIn	Gibt Wahrheitswert darüber zurück, ob sich die Maus gerade irgendwo im Haupt- oder in einem Submenü befindet
.managesOverheight	Für das Ab-/Einschalten der automatischen Überhöhenbehandlung >> Ist standardmäßig auf 1 (true)
.options[]	Array mit allen im Menü vorhandenen Menüoptionen (Option-Objekte) >> Zugriff über Aufschrift oder Index möglich
.parentOption	Menüoption des eventuellen Elternmenüs, von der aus dieses Menü aufgeklappt wird >> Typ: Option-Objekt
.activeOption	Momentan im betreffenden (Sub-)Menü angesteuerte Option >> Typ: Option-Objekt
.activeOptionGlobal	Momentan im Wurzel- oder irgendeinem der Submenüs angesteuerte Option >> Typ: Option-Objekt >> Wird nur im Wurzelmenü angelegt
.submenus	Array mit den zu dem jeweiligen (Sub-)Menü gehörenden Submenus (sonst 'null')
.allSubmenus	Enthält Array mit allen beliebig tief verschachtelten Submenüs (sonst 'null') - derzeit nur für das Wurzelmenü
.html	Enthält das Menü als HTML-Element >> Grafikstile hierüber definierbar, z.B.: myMenu.html.style.color = "blue";
.htmlPaddingStyle	Ausnahme: Padding-Definitionen sollten hier gesetzt werden >> Sonst nehmen Optionen-Aktivitätsmarkierungen nicht ganze Breite ein
.activatedHtml	Enthält Grafikstile, die angesteuerte Optionen dieses Menüs haben sollen >> Nutzungsbeispiel: myMenu.activatedHtml.style.backgroundColor = "gray";
.lastOptionId	Enthält ID der zuletzt generierten Option
.enableKeyControls([truthValue])	Ermöglicht Steuerung mit Pfeiltasten >> Falls 'true' angegeben (optional), wird Buchstabensteuerung für alle tieferen Submenüs aktiviert, sonst nur für das Einzelmenü ohne Submenüs
.disableKeyControls([truthValue])	Schaltet Steuerbarkeit mit Pfeiltasten ab >> Falls 'true' angegeben (optional), wird Buchstabensteuerung für alle tieferen Submenüs deaktiviert, sonst nur für das Einzelmenü ohne Submenüs
.enableCharControls()	Ermöglicht Steuerung mit Buchstabentasten (auch Zahlen und Zeichen; nur fürs ganze Menü statt einzelner Submenüs)
.disableCharControls()	Schaltet Steuerbarkeit mit Buchstabentasten ab (nur fürs ganze Menü statt einzelner Submenüs)
.activateMouseControls()	Ermöglicht Steuerung per Maus(zurzeit nur als interne Methode genutzt)
.getRootMenu()	Gibt das Hauptmenü, das kein Submenü ist, zurück.
.getActiveOption()	Gibt die derzeit angesteuerte Option dieses Menüs an.
.listOptions()	Gibt Liste aller Optionen dieses Menüs und seiner Submenüs zurück >> Typ: String
.display([x,y [,positionVal]])	Zeigt das Menü an

	>> 'positionVal' muss als "fixed" angegeben werden, wenn Scrollings das Menü nicht mitverschieben sollen
.hide([message])	Stellt das Schließen des Menüs und seiner Submenüs durch Verbergen dar.
	>> wird der String "noDeactivate" übergeben, wird das menü ohne Deaktivierung der evtl. noch aktiven Option verborgen
.clone()	Erzeugt einen Clone des Menüs und gibt diesen zurück.
.remove()	Löscht das Menü und seine Submenüs (entfernt auch Eventhandler & verborgene HTML-Elemente vollständig)
.addShadow(opc, ab, cd)	Statt Menü incl. aller Submenüs mit Schatten aus (analog id(element).addShadow())
.addOption(label/pathString)	Fügt dem Menü eine Option hinzu
	>> Auch in Untermenüs hinein, wenn Argument vom Format "Optionsname|Optionsname|..."
	>> Wenn im 'pathString' vorletztes Label zu einer submenülosen Option gehört, wird Submenü automatisch erzeugt
	>> Bei großen komplexen Menüs, die per Schleife erzeugt werden o.ä.: vor Aufruf >> [menü].addOption.noDOM = true << (schneller, da DOM-Anlegung nicht sofort, sondern erst beim Displaying)
.addOptions(label1 [, label2, label3,...])	Fügt dem Menü (auch mehrere) Optionen (Typ Option-Objekt) hinzu
	>> Zugriff danach: <ContextMenu>.options[<Index/Aufschrift>]
	>> Kann auch Arrays abarbeiten:
	var arr = new Array("Laden", "Speichern", "Verlassen");
	myMenu.addOptions(arr);
	>> Gibt derzeit eine Option zurück - sinnvoll nur, wenn mit dieser Fkt. nur ein einziges Label übergeben wurde (Achtung: addOption (ohne s) baut hierauf auf!)
	(schneller, da DOM-Anlegung nicht sofort, sondern erst beim Displaying)
	>> Bei großen komplexen Menüs, die per Schleife erzeugt werden o.ä.: vor Aufruf >> [menü].addOptions.noDOM = true << (schneller, da DOM-Anlegung nicht sofort, sondern erst beim Displaying)
.insertOption(label, i)	Erzeugt eine neue Option und fügt sie an der Stelle i ein
	>> die anderen Optionen ab dieser Stelle rücken um eine Stelle nach unten
.deleteOption(index/label)	Löscht eine Option (grafisch & intern)
	>> Submenü(s) der Option werden n i c h t mitgelöscht
.getOption(pathString)	Ermöglicht Zugriff auf eine Option per Übergabe eines Strings vom Format "Optionsname|Optionsname|..."
.getOptionByArray(pathArray)	Ähnlich .getOption()
.bindToClipboard(text)	Bestimmt einen Text, der beim Klick auf die Option in die Zwischenablage gespeichert wird
.knows(object)	Gibt an, ob ein Objekt oder HTML-Element zum Menü gehört
	>> Überprüfung derzeit nur beim Wurzelmenü beginnend, nicht ab einem Submenü
<Option>.label	Enthält Aufschrift der jeweiligen Menüoption
	>> Typ: String
<Option>.id	Enthält Kennummer der Option als fortlaufende Zahl (erstgenerierte Option = 0)
<Option>.parentMenu	Enthält das die Option umgebende Menü
	>> Typ: ContextMenu-Objekt
<Option>.precOption	Enthält Vor-Nachbarn der Option
	>> Typ: Option-Objekt
<Option>.nextOption	Enthält Nach-Nachbarn der Option
	>> Typ: Option-Objekt
<Option>.isActive	Enthält Wahrheitszahlwert (0/1) darüber, ob die Option momentan angesteuert ist.
<Option>.isDisabled	Enthält Wahrheitszahlwert (0/1) darüber, ob die Option unansteuerbar ist
<Option>.isTickable	Enthält Wahrheitszahlwert (0/1) darüber, ob die Option "ankreuzbar" ist
	>> Ankreuzfeld dann für CSS unter ".tickBox" verfügbar
<Option>.isTicked	Enthält Wahrheitszahlwert (0/1) darüber, ob die Option "angekreuzt" ist
<Option>.submenu	Enthält Submenü, das sich aus der Option aufklappt
	>> Typ: ContextMenu-Objekt
<Option>.html	Enthält die Option als HTML-Element.
	>> Darum auch individuelle Grafikstile für eine einzige Option möglich.
	>> Nutzungsbsp.: myMenu.options[<Index/Aufschrift>].html.style.backgroundColor = "green";
<Option>.addSubmenu()	Fügt der Option ein Submenü (Typ ContextMenu-Objekt) hinzu
	>> Zugriff danach: <ContextMenu>.options[<Index/Aufschrift>].submenu
	>> Gibt das neue Submenu auch zurück
<Option>.openSubmenu()	Öffnet das Untermenü dieser Option ohne Interaktivität des Nutzers
<Option>.activate()	Aktiviert diejeweilige Option, als ob sie angesteuert würde.
<Option>.deactivate()	Deaktiviert diejeweilige Option.
<Option>.disable()	Macht die Option unaktiviertbar
	>> evtl. gut für Zwischentitel u.a.
<Option>.getIndex()	Gibt den Index zurück, an dem die Option im Optionen-Array des umgebenden (Sub-)Menüs steht
<Option>.makeTickable()	Macht den Menüpunkt mit Häkchen ankreuzbar
<Option>.tick()	Kreuzt den Menüpunkt mit Häkchen an
<Option>.untick()	Macht Häkchensetzung rückgängig
<Option>.separate([color1 [, color2]])	Trennt die Option von den vorhergehenden mit einem horizontalen Trennstrich
	>> color1, color2:Optionale Farben für oberen & unteren Strichrand (gut für Reliefeffekte)
	>> Trennstrich-Stil über Klassenname "separator" nachträglich bearbeitbar, z.B.: _(myMenu.options["..."].html).cl("separator").style.borderTop = "...";
<Option>.separateUp([color1 [, color2]])	Wie <Option>.separate(), jedoch wird der Strich unterhalb statt oberhalb des Optionsinhalts eingesetzt
ContextMenu.knows(obj)	Gibt an, ob sich ein HTML-Element in einem Kontextmenü befindet, aber auch, ob eine Variable eine Instanz von ContextMenu oder einer Option davon ist..
useSubmenus()	Ermöglicht in HTML-ul-Navigationslisten aufklappbare Submenüs
	>> Sollte am Fuße des BODYs eingesetzt werden (d.h. nachdem alle HTML-Elemente der Submenu-Klasse geladen sind)
	>> Aufzuklappender Menüpunkt muss Klassenbezeichnung "submenu" haben
	>> Tag des aufklappenden Menüpunkts muss Submenüpunkte umschließen!
	>> Schema der strukturellen Voraussetzung:

```
<ul>
    <li class="submenu"><div><a>Überschrift</a>
        <ul>
            <li><div><a href="">Punkt 1</a></div></li>
            <li><div><a href="">Punkt 2</a></div></li>
            <li><div><a href="">Punkt ...</a></div></li>
            ...
        </ul>
    </div></li>
</ul>
```

enterSubmenu(node)	Arbeitsfunktion von useSubmenus( ); klappt Submenü auf
exitSubmenu(node)	Arbeitsfunktion von useSubmenus( ); schließt Submenü

Events:

Event.add(origHName, extendingF [, extenderName])	Erweitert einen (Event-)Handler oder eine Variable, in der eine Fkt. gespeichert ist, um eine andere Funktion, statt ihn/sie zu ersetzen >> origHName: name des zu erweiternden Handlers (als String übergeben! Sollte höchstens ' statt " beinhalten) >> extendingF: Funktion, um die der Handler erweitert werden soll >> extenderName: Beliebiger Name der hinzugekommenen Funktion, die anhand dessen in 'Event.drop()' gezielt deaktiviert werden kann >> Sinnvoll bei Eventhandlern, die nicht überschrieben werden sollen >> 'this' zeigt nicht auf das Objekt, dem der Event zugewiesen wurde, dafür aber 'that', wenn es als formaler Parameter im Fkt.-Kopf angegeben war. >> Bei IE statt "return false/true": window.event.returnValue = false; >> Anwendungsbeispiel: document.onclick = function(e){ alert("Dies ist die erste Reaktion."); }; Event.add('document.onclick', function(){ alert("Und dies ist die erweiterte Reaktion."); });
Event.drop(hName, extenderName)	Deaktiviert eine mit Event.add() an einen Handler angefügte Funktion >> extenderName: in Event.add() der Funktion gegebener Name >> Falls in Event.add() kein Name vergeben wurde, muss hier Ziffer angegeben werden (i.S.v.: x-te hinzugefügte Funktion) >> Alternative Form: Event.drop(extenderName)
Event.set(origHName, fnc, extenderName)	Ersetzt eine der Erweiterungen eines (Event-)Handlers oder einer Variable, in der eine Fkt. gespeichert ist, durch eine andere Funktion >> Alternative Form: Event.set(fnc, extenderName) >> Falls der Eventhandler oder die benannte erweiternde Funktion nicht existiert, verhält sich die Funktion wie Event.add().
Event.exists(hName, extName)	Gibt an, ob eine bestimmte Handlererweiterung schon existiert
Event.guard(hName)	Überwacht einen Eventhandler, um eine Warnung auszugeben, falls der Programmierer ihn überschreibt, statt Event.add() zu nutzen.
Event.add2(obj, evType, fn, useCapture)	Alternative zu Event.add() >> Ist etwas schneller, jedoch noch nicht kompatibel zu Event.drop() & Event.exists(). >> Vorteil ggb. als Event.add(): 'this' zeigt auf 'obj' >> 'evType': Handlername ohne 'on'-Präfix >> Verarbeitet auch '(on)attrchange'

getTarget()	Gibt das dem behandelten Element untergeordnete Element zurück, an dem das Ereignois geschehen ist.
	>> Anwendungsbeispiel:
	document.onmouseover = function(){
	ooo(getTarget().innerHTML);
	};
	>> Mit Event.add muss der Ereignis-Parameter in der Umgebungsfunktion angegeben werden:
	Event.add("document.onmouseover", function(e){
	ooo(getTarget().innerHTML);
	});

Eingabegeräte & Interaktion:

---

window.onF6	Reagiert auf Drücken der F6 Taste, auch wenn IFrame-Inhalt fokussiert ist >> Im Firefox muss bei Benutzung dieses Handlers doppelt in die Location Bar geklickt werden, um diese benutzen zu können.
window.onaddressbarclick	Reagiert auf Hineinklick in addressbar (etwas unzuverlässig, größere zeitl. Abstände zwischendurch und vorheriger Wiederreinklick ins Dokument nötig) >> Reagiert in FF momentan auch auf Mausrädchenklick auf Home-Button >> Ideen zur Präzisierung des Eventhandlers im IE: Ausnutzen von: - wenn Dokumenträndер links&rechts genau wie im Maximiertmodus plaziert: Browserfenster höchstwahrscheinlich im Maximiertmodus => event.screenX & event.screenY nutzbar (statt clientX/client>) - Zusätzliche Toolbars haben vermutlich einheitliche Höhen => Aus Höhe des oberen Dokumentrands kann im Max.-modus auf Anzahl & Lage der Toolbars geschlossen werden - Browserfenster per Maus nicht beliebig nach oben verschiebbar => Ab bestimmter Dokumentrandhöhe wird es feststellbar, dass keine Toolbars installiert sind und addressbarlage wird sehr genau bestimmbar - Nutzen von Cookies zum Speichern schon festgestellter Browsergestaltinformationen
window.onmousetabchange	Reagiert auf Mausgesteuerten Wechsel des Tabs >> Blockiert im FF momentan die addresszeile, so dass zur Blockade-Aufhebung schneller Mehrfachklick nötig ist
window.onback	Reagiert auf Betätigung des Zurück-Knopfes >> Für IFrame-Navigation gedacht - muss evtl. für Ankerlink-Navigation getestet/optimiert werden
window.onforward	Reagiert auf Betätigung des Vorwärts-Knopfes >> Für IFrame-Navigation gedacht - muss evtl. für Ankerlink-Navigation getestet/optimiert werden
window.onhistorychange	Reagiert z.B. auf Wechsel der Seite in einem IFrame
document.onenter	Dokumentweiter Eventhandler für die Entertaste
document.onuparrow	Dokumentweiter Eventhandler für die Pfeil-nach-oben-Taste
document.ondownarrow	Dokumentweiter Eventhandler für die Pfeil-nach-unten-Taste
document.onrightarrow	Dokumentweiter Eventhandler für die Pfeil-nach-rechts-Taste
document.onleftarrow	Dokumentweiter Eventhandler für die Pfeil-nach-links-Taste
document.onmouseleftall	Ergänzung/Alternative zu 'onmouseout', da letzteres auch feuert, wenn Zeiger nur über Textfelder oder IFrames gefahren
document.onmousestop	Feuert, wenn Maus angehalten wird
usedKey1(k [, a [, b]])	Ermittelt, ob eine bzw. welche Taste gedrückt wurde >> ohne a & b: Gibt gedrückten Code der betätigten Taste zurück >> mit a: Gibt zurück, OB Tastencode a gedrückt worden ist >> mit a & b: Gibt zurück, OB die Tastencodes a und b gleichzeitig gedrückt worden sind >> Einzubauen in: xyz.onkeydown = function(k){ ... }; >> Var. "k" wichtig!

>> auch mit onkeypress, onkeyup usw.

usedKey([a [, b [, c[, d]]]]) Wie usedKey1(), jedoch mit bis zu 4 gleichzeitigen Tastendrücken und ohne k-Parameter
>> Einzubauen in: xyz.onkeydown = function(k){ ... };
>> k-Parameter noch in umgb. Funktion wichtig
>> Experimentalstadium
>> nutzt deprecated-Eigenschaft 'caller'
>> in FF auch für Maustasten

getKeyName(keycode) Gibt einen Keycode als Tastenbezeichnung zurück
>> Empfohlene Anwendung: '... getKeyName(e.keyCode) ...'

getChar(keycode) Gibt das gedrückte Z e i c h e n an
>> Falls es keine Zeichentaste war (sondern z.B. Strg, Shift, o.ä.), wird "" zurückgegeben

usedButton() Gibt zurück, welche Maustaste geklickt wurde
>> Einzubauen z.B. in: xyz.onmousedown = function(k){ ... };
>> k-Parameter noch in umgb. Funktion wichtig
>> nutzt deprecated-Eigenschaft 'caller'
>> funktioniert nicht im Opera
>> mgl. Rückgabewerte: "left", "middle", "right"

onAlt(key, func) Legt Reaktion für einen Alt-Shortcut fest
>> Nur für echte Buchstaben & Ziffern; im IE auch noch Space-Taste möglich (" ")
>> In FF muss Shift-Taste zusätzlich benutzt werden

getPressedKeysNum() Gibt die Anzahl der Tasten zurück, die momentan gleichzeitig gedrückt werden.

new MouseListener() Überwacht die Maus
>> Schon verfügbar in der Bibliotheks-Variable 'Mouse'
Mouse.getX() Gibt die aktuelle Maus-X-Position zurück (gemessen an Rändern des Anzeigebereichs)
Mouse.getY() Gibt die aktuelle Maus-Y-Position zurück (gemessen an Rändern des Anzeigebereichs)
Mouse.getScrX() Gibt die aktuelle Maus-X-Position zurück (gemessen an Bildschirmrändern)
Mouse.getScrY() Gibt die aktuelle Maus-Y-Position zurück (gemessen an Bildschirmrändern)
Mouse.isAbove(element) Gibt zurück, ob der Mauszeiger sich über dem Element befindet
>> vorher: Mouse.isIn
Mouse.getElement() Gibt Element zurück, das sich unterhalb des Mauszeigers befindet
Mouse.catchAll() Um auch über IFrames mit *domain-externen* Seiten die richtigen Mauskoordinaten wiedergeben zu können
>> Nur ohne Scrolling
Mouse.onrightclick Eventhandler. Feuert, wenn rechte Maustaste gedrückt
Mouse.onmiddleclick Eventhandler. Feuert, wenn mittlere Maustaste / Rädchen gedrückt
Mouse.onleftclick Eventhandler. Feuert, wenn linke Maustaste gedrückt
Mouse.onoveriframe Eventhandler. Feuert, wenn Maus sich über einem beliebigen IFrame (auch mit externen Inhalten) befindet.
getClickTarget(e) Gibt Knoten, auf den geklickt wurde, zurück
>> Anwendung:

```
document.onclick = function(e){
    ... getClickTarget(e) ... ;
};
```

Ajax:

Ajax.getObject(key) Gibt XMLHttpRequest-Objekt browserübergreifend zurück
>> Darf aus Sicherheitsgründen nur innerhalb einer Funktion benutzt werden; zurückgegebenes Objekt darf nicht global oder als public-Member eines globalen Objekts deklariert werden.

Ajax.getKey() Gibt den Schlüssel an, der für Ajax-Operationen als 'key'-Parameter von 'Ajax.getObject()', 'XMLHttpRequest()' und ActiveXObject() einzusetzen ist
>> Darf nicht außerhalb einer Funktion eingesetzt werden.
>> Rückgabewert sollte nur als lokale Variable oder Funktionsparameter gespeichert werden, nie global oder als public-Eigenschaft eines global erreichbaren Objekts

FileSystem Möglichkeit zur Manipulation von Dateien (PHP-Server und "ajaxFileSystem.php" erforderlich) ein
>> Aus Sicherheitsgründen dürfen'FileSystem' und seine Methoden nur innerhalb einer Funktion benutzt werden, und zwar ein- und derselben (besonders '.getKey()')!
>> Abkürzung 'fs' statt FileSystem möglich
>> Nutzung der Funktionen z.B. so: FileSystem.makeDir(...) oder fs.makeDir()
>> FileSystem-Funktion darf nicht unbenutzt in der Bibliothek 'herumliegen' (sonst löschen!)
>> Derzeit muss 'ajaxFileSystem.php' im Verzeichnis jeder zugreifenden HTML-Datei liegen - bite noch zentralisieren, falls möglich
>> Anwendungsbeispiel:

```
function beliebigeUmgebendeFunktion(){
    var key = fs.getKey(); // Wichtig!
    fs.write("test.txt", "Kleiner Inhalt", key).onready = function(){
        ooo(fs.getResponse());
    };
}
```

.[fnc].onready Eventhandler, der feuert, wenn das System bereit für weitere Aufgaben ist
>> Anwendung z.B.: fs.makeDir("xyz.dat").onready = function(){ ... };
.getKey() Gibt den Schlüssel an der in den 'key'-Parameter der anderen Funktionen eingesetzt weren kann
>> Muss aus Sicherheitsgründen sofort nach Starten der Website oder Aufbau des Bodys eingesetzt werden, sonst kann der Schlüssel gestohlen werden.
>> Kann nur ein einziges Mal benutzt werden!
>> Nur innerhalb einer Funktion.
>> Rückgabewert sollte nur als lokale Variable oder Funktionsparameter gespeichert werden, nie global oder als public-Eigenschaft eines global erreichbaren Objekts
.makeDir(dirname, key) Erzeugt ein Verzeichnis
.deleteDir(dirname, key) Löscht ein Verzeichnis (auch wenn es Inhalte hat)
.emptyDir(dirname, key) Leert ein Verzeichnis
.makeFile(filename,data, key) Erzeugt neue Datei mit beliebigem Dateinamen (überschreibt keine vorhandenen gleichnamigen Dateien)
.appendTo(filename,data, key) Schreibt in vorhandene Datei (Text wird angehängt)

.write(filename,data, key)	Überschreibt vorhandene Datei
	>> Falls Datei nicht vorhanden, wird automatisch neue angelegt
.rename(filename,newName, key)	Benennt Datei oder Verzeichnis um
	>> Bei Nutzung unterschiedlicher Pfadangaben in den beiden Parametern wirkt die Funktion verschiebend.
	>> Darum: Für bloßes Umbenennen sollte in beiden Parametern die selbe Pfadangabe genannt werden.
.deleteFile(filename, key)	Löscht Datei
.execute(phpcode, key)	Führt PHP-Code aus
	>> Voraussetzung: Auf ajaxFileystem.php wir d nicht domainüberschreitend zugegriffen
.search(filename)	Fragt nach Existenz einer Datei
	>> Abfrage des Ergebnisses in Code, der im Eventhandler "onready" gespeichert wird
	>> Abfrage per .exists()
	>> Auch für Verzeichnis-Existenzprüfung geeignet (dann muss [?] Verzeichnis mit abschließendem Schrägstrich in 'filename' gekennzeichnet werden)
.exists()	Liefert endgültige Antwort auf .search()-Existenzfrage als 0-/1-Wert
	>> Wird im Code eingesetzt, der im Eventhandler "onready" gespeichert wird
.request(fName[, urlPars[, key]])	Fordert Inhalt der Datei an
	>> Abfrage des Ergebnisses in Code, der im Eventhandler "onready" gespeichert wird
	>> Abfrage per getResponse() oder getXMLResponse()
	>> urlPars: Post-Daten im URL-Parameter-Notation (z.B. 'x=38&s=hallo'). Bei nichtvorhandenen URL-Parametern 'null' übergeben, wenn nötig.
.knock(fName, urlPars, key)	"Meldet" sich bei einer Datei lediglich und erwartet keinen Response.
.requestFileDate(filename)	Fragt nach letztem Änderungsdatum einer Datei
	>> Abfrage des Ergebnisses in Code, der im Eventhandler "onready" gespeichert wird
	>> Abfrage per getResponse()
.getResponse()	Liefert Ergebnisse der folgenden Anfragen: .request(), .requestFileDate()
.getXMLResponse(tagName)	Liefert Ergebnisse der folgenden Anfragen als Inhalt des gewünschten XML-tags: .request()
	>> nur für Dateien mit .xml-Endung
.error()	Gibt 1 bzw. -wahr- zurück, falls bei einer der Dateioperationen ein Fehler aufgetreten ist
	>> Schnellere Alternative zu .exists()

new AjaxEventSystem()	Zum Austauschen von Events zwischen verschiedenen Clients
.onevent	Eventhandler, durch den festgestellt wird, ob ein Ereignis geschehen ist
.getEventType()	Gibt Typ des Events zurück
	>> Wird in der reagierenden Funktion eingesetzt, die in "onevent" definiert wurde
.getEventContent()	Gibt Inhalt des Events zurück
	>> Wird in der reagierenden Funktion eingesetzt, die in "onevent" definiert wurde
.getEventSource()	Gibt Quelle des Events zurück
	>> Wird in der reagierenden Funktion eingesetzt, die in "onevent" definiert wurde
.fire(content[,type[,source]])	Erzeugt Event
	>> Wichtig wg. Firefox-bug: Falls diese Methode in Intervall angewendet werden soll, dann Intervall mit setTimeout simulieren!
	>> Programmierhinweis: Bei mehreren Clients können Kollisionen auftauchen. Mgl. Abhilfe: in ajaxFileSystem.php beim makeFile() "Mülldatei" erzeugen, wenn Datei schon vorhanden. EventSystem prüft nach Feuern auf Vorhandensein der Mülldatei und feuert in dem Fall nochmal
new ServerInfo()	Zum Erfahren von Infos vom und über den Server
	>> Code sollte stets im onready Eventhandler fortgesetzt werden
.requestInfo(headerName)	Bittet um eine Info (Name eines typischen Headers ist anzugeben, z.B. "Date" oder "Server")
	>> Ergebnisse auslesbar mit .getInfo() in einem onready-Block
.getInfo()	Gibt Ergebnis des Requests an
.onready	Eventhandler, der feuert, wenn das System bereit für weitere Aufgaben ist

Textcursor & -markierung:

(für Weiterprogrammierung evtl. nützlich: http://the-stickman.com/web-development/javascript/finding-selection-cursor-position-in-a-textarea-in-internet-explorer/)

Cursor.paste(str1 [,str2])	Fügt Text an Curor-Position ein (auch für IE!)
	>> Falls zweiter String angegeben und Textstück markiert: Umschließungswirkung ("abcEINSdefgZWEIhij")
	>> Wohl nicht für andere außer IE & FF
Cursor.getPos(["start"|"end"|0|1])	Gibt aktuelle Textcursor-Position an
	>> Ohne Parameter / mit Param. "start" / mit Param. 0: Gibt Position des Markierungsanfangs an
	>> Mit Parm. "end" / mit Param. 1: Gibt Position des Markierungsendes an
	>> Nur für Eingabefelder (Input & Textareas) - falls kein Eingabefeld aktuell fokussiert, wird 'null' zurückgegeben
Cursor.to(pos [,pos2])	Setzt Textcursor an gewünschte Position
	>> Markierungswirkung bei Angabe von 'pos2'
	>> Nur für Eingabefelder (z.B. Textareas)
Cursor.toEnd()	Setzt Textcursor an das Ende
Cursor.unselect()	Demarkiert Text, egal ob in Texteingabefeld oder im sonstigen Dokument
Cursor.save()	Speichert die Markierung (nicht für Eingabefelder, sondern nur sonstiges Dokument)
Cursor.restore()	Stellt Markierung, falls z.B. durch Klick woandershin verlorengegangen, wieder her (nicht für Eingabefelder, sondern nur sonstiges Dokument)

Cookies:

Cookie.set(label, value [,duration])	Setzt einen Cookie-Wert mit Label, anhand dessen er später wieder gelesen werden kann
	>> duration: Gültigkeitsdauer - wenn nicht gesetzt, 10 Jahre
Cookie.get(label)	Liest den Cookie-Wert eines Labels

Cookie.drop(label)	Löscht ein Cookie mit dem angegebenen Label
Cookie.big.set(label, value)	Setzt ein Supercookie (größere Speicherkapazität, höhere Persistenz) >> Speicherkapazitäten: IE 6 & 7 ca. 0,9 MB; IE 8 ca. 10 MB; FF ca. 5 MB >> Speichert auch Arrays, die von 'Cookie.big.get()' auch als Arrays zurückgegeben werden (es werden aber nur S t r i n g-Arrayelemente zurückgegeben!)
Cookie.big.get(label)	Liest den Wert eines Supercookies >> Falls Label nicht existiert, wird 'null' zurückgegeben
Cookie.big.drop([label])	Löscht eine Cookie-Variable komplett (so dass eine Abfrage per Cookie.big.get() 'null' ergibt) >> Ohne 'label'-Angabe: Alle Cookie-Variablen werden gelöscht

User-Authentication:

Auth.start()	Startet den LogIn-Dialog bzw. -Vorgang (z.B. für Buttons mit der Aufschrift "Log In") >> Ein Dialog wird eröffnet, der gleichzeitig als Neuan- als auch als Rückmeldung benutzt werden kann. >> Ein Cookie namens "username" wird gesetzt >> Für den User wird ein verzeichnis 'users/<Benutzername>' eingerichtet >> Fortsetzung des Codes sollte im onready-Handler geschehen: Auth.start().onready = function(){ ... } >> ID des Username-Eingabefelds: 'usernameInput'; Button-ID: 'usernameConfirmButton' >> ID der Passwort-Eingabefelder: 'passwordInput'; Button-ID: 'pwConfirmButton'
Auth.isKnownClient()	Gibt an, ob der aktuelle Benutzer registriert ist (auch wenn er technisch noch nicht komplett eingeloggt worden ist) >> Ist derzeit nichts weiter als die Überprüfung des Existenz des Cookies "username"
Auth.logOut()	Meldet den User ab. >> Cookie "username" wird gelöscht
Auth.addPassword()	Fordert den User zum Ausdenken und Eingeben eines Passwort auf und schützt den Usernamen damit, so dass beim nächsten Einloggen immer eine Passworteingabeaufforderung erscheint. >> Für den User wird ein verzeichnis 'users/<Benutzername>/<Passwort>' eingerichtet >> Nur sinnvoll, wenn User gerade passwortlos eingeloggt >> Fortsetzung des Codes sollte im onready-Handler geschehen: Auth.addPassword().onready = function(){ ... }
Auth.protect(fnc, resumeAfterLogIn)	Schützt einen bestimmten Code-Abschnitt, so dass er nur im eingeloggten Zustand ausgeführt wird. >> 'resumeAfterLogin': Wahrheitswert - true falls nach dem eventuellen LogIn-Dialog 'fnc' ausgeführt werden soll, sonst passiert nach einem solchen gesonderten Einloggen nichts.
Auth.read(filename)	Liest eine Datei im (evtl. passwortgeschützten) Verzeichnis des Users >> Fortsetzung des Codes sollte im onready-Handler geschehen, z.B.: Auth.read(filename).onready = function(){ alert(Auth.data);} >> Gelesene Daten befinden sich in Auth.data >> Bei nicht gefundener Datei enthält Auth.data den Wert "FileSystemError" >> Der Aufruf dieser Methode sollte immer auf Auslösung des onlogin-Handlers zurückgehen (sonst Konflikte).
Auth.write(filename, data)	Schreibt eine Datei im (evtl. passwortgeschützten) Verzeichnis des Users >> Gleichnamige evtl. vorhandene Datei wird überschrieben >> Fortsetzung des Codes sollte im onready-Handler geschehen: Auth.write(filename, data).onready = function(){ ... } >> Der Aufruf dieser Methode sollte immer auf Auslösung des onlogin-Handlers zurückgehen (sonst Konflikte).
Auth.erase(filename)	Löscht eine Datei im (evtl. passwortgeschützten) Verzeichnis des Users >> Fortsetzung des Codes sollte im onready-Handler geschehen: Auth.write(filename, data).onready = function(){ ... }
Auth.contents	Array, der die Dialogtexte-/Inhalte enthält (als HTML-Code) >> Auth.contents[0]: Titel des LogIn-Fensters >> Auth.contents[1]: Inhalt vor dem Username-Eingabefeld >> Auth.contents[2]: Inhalt nach dem Username-Eingabefeld >> Auth.contents[3]: Fehlermeldung bei falschem Eingabeformat >> Auth.contents[5]: Titel des Fensters mit der verpflichtenden Passwort-Eingabeaufforderung; >> Auth.contents[6]: Inhalt vor dem Passwort-Eingabefeld >> Auth.contents[7]: Inhalt nach dem Passwort-Eingabefeld >> Auth.contents[8]: Fehlermeldung bei falschem Passwort >> Auth.contents[9]: Titel des Fensters mit der freiwilligen Passwort-Eingabeaufforderung >> Auth.contents[10]: Inhalt vor dem Passwort-Eingabefeld >> Auth.contents[11]: Inhalt nach dem Passwort-Eingabefeld >> Auth.contents[12]: Fehlermeldung bei falschem Passwort-Format >> Die Zeichenfolge '<Auth.username>' (mit Spitzklammern) im String wird automatisch mit dem aktuellen Usernamen ergänzt
Auth.data	Enthält durch Auth.read() gelesene Daten >> Bei nicht gefundener Datei enthält Auth.data den Wert "FileSystemError"
Auth.username	Beinhaltet den Benutzernamen des aktuellen Nutzers
Auth.hasPassword	Beinhaltet Wahrheitswert darüber, ob zu 'Auth.username' ein Passwort gehört
Auth.isFirstLogInEver	Beinhaltet Wahrheitswert darüber, ob dieser Login gleichzeitig eine Erstregistrierung ist >> Sollte im onlogin-Handler eingesetzt werden.
Auth.isMailBased	'true', wenn als Usernamen nur E-mail-Adressen akzeptiert werden sollen, sonst 'false'
Auth.logInMB	Beinhaltet die Message-Box aus Auth.start();
Auth.onlogin	Eventhandler, der feuert, wenn ein Log-In erfolgreich durchgeführt worden ist (im Fall eines vorhandenen Passwortes nach dem Akzeptieren desselben)
Auth.onlogout	Eventhandler, der im Log-Out-Fall feuert

Arbeitsfunktionen von 'Auth':	checkProtectionExistence() (+ .onready, .exists)
	checkUsername(); (+ .onready, .isNewUsername)
	hasRightFormat()
	logIn() (+ .onready)
	register() (+ .onready)
	requirePasswordInput() (+ .onready, .getPassword())
	setUnloadNotifier()
	writePwSubmitter()

URL-Parameter:

URLParams.get(varName[, url] [, sign])	Liest URL-Parameter aus und gibt den Wert zurück
	>> Ohne die Angabe einer URL wird die aktuelle URL geprüft
	>> Wenn URL überhaupt keine Parameter enthält, wird 'null' zurückgegeben
	>> Mögliche Verwendungsformate:
	val = urlParameter("q"); // Sucht nach q-Wert in Browserfenster-URL ab ?-Zeichen
	val = urlParameter("q", "#"); // Sucht nach q-Wert in Browserfenster-URL ab #-Zeichen
	val = urlParameter("q", "http://www.xyz.de/index.php?q=100"); // Sucht nach q-Wert in beliebiger URL ab ?-Zeichen
	val = urlParameter("q", "http://www.xyz.de/index.php#q=100", "#"); // Sucht nach q-Wert in beliebiger URL ab #-Zeichen
URLParams.set(varName, val, url)	Ändert den Wert eines vorhandenen URL-Parameters oder setzt nicht vorhandenen Parameter neu und gibt die URL dann zurück
	>> Anders als bei get()-Methode momentan nur für "?"-Zeichen und übergebener URL
URLParams.drop(varName, url)	Löscht einen Parameterterm aus übergebener URL
	>> Anders als bei get()-Methode momentan nur für "?"-Zeichen und übergebener URL
URLParams.extend(varName, val, url)	Erweitert den Wert eines vorhandenen URL-Parameters per Kommatrennung (aus x=3 wird z.B. x=3,fx) und gibt URL dann zurück
	>> Anders als bei get()-Methode momentan nur für "?"-Zeichen und übergebener URL
URLParams.retract(varName, val, url)	Gegenstück zu extend()-Methode, läscht aus Kommagetrennter Werte-Kette den ausgesuchten Wert und lässt Parameter-Term-Rest stehen

Multimedia & Animation:

new DrawingKit()	Stellt Werkzeug(e) zum Zeichnen zur Verfügung
.onready	Eventhandler, der feuert, wenn nach der Objekterzeugung das DrawingKit bereit ist (Anwendung obligatorisch)
.setViewBox(x1,y1,w,h)	Beschränkt Sichtbarkeit der Zeichnungen auf bestimmten Bereich
.setColor(color)	Stellt die Farbe ein, mit der gezeichnet werden soll
.setFillColor(color)	Stellt Füllfarbe ein; Durchsichtigkeit mit Argument "none" (Voreinstellung)
.setStrokeWidth(width)	Stellt Stiftdicke ein; Voreinstellung 1
.drawLine(x1,y1,x2,y2)	Zeichnet eine Linie
.drawCurve(x1,y1,x2,y2,x3,y3,x4,y4)	Zeichnet eine Kurve (2. & 3. Punkt sind Stützpunkte)
.drawEllipse(x,y,rx,ry)	Zeichnet eine Ellipse
.drawCircle(x,y,r)	Zeichnet einen Kreis
.drawRect(x,y,w,h,rx,ry)	Zeichnet Rechteck; bei Angabe von rx & ry mit abgerundeten Ecken außer im IE (noch)
.drawPolyline(points)	Zeichnet Kette von Linien (braucht Array als Argument (points)!)
.hideCanvas()	Verbirgt Zeichnungen (und ermöglicht Maus-Interaktion mit dahinterliegenden Elementen)
.showCanvas()	Macht Zeichnungen (wieder) sichtbar
moveText(speed, text, x1, y1, x2, y2, id)	Bewege Text von A(x1,y1) nach B(x2,y2)
movePicture(speed, url, x1, y1, x2, y2, id)	Bewege Bild von A(x1,y1) nach B(x2,y2)
slideshow_addPic(picFile)	Füge Bilddatei zu Slideshow hinzu
slideshow_visualize(breite, hoehe)	Setze Slideshow-Monitor nach slideshow_addPic()-Liste ins Fenster (Parameter=Bilderabmessungen)
slideshow_switch()	Schalte zum nächsten Bild
	>> sollte nicht zu kurz nach slideshow_visualize eingesetzt werden
slideshow_jump(pos)	Bewege Bilderkette nach links (Parameter=Strecke in Pixeln)
	>> Parameterempfehlung: Bilderbreite + 25
playSound(soundFilename)	Spielt eine Tondatei ab (z.B. mp3)
	>> Nutzt MediaPlayer-PlugIn, darum nur für Windows
	>> Gut für Firefox
	>> Voller (nicht relativer) Pfad muss angegeben werden (wg. FF)
	>> Sollte erst nach Aufbau des BODY-Elements eingesetzt werden
preloadSound(soundFilename)	Lädt Sounddatei vor
	>> komplette URL im Sounddateinamen nötig!
playPreloadedSound(volume)	Spielt vorgeladene Sounddatei ab
	>> volume: Wert von 0 bis 10 (ganz leise bis ganz laut; auch Kommazahlen mgl.)

Performance- & Speed-Tests:

Performance.rateSpeed()	Bewertet die Geschwindigkeit des PCs bzw. der VM mt einer von 5 Bewertungen: "very slow", "slow", "middle", "fast", "very fast" (= Rückgabewerte)
Performance.rateMemory()	Bewertet die Speicherperformance mt einer von 5 Bewertungen von "very low", "low", "middle", "high", "very high" (= Rückgabewerte)
Performance.rateGraphics([col])	Bewertet die Grafikperformance mit einer von 5 Bewertungen: "very slow", "slow", "middle", "fast", "very fast"
	>> Anwendung: Performance.rateGraphics().onready = function(){

```
                                        alert(this.graphicSpeed);
                                    };
```

>> col: Farbe der Testbox (optional; Standard "white")

.graphicSpeed	Beinhaltet Grafikgeschwindigkeitsbewertung (in onready-Funktion zu benutzen)
new SpeedTester()	Zum Herausfinden der Verbindungsgeschwindigkeit (derzeit nur Downstream)
	>> Ist auch von Servergeschwindigkeit abhängig
.requestSpeed()	Fragt nach aktuell möglichen kbps
	>> Programmrest muss nach Instantiierung im Code fortgesetzt werden, der im onready-Eventhandler gespeichert wird
.getSpeed()	Gibt die mit .requestSpeed() erfragten möglichen kbps an
	>> Einsatz im onready-Handler
.onready	Eventhandler, der feuert, wenn das System bereit für weitere Aufgaben ist
	>> Anwendung z.B.: "spt.onready = methodeXY;"

Entwicklerwerkzeuge:

ooo(text)	Ausgabe für Testausgaben & Debugging
o(text [, geoTuples])	wie o(), jedoch einzeilig (gut für Statusüberprüfungen)
	>> geoTuples: Koordinaten- und Dimensionsangaben als String im Format "x327 y267 w278 h981" (jede Angabe optional, Reihenfolge egal)
OOO(text)	Wie ooo(), jedoch in externem Fenster.
O(text [, geoTupels])	Wie o(), jedoch in externem Fenster.
output(text)	Blendet Ausgabebereich für Programmierer ein, innerhalb des Dokuments
	(Alternative zu alert)
outputWindow(text)	Blendet Ausgabebereich für Programmierer ein, in neuem Fenster
	(Alternative zu alert)
debugMessage(output)	Lässt Programmablauf Schritt für Schritt nachvollziehen
debugMessagesOFF()	Schaltet Programmverfolgungsanzeige aus, auch wenn debugMessage()-Zeilen im Code noch stehen
debugMessagesON()	Macht debugMessagesOFF() rückgängig
debugMessagesConfirmOFF()	Zum Protokollieren ohne erneute Bestätigung vor jedem nächsten Schritt
debugable(commands)	Sorgt in Stringcodes dafür, dass debugMessage() auch von dort den richtigen Namen der umgebenden Fkt. angibt
	>> Sonst gibt er "[Funktion anonymous]" u.ä.
	>> Gebrauchsweise z.B.: window.setTimeout(debugable('meineFunktion();'),100);
listProperties(obj)	Erstellt eine Liste (Typ String) der Attribute eines Objekts und der dazugehörigen Werte

Sonstiges:

onbodyload	Eventhandler, dessen Code erst ausgeführt wird, wenn das Body-Element vorhanden ist.
isIE	Enthält Wahrheitswert darüber, ob der IE vorliegt
	>> Varianten: isIE6, isIE7, isIE8
isFF	Enthält Wahrheitswert darüber, ob der Firefox vorliegt
[Iframe].belongsToSystem	Ist im IFrame auf 1 gesetzt, falls dieser vom Framework stammt und fehlt, falls nicht.
[Object].hasProperties()	Gibt an, ob einem Objekt bereits eine Eigenschaft zugewiesen worden ist
[Object].clone()	Klont ein Objekt
	>> Style-Objekt wird in FF, nicht aber in IE geklont
undef([arg])	Gibt undefined-Typ zurück (gut zur Auslassung von Parametern bei Funktionsaufrufen)
	>> Bei Angabe eines Paramters prüft er, ob dieser "undefined" ist und gibt dann 'true' zurück.
hideGlobal(varNameStr [, val]).allow(fncs)	Schützt globale Variablen, so dass Sie nur noch von authorisierten Funktionen gelesen werden können, und auch von diesen nur durch get_[varName](),
	>> varNameStr: Variablenname in Anführungszeichen gefasst.
	>> Wenn 'val' angegeben (optional), dann braucht die Variable nicht schon vorher zu existieren.
	>> Beispiel:

```
var g1 = "geheim";
function myFnc(){
        ooo(get_g1());
}
hideGlobal(g1).allow(myFnc);
```

getChangedAttributes(obj1, obj2)	Gibt Array mit den Attributen zurück, die in obj2 anders als in obj1 oder gegenüber obj1 neu hinzugekommen sind.
	>> Rückgabe-Array echt und assoziativ zugleich: Per Nummernindex Attributnamen, per Attributnamen die Werte.
	>> Nebeneffekt der Echtheit des Arrays: per ".length" ist die Anzahl der Attribute, die sich geändert haben bzw. neu hinzugekommen sind, feststellbar.
getHtmlAttributesObj(str)	Generiert aus einem als Stringcode übergebenen HTML-Element ein (Nicht-HTML-)Objekt, dass nur die im Code angegebenen Attribute samt Werte enthält
new UserInfo()	Erzeugt Objekt, das einige (approximative) Infos über den User angeben kann
.onready	Eventhandler, in welchem der Rest des Programms fortgesetzt werden muss
.city	Gibt Stadt an, in der sich der User wahrcheinlich aufhält
.region	Gibt Region/Bundesland an, in der sich der User wahrcheinlich aufhält

```
        .country                        Gibt Land an, in der sich der User wahrcheinlich aufhält

    cX(width)                           Gibt eine X-Koordinate so an, dass eine Box / ein Fenster mit der Breite 'width' horizontal zentriert würde
    cY(height)                          Gibt eine Y-Koordinate so an, dass eine Box / ein Fenster mit der Höhe 'height' vertikal zentriert würde
    rX(width)                           Gibt eine Y-Koordinate so an, dass eine Box / ein Fenster mit der Breite 'width' rechtsbündig angezeigt würde

    makeBox(x,y,w,h,color)              Erzeugt eine DIV-Box mit den gewünschten Eigenschaften und gibt diese zurück
                                        >> Wenn nur eine Farbe angegeben ist, wird der ganze Anzeigenbereich abgedeckt (alternativ zu coverDisplay() )

    coverDisplay([color])               Deckt den ganzen Anzeigebereich zu
                                        >> Gibt bearbeitbares DIV-Element zurück, das sich durch '[Element].remove()' entfernen lässt

    fontExists(font)                    Gibt Wahrheitswert darüber zurück, ob der angegebene Font im Browser verfügbar ist

    post(url, paramStr [, target])      Steuert eine Seite POST-Daten übergebend an
                                        >> Bsp.: post("http://www.seite.de", "a=xyz&b=321&c=bla");
                                        >> target: Ausgabeziel (z.B. IFrame-Name übergeben für Ausgabe in IFrame, oder Fenstername in [window].name)

    printURL(url)                       Drucke beliebige URL (auch Bilddateien angebbar)

    loadInvisible(url)                  Lädt beliebige URL oder Datei unsichtbar
                                        >> Format: variableXY = loadInvisible(...)
                                        >> Nachträgliches Ändern mit: variableXY.src = "...";

    invisibleINPUT(idStr)               Setzt unsichtbares Input-Feld mit der in 'idStr' übergebenen ID, falls diese nicht vergeben ist
                                        >> Existenz braucht also nicht vorher abgefragt zu werden
                                        >> Gibt das Input-Feld als HTML-Element zurück

    browserName([name])                 Gibt Browsernamens ohne Version an - zuverlässiger/einfacher als mit 'navigator.appName' u.a.
                                        >> Mgl. Rückgaben: "Internet Explorer", "Firefox", "Opera", "Safari", "Chrome", "Konqueror", "Netscape", "unknown"
                                        >> Mit Argument 'name': Gibt an, ob der Browser den angegebenen Namen hat
                                        >> Prüft bei Argument-Angabe mit oder ohne Versionsnummer, z.B. 'if(browserName("Internet Explorer 6"))...'

    preciseBrowserName()                    Präzise Angabe des Browsernamens und der Version
                                        >> Bitte noch testen insb. für IE7+IE8

    noBrowserExcept(bro1, bro2, bro3, ..., ...)  Schließt anzugebende Browser aus
                                        >> Akzeptierte Werte (Strings): "Microsoft Internet Explorer", "Mozilla Firefox", "Safari", "Opera", "Konqueror", "Netscape Navigator"
                                        >> Bei einigen sogar Versionsangabe möglich, z.B. "Microsoft Internet Explorer 5" etc.
                                        >> Derzeit maximal acht Browser

    preventFrameKilling()               Verhindert Framekilling

    getAssignmentTuples(str)            Macht z.B. aus "x100 y200" ein Objekt nach dem Muster obj.x und obj.y mit den Werten 100 und 200

    to_dd_mm_yyyy(dateObj/millisec)     Setzt eine Datumsangabe in das Format des Beispiels 01.01.2010

    getDateOfNext(day)                  Ermittelt z.B. das Datum des nächsten Samstags etc.
                                        day: vom Typ Number - Sonntag hat 0, Montag 1 usw.

    history.setLength(length[, fnc])    Verändert Wert in "history.length" gezielt.
                                        >> Programmfortsetzung in 'fnc'
                                        >> In FF & Chrome für 'length' keine Werte höher als 50
                                        >> Sollte nicht sofort mit dem Body-Load gestartet werden, sondern z.B. mit einem kleinen setTimeout
                                        >> Nicht für IE

    history.walk(steps)                 Ähnlich wie history.go(), läuft jedoch auch, wenn mehr Schritte als ín History übrige Seiten angegeben sind
                                        >> Negativangabe in steps bewirken Rückwärtsblättern
                                        >> Ist langsamer als history.go()

    skypeInstalled()                    Gibt Wahrheitswert darüber zurück, ob Skype installiert ist
                                        >> Momentan nur für Firefox & Windows-IE

    javaActive(codebase)                Gibt Wahrheitswert darüber zurück, ob Java installiert UND aktiviert ist
                                        >> Braucht "JavaDetecting.class", dessen Pfad mit 'codebase' angegeben wird
                                        >> Alternative zu navigator.javaEnabled()
                                        >> IE 7 gibt Active-X-Warungen => codebase-Angabe aus Applet-String raus
                                        >> für evtl. Anpassungen siehe: http://frank.neatstep.com/node/93

    googleAdsTo(x,y, i)                 Positioniert Google-Anzeigen
                                        >> i ist die Nummer des Anzeigenblocks (je älter, desto niedriger)
                                        >> funktioniert in letzter Zeit schlecht
    ----------------------------------------------------------------------------------------------------------------

*/


/* ############################################################################################## */
var Turboid = {};
var browserType = navigator.appName;
var wahr = new Boolean(true);
var falsch = new Boolean(false);
var tempGlobal01, tempGlobal02, tempGlobal03, tempGlobal04, tempGlobal05, tempGlobal06, tempGlobal07, tempGlobal08, tempGlobal09, tempGlobal10;
var isIE, currentlyFocussedElement;
```

```
var zeroClipboardIsActive = 0;
var draggableIsClicked = 0; // für id().makeDraggable()
var _hoverpopup_ = null; var hoverpopupListenerExists = 0;
var contactListenerTargets = [];
var iframeFocusAllowed = 1; // Für Chrome
/* ################################################################################################ */


function functionsLibraryTest(){
	alert("functionsLibrary works!");
}



Turboid.isOnlyOne = function(){
	return !(top!=self && top.Turboid);
}

function undef(arg){
	if(undef.arguments && undef.arguments.length==1){
		if((typeof arg)=="undefined")
			return true;
		else
			return false;
	}
}

//  ___________________ \/ ___________________ \/ ___________________ \/ ___________________ \/ ___________________ \
//  ==  P R O T O T Y P E S  ==


	// Object-Prototypes:

	Object.prototype.clone = function() {
		var retObj = {};
		for(var k in this){ if(keyOk(k)){
			retObj[k] = this[k];
		}}
		return retObj;
	}

	Object.prototype.hasProperties = function(){
		for(var i in this){if(keyOk(i)){
			return 1;
		}}
		return 0;
	}

	Object.prototype.setByRef = function(val){ // From sirdarckcat.blogspot.com
		this.valueOf=this.toSource=this.toString=function(){
			return val;
		};
		return val;
	};

	Object.prototype.tempchange = function(varName, val, msec){
		var obj = this;
		if(undef(listeningNumbers[varName]))
			listeningNumbers[varName] = 0;
		if(undef(obj[varName+"_origVal"]))
			obj[varName+"_origVal"] = obj[varName];
		obj[varName] = val;
		listeningNumbers[varName]++;
		setTimeout(function(){
			if(listeningNumbers[varName]==1){
				obj[varName] = obj[varName+"_origVal"];
			}
			listeningNumbers[varName]--;
		},msec);
	}; var ieService = {   }; window.tempchange = ieService.tempchange;


	Object.prototype.getLength = function(){
		var n = 0;
		from(this, function(k){
			n++;
		})
		return n;
	}

	// Array-Prototypes:

	Array.prototype.insertAt = function(i,val){
```

```
        var part1 = this.slice(0,i);
        var part2 = this.slice(i,this.length);
        part1.push(val);
        var result = part1.concat(part2);
        this.push(0);
        for(var i = 0; i < result.length; i++){
            this[i] = result[i];
        }
    }

    Array.prototype.drop = function(i){
        var result = (this.slice(0,i)).concat(this.slice(i+1,this.length));
        this.pop();

        for(var i = 0; i < this.length; i++){
            this[i] = result[i];
        }
        return this;
    }

    Array.prototype.dropByValue = function(val){
        var beg = 0;
        var tempArr = new Array(0);
        for(var i = 0; i < this.length; i++){
            if(this[i]==val){
                tempArr = tempArr.concat(this.slice(beg,i));
                beg = i+1;
            }
            if(i==this.length-1)
                tempArr = tempArr.concat(this.slice(beg,i+1));
        }
        for(var i = 0; i < tempArr.length; i++){
            this[i] = tempArr[i];
        }
        var laps = this.length;
        for(var i = tempArr.length; i < laps; i++){
            this.pop();
        }
        return this;
    };

    Array.prototype.getKeyAtPoint = function(ind){
        var c=0;
        for(var i in this){
            if(keyOk(i)){
                if(c==ind)
                    return(i);
                c++;
            }
        }
    };


    Array.prototype.getValueAtPoint = function(ind){
        var c=0;
        for(var i in this){
            if(keyOk(i)){
                if(c==ind){
                    return this[i];
                }
                c++;
            }
        }
    }

    Array.prototype.getIndexOfKey = function(key){
        var c = 0;
        for(var i in this){
            if(keyOk(i)){
                if(i==key)
                    return c;
                c++;
            }
        };
    }


    Array.prototype.getIndex = function(value){
        for(var i=0; i<this.length; i++){
            if(this[i]==value){
                return i;
                break;
            }
        }
```

```
        return null;
    }

    Array.prototype.contains = function(value){
        for(var i=0; i < this.length; i++){
            if(this[i]==value){
                return 1;
            }
        }
        return 0;
    }

    Array.prototype.getNext = function(){
        if(!isNaN(this.currentPoint)){
            if(this.currentPoint==this.length-1)
                return null;
            else
                this.currentPoint++;
        } else {
            this.currentPoint = 0;
        }
        return this[this.currentPoint];
    };

    Array.prototype.setNext = function(next){
        this.currentPoint = next-1;
    };

    Array.prototype.getCurrent = function(next){
        return this[this.currentPoint];
    };

    Array.prototype.getPrev = function(){
        if(!isNaN(this.currentPoint)){
            if(this.currentPoint==0)
                return null;
            else
                this.currentPoint--;
        } else {
            this.currentPoint = 0;
        }
        return this[this.currentPoint];
    };

    Array.prototype.avg = function() {
        var av = 0;
        var cnt = 0;
        for (var i = 0; i < this.length; i++) {
            var e = +this[i];
            if(!e && this[i] !== 0 && this[i] !== '0')
                e--;
            if (this[i] == e) {
                av += e; cnt++;
            }
        }
        return av/cnt;
    };

    // Function-Prototypes:

    Function.prototype.getHead = function(){
        fkt = this + "";
        return fkt.substring(0,fkt.indexOf('{'));
    };

    Function.prototype.getBody = function(){
        fkt = this + "";
        return fkt.substring(fkt.indexOf("{")+1, fkt.lastIndexOf("}"));
    }

    Function.prototype.getName = function(){
        fkt = this + "";
        return trimString(fkt.substring(9, fkt.indexOf("(")));
    }

    Function.prototype.getParameters = function(){
        fkt = this + "";
        return trimString(fkt.substring(fkt.indexOf("(")+1, fkt.indexOf(")")));
    }

    Function.prototype.clone = function() {
      var that = this;
      var temp = function temporary() { return that.apply(this, arguments); };
      for( key in this ) {
```

```
            temp[key] = this[key];
        }
        return temp;
    };


    Function.prototype.get = function(varName){
        var r = this[varName];
        delete this[varName];
        return r;
    }

    Function.prototype.isEmpty = function(){
        return (this.getBody().replace(/ /g, "").length<2);
    };

/*
\_____________/\_____________/\_____________/\_____________/\_____________/*/




function keyOk(key){
    if(!(key in Object.prototype) && !(key in Array.prototype) && key!="undefined")
        return 1;
    else
        return 0;
}


//_____________\_____________\_____________\_____________\_____________\
//   ==DEFERRING FEATURE==

    listeningNumbers = {};


    function defer(fnc, msec){
        if(undef(listeningNumbers[fnc]))
            listeningNumbers[fnc] = 0;
        listeningNumbers[fnc]++;
        setTimeout(function(){
            if(listeningNumbers[fnc]==1){
                fnc();
            }
            listeningNumbers[fnc]--;
        },msec);
    }


/*   TEMPCHANGE FEATURE(end)
\_____________/\_____________/\_____________/\_____________/\_____________/*/


//_____________\_____________\_____________\_____________\_____________\
//   ==INCLUDE-FEATURE==

    function fireOnReadyHandler(scriptName){
        if((typeof include[scriptName].onready)=="function"){
            setTimeout(function(){
                include[scriptName].onready();
                include[scriptName].onready=function(){};
            },100);
        }
}

function include(scriptName, pFnc){

    include[scriptName] = {};

    if((typeof pFnc)=="undefined")
        var fnc = function(){};
    else
        var fnc = function(){ availableIncludes.push(scriptName); pFnc();};

    if(includes.contains(scriptName)){
        if(availableIncludes.contains(scriptName)){
            fnc();
        } else {
            loop(function(){
                if(availableIncludes.contains(scriptName))
                    loop.exit();
            });
            loop.then(function(){
                fnc();
```

```
                });
            }
        } else {
            includes.push(scriptName);
            var scriptElm = document.createElement("script");
            scriptElm.setAttribute("language","JavaScript");
            scriptElm.setAttribute("src",scriptName);
            var bd = document.getElementsByTagName("body")[0];
            bd.appendChild(scriptElm);
            // most browsers
            scriptElm.onload = function(){ fireOnReadyHandler(scriptName); fnc();};
            // IE:
            scriptElm.onreadystatechange = function() {
                if(this.readyState == "complete"){
                    fireOnReadyHandler(scriptName);
                    fnc();
                }
                if(this.readyState == "loaded"){
                    setTimeout(function(){
                        fireOnReadyHandler(scriptName);
                        fnc();
                    },1000);
                }
            }
        }
    }; var includes = []; var availableIncludes = [];


    include.register = function(){
        var args = include.register.arguments;

        for(var i = 0; i < args.length-1; i++){          // Arbeite übergebene Argumente ab
            var retVal, isReady;

            var file = args[args.length-1];          // Letztes Argument ist der Name der Datei mit den Funktionen.
                                                                                    // Für jede Funktion wird ihr scheinbares Vorhandensein vorgekaukelt, ...
            window[args[i]] = function(x){ return function(p0, p1, p2, p3, p4, p5, p6, p7, p8, p9, p10, p11, p12, p13, p14, p15, p16, p17, p18, p19){
                var fncName = args[x];                                              // ... mit einer namensgleichen Funktion, die in Wirklichkeit nur...
                                                                                    // ... die Datei mit den echten Funktionen laden soll. Sobald eine der Scheinfunktionen ...
                if(!include.register.files.contains(file)){
                    include.register.files.push(file);
                    include(file, function(){                                       // ... dies getan hat, wird sie beim Laden der Datei durch die echte Funktion ja überschrieben.
                        retVal = (window[fncName])(p0, p1, p2, p3, p4, p5, p6, p7, p8, p9, p10, p11, p12, p13, p14, p15, p16, p17, p18, p19);          // Die echte wird ausgeführt und
ihr Rückgabewert gesichert.
                        var c = {};
                        for(var j = 0; j < args.length-1; j++){                     // Jede andere neu geladene Funktion wird so umgeschrieben, ...
                            c[args[j]] = window[args[j]].clone();                   // ... dass bei ihrem Aufruf ihren Rückgabewert in ihrer Property '.retVal' selbst speichert ...
                            window[args[j]] = function(y){ return function(p0, p1, p2, p3, p4, p5, p6, p7, p8, p9, p10, p11, p12, p13, p14, p15, p16, p17, p18, p19){
                                window[args[y]].retVal = c[args[y]](p0, p1, p2, p3, p4, p5, p6, p7, p8, p9, p10, p11, p12, p13, p14, p15, p16, p17, p18, p19);
                                var obj = {};                                       // ... und ein Alibi-Objekt zurückgibt, um die Fortsetzung des Code-Rests ...
                                obj.resume = function(fnc){                         // ... in [Funktion].resume(...) zu ermöglichen.
/* setTimeout(function(){ */ fnc();       //},111);
                                };
                                return obj;
                            };}(j);
                        }
                        isReady = 1;
                    });
                    var obj = {};                                   // Die erste aufgerufene der neuen Funktionen gibt für das resuming auch ein Objekt zurück,...
                    obj.resume = function(fnc){                             // ... muss aber als einzige mit der Ausführung des Resuming-Codes bis zur Fertigladung der Include-Datei warten,..
                        include(file, function(){                   //  ... da sie es ist, die "sich" erst aus der Datei laden muss und darum nicht sofort bereit steht. (Die Include-Fkt.
                            window[fncName].retVal = retVal;                // ... beinhaltet bereits den Warteprozess.)
                            fnc();
                        });
                    };
                    return obj;
                } else {
                    // Falls schon beim allerersten Mal versucht wird, zwei oder mehr der ausgelagerten Funktionen direkt hintereinander aufzurufen...
                    // dann obige Prozedur verhindern, da sonst jede Funktion mindestens doppelt geklont und evtl. vorherige Rückgabewerte überschreiben löschen würde:
                    include(file, function(){
                        window[fncName](p0, p1, p2, p3, p4, p5, p6, p7, p8, p9, p10, p11, p12, p13, p14, p15, p16, p17, p18, p19).resume(function(){
                            isReady = 1;
                            //ooo(window[fncName].retVal);
                        });

                    });
                    var obj = {};
                    obj.resume = function(fnc){
                        include(file, function(){
                            fnc();
                        });
                    };
                    return obj;
                }
            }
```

```
                };}(i);
            }
        }; include.register.files = [];



/*
\_____________/\_____________/\_____________/\_____________/\_____________/*/

function getNameOfChangedStyle(a, b){
    a = replaceStrings(":", ";", a);
    a = replaceStrings(" ", "", a);
    a = a.split(";");
    b = replaceStrings(":", ";", b);
    b = replaceStrings(" ", "", b);
    b = b.split(";");
    if(a.length!=b.length){
        if(a[0]!=b[0]){
            if(a.length > b.length)
                return a[0];
            else
                return b[0];
        } else if(a[a.length-3]!=b[b.length-3]) {
            if(a.length > b.length)
                return a[a.length-3];
            else
                return b[b.length-3];
        } else {
            for(var i = 0; i < a.length; i++){
                if(a[i]!=b[i] && a.length>b.length)
                    return a[i];
                if(a[i]!=b[i] && a.length<b.length)
                    return b[i];
            }
        }
    } else {
        for(var i = 0; i < a.length; i++){
            if(a[i]!=b[i])
                return a[i-1];
        }
    }
    return null;
}

function getChangedAttributes(obj1, obj2){
    var retArr = [];
    from(obj2, function(k){
        if(k!="style" && (undef(obj1[k]) || obj1[k]!=obj2[k])){
            retArr.push(k);
            retArr[k] = obj2[k];
        }
    });
    return retArr;
}


function getHtmlAttributesObj(str){
    var str = str.substring(0,str.indexOf(">")+1)+"</"+str.substring(1,str.indexOf(" "))+">"; // Element vom Inhalt befreien (für Performance beim Einfügen in den hiddenCont_system).
    var retObj = {};
    hiddenCont_system.innerHTML = str;
    for(var i in hiddenCont_system.firstChild){ if(keyOk(i)){
        if(isHtmlAttribute(i) && i!="dataset" && i!="style" && hiddenCont_system.firstChild[i]!="")
            retObj[i] = hiddenCont_system.firstChild[i];
    }}
    if(hiddenCont_system.firstChild.style[0]){
        retObj.style = {};
        for(var i = 0; i < hiddenCont_system.firstChild.style.length; i++){
            retObj.style[hiddenCont_system.firstChild.style[i]] = hiddenCont_system.firstChild.style[hiddenCont_system.firstChild.style[i]];
        }
    }

    return retObj;

}
var hiddenCont_system = document.createElement("div");

//_____________\_____________\_____________\_____________\_____________\
//    ==HANDLER EXTENDING FEATURE==


        var Event = {};

        //····························\/····························\/····························\
        // Assisting functions:
```

```
        function isNormalFunction(fnc){
            //setTimeout(function(){ ooo((fnc + "").indexOf("origF(")==-1 && (fnc + "").indexOf("function")!=-1);  ooo(fnc);},333);
            return ((fnc + "").indexOf("origF(")==-1 && (fnc + "").indexOf("function")!=-1);
        }

        function getfVarStr(extenderName){
            for(var key1 in extendingHandlers){if(keyOk(key1)){
                for(var key2 in extendingHandlers[key1]){if(keyOk(key2)){
                    if(key2==extenderName){
                        return key1;
                    }
                }}
            }}
        }

    /* Assisting functions (end)
    /\..... ...../ \..... ...../ \..... ...../*/



    Event.add = function(fVarStr, extendingF, extenderName){

        var isFirstExtension = 0;
        var objName = fVarStr.indexOf(".")>0?fVarStr.split(".")[0]:"window"; // Filtert den Objektnamen vor dem Punkt aus fVarStr, damit er per 'that' referenziert werden kann.
        if(fVarStr.indexOf("")>0)
            var fVarStr = fVarStr.replace(/"/g, """);
        if(undef(extenderName))
            extenderName = randomString(10);
    eval('var f = '+fVarStr+';');

    // Liste der diesem Eventhandler hinzugefügten Funktionen um neue Funktion erweitern, um diese Funktionen später mit Event.drop( ) herausnehmen zu können:
    if(undef(extendingHandlers[fVarStr])){
        extendingHandlers[fVarStr] = [];
        isFirstExtension = 1;
    }
    if((f+"").indexOf("origF(")==-1 && (f+"").indexOf("function")!=-1)
        extendingHandlers[fVarStr]["veryFirstFunction"] = f;

    extendingHandlers[fVarStr][extenderName] = extendingF;

    // Alarmiere evtl. das Event.guard-Feature bei unerwarteten Überschreibungen:
    if(guardedHandlers[fVarStr] && isNormalFunction(f) && f!=guardedHandlers[fVarStr] )
        Event.guard[fVarStr]();

    // Eigentliche Erweiterungsoperation:
    var extendingFparams = extendingF.getParameters();

    eval('if(!'+fVarStr+') '+fVarStr+'=function('+extendingF.getParameters()+'){};');

    if(isFirstExtension)
        params = extendingFparams;
    else
        eval('var params = '+fVarStr+'.getParameters();');

    if(/\bthat\b/.exec(extendingFparams))
        var params2 = extendingFparams.replace(/\bthat\b/, objName); // Ermöglicht die Referenzierung des den Event bekommenden Objekts per 'that'.
    else
        var params2 = params;


    // Wenn anzufügende Funktion mehr formale Parameter besitzt als der bestehende Handler, soll die Synthesis, welche die anzufügende Funktion ja aufruft, dieselbe Parameterstruktur haben:
    var extendingFparamsLength = extendingFparams==""?0:extendingFparams.split(",").length;
    var paramsLength = params==""?0:params.split(",").length;
    if(extendingFparams.split(",").contains("that"))
        extendingFparamsLength--;

    if(extendingFparamsLength > params){  // Ermöglicht die Referenzierung des den Event bekommenden Objekts per 'that'.
        params = extendingFparams;

        params2 = params;
        params = params.replace(/ /g, "").split(",").dropByValue("that").join(",");

        if(/\bthat\b/.exec(extendingFparams))
            params2=params2.replace(/\bthat\b/, objName);
    }

    eval('var origF='+fVarStr+';                                    ');
    eval('var synthesis = function('+params+'){                     '+
    '           origF('+params+');                              '+
    '           extendingHandlers[fVarStr][extenderName]('+params2+');          '+
    '}                                                          ');
    eval(fVarStr +'= synthesis;                                     ');
```

```
};  var extendingHandlers = [];


Event.set = function(){

    //…… \ /…… \ /…… \
    // Arguments:

    var args = Event.set.arguments;
    var end = 0;

    if(args.length==3){
        var fVarStr = args[0];
        var fnc = args[1];
        var extenderName = args[2];
    } else {
        var fnc = args[0];
        var extenderName = args[1];
        var fVarStr = getfVarStr(extenderName);
    }

    /* Arguments (end)
    /\…… ……/ \…… ……/ \…… ……/*/


    if(!extendingHandlers[fVarStr] || !extendingHandlers[fVarStr][extenderName]){
        Event.add(fVarStr, fnc, extenderName);
        return true;
    }


    if((typeof extenderName)=="number")
        extenderName = extendingHandlers[fVarStr].getKeyAtPoint(extenderName);

    if(extendingHandlers[fVarStr][extenderName]!=guardedHandlers[fVarStr])
        extendingHandlers[fVarStr][extenderName] = fnc;
    else
        alert('Error: You tried to replace a protected function in "'+fVarStr+'".');

};


Event.drop = function(){

    //…… \ /…… \ /…… \
    // Arguments:

    var args = Event.drop.arguments;
    var end = 0;

    if(args.length==2){
        var fVarStr = args[0];
        var extenderName = args[1];
    } else {
        var extenderName = args[0];
        var fVarStr = getfVarStr(extenderName); // (Übrigens: Falls extenderName-Schlüssel nicht existiert wird "undefined" zurückgegeben.)
    }

    /* Arguments (end)
    /\…… ……/ \…… ……/ \…… ……/*/

    if((typeof extenderName)=="number")
        extenderName = extendingHandlers[fVarStr].getKeyAtPoint(extenderName);

    if(_(fVarStr)){
        if(extendingHandlers[fVarStr][extenderName]!=guardedHandlers[fVarStr])
            extendingHandlers[fVarStr][extenderName] = function(){};
        else
            alert('Error: You tried to drop a protected function in "'+fVarStr+'".');
    } else {
        return false; // Fehlerrückgabe.
    }

};


Event.exists = function(fVarStr, extenderName){
    if(_(extendingHandlers)){
        if(extendingHandlers[fVarStr]){
            if(extendingHandlers[fVarStr][extenderName] && !extendingHandlers[fVarStr][extenderName].isEmpty())
                return 1;
            else
                return 0;
```

```
            } else {
                return 0;
            }
        } else {
            return 0;
        }
};


Event.guard = function(fVarStr){
    var watcherName = randomString(5);

    eval('var fnc = '+fVarStr+';');

    if(undef(extendingHandlers[fVarStr]))
        extendingHandlers[fVarStr] = [];

    guardedHandlers[fVarStr] = fnc;

    Event.guard[fVarStr] = function(){
        alert('Error: "'+fVarStr+'" has been completely overwritten. Please use "Event.add()" instead of overwriting.');
    };

    eval( 'loop.timer.addSlow(function(){'+
        '    if(guardedHandlers[fVarStr] != '+fVarStr+' && ('+fVarStr+'+"").indexOf("origF(")==-1){'+
        '        loop.timer.drop(watcherName);  '+
        '        alert("Error: \\"'+fVarStr+'\\" has been completely overwritten. Please use \\"Event.add()\\" instead of overwriting.");'+
        '    }'+
        '}, watcherName);'
    );
};  var guardedHandlers = [];


Event.add2 = function(elm, evType, fn, useCapture) {
    if (elm.addEventListener) {
        if(evType=="attrchange"){
            evType = "DOMAttrModified";
            elm.prevCssText = elm.style.cssText;
            elm.addEventListener(evType, function(e){
                if(_(elm).hasAttributeNode(e.relatedNode)){
                    if(e.attrName=="style"){
                        var nameOfChangedStyle = getNameOfChangedStyle(elm.prevCssText, elm.style.cssText);
                        if(nameOfChangedStyle){
                            var obj = {};
                            obj.attrName = "style."+toCamelCase(nameOfChangedStyle);
                            obj.relatedNode = e.relatedNode;
                            obj.newValue = elm.style[nameOfChangedStyle];
                            e = obj;
                        }
                        elm.prevCssText = elm.style.cssText;
                    }
                    fn(e);
                }
            }, useCapture);
        } else {
            elm.addEventListener(evType, fn, useCapture);
        }
        return true;
    }
    else if (elm.attachEvent) {
        if(evType=="attrchange"){
            evType = "propertychange";
            var r = elm.attachEvent('on' + evType, function(){
                if(event.propertyName.indexOf("style.")==0 || _(event.propertyName).isOneOf("abbr", "accept-charset", "accept", "accesskey", "action", "align", "alink", "alt",
"archive", "archive", "axis", "background", "bgcolor", "border", "cellpadding", "cellspacing", "char", "charoff", "charset", "checked", "cite", "class", "classid", "clear", "code", "codebase", "codetype",
"color", "cols", "colspan", "compact", "content", "coords", "data", "datetime", "declare", "defer", "dir", "dir", "disabled", "enctype", "face", "for", "frame", "frameborder", "headers", "height", "href",
"hreflang", "hspace", "http-equiv", "id", "ismap", "label", "label", "lang", "language", "link", "longdesc", "longdesc", "marginheight", "marginwidth", "maxlength", "media", "method", "multiple",
"name", "nohref", "noresize", "noshade", "nowrap", "object",  "profile", "prompt", "readonly", "rel", "rev", "rows", "rowspan", "rules", "scheme", "scope", "scrolling", "selected", "shape", "size",
"span", "span", "src", "standby", "start", "summary", "tabindex", "target", "text", "title", "type", "usemap", "valign", "value", "valuetype", "version", "vlink", "vspace", "width")){
                    var e = {};
                    e.attrName = event.propertyName;
                    if(e.attrName.indexOf("style.")==0){
                        e.relatedNode = event.srcElement.getAttributeNode("style");
                        e.newValue = elm.style[replaceStrings("style.", "", e.attrName)];
                    } else {
                        e.relatedNode = event.srcElement.getAttributeNode(e.attrName);
                        e.newValue = elm[e.attrName];
                    }

                    fn(e);
                }
            });
        } else {
            var r = elm.attachEvent('on' + evType, fn);
```

```
                }


                return r;
            }
            else {
                if((typeof elm['on' + evType])=="function")
                    fn = new Function(elm['on' + evType].getBody() + fn.getBody());
                elm['on' + evType] = fn;
            }
        };




/*
\_____________/\_____________/\_____________/\_____________/\_____________/*/


//_____________\/_____________\/_____________\/_____________\/_____________\
//    ==L O O P - F E A T U R E==

        loop.exit = function(){
            breakedLoops.push(loop.exit.caller);
        }; var breakedLoops = [];


    function loop(fkt,interv,lapsNumber){
        allLoops.push(fkt);
        var doneLaps = 0;
        if(lapsNumber)
            lapsNumber--;   // Sonst macht er eine Runde zu viel.
        if(undef(interv))
            var interv = 1;  // Wenn kein Intervall angegeben ist, dann nimm 1.
        var schl = function(){
            if(!breakedLoops.contains(fkt)){
                if(undef(lapsNumber) || doneLaps<=lapsNumber)
                    setTimeout(fkt,1);
                if(_(lapsNumber) && doneLaps>=lapsNumber){
                    schl = "";
                    breakedLoops.push(fkt);
                }
                setTimeout(schl,interv);
                doneLaps++;
            }
        }; schl();
    } var allLoops = [];

    loop.then = function(fkt){
        var lastLoop = allLoops[allLoops.length-1];
        var schleife = function(){
            if(breakedLoops.contains(lastLoop)){
                schleife = "";
                setTimeout(fkt,1);
            }
            setTimeout(schleife,1);
        }; schleife();
    }

    loop.turbo = function(fkt,acc,lapsNumber){
        //¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\
        // Acceleration:
        if(undef(acc)){
            var acc = 1;
        }
        if(_(lapsNumber)){
            var additionalLaps = lapsNumber%(10*acc);
            lapsNumber = lapsNumber / (10*acc);
        } else {
            var additionalLaps = 0;
        }
        var origFkt = fkt;
        fkt = function(){
            for(var i = 0; i < (10*acc); i++){
                if(breakedLoops.contains(origFkt)) break;
                origFkt();
            }
        };
        /* Acceleration (end)
        //\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/*/
        allLoops.push(origFkt);
        var doneLaps = 0;
```

```
            if(lapsNumber)
                lapsNumber--;   // Sonst macht er eine Runde zu viel.
            var schl = function(){
                if(!breakedLoops.contains(origFkt)){
                    if(undef(lapsNumber) || doneLaps<=lapsNumber)
                        setTimeout(fkt,1);
                    if(_(lapsNumber) && doneLaps>=lapsNumber){
                        schl = "";
                        for(var i = 0; i < additionalLaps+1; i++){
                            origFkt();
                        }
                        breakedLoops.push(fkt);
                    }
                    setTimeout(schl,1);
                    doneLaps++;
                }
            }; schl();
        }


/*
\_____________/\_____________/\_____________/\_____________/\_____________/*/


var scriptTime = 0;


//              \/              \/              \/              \/              \
//   ==LOOP  QUEUE  FEATURE==


    var loopQueueFunctions = [];
    var loopSlowQueueFunctions = [];

    loop(function(){
        for(var k in loopQueueFunctions){ if(keyOk(k)){
            (loopQueueFunctions[k])();
        }}
        for(var k in loopSlowQueueFunctions){ if(keyOk(k)){
            if(isIE){
                if(scriptTime%(62*20)==0)
                    (loopSlowQueueFunctions[k])();
            } else {
                if(scriptTime%(50*20)==0)
                    (loopSlowQueueFunctions[k])();
            }
        }}

    }, 50);


    loop.timer = {};

    loop.timer.add = function(fnc, fncLabel){
        if(undef(fncLabel))
            fncLabel = randomString(4);
        loopQueueFunctions[fncLabel] = fnc;
    };


    loop.timer.addSlow = function(fnc, fncLabel){
        if(undef(fncLabel))
            fncLabel = randomString(4);
        loopSlowQueueFunctions[fncLabel] = fnc;
    };




    loop.timer.drop = function(fncLabel){
        if(!isNaN(fncLabel))
            fncLabel = loopQueueFunctions.getKeyAtPoint(fncLabel);
        if(!isNaN(fncLabel))
            fncLabel = loopSlowQueueFunctions.getKeyAtPoint(fncLabel);

        delete loopSlowQueueFunctions[fncLabel];
        delete loopQueueFunctions[fncLabel];
    };


/*   LOOP  QUEUE  FEATURE(end)
\_____________/\_____________/\_____________/\_____________/\_____________/*/

loop.timer.add(function(){
```

```
	if(isIE)
		scriptTime+=62;
	else
		scriptTime+=50;

});

//_____________\_____________\_____________\_____________\_____________\
//    ==DELAY FEATURE==

		var delayFunctionsArrays = [];
		for(var i = 0; i < 20; i++){
			delayFunctionsArrays[i] = [];
		}
		var dfaIndex = 0;


		loop.timer.add(function(){
			if(dfaIndex == 19)
				dfaIndex = -1;
			dfaIndex ++;
			for(var i = 0; i < delayFunctionsArrays[dfaIndex].length; i++){
				(delayFunctionsArrays[dfaIndex][i])();
			}
			delayFunctionsArrays[dfaIndex] = null;
			delayFunctionsArrays[dfaIndex] = [];
		});



		function delay(fnc, sec){
			if(sec<1 && !((sec*100)%10)){
				if(isIE){
					var index = (sec*1000)/50+dfaIndex;
					if(index>19)
						index-=20;
					delayFunctionsArrays[index].push(fnc);
				} else {
					setTimeout(fnc, sec*1000);
				}
			}
		}


/*    DELAY FEATURE(end)
\_____________/\_____________/\_____________/\_____________/\_____________/*/

	function from(list, fnc){
		for(var k in list){ if(keyOk(k)){
			fnc(k);
		}}
	};


	function hideGlobal(varNameStr, val){
		var pw = window[varNameStr];
		if(undef(pw))
			pw = window[varNameStr] = val;

		var args = hideGlobal.arguments;
		var retObj = {};
		retObj.allow = function(){
			if(!(this.allow.arguments[0] instanceof Function) && ((this.allow.arguments[0] instanceof Object) || (this.allow.arguments[0] instanceof Array))){ // Falls ein Array an
Funktionen übergeben wird...
				var arr = this.allow.arguments[0];

				this.authorizedFncs = {};

				for(var i in arr){ if(keyOk(i)){
					this.authorizedFncs[i] = arr[i];
					this.authorizedFncs.length = i+1;
				}}



			} else {

				this.authorizedFncs = this.allow.arguments;
			}
		};

		window["get_"+varNameStr] = function(){
			retObj.authorizedFncs.contains = Array.prototype.contains;
```

```
            if(retObj.authorizedFncs.contains(window["get_"+varNameStr].caller))
                return pw;
        };

        window[varNameStr] = window[randomString(5)];  // Setze die zu verbergende globale Variable auf "undefined".

        return retObj;

    }

//_____________\_____________\_____________\_____________\_____________\
//    ==ONLOADEDBODY FEATURE==

    var onbodyload = function(){ var x = "origF("; }; // nur zu Markierungszwecken
    loop(function(){
        if((typeof onbodyload)=="function" && document.getElementsByTagName("body")[0]){
            loop.exit();
            onbodyload();
        }
    });


/*    ONLOADEDBODY FEATURE(end)
\_____________/\_____________/\_____________/\_____________/\_____________/*/


//_____________\_____________\_____________\_____________\_____________\
//
//    ==DOCUMENT-POSITION TO SCREEN==


    if(browserName()=="Internet Explorer"){
        loop(function(){
            thisDocumentWidth = Mouse.getScrX() - Mouse.getX();
            thisDocumentHeight = Mouse.getScrY() - Mouse.getY();
            if(thisDocumentWidth || thisDocumentHeight){
                loop.exit();
            }
        },111);
    } thisDocumentWidth = 0; thisDocumentHeight = 0;

    document.getX = function(){
        if(browserName()=="Internet Explorer"){
            if(thisDocumentWidth){
                var x = thisDocumentWidth;
                thisDocumentWidth = 0;
            } else {
                var x = Mouse.getScrX() - Mouse.getX();
            }
        } else {
            var winX = (document.all)?window.screenLeft:window.screenX;
            var x = winX + (window.outerWidth - getViewportWidth());
        }
        return x;
    };

    document.getY = function(){
        if(browserName()=="Internet Explorer"){
            if(thisDocumentHeight){
                var y = thisDocumentHeight;
                thisDocumentHeight = 0;
            } else {
                var y = Mouse.getScrY() - Mouse.getY();
            }
        } else {
            var winY = (document.all)?window.screenTop:window.screenY;
            var y = winY + (window.outerHeight -(getViewportHeight()+23));
        }
        return y;
    };

/*
\_____________/\_____________/\_____________/\_____________/\_____________/*/


//_____________\_____________\_____________\_____________\_____________\
//    ==BROWSER DETECTION==

    var browserInformation = new function BrowserCheck () {
     var searchBrowserInfo = function () {
       var bname, ver;
       var bs = navigator.userAgent;
       var browserCheck = [ { identification: 'Firefox', name: 'Firefox', version: 'Firefox/\([0-9.]\+\)' },
```

```
            { identification: 'Konqueror', name: 'Konqueror', version: 'Konqueror/\([0-9.]\+\)' },
            { identification: 'MSIE', name: 'Internet Explorer', version: 'MSIE \([0-9.]\+\)' },
            { identification: 'Camino', name: 'Camino', version: 'Camino/\([0-9.]\+\)' },
            { identification: 'Opera', name: 'Opera', version: 'Opera/\([0-9.]\+\)' },
            { identification: 'Netscape', name: 'Netscape', version: 'Netscape[0-9]\?/\([0-9.]\+\)' },
            { identification: 'Safari', name: 'Safari', version: 'Safari/\([0-9.]\+\)' },
            { identification: 'Gecko', name: 'Mozilla', version: 'rv:\([0-9.]+\)' }
          ];
  var i = 0;
  while (!bname && browserCheck[i]) {
   if (bs.indexOf (browserCheck[i].identification) != -1) {
    bname = browserCheck[i].name;
    if (bs.match (RegExp (browserCheck[i].version)) != -1)
     ver = RegExp.$1;
   }
   i++;
  }
  return { name: bname || 'unbekannt', version: ver || 'unbekannt' };
 }
 var browser = searchBrowserInfo ();
 this.getBrowserName = function () {
  return browser.name;
 }
 this.getBrowserVersion = function () {
  return browser.version;
 }
};

function browserName(question){
    if(undef(question)){
        if (window.opera)
            return "Opera";
        else if(navigator.userAgent.indexOf("Chrome")>-1)
            return "Chrome";
        else if(navigator.userAgent.indexOf("Netscape")>-1)
            return "Netscape";
        else if(navigator.userAgent.indexOf("MSIE")>-1)
            return 'Internet Explorer';
        else if(navigator.userAgent.indexOf("Safari")>-1 && navigator.userAgent.indexOf("Chrome")==-1)
            return 'Safari';
        else if(navigator.userAgent.indexOf("Firefox")>-1)
            return 'Firefox';
        else if(navigator.userAgent.indexOf("Konqueror")>-1)
            return 'Konqueror';
        else
            return 'unknown';
    } else {
        if((browserInformation.getBrowserName()+" "+browserInformation.getBrowserVersion()).indexOf(question)==0)
            return 1;
        else
            return 0;
    }
};  isIE = browserName("Internet Explorer"); isIE6 = browserName("Internet Explorer 6"); isIE7 = browserName("Internet Explorer 7"); isIE8 = browserName("Internet Explorer 8");
isIE9=browserName("Internet Explorer 9");
    isFF = browserName("Firefox"); isOpera = browserName("Opera");

function preciseBrowserName(){
    var browserName = "";
    if(document.ids)browserName = 'nc4';
    else if( document.all && !document.getElementById )browserName = 'Microsoft Internet Explorer 4';
    else if(window.pkcs11&&window.XML)browserName = 'Mozilla Firefox';
    else if(navigator.userAgent.indexOf("Safari")>-1) browserName="Safari";
    else if( window.getSelection && window.atob )browserName = 'Netscape Navigator 7';
    else if( window.getSelection && !document.compatMode )browserName = 'Netscape Navigator 6';
    else if( window.clipboardData && document.compatMode )
      browserName = window.XMLHttpRequest? 'Microsoft Internet Explorer 7' : 'Microsoft Internet Explorer 6';
    else if( window.clipboardData ){browserName = 'Microsoft Internet Explorer 5';
        if( !document.createDocumentFragment ) x+='.5';
        if( document.doctype && !window.print ) x+='m';}
    else if( document.images && !document.all ) browserName = 'Netscape Navigator 3';
    else if(document.clientWidth&&!window.RegExp)browserName = 'Konqueror 2';
    else if( window.opera && !document.createElement )browserName = 'Opera 5';
    else if( window.opera && window.getComputedStyle ) {
            if(document.createRange)browserName = 'Opera 8';
              else if(window.navigate)browserName = 'Opera 7.5';
                        else browserName = 'Opera 7.2';                }
    else if( window.opera && document.compatMode )browserName = 'Opera 7';
    else if( window.opera && document.releaseEvents )browserName = 'Opera 6';
    else if( document.contains && !window.opera )browserName = 'Opera 3';
    else if( document.getElementById && !document.all ) browserName = 'Opera 4';
    else browserName = '???';
    return browserName;
}
```

```
function noBrowserExcept(a,b,c,d,e,f,g,h){
	var thisBrowser = preciseBrowserName();
	var thisBrowserIsOk = 0;
	if(_(a)){
		if(thisBrowser.indexOf(a) > -1){
			thisBrowserIsOk = 1;
		} else{
			if(_(b)){
				if(thisBrowser.indexOf(b) > -1){
					thisBrowserIsOk = 1;
				} else{
					if(_(c)){
						if(thisBrowser.indexOf(c) > -1){
							thisBrowserIsOk = 1;
						} else{
							if(_(d)){
								if(thisBrowser.indexOf(d) > -1){
									thisBrowserIsOk = 1;
								} else{
									if(_(e)){
										if(thisBrowser.indexOf(e) > -1){
											thisBrowserIsOk = 1;
										} else{
											if(_(f)){
												if(thisBrowser.indexOf(f) > -1){
	thisBrowserIsOk = 1;
} else{
	if(_(g)){
		if(thisBrowser.indexOf(g) > -1){
			thisBrowserIsOk = 1;
		} else{
			if(_(h)){
				if(thisBrowser.indexOf(h) > -1)
					thisBrowserIsOk = 1;
			}
		}
	}
}
											}
										}
									}
								}
							}
						}
					}
				}
			}
		}
	}
	var allowedBrowsers = "";
	if(_(a)){
		allowedBrowsers += "- " + a + "\r\n";
		if(_(b)){
			allowedBrowsers += "- " +  b + "\r\n";
			if(_(c)){
				allowedBrowsers += "- " +  c + "\r\n";
				if(_(d)){
					allowedBrowsers += "- " +  d + "\r\n";
					if(_(e)){
						allowedBrowsers += "- " +  e + "\r\n";
						if(_(f)){
							allowedBrowsers += "- " +  f + "\r\n";
							if(_(g)){
								allowedBrowsers += "- " +  g + "\r\n";
								if(_(h)){
									allowedBrowsers += "- " +  h + "\r\n";
								}
							}
						}
					}
				}
			}
		}
	}
	if(!thisBrowserIsOk){
		alert(	"Please use one of the following browsers:\r\n"+ allowedBrowsers);
		location.href = "about:blank";
	}
}
```

```
/*    BROWSER DETECTION(end)
\_____________/\_____________/\_____________/\_____________/\_____________/*/


//_____________\/_____________\/_____________\/_____________\/_____________\
//
//    ==PRINTING-FEATURE==


    function printURL(url){
        var fensteroptionen;
        fensteroptionen = "width=600,height=550,resizable=yes,scrollbars=yes,status=no,menubar=no,toolbar=no,location=no,directories=no,";
        fensteroptionen+= "left=" + screen.width+10;
        var druckfenster = window.open(url,"Drucken",fensteroptionen);
        druckfenster.print();
        druckfenster.close();
    }


    function printNode(htmNode){
        var fensteroptionen;
        fensteroptionen = "width=600,height=550,resizable=yes,scrollbars=yes,status=no,menubar=no,toolbar=no,location=no,directories=no,";
        fensteroptionen+= "left=" + screen.width+10-500;

        if(browserType=="Microsoft Internet Explorer"){
            var druckfenster = window.open("infrastruktur/newWindowOldStyle.htm","Drucken",fensteroptionen);
        } else {
            var druckfenster = window.open("about:blank","Drucken",fensteroptionen);
            druckfenster.document.writeln("<html><head id=\"kopf\"> </head><body id=\"korpus\"></body></html>");
            druckfenster.document.all.kopf.innerHTML = document.all.kopf.innerHTML;
        }
        druckfenster.document.all.korpus.innerHTML = htmNode.innerHTML;
        druckfenster.print();
        druckfenster.close();
    }
/*
\_____________/\_____________/\_____________/\_____________/\_____________/*/


    isHtmAttr = {};
    isHtmAttr["abbr"] = 1; isHtmAttr["accept-charset"] = 1; isHtmAttr["accept"] = 1; isHtmAttr["accesskey"] = 1; isHtmAttr["action"] = 1; isHtmAttr["align"] = 1; isHtmAttr["alink"] = 1;
isHtmAttr["alt"] = 1; isHtmAttr["archive"] = 1; isHtmAttr["archive"] = 1; isHtmAttr["axis"] = 1; isHtmAttr["background"] = 1; isHtmAttr["bgcolor"] = 1; isHtmAttr["border"] = 1; isHtmAttr[
"cellpadding"] = 1; isHtmAttr["cellspacing"] = 1; isHtmAttr["char"] = 1; isHtmAttr["charoff"] = 1; isHtmAttr["charset"] = 1; isHtmAttr["checked"] = 1; isHtmAttr["cite"] = 1; isHtmAttr["class"]
 = 1;isHtmAttr["className"] = 1; isHtmAttr["classid"] = 1; isHtmAttr["clear"] = 1; isHtmAttr["code"] = 1; isHtmAttr["codebase"] = 1; isHtmAttr["codetype"] = 1; isHtmAttr["color"] = 1;
isHtmAttr["cols"] = 1; isHtmAttr["colspan"] = 1; isHtmAttr["compact"] = 1; isHtmAttr["content"] = 1; isHtmAttr["coords"] = 1; isHtmAttr["data"] = 1; isHtmAttr["datetime"] = 1; isHtmAttr[
"declare"] = 1; isHtmAttr["defer"] = 1; isHtmAttr["dir"] = 1; isHtmAttr["dir"] = 1; isHtmAttr["disabled"] = 1; isHtmAttr["enctype"] = 1; isHtmAttr["face"] = 1; isHtmAttr["for"] = 1; isHtmAttr[
"frame"] = 1; isHtmAttr["frameborder"] = 1; isHtmAttr["headers"] = 1; isHtmAttr["height"] = 1; isHtmAttr["href"] = 1; isHtmAttr["hreflang"] = 1; isHtmAttr["hspace"] = 1; isHtmAttr[
"http-equiv"] = 1; isHtmAttr["id"] = 1; isHtmAttr["ismap"] = 1; isHtmAttr["label"] = 1; isHtmAttr["label"] = 1; isHtmAttr["lang"] = 1; isHtmAttr["language"] = 1; isHtmAttr["link"] = 1;
isHtmAttr["longdesc"] = 1; isHtmAttr["longdesc"] = 1; isHtmAttr["marginheight"] = 1; isHtmAttr["marginwidth"] = 1; isHtmAttr["maxlength"] = 1; isHtmAttr["media"] = 1; isHtmAttr["method"
] = 1; isHtmAttr["multiple"] = 1; isHtmAttr["name"] = 1; isHtmAttr["nohref"] = 1; isHtmAttr["noresize"] = 1; isHtmAttr["noshade"] = 1; isHtmAttr["nowrap"] = 1; isHtmAttr["object",  "profile"]
 = 1; isHtmAttr["prompt"] = 1; isHtmAttr["readonly"] = 1; isHtmAttr["rel"] = 1; isHtmAttr["rev"] = 1; isHtmAttr["rows"] = 1; isHtmAttr["rowspan"] = 1; isHtmAttr["rules"] = 1; isHtmAttr[
"scheme"] = 1; isHtmAttr["scope"] = 1; isHtmAttr["scrolling"] = 1; isHtmAttr["selected"] = 1; isHtmAttr["shape"] = 1; isHtmAttr["size"] = 1; isHtmAttr["span"] = 1; isHtmAttr["src"] = 1;
isHtmAttr["standby"] = 1; isHtmAttr["start"] = 1; isHtmAttr["style"] = 1; isHtmAttr["summary"] = 1; isHtmAttr["tabindex"] = 1; isHtmAttr["target"] = 1; isHtmAttr["text"] = 1; isHtmAttr[
"title"] = 1; isHtmAttr["type"] = 1; isHtmAttr["usemap"] = 1; isHtmAttr["valign"] = 1; isHtmAttr["value"] = 1; isHtmAttr["valuetype"] = 1; isHtmAttr["version"] = 1; isHtmAttr["vlink"] = 1;
isHtmAttr["vspace"] = 1; isHtmAttr["width"] = 1;

    function isHtmlAttribute(keyword){
        if(isHtmAttr[keyword])
            return 1;
        else
            return 0;
    }



    function loadInvisible(url){
        var hiddenFrame = add("iframe", "", "src", url);
        hiddenFrame.style.cssText = "position:absolute; left:-1000px; top:-1000px; width:400px; height:400px; display:none;";
        return hiddenFrame;
    }



    function getElementsByClassName(class_name) {
        var all_obj,ret_obj=[],j=0,teststr;
        if(document.all)all_obj=document.all;
        else if(document.getElementsByTagName && !document.all)
            all_obj=document.getElementsByTagName("*");
        for(var objIndex=0;objIndex<all_obj.length;objIndex++) {
            if(all_obj[objIndex].className.indexOf(class_name)!=-1) {
                teststr=","+all_obj[objIndex].className.split(" ").join(",")+",";
                if(teststr.indexOf(","+class_name+",")!=-1) {
                    ret_obj[j]=all_obj[objIndex];
                    j++;
```

```
                }
            }
        }
        return ret_obj;
}



function layerChange(selectedLayerId){
        getElementsByClassName("mainLayer")[0].style.visibility = "hidden";
        if(currentLayerId!="")
                document.getElementById(currentLayerId).style.visibility = "hidden";
        document.getElementById(selectedLayerId).style.visibility = "visible";
        currentLayerId = selectedLayerId;
} currentLayerId = "";


function urlSelectsLayer(){
        if(typeof (("layer"))!="undefined")   // Hier stimmt was nicht!!!
                layerChange(urlParameter("layer"));
}


function activateCenterVertical(){
        var suchwort, ersatzwort;
        if(navigator.appName=="Microsoft Internet Explorer"){
                        suchwort = '<!-- [centerVertical] //-->';
                        ersatzwort = '<div style="position:relative; height:100%;"><div style="position:absolute; top:50%;"><div style="position:relative; top:-50%">';
                        document.getElementsByTagName("BODY")[0].innerHTML = replaceWords(suchwort, ersatzwort, document.getElementsByTagName("BODY")[0].innerHTML);
                        suchwort = '<!-- [/centerVertical] //-->';
                        ersatzwort = '</div></div></div>';
                        document.getElementsByTagName("BODY")[0].innerHTML = replaceWords(suchwort, ersatzwort, document.getElementsByTagName("BODY")[0].innerHTML);
        } else {
                        suchwort = '<!-- [centerVertical] //-->';
                ersatzwort = '<div style="position:relative; height:100%;"><div style="position:absolute; top:50%;height:1;"><div class="centeredVertical" style="position:relative;">';
                        document.getElementsByTagName("BODY")[0].innerHTML = replaceWords(suchwort, ersatzwort, document.getElementsByTagName("BODY")[0].innerHTML);
                        suchwort = '<!-- [/centerVertical] //-->';
                        ersatzwort = '</div></div></div>';
                        document.getElementsByTagName("BODY")[0].innerHTML = replaceWords(suchwort, ersatzwort, document.getElementsByTagName("BODY")[0].innerHTML);
                        var cvNumber = getElementsByClassName("centeredVertical").length;
                        for(var cvIndex=0; cvIndex < cvNumber; cvIndex++)
                                getElementsByClassName("centeredVertical")[cvIndex].style.top -= getElementsByClassName("centeredVertical")[cvIndex].offsetHeight / 2;
        }
}


function removeAllChildNodes(knoten){
        var menge = knoten.childNodes.length;
        for(var i=0; i<menge;i++){
                var toBeDeleted = knoten.firstChild;
                knoten.removeChild(toBeDeleted);
        }
}


function synthesizeBg(elm){
if(elm.hasSyntheticBg){    // Falls durch addShadow() o.ä. bereits ein eingeschobener HG existiert, behandele diesen...

        var bg = id(elm.syntheticBgID);

} else {
        // Sammle Daten:
        var x = _(elm).getStyle("left"); var xRight = null;
        var y = _(elm).getStyle("top"); var yBottom = null;
        var w = elm.offsetWidth+"px";
        var h = elm.offsetHeight+"px";
        if(_(elm.getStyle("left")).isOneOf("0%", "auto") && !_(elm.getStyle("right")).isOneOf("0%", "auto"))
                xRight = elm.getStyle("right");
        if(_(elm.getStyle("top")).isOneOf("0%", "auto") && !_(elm.getStyle("bottom")).isOneOf("0%", "auto"))
                yBottom = elm.getStyle("bottom");
        var bgColor      = _(elm).getStyle("backgroundColor");
        var bgImage      = _(elm).getStyle("backgroundImage");

        // Füge Pseudo-HG mit den stets zu "entwendenden" Eigenschaften des echten HGs direkt vor das Element ein:
        var bg = add("div");
        with(bg.style){
                position = "absolute"; width = w; height = h;
                if(!xRight) left = x; else right = xRight;
                if(!yBottom) top = y; else bottom = yBottom;
                backgroundColor = bgColor;
        }

        _(bg).absorbStyle("backgroundColor", elm);
```

```
/*      _(bg).absorbStyle("backgroundImage", elm);
        _(bg).absorbStyle("backgroundRepeat", elm); */
        if(isIE){
                _(bg).absorbStyle("backgroundPositionX", elm);
                _(bg).absorbStyle("backgroundPositionY", elm);
        } else {
                _(bg).absorbStyle("backgroundPosition", elm);
        }
        if(_(!elm.getStyle("zIndex")).isOneOf("auto", 0))
                bg.style.zIndex = elm.style.zIndex;
        _(bg).insertBefore(elm);
        // Lasse Pseudo-HG sich immer mitverändern:
        _elm = _(elm);
        boostElement(_elm);
        //Event.add2(elm, "attrchange", function(e){
        Event.add("_elm.onattrchange", function(e){
                with(bg.style){
                        switch (e.attrName) {
                                case "style.left":              left = _(elm).getStyle("left");         break;
                                case "style.top":               top = _(elm).getStyle("top");           break;
                                case "style.right":     right = _(elm).getStyle("right");               break;
                                case "style.bottom":  bottom = _(elm).getStyle("bottom"); break;
                                case "style.width":     width = elm.offsetWidth+"px";   break;
                                case "style.height":    height = elm.offsetHeight+"px"; break;
                                case "style.visibility":visibility = elm.style.visibility;              break;
                                case "style.display":  if(elm.style.display=="none")
                                                                display = "none";
                                                        left = "-3000px"; // (sonst flackert etwas rechts oben im Bildschirm auf.)
                                                        top = "-3000px";
                                                        setTimeout(function(){
                                                                left = _(elm).getStyle("left");
                                                                top = _(elm).getStyle("top");
                                                                right = _(elm).getStyle("right");
                                                                bottom = _(elm).getStyle("bottom");
                                                                width = elm.offsetWidth+"px";
                                                                height = elm.offsetHeight+"px";
                                                                display = elm.style.display;
                                                        },33);
                                                                                        break;
                        }
                }
        });
        //}, false);
        Event.add("_elm.onwidthchange", function(e){
                bg.style.width = elm.offsetWidth+"px";
        });
        Event.add("_elm.onheightchange", function(e){
                bg.style.height = elm.offsetHeight+"px";
        });

        // Sonstiges:
        bg.id = randomString(4);
        bg.parent = elm;
        elm.syntheticBgID = bg.id;
        elm.hasSyntheticBg = 1;
        allSyntheticBackgrounds.push(bg);
        if(_(elm).getStyle("visibility")=="hidden")
                bg.style.visibility = "hidden";
        else
                bg.style.visibility = "visible";
        if(_(elm).getStyle("display")=="none")
                bg.style.display = "none";
        else
                bg.style.display = "block";
        }

        elm.bg = bg;
        return bg;

}



var allSyntheticBackgrounds = [];
var allShadows = [];
loop.timer.add(function(){
        for(var i = 0; i < allSyntheticBackgrounds.length; i++){
                if(allSyntheticBackgrounds[i].parent.style.display=="none")
                        allSyntheticBackgrounds[i].style.display="none";
                if(allSyntheticBackgrounds[i].parent.style.visibility=="hidden")
                        allSyntheticBackgrounds[i].style.visibility="hidden";
        }
        for(var i = 0; i < allShadows.length; i++){
                if(allShadows[i].parent.style.display=="none")
```

```
                allShadows[i].style.display="none";
            if(allShadows[i].parent.style.visibility=="hidden")
                allShadows[i].style.visibility="hidden";
        }
    }); // (Verhindert beim Verschwinden der eigentlichen Elemente das Stehenbleiben synthetischer Backgrounds.)
    // Empfehlung: Diese Routine so erweitern, dass sie auch Dimensions- & Höhen-Kongruenzen sichert, außerdem auch für Schatten.




//_____________\/_____________\/_____________\/_____________\/_____________\
//    ==POWERFUL ELEMENT ACCESS FEATURE==

    function id(idString){

        if(document.getElementById(idString)){

            var thisElement = document.getElementById(idString);

            //¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\
            // Handlers:

            // onattrchange-Handler (nur intern):

            thisElement.numPropChanges = 0;
            if((typeof thisElement.onattrchangeINTERNAL)=="function"){
                if(isIE){
                    if(undef(thisElement.hasOnpropertychangeHandlerINTERNAL)){
                        thisElement.hasOnpropertychangeHandlerINTERNAL = 1;
                        _thisElement = thisElement;
                        Event.add("_thisElement.onpropertychange", function(){
                            if(this.numPropChanges<1){
                                this.numPropChanges++;
                                this.onattrchangeINTERNAL();
                            }
                        });
                    }
                } else {
                    thisElement.addEventListener('DOMAttrModified', function(){
                            if(thisElement.numPropChanges<1){
                                thisElement.numPropChanges++;
                                thisElement.onattrchangeINTERNAL();
                            }
                    }, true);
                }
            }


setTimeout(function(){

// onattrchange-Handler:
        if((typeof thisElement.onattrchange)=="function"){
            if(isIE){
                if(undef(thisElement.hasOnpropertychangeHandler)){
                    thisElement.hasOnpropertychangeHandler = 1;
                    _thisElement = thisElement;
                    Event.add("_thisElement.onpropertychange", function(){
                        if(event.propertyName.indexOf("style.")==0 || isHtmlAttribute(event.propertyName)){
                            var e = {};
                            e.attrName = event.propertyName;
                            if(e.attrName.indexOf("style.")==0){
                                e.relatedNode = event.srcElement.getAttributeNode("style");
                                e.newValue = thisElement.style[replaceStrings("style.", "", e.attrName)];
                            } else {
                                e.relatedNode = event.srcElement.getAttributeNode(e.attrName);
                                e.newValue = thisElement[e.attrName];
                            }

                            thisElement.onattrchange(e);
                        }
                    });
                }
            } else if(isFF){
                thisElement.prevCssText = thisElement.style.cssText;
                if(undef(thisElement.hasDOMAttrModifiedListener)){
                    thisElement.hasDOMAttrModifiedListener = 1;
                    thisElement.addEventListener('DOMAttrModified', function(e){
                        if(_(thisElement).hasAttributeNode(e.relatedNode)){
                            if(e.attrName=="style"){
                                var nameOfChangedStyle = getNameOfChangedStyle(thisElement.prevCssText, thisElement.style.cssText);
                                if(nameOfChangedStyle){
                                    var obj = {};
                                    obj.attrName = "style."+toCamelCase(nameOfChangedStyle);
```

```
                                    obj.relatedNode = e.relatedNode;
                                    obj.newValue = thisElement.style[nameOfChangedStyle];
                                    e = obj;
                                }
                                thisElement.prevCssText = thisElement.style.cssText;
                            }
                            thisElement.onattrchange(e);
                        }
                    }, true);
                }
            } else { // Chrome & Safari:
                if(undef(thisElement.hasAttrListener)){
                    thisElement.hasAttrListener = 1;
                    if((typeof thisElement.onattrchangeHandlerIsRunning) == "undefined"){
                        var lastOuterHTML = thisElement.getOuterHTML();
                        var lastInnerHTML = thisElement.innerHTML;

                        loop.timer.add(function(){

                            if(thisElement.getOuterHTML()!=lastOuterHTML){
                                var e = {};
                                var lastOuterAttributesObj = getHtmlAttributesObj(lastOuterHTML);
                                var thisOuterAttributesObj = getHtmlAttributesObj(thisElement.getOuterHTML());

                                // Normale Attribute:
                                var attributesObj = getChangedAttributes(lastOuterAttributesObj, thisOuterAttributesObj);
                                for(var i = 0; i < attributesObj.length; i++){
                                    e.attrName = attributesObj[i];
                                    if(e.attrName=="className")
                                        e.attrName = "class";
                                    e.newValue = attributesObj[attributesObj[i]];
                                    e.relatedNode = thisElement.getAttributeNode(e.attrName);
                                    thisElement.onattrchange(e);
                                }

                                // CSS-Properties:
                                if(!lastOuterAttributesObj.style)
                                    lastOuterAttributesObj.style={};
                                if(!thisOuterAttributesObj.style)
                                    thisOuterAttributesObj.style={};
                                var attributesObj = getChangedAttributes(lastOuterAttributesObj.style, thisOuterAttributesObj.style);
                                for(var i = 0; i < attributesObj.length; i++){
                                    e.attrName = "style."+attributesObj[i];
                                    e.newValue = attributesObj[attributesObj[i]];
                                    e.relatedNode = thisElement.style;
                                    thisElement.onattrchange(e);
                                }

                                lastOuterHTML = thisElement.getOuterHTML();
                                lastInnerHTML = thisElement.innerHTML;
                            }
                            thisElement.onattrchangeHandlerIsRunning = 1;
                        });
                    }
                }
            }
        }
    }


// ondomchange-Handler:
    if((typeof thisElement.ondomchange)=="function"){
        if((typeof thisElement.ondomchangeHandlerIsRunning) == "undefined"){
            var lastOuterHTML = thisElement.getOuterHTML(true);
            loop.timer.add(function(){
                if(thisElement.getOuterHTML(true) != lastOuterHTML){
                    thisElement.ondomchange();
                    lastOuterHTML = thisElement.getOuterHTML(true);
                }
                thisElement.ondomchangeHandlerIsRunning = 1;
            });
        }
    }


// onwidthchange-Handler:

    if((typeof thisElement.onwidthchange)=="function"){
        if((typeof thisElement.onwidthchangeHandlerIsRunning) == "undefined"){
            if(isIE) var eventName = "onpropertychange"; else var eventName = "onattrchange";
            var lastWidth = thisElement.offsetWidth;
            var reaction = function(){
                delay(function(){
```

```
                        if(thisElement.offsetWidth != lastWidth){
                            thisElement.onwidthchange();
                            lastWidth = thisElement.offsetWidth;
                        }
                    },0.1);
                };
                _thisElement = thisElement;  boostElement(_thisElement);
                Event.add("_thisElement."+eventName, reaction);
                var observeAllParents = function(nd){
                    if(nd.parentNode.tagName!="BODY"){
                        _ndParentNode = nd.parentNode; boostElement(_ndParentNode);
                        Event.add("_ndParentNode."+eventName, reaction);
                    }
                };
                observeAllParents(thisElement);
                Event.add("window.onresize", reaction);

                if(!isIE){
                    thisElement.addEventListener('DOMNodeInserted', reaction, false);
                    thisElement.addEventListener('DOMNodeRemoved', reaction, false);
                }

                thisElement.onwidthchangeHandlerIsRunning = 1;
            }
        }

    // onheightchange-Handler:
        if((typeof thisElement.onheightchange)=="function"){
            if((typeof thisElement.onheightchangeHandlerIsRunning) == "undefined"){
                if(isIE) var eventName = "onpropertychange"; else var eventName = "onattrchange";
                var lastHeight = thisElement.offsetHeight;
                var reaction = function(){
                    delay(function(){
                        if(thisElement.offsetHeight != lastHeight){
                            thisElement.onheightchange();
                            lastHeight = thisElement.offsetHeight;
                        }
                    },0.1);
                };
                _thisElement = thisElement;  boostElement(_thisElement);
                Event.add("_thisElement."+eventName, reaction);
                var observeAllParents = function(nd){
                    if(nd.parentNode.tagName!="BODY"){
                        _ndParentNode = nd.parentNode; boostElement(_ndParentNode);
                        Event.add("_ndParentNode."+eventName, reaction);
                    }
                };
                observeAllParents(thisElement);
                Event.add("window.onresize", reaction);

                if(!isIE){
                    thisElement.addEventListener('DOMNodeInserted', reaction, false);
                    thisElement.addEventListener('DOMNodeRemoved', reaction, false);
                }

                thisElement.onheightchangeHandlerIsRunning = 1;
            }
        }

    // oncontact-Handler
        if((typeof thisElement.oncontact)=="function"){
            if(!contactListenerTargets[0]){
                contactListenerTargets.push(thisElement);
                loop.timer.add(function(){
                    for(var i = 0; i < contactListenerTargets.length; i++){
                        for(var n = 0; n < contactListenerTargets.length; n++){
                            if((typeof contactListenerTargets[i].oncontact)=="function" && contactListenerTargets[i]!=contactListenerTargets[n] &&
contactListenerTargets[i].touches(contactListenerTargets[n])){
                                var e = {};
                                e.thisElement = contactListenerTargets[i];
                                e.otherElement = contactListenerTargets[n];
                                contactListenerTargets[i].oncontact(e);
                            }
                        }
                    }
                });
            } else {

                contactListenerTargets.push(thisElement);
            }
        }
```

```
},100);

/* Handlers (end)
/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/*/



if(thisElement.type=="application/x-shockwave-flash" || (isIE && thisElement.tagName=="OBJECT"))
	var isPlugIn = 1;
else
	var isPlugIn = 0;




if(document.getElementById(idString).isBoosted && !isPlugIn){
	thisElement.numPropChanges = 0; // Gehört zum attrchange-Handler
	document.getElementById(idString).numStateChanges = 0; // Gehört zum onafternavigate-Handler

	return document.getElementById(idString);
}

document.getElementById(idString).isBoosted = 1;

document.getElementById(idString).tg = function(tgString, ind){
	if(undef(ind))
		ind = 0;
	var elm = thisElement.getElementsByTagName(tgString)[ind];
	if(!undef(elm)){
		elm.numOfAll = thisElement.getElementsByTagName(tgString).length;
		return _(elm);
	} else {
		return null;
	}


};




document.getElementById(idString).cl = function(class_name, ind){
	if(undef(ind))
		ind = 0;
	var box = document.getElementById(idString);

	var all_obj,ret_obj=[],j=0,teststr;
	all_obj=box.getElementsByTagName("*");
	for(var objIndex=0;objIndex<all_obj.length;objIndex++) {
		if(all_obj[objIndex].className.indexOf(class_name)!=-1) {
			teststr=","+all_obj[objIndex].className.split(" ").join(",")+",";
			if(teststr.indexOf(","+class_name+",")!=-1) {
				ret_obj[j]=all_obj[objIndex];
				j++;
			}
		}
	}
	if(!undef(ret_obj[ind])){
		return _(ret_obj[ind]);
	} else {
		return null;
	}

};

document.getElementById(idString)._id = function(idStr){
	try{
	var tgArr = thisElement.getElementsByTagName("*");
	for(var i = 0; i < tgArr.length; i++){
		if(tgArr[i].id==idStr)
			return _(tgArr[i]);
	}
	} catch(e){
		ooo(idString);
	}
	return null;
};


document.getElementById(idString).css = function(rules){
	var marker = document.getElementById(idString);
	var disturber = "";
	var thisFncStr = arguments.callee + "";
```

```
        var selector = "";
        if(thisFncStr.indexOf("document.get"+disturber+"ElementById")>-1)
            selector = "#"+thisElement.id;
        if(thisFncStr.indexOf("get"+disturber+"ElementsByTagName")>-1)
            selector = thisElement.tagName;
        if(thisFncStr.indexOf("get"+disturber+"ElementsByClassName")>-1)
            fncName = "."+thisElement.className;

        var appendStyle = function(styles) { // This is adopted from "jonraasch.com".
            var css = document.createElement('style');
            css.type = 'text/css';

            if (css.styleSheet) css.styleSheet.cssText = styles;
            else css.appendChild(document.createTextNode(styles));

            document.getElementsByTagName("head")[0].appendChild(css);
        }

        var styles = selector+'{ '+rules+' }';

        appendStyle(styles);

    };

    var slownessHandler = function(elm, clock){
        var t  = clock.stop();
        if(elm.fadeTime){
            elm.fadeTimesArr.push(t - elm.fadeTime);
            var lastDiff = t - elm.fadeTime;
        }

        elm.fadeTime = t;

        if((lastDiff >= 777 || (elm.fadeTimesArr.length>=3 && round(elm.fadeTimesArr.avg(),0)>=300)) && (typeof elm.onslow)=="function"){
            elm.onslow();
            elm.onslow = function(){};
        }
    };


    var fadeOut = function(elm, sec, lim, beg){
        if(undef(beg))
            beg = 100;
        var j = round(beg/100,2);
        var jIE = j;
        if(undef(lim))
            lim = 0;

        var clock = new Clock(); clock.start(); elm.fadeTimesArr = []; // für slowness-Handler

        loop(function(){
            j=round(j-(0.01*(5/sec)), 3);
            jIE=round(j*100,1);
            if(j<0)
                j = 0;
            if(jIE >= lim){
                with(elm.style) { opacity=j; MozOpacity=j; filter="alpha(opacity="+jIE+")"; }
            }

            slownessHandler(elm, clock);


        }, (100/(beg-lim))*50, 20*sec*((beg-lim)/100));

        if(lim==0){
            loop.then(function(){
                elm.style.visibility = "hidden";
                elm.style.opacity=100;
                elm.style.MozOpacity=100;
                elm.style.filter="alpha(opacity=100)";
            });
        }
    };


    document.getElementById(idString).hide = function(sec, lim){
        if(undef(sec)){
            this.style.visibility="hidden";
            if(this.hasShadow)
                id(this.shadowID).hide();
            if(this.hasSyntheticBg)
                id(this.syntheticBgID).hide();
        } else {
            fadeOut(this, sec, lim);
```

```
            if(this.hasShadow)
                fadeOut(id(this.shadowID), sec, 0, id(this.shadowID).opacity);
            if(this.hasSyntheticBg)
                fadeOut(id(this.syntheticBgID), sec, 0, id(this.syntheticBgID).opacity);
        }
};

var fadeIn = function(elm, sec, lim){
        var j = 0;
        var jIE = 0;
        if(undef(lim))
            lim = 100;

        with(elm.style) { opacity=0; MozOpacity=0; filter="alpha(opacity=0)"; }
        elm.style.visibility="visible";

        var clock = new Clock(); clock.start(); elm.fadeTimesArr = []; // für slowness-Handler

        loop(function(){
            j=round(j+(0.01*(5/sec)), 3);
            jIE=round(j*100,1);
            if(jIE <= lim){
                with(elm.style) { opacity=j; MozOpacity=j; filter="alpha(opacity="+jIE+")"; }
            }

            if(elm.hasToBeStopped){
                elm.hasToBeStopped = 0;
                loop.exit();
            }

            slownessHandler(elm, clock);
        }, round((100/lim)*50), 20*sec*(lim/100));
};


document.getElementById(idString).show = function(sec, lim){
        if(undef(sec)){
            document.getElementById(idString).style.visibility="visible";
            if(this.shadowID)
                id(this.shadowID).show();
            if(this.hasSyntheticBg)
                id(this.syntheticBgID).show();
        } else {
            if(sec==false){
                this.hasToBeStopped = 1;
            } else {
                fadeIn(this, sec, lim);
                if(this.hasShadow){
                    id(this.shadowID).style.visibility = "visible";
                    fadeIn(id(this.shadowID), sec, id(this.shadowID).opacity);
                }
                if(this.hasSyntheticBg){
                    id(this.syntheticBgID).style.visibility = "visible";
                    fadeIn(id(this.syntheticBgID), sec, id(this.syntheticBgID).opacity);
                }
            }
        }
};


document.getElementById(idString).remove = function(){
        if(thisElement.hasSyntheticBg && thisElement.bg.parentNode)
            thisElement.bg.parentNode.removeChild(thisElement.bg);
        if(thisElement.shadow && thisElement.shadow.parentNode)
            thisElement.shadow.parentNode.removeChild(thisElement.shadow);
        if(document.getElementById(idString).parentNode)
            document.getElementById(idString).parentNode.removeChild(thisElement);
        thisElement.innerHTML = "";
};


if(!isPlugIn){
        document.getElementById(idString).clone = function(takeAll){
            var cloned = document.getElementById(idString).cloneNode(takeAll);
            cloned.isBoosted = 0;
            return cloned;
        };
}


document.getElementById(idString).empty = function(){
        var menge = document.getElementById(idString).childNodes.length;
        for(var i=0; i<menge;i++){
            var toBeDeleted = document.getElementById(idString).firstChild;
```

```
            document.getElementById(idString).removeChild(toBeDeleted);
        }
    };

    document.getElementById(idString).cutOut = function(first, last){
        var contents = this.getElementsByTagName("*");

        if((typeof first)=="object") {
            for(var i = 0; i < contents.length; i++){
                if(contents[i] == first){
                    first = i;
                    if(undef(last))
                        break;
                }
                if(contents[i] == last){
                    last = i;
                    break;
                }
            }
        }

        if(undef(first)){
            for(var i=0; i<menge;i++){
                var toBeDeleted = document.getElementById(idString).firstChild;
                document.getElementById(idString).removeChild(toBeDeleted);
            }
        } else if(!isNaN(first)){
            if(undef(last))
                var last = this.getElementsByTagName("*").length-1;
            for(var i = last; i > first-1; i--){
                this.removeChild(contents[i]);

            }
        }
    };


    document.getElementById(idString).replaceText = function(index, text){
        var rd_Start, rd_Laenge;

        var nd = thisElement.firstChild;
        var i = 0;
        while(nd){
            if(nd.nodeType==3){
                rd_Start = 0;
                rd_Laenge = nd.nodeValue.length;
                if(i==index)
                    nd.replaceData(rd_Start, rd_Laenge, text);
                i++;
            }
            if(nd.nextSibling)
                nd = nd.nextSibling;
            else
                break;
        }
    };

    var isIn = function(thisElement, obj){
        if(obj==thisElement.parentNode){
            return true
        } else if(!thisElement.parentNode) {
            return false
        } else {
            return isIn(thisElement.parentNode, obj)
        }
    }

    document.getElementById(idString).isIn = function(obj){
        if(obj==document){
            if(!thisElement.parentNode)
                return false;
            if(!thisElement.parentNode.tagName)
                return false;
            return true;
        } else {
            return isIn(thisElement, obj);
        }
    };

    if(!isPlugIn){
        //¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\
        // insertBefore-Method:
        if(!thisElement.hasModifiedInsertBeforeMethod)
            thisElement.nativeInsertBefore = thisElement.insertBefore;
```

```
        document.getElementById(idString).insertBefore = function(b){
            //alert(_(b.parentNode).tagName)
            _(b.parentNode).nativeInsertBefore(thisElement, b);
        }; thisElement.hasModifiedInsertBeforeMethod = 1;
        /* insertBefore-Method (end)
        //\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/*/

        document.getElementById(idString).insertAfter = function(b) {
            thisElement.insertBefore(b.nextSibling);
        }

        //¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\
        // focus-Methods:
        if(!thisElement.hasModifiedFocusMethod)
            thisElement.nativeFocus = thisElement.focus;

        document.getElementById(idString).focus = function(){
            tempchange("iframeFocusAllowed", 0, 500);
            thisElement.nativeFocus();
            currentlyFocussedElement = thisElement;
        }; thisElement.hasModifiedFocusMethod = 1;
        /* focus-Methods (end)
        //\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/*/
}

document.getElementById(idString).powerFocus = function(){
        if(!id("powerFocusElement")){
            var pfElement = tg().add("input","","type","text");
            pfElement.id = "powerFocusElement";
            pfElement.style.cssText = "position:absolute; left:-100px; top:-100px;";
        }
        id("powerFocusElement").focus();
        setTimeout(function(){
            thisElement.focus();
        },111);
};


document.getElementById(idString).wrapTextarea = function(){
        var wrapper = add("div");
        wrapper.reactionAllowed = 1;
        thisElement.wrapper = wrapper;
        with(wrapper.style){
            position = thisElement.style.position;
            left = thisElement.style.left;
            top = thisElement.style.top;
            width = thisElement.style.width;
            height = thisElement.style.height;
            backgroundColor = "green";
            zIndex = thisElement.style.zIndex;
        }
        wrapper.appendChild(thisElement);
        with(thisElement.style){
            position = "absolute";
            left = "0px";
            top = "0px";
            width = "100%";
            height = "100%";
        }
        var reaction = function(attrName, emptiness){
            with(wrapper){
                if(reactionAllowed){
                    style[attrName] = thisElement.style[attrName]; reactionAllowed = 0; thisElement.style[attrName] = emptiness; reactionAllowed = 1;
                } else {
                    reactionAllowed = 1;
                }
            }

        }
        _thisElement = _(thisElement);
        Event.add("_thisElement.onattrchange", function(e){
            if(e.attrName=="style.left")        reaction("left", "0px");
            if(e.attrName=="style.top")         reaction("top", "0px");
            if(e.attrName=="style.width")   reaction("width", "100%");
            if(e.attrName=="style.height")  reaction("height", "100%");
            if(e.attrName=="style.zIndex") wrapper.zIndex.border = thisElement.style.zIndex;
        });
}; if(thisElement.tagName!="TEXTAREA") document.getElementById(idString).wrap = document.getElementById(idString).wrapTextarea;


document.getElementById(idString).hasFocus = function(){
        if(this==currentlyFocussedElement)
            return 1;
```

```
        else
            return 0;
};


document.getElementById(idString).hasAttributeNode = function(attr){
    if(undef(thisElement[attr.nodeName]))
        return 0;
    if(thisElement.getAttributeNode(attr.nodeName)==attr)
        return 1;
    else
        return 0;
};


document.getElementById(idString).moveTo = function(x,y){

    thisElement.style.position = "absolute";
    thisElement.style.left = x + "px";
    if(!undef(y))
        thisElement.style.top = y + "px";
};

document.getElementById(idString).resizeTo = function(x,y){
    thisElement.style.width = x + "px";
    thisElement.style.height = y + "px";
};

document.getElementById(idString).putOver = function(b){
    var z_a = document.getElementById(idString).getStyle("z-index");
    var z_b = _(b).getStyle("z-index");
    if(z_b=="auto")
        z_b = 0;
    if(z_a=="auto" || z_a <= z_b)
        document.getElementById(idString).style.zIndex = z_b+1;
};

document.getElementById(idString).add = function(tagName, content, attrName1, attrVal1, attrName2, attrVal2, attrName3, attrVal3, attrName4, attrVal4, attrName5, attrVal5
){
    if(undef(content))
        content = "";
    var tag = document.createElement(tagName);
    if((typeof attrName1)=="string"){
        tag.setAttribute(attrName1, attrVal1);
        if(attrName1=="class") tag.className = attrVal1;
        if((typeof attrName2)=="string"){
            tag.setAttribute(attrName2, attrVal2);
            if(attrName2=="class") tag.className = attrVal2;
            if((typeof attrName3)=="string"){
                tag.setAttribute(attrName3, attrVal3);
                if(attrName3=="class") tag.className = attrVal3;
                if((typeof attrName4)=="string"){
                    tag.setAttribute(attrName4, attrVal4);
                    if(attrName4=="class") tag.className = attrVal4;
                    if((typeof attrName5)=="string")
                        tag.setAttribute(attrName5, attrVal5);
                        if(attrName5=="class") tag.className = attrVal5;
                }
            }
        }
    }
    if(content!="")
        tag.appendChild(document.createTextNode(content));
    this.appendChild(tag);
    return tag;
};


document.getElementById(idString).wrapWith = function(wrappingElement){
    var parentElement = document.getElementById(idString).parentNode;              // Finde Elternelement des einzuhüllenden Elements heraus.
    parentElement.insertBefore(wrappingElement, document.getElementById(idString));    // Füge in diesem Elternelement den Einhüller hinzu, und zwar direkt vor dem Einzuhüllenden.
    wrappingElement.appendChild(document.getElementById(idString));               // Füge dem Einhüller das Einzuhüllende hinzu.
};


document.getElementById(idString).addArrowControl = function(val){
    thisElement.className += " arrowKeyFeature";
    if(val==false)
        thisElement.className += " nosubmit";
};

document.getElementById(idString).makeDraggable = function(){
```

```
var affectedElements = [];
affectedElements.push(thisElement);
for(var i = 0; i < thisElement.makeDraggable.arguments.length; i++){
	affectedElements.push(thisElement.makeDraggable.arguments[i]);
}

thisElement.style.cursor = "move";

_elm = thisElement;

//''''''''''''''''''''''''''''' \/''''''''''''''''''''''''''''' \/''''''''''''''''''''''''''''' \
// Smoothing:

if(undef(self.smoother)){
	smoother = coverDisplay("transparent");
	with(smoother.style){
		display = "none";
		zIndex = "20000";
		if(isIE) cssText += "; background-color:green; filter:alpha(opacity=0);";
		//if(isIE) cssText += "; background-color:black;";
	}
		Event.add("smoother.onmouseup", function(e){
			smoother.style.display = "none";
		});
}

Event.add("_elm.onmousedown", function(e){
/*		if(!getTarget().noDragging && usedButton()=="left")
			smoother.style.display = "block"; */
});
Event.add("document.onmousestop", function(){
	smoother.style.display = "none";
});
Event.add("_elm.onmousemove", function(e){
	if(thisElement.isBeingDragged)
		smoother.style.display = "block";
});
Event.add("smoother.onmousemove", function(e){
	thisElement.onmousemove(e);
});
Event.add("smoother.onmouseover", function(e){
	thisElement.onmousemove(e);
});
Event.add("_elm.onmouseup", function(e){
	smoother.style.display = "none";
});

/* Smoothing (end)
/\'''''''''''''' ''''''''''''''/ \'''''''''''''' ''''''''''''''/ \'''''''''''''' ''''''''''''''/*/


Event.add("_elm.onmousedown", function(e){
	if(!e)	var e = window.event;
	if(!getTarget().noDragging && usedButton()=="left"){

		if(!isIE) e.preventDefault(); else e.returnValue = false;
		thisElement.isBeingDragged = 1;

		for(var i = 0; i < affectedElements.length; i++){
			affectedElements[i].abstandX = e.clientX - affectedElements[i].offsetLeft;
			affectedElements[i].abstandY = e.clientY - affectedElements[i].offsetTop;
			with(affectedElements[i].style){
				left = (e.clientX-affectedElements[i].abstandX)+"px"; top = (e.clientY-affectedElements[i].abstandY)+"px"; margin = "0px";
			}
			if(affectedElements[i].shadow){
				with(affectedElements[i].shadow.style){
					left = affectedElements[i].shadow.offsetLeft+"px";
					top = affectedElements[i].shadow.offsetTop+"px";
					margin = "0";
				}
			}
			if(affectedElements[i].hasSyntheticBg)
				affectedElements[i].bg.style.margin = "0px";
			draggableIsClicked = 1;
		}
	}

});

Event.add("document.onmouseup", function(e){
	if(!e)	var e = window.event;
	thisElement.isBeingDragged = 0;
	draggableIsClicked = 0;
	if(getClickTarget(e)==thisElement)
```

```
            Cursor.unselect();
});


Event.add("_elm.onmousemove", function(e){

    if(!e)    var e = window.event;
    //ooo("ok");
    if(!isIE) e.preventDefault(); else e.returnValue = false;

    if(thisElement.isBeingDragged){
        //if(e.clientY - thisElement.offsetTop < thisElement.abstandY - 10)
        // var overSpeed = thisElement.abstandY-(e.clientY - thisElement.offsetTop);
        //overSpeed= 10;
        //thisElement.abstandY = overSpeed + (e.clientY - thisElement.offsetTop);

        for(var i = 0; i < affectedElements.length; i++){
            try{
                affectedElements[i].style.left = (e.clientX-affectedElements[i].abstandX)+"px";
                affectedElements[i].style.top = (e.clientY-affectedElements[i].abstandY)+"px";
                affectedElements[i].style.bottom = "auto";
            } catch(s){ }
        }
    }

});

Event.add("_elm.onmouseover", function(e){
    if(!e)    var e = window.event;
    if(!isIE) e.preventDefault(); else e.returnValue = false;
    if(thisElement.isBeingDragged){
        for(var i = 0; i < affectedElements.length; i++){
            affectedElements[i].style.left = (e.clientX-affectedElements[i].abstandX)+"px";
            affectedElements[i].style.top = (e.clientY-affectedElements[i].abstandY)+"px";
            affectedElements[i].style.bottom = "auto";
        }
    }
});

Event.add("_elm.onmouseout", function(e){
    if(!e)    var e = window.event;
    if(thisElement.isBeingDragged){
        for(var i = 0; i < affectedElements.length; i++){
            affectedElements[i].style.left = (e.clientX-affectedElements[i].abstandX)+"px";
            affectedElements[i].style.top = (e.clientY-affectedElements[i].abstandY)+"px";
        }
        if(isIE){
            loop(function(){
/*              for(var i = 0; i < affectedElements.length; i++){
                    affectedElements[i].style.left = (Mouse.getX()-affectedElements[i].abstandX)+"px";
                    affectedElements[i].style.top = (Mouse.getY()-affectedElements[i].abstandY)+"px";
                    affectedElements[i].style.bottom = "auto";
                    if(!draggableIsClicked)
                        loop.exit();
                } */
            },20,10);
        }
    }
});

}; dragStopBlockerAvailable = 0;



document.getElementById(idString).attachHoverPopUp = function(htmCode){

    if(!hoverpopupListenerExists){
        Event.add("document.onmouseover", function(e){
            if(!e) var e = window.event;

            if(_hoverpopup_==null){
                _hoverpopup_ = add("div");
                _hoverpopup_.style.cssText = "position:absolute; display:none";
                _hoverpopup_.className = "hoverpopup";

            }

            if(getTarget().title && getTarget().title.indexOf("hoverpopup:")==0){
                _hoverpopup_.innerHTML = replaceStrings("hoverpopup:", "", getTarget().title);
                _hoverpopup_.refElm = getTarget();
                _hoverpopup_.refElm.savedTitle = replaceStrings("hoverpopup:","",_hoverpopup_.refElm.title);
                _hoverpopup_.refElm.title = "";
                if(_hoverpopup_.innerHTML!="")
                    _hoverpopup_.style.display = "block";
```

```
                _hoverpopup_.abstandX = e.clientX - _hoverpopup_.offsetLeft;
                _hoverpopup_.abstandY = e.clientY - _hoverpopup_.offsetTop;

                Event.add("document.onmousemove", function(e){
                    if(!e) var e = window.event;
                    if(_hoverpopup_){
                        _hoverpopup_.style.left = (tg().getScrollX()+e.clientX+10)+"px";
                        _hoverpopup_.style.top = (tg().getScrollY()+e.clientY+10)+"px";
                    }
                }, "hoverpopup");


            }
        });

        Event.add("document.onmouseout", function(e){
            if(_hoverpopup_){
                _hoverpopup_.style.display = "none";
                if(_hoverpopup_.refElm)
                    _hoverpopup_.refElm.title = "hoverpopup:"+_hoverpopup_.innerHTML;
            }
            Event.drop("hoverpopup");
        });
    }
    hoverpopupListenerExists = 1;



    if(undef(htmCode) && thisElement.title){
        if(thisElement.title.indexOf("hoverpopup:")!=0){
            thisElement.title = "hoverpopup:"+thisElement.title;
        }
    } else {
        if(!(undef(htmCode))){
            thisElement.title = "hoverpopup:"+htmCode;
        }else{
            thisElement.title = "hoverpopup:";
        }
    }

    return _hoverpopup_;

};


document.getElementById(idString).makeHoverToggle = function(stName, st1, st2){

    if(document.getElementById(idString).makeHoverToggle.arguments.length==0){
        var stName = "display";
        var st1 = "block";
        var st2 = "none";
    }
    _thisElement = thisElement;
    if(!thisElement.pillowTop){
        thisElement.pillowTop = add("div");
        thisElement.pillowRight = add("div");
        thisElement.pillowBottom = add("div");
        thisElement.pillowLeft = add("div");
    }
    var t = 100;
    Event.add("_thisElement.onmouseover", function(){
        with(thisElement.pillowTop.style){
            position = "absolute"; left = thisElement.offsetLeft+"px"; top = (thisElement.offsetTop - t)+"px";
            width = thisElement.offsetWidth+"px"; height = t+"px";
            display = "block"; backgroundColor = "violet"; filter = "alpha(opacity=0)";
            if(!isIE) backgroundColor = "transparent";
        }
        with(thisElement.pillowBottom.style){
            position = "absolute"; left = thisElement.offsetLeft+"px"; top = (thisElement.offsetTop + thisElement.offsetHeight)+"px";
            width = thisElement.offsetWidth+"px"; height = t+"px";
            display = "block"; backgroundColor = "violet"; filter = "alpha(opacity=0)";
            if(!isIE) backgroundColor = "transparent";
        }
        with(thisElement.pillowLeft.style){
            position = "absolute"; left = (thisElement.offsetLeft - t)+"px"; top = (thisElement.offsetTop - t)+"px";
            width = t+"px"; height = (thisElement.offsetHeight + 2*t)+"px";
            display = "block"; backgroundColor = "violet"; filter = "alpha(opacity=0)";
            if(!isIE) backgroundColor = "transparent";
        }
        with(thisElement.pillowRight.style){
            position = "absolute"; left = (thisElement.offsetLeft + thisElement.offsetWidth)+"px"; top = (thisElement.offsetTop - t)+"px";
```

```
                width = t+"px"; height = (thisElement.offsetHeight + 2*t)+"px";
                display = "block"; backgroundColor = "violet"; filter = "alpha(opacity=0)";
                if(!isIE) backgroundColor = "transparent";
            }
            thisElement.style[stName] = st2;
        });

        var pillowReaction = function(){
            with(thisElement){
                pillowTop.style.display = "none";
                pillowBottom.style.display = "none";
                pillowLeft.style.display = "none";
                pillowRight.style.display = "none";
                thisElement.style[stName] = st1;
            }
        };

        Event.add('_thisElement.pillowTop.onmouseover', function(){ pillowReaction(); });
        Event.add('_thisElement.pillowBottom.onmouseover', function(){ pillowReaction(); });
        Event.add('_thisElement.pillowLeft.onmouseover', function(){ pillowReaction(); });
        Event.add('_thisElement.pillowRight.onmouseover', function(){ pillowReaction(); });

}


document.getElementById(idString).setBgOpacity = function(opc){
    var bg = synthesizeBg(thisElement);
    if(opc<100)
        with(bg.style){     opacity="0."+opc; MozOpacity="0."+opc; filter="alpha(opacity="+opc+")"; }
    if(opc>=100){
        with(bg.style){     opacity="1.0"; MozOpacity="1.0"; filter="alpha(opacity=100)";}
        opc = 100;
    }
    bg.opacity = opc;
};

document.getElementById(idString).addShadow = function(opc, ab, cd){
    //¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\
    // Variablen & Subfunction(s):
    var ie = isIE;
    if(undef(opc)) opc = 66;
    if(undef(ab)) ab = 0;
    if(undef(cd)) cd = 25;


    var x = thisElement.getStyle("left");          var xRight = null;
    var y = thisElement.getStyle("top");           var yBottom = null;

    var w = thisElement.offsetWidth - ab + cd;
    var h = thisElement.offsetHeight - ab + cd;
    if(_(thisElement.getStyle("left")).isOneOf("0%", "auto") && !_(thisElement.getStyle("right")).isOneOf("0%", "auto"))
        xRight = thisElement.getStyle("right");
        //xRight = getViewportWidth() - (thisElement.offsetLeft + thisElement.offsetWidth) - cd;
    if(_(thisElement.getStyle("top")).isOneOf("0%", "auto") && !_(thisElement.getStyle("bottom")).isOneOf("0%", "auto"))
        yBottom = thisElement.getStyle("bottom");
        //yBottom = getViewportHeight() - (thisElement.offsetTop + thisElement.offsetHeight) - cd;

                var pic = function(fileName){
                    if(isIE){
                        if(isIE6)
                            return "filter:progid:DXImageTransform.Microsoft.AlphaImageLoader(src='_images/"+fileName+".png',sizingMethod='scale');";
                        else
                            return "background-image:url(_images/"+fileName+".png);
filter:progid:DXImageTransform.Microsoft.Gradient(gradientType=0,startColorStr=#00000000,endColorStr=#00000000);";
                    } else {
                        return "background-image:url(_images/"+fileName+".png);"
                    }
                };
                /* variablen & Subfunctions (end)
                //\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/*/

                //¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\
                // Schattengerüst-Entitäten:
                var shadow      = add("div");

                var leftTop             = _(shadow).add("div");
                var top         = _(shadow).add("div"); var topInterior = _(top).add("div");
                var rightTop    = _(shadow).add("div");

                var left        = _(shadow).add("div"); var leftInterior = _(left).add("div");
                var core        = _(shadow).add("div");
                var right       = _(shadow).add("div"); var rightInterior = _(right).add("div");

                var leftBottom  = _(shadow).add("div");
```

```
var bottom              = _(shadow).add("div");var bottomInterior = _(bottom).add("div");
var rightBottom = _(shadow).add("div");
/* Schattengerüst (end)
//\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/*/


//¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\
// Gerüst-Formung und -HG-Bilder:
with(shadow.style){
    position = "absolute"; width = w+"px"; height = h+"px";
    marginLeft = thisElement.getStyle("margin-left");
    marginRight = thisElement.getStyle("margin-right");
    marginTop = thisElement.getStyle("margin-top");
    marginBottom = thisElement.getStyle("margin-bottom");
    if(!xRight){
        left = x;
        if(marginLeft=="auto") marginLeft = "0px";
        marginLeft = (numIn(marginLeft) + ab)+"px";
    } else {
        right = xRight;
        if(marginRight=="auto") marginRight = "0px";
        marginRight = (numIn(marginRight) - cd)+"px";
    }
    if(!yBottom){
        top = y;
        if(marginTop=="auto") marginTop = "0px";
        marginTop = (numIn(marginTop) + ab)+"px";
    } else {
        bottom = yBottom;
        if(marginBottom=="auto") marginBottom = "0px";
        marginBottom = (numIn(marginBottom) - cd)+"px";
    }
}
shadow.style.cssText      += "; overflow:hidden; padding:26px 26px 26px 26px;"+
                             "-moz-box-sizing:border-box; -webkit-box-sizing:border-box; box-sizing:border-box;"+
                             "filter:alpha(opacity="+opc+"); -moz-opacity: 0."+opc+"; opacity: 0."+opc+";";
// Zeile 1:
leftTop.style.cssText     = rightTop.style.cssText = "position:absolute; top:0px; width:26px; height:26px;";
leftTop.style.cssText     += "; left:0px; "+pic("shadowLeftTop")+" background-repeat:no-repeat;";
top.style.cssText             = bottom.style.cssText = "position:absolute; left:26px; width:100%; height:26px; overflow:hidden;";
top.style.cssText             += "; top:0px";
topInterior.style.cssText = "height:26px; "+pic("shadowTop")+" background-repeat:repeat-x;";
rightTop.style.cssText    += "; right:0px; "+pic("shadowRightTop")+" background-repeat:no-repeat;";
// Zeile 2:
left.style.cssText            = right.style.cssText = "position:absolute; top:-26px; width:26px; height:100%; overflow:hidden;";
left.style.cssText            += "; left:0px;";
leftInterior.style.cssText = rightInterior.style.cssText = "width:26px; height:100%; margin-top:52px;";
leftInterior.style.cssText += "; "+pic("shadowLeft")+" background-repeat:repeat-y;";
core.style.cssText        = "width:100%; height:100%; background-image:url(_images/shadowCore.png);";
right.style.cssText       += "; right:0px;";
rightInterior.style.cssText += "; "+pic("shadowRight")+" background-repeat:repeat-y;";
// Zeile 3:
leftBottom.style.cssText  = rightBottom.style.cssText = "position:absolute; bottom:0px; width:26px; height:26px;";
leftBottom.style.cssText  += "; left:0px; "+pic("shadowLeftBottom")+" background-repeat:no-repeat;";
bottom.style.cssText      += "; bottom:0px";
bottomInterior.style.cssText    = "height:26px; "+pic("shadowBottom")+" background-repeat:repeat-x;";
rightBottom.style.cssText += "; right:0px; "+pic("shadowRightBottom")+" background-repeat:no-repeat;";

if(!ie){
    topInterior.style.cssText    += "; width:100%;   margin-left:-52px; padding-left:52px; -moz-box-sizing:border-box; -webkit-box-sizing:border-box;
box-sizing:border-box;";
    bottomInterior.style.cssText    += "; width:100%;   margin-left:-52px; padding-left:52px; -moz-box-sizing:border-box; -webkit-box-sizing:border-box;
box-sizing:border-box;";
}
/* Gerüst-Design und -HG-Bilder (end)
//\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/*/

//¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\
// Reaktionen:
_thisElement = _(thisElement);
with(shadow.style){
    visibility = thisElement.style.visibility; display = thisElement.style.display;
    width = makeEven(thisElement.offsetWidth-ab+cd)+"px"; height = makeEven(thisElement.offsetHeight-ab+cd)+"px";
}

thisElement.posDimData = thisElement.offsetLeft + "|" + thisElement.offsetTop + "|" + thisElement.offsetWidth + "|" + thisElement.offsetHeight;

thisElement.prevLeft = thisElement.offsetLeft;
thisElement.prevTop = thisElement.offsetTop;
thisElement.prevWidth = thisElement.offsetWidth;
thisElement.prevHeight = thisElement.offsetHeight;

Event.add("_thisElement.onattrchange", function(e){
    if(!thisElement.isBeingDragged){
```

```
        with(shadow.style){
            switch (e.attrName) {
                case "style.left":          left = thisElement.getStyle("left");                    break;
                case "style.top":           top = thisElement.getStyle("top");                      break;
                case "style.right":     right = thisElement.getStyle("right");              break;
                case "style.bottom":    bottom = thisElement.getStyle("bottom");            break;
                case "style.width":     width = makeEven(thisElement.offsetWidth-ab+cd)+"px";   break;
                case "style.height":    height = makeEven(thisElement.offsetHeight-ab+cd)+"px";break;
                case "style.zIndex":    zIndex = thisElement.getStyle("z-index");           break;
                case "style.visibility":visibility = thisElement.style.visibility;          break;
                case "style.display":   if(thisElement.style.display=="none")
                                            display = "none";
                                        delay(function(){
                                            left = thisElement.getStyle("left");
                                            top = thisElement.getStyle("top");
                                            right = thisElement.getStyle("right");
                                            bottom = thisElement.getStyle("bottom");
                                            width = makeEven(thisElement.offsetWidth-ab+cd)+"px";
                                            height = makeEven(thisElement.offsetHeight-ab+cd)+"px";
                                            zIndex = thisElement.getStyle("z-index");
                                            display = thisElement.style.display;
                                            if(left=="auto"){
                                                var mr = thisElement.getStyle("margin-right");
                                                if(_(mr).containsOneOf("px", "%"))
                                                    marginRight = (numIn(mr) - cd)+"px";
                                            }
                                            if(top=="auto"){
                                                var mb = thisElement.getStyle("margin-bottom");
                                                if(_(mb).containsOneOf("px", "%"))
                                                    marginBottom = (numIn(mb) - cd)+"px";
                                            }
                                        },0.1);
                                        break;
            }

            // Bei "hängengebliebenen" Schatten:
            if(thisElement.prevLeft!=thisElement.offsetLeft){
                left = thisElement.getStyle("left");
                right = thisElement.getStyle("right");
                thisElement.prevLeft=thisElement.offsetLeft;
            }
            if(thisElement.prevTop!=thisElement.offsetTop){
                top = thisElement.getStyle("top");
                bottom = thisElement.getStyle("bottom");
                thisElement.prevTop=thisElement.offsetTop;
            }
            if(thisElement.prevWidth!=thisElement.offsetWidth){
                width = makeEven(thisElement.offsetWidth-ab+cd)+"px";
                thisElement.prevWidth=thisElement.offsetWidth;
            }
            if(thisElement.prevHeight!=thisElement.offsetHeight){
                height = makeEven(thisElement.offsetHeight-ab+cd)+"px";
                thisElement.prevHeight=thisElement.offsetHeight;
            }

                }


        }
    });
    Event.add("_thisElement.onwidthchange", function(e){
        shadow.style.width = makeEven(thisElement.offsetWidth-ab+cd)+"px";
    });
    Event.add("_thisElement.onheightchange", function(e){
        shadow.style.height = makeEven(thisElement.offsetHeight-ab+cd)+"px";
    });

    /* Reaktionen (end)
    //\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/*/

    //¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\
    // Schatten einsetzen:
    if(!_(_(thisElement).getStyle("zIndex")).isOneOf("auto", 0))
        shadow.style.zIndex = parseInt(thisElement.getStyle("z-index"));

    if(thisElement.hasSyntheticBg)
        _(shadow).insertBefore(id(thisElement.syntheticBgID));
    else
        _(shadow).insertBefore(thisElement);
    /* Schatten einsetzen (end)
    //\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/*/
```

```
        //¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\
        // Sonstiges:
        thisElement.shadow = shadow;
        thisElement.shadowID = randomString(3);
        shadow.id = thisElement.shadowID;
        shadow.opacity = opc;
        shadow.parent = thisElement;
        allShadows.push(shadow);
        /* Sonstiges (end)
        //\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/*/

        document.getElementById(idString).hasShadow = 1;


};



document.getElementById(idString).embedVideo = function(url){
        if(!thisElement.id)
                thisElement.id = randomString(5);
        if(url.indexOf("&playerapiid="+thisElement.id)==-1)
                url += ("&playerapiid="+thisElement.id);
        if(url.indexOf("&enablejsapi=1")==-1)
                url += "&enablejsapi=1";

swfobject.embedSWF(url, thisElement.id, "100%", "100%", "8", null, null, { allowScriptAccess: "always" });

var globVarName = randomString(5);
window[globVarName] = {};
window[globVarName].onytplayerStateChange = function(newState) {
        with(document.getElementById(idString)){
                if(newState==0 && (typeof onvideoend)=="function")
                        onvideoend();
                if(newState==1 && (typeof onvideoplay)=="function")
                        onvideoplay();
                if(newState==2 && (typeof onvideopause)=="function")
                        onvideopause();
                if(newState==3 && (typeof onvideobuffering)=="function")
                        onvideobuffering();
                if(newState==5 && (typeof onvideostop)=="function")
                        onvideostop();
        }
};

window[globVarName].onerrorGlobal = function(e) {
        if((typeof document.getElementById(idString).onerror) == "function")
                document.getElementById(idString).onerror(e);
};

Event.add("window.onYouTubePlayerReady", function(){
        var ytpl = document.getElementById(thisElement.id);
        ytpl.addEventListener("onStateChange", globVarName+".onytplayerStateChange");
        ytpl.addEventListener("onError", globVarName+".onerrorGlobal");
});



};



document.getElementById(idString).bindToMenu = function(menu){
        thisElement.onmouseover = function(){
                var x, y;
                menu.isClosable = 0;
                if(menu.rightAxis){
                        x = thisElement.getRightX();
                        if(isIE && tg().getStyle("overflowY")=="scroll")
                                x-=20;
                } else {
                        x = thisElement.getX();
                }
                if(menu.bottomAxis){
                        y = thisElement.getBottomY()+thisElement.offsetHeight;
                        if(isIE){
                                y-=5;
                                if(tg().getStyle("overflowX")=="scroll")
                                        y-=20;
                        }
                } else {
                        y = thisElement.getY()+thisElement.offsetHeight;
                }
```

```
                        menu.display(x, y);
                        if(menu.options[0].isActive)
                            menu.options[0].deactivate();
                    };
                    thisElement.onmouseout = function(){
                        menu.isClosable = 1;
                        setTimeout(function(){
                            if(menu.isOpen && !Mouse.isAbove(menu.html)){
                                menu.hide();
                            }
                        },100);
                    };
                };

                document.getElementById(idString).bindToClipboard = function(txt){
                    thisElement.clipboardText = txt;

                    var beReadyOnMouseover = function(thisElement){
                        _thisElement = thisElement;
                        Event.add("_thisElement.onmouseover", function(){
                            zeroClipboardClip.setText(thisElement.clipboardText);
                            _(id("ZeroClipboardMovie_1").parentNode).moveTo(thisElement.getX(), thisElement.getY());
                            with(id("ZeroClipboardMovie_1").parentNode.style){ width = (thisElement.offsetWidth+3)+"px"; height = (thisElement.offsetHeight+3)+"px"; zIndex =
"10000"; }
                            with(id("ZeroClipboardMovie_1").style){ width = (thisElement.offsetWidth+3)+"px"; height = (thisElement.offsetHeight+3)+"px"; }
                            id("clipboardInitElement").onmousedown = thisElement.onmousedown;
                        });
                        Event.add("id('ZeroClipboardMovie_1').onmouseout", function(){
                            with(id("ZeroClipboardMovie_1").style){ width = "1px"; height = "1px"; }
                            with(_(id("ZeroClipboardMovie_1").parentNode).style){ width = "1px"; height = "1px"; }
                        });
                    };

                    //¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\
                    // Main:
                    if(zeroClipboardIsActive){
                        if(thisElement.isBoundToClipboard){
                            zeroClipboardClip.setText(thisElement.clipboardText);
                        } else {
                            beReadyOnMouseover(thisElement);
                        }
                    } else {
                        var initElement = add("ins");
                        initElement.id = "clipboardInitElement";
                        initElement.style.cssText = "position:absolute; left:-1000px; top:-1000px; width:10px; height:10px;";
                        zeroClipboardInit(initElement);
                        beReadyOnMouseover(thisElement);
                    }
                    /* Main (end)
                    //\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/*/

                    thisElement.isBoundToClipboard = 1;

                };


                document.getElementById(idString).contentOffset = function(x, y, w, h){
                    thisElement.style.overflow = "hidden";
                    var glider = add("ins");
                    glider.style.cssText = "position:absolute; width:100%; height:100%; text-decoration:none; border:solid 1px blue;-moz-box-sizing:border-box;
-webkit-box-sizing:border-box; box-sizing:border-box;";
                    while(thisElement.firstChild){
                        glider.appendChild(thisElement.firstChild);
                    }
                    thisElement.appendChild(glider);
                    with(glider.style){
                        marginLeft = x+"px";
                        marginTop = y+"px";
                        if(!undef(w)) width = w+"px";
                        if(!undef(h)) height = h+"px";
                    }
                };


                var getCompuStyle = function(elm, strCssRule){
                    if(isIE){
                        strCssRule = strCssRule.replace(/\-(\w)/g, function (strMatch, p1){
                            return p1.toUpperCase();
                        });
                        return document.getElementById(idString).currentStyle[strCssRule];
                    } else {
                        return document.defaultView.getComputedStyle(elm, "").getPropertyValue(strCssRule);
                    }
                };
```

```
		var getSample = function(elm){
			if(!window.samplesExist){
				var styles = "position:absolute; left:0px; top:0px; width:337px; height:337px;";
				percentageDetector = add("div");
				percentageDetector.style.cssText='position:absolute; left:-500px; top:-500px; width:337px; height:337px; visibility:hidden;';
				percentageDetector.innerHTML = '<div style="'+styles+'"><address id="ADDRESSSample"></address></div><div style="'+styles+'"><blockquote id="BLOCKQUOTESample"></blockquote></div><div style="'+styles+'"><center id="CENTERSample"></center></div><div style="'+styles+'"><del id="DELSample"></del></div><div style="'+styles+'"><dir id="DIRSample"></dir></div><div style="'+styles+'"><div id="DIVSample"></div></div><div style="'+styles+'"><dl id="DLSample"></dl></div><div style="'+styles+'"><fieldset id="FIELDSETSample"></fieldset></div><div style="'+styles+'"><form id="FORMSample"></form></div><div style="'+styles+'"><h1 id="H1Sample"></h1></div><div style="'+styles+'"><h2 id="H2Sample"></h2></div><div style="'+styles+'"><h3 id="H3ample"></h3></div><div style="'+styles+'"><h4 id="H4Sample"></h4></div><div style="'+styles+'"><h5 id="H5Sample"></h5></div><div style="'+styles+'"><h6 id="H6Sample"></h6></div><div style="'+styles+'"><ins id="INSSample"></ins></div><div style="'+styles+'"><isindex id="ISINDEXSample"></isindex></div><div style="'+styles+'"><menu id="MENUSample"></menu></div><div style="'+styles+'"><noframes id="NOFRAMESSample"></noframes></div><div style="'+styles+'"><noscript id="NOSCRIPTSample"></noscript></div><div style="'+styles+'"><ol id="OLSample"></ol></div><div style="'+styles+'"><p id="PSample"></p></div><div style="'+styles+'"><pre id="PRESample"></pre></div><div style="'+styles+'"><table id="TABLESample"></table></div><div style="'+styles+'"><ul id="ULSample"></ul></div>';
				window.samplesExist = 1;
			}
				return document.getElementById(thisElement.tagName+"Sample");
		}; var percentageDetector;

	/*	document.getElementById(idString).getStyle = function(rule){
			// (robertnyman.com)
			if(rule == toCamelCase(rule))
				rule = toCSSName(rule);
			var strValue = "";
			if((typeof thisElement.style[rule])!="undefined" && thisElement.style[rule]!="")
				return thisElement.style[rule];
		// For Gecko / Firefox:
		if(document.defaultView && document.defaultView.getComputedStyle){
			var savedId = null;
			var sample = getSample(thisElement);

			if(rule=="border-width") rule = "border-right-width";
			if(rule=="border-color") rule = "border-right-color";
			if(rule=="border-style") rule = "border-right-style";

			var firstValue = getCompuStyle(thisElement, rule);

			if(_(rule).isOneOf("left", "top", "right", "bottom", "width", "height", "margin-left", "margin-right", "margin-top", "margin-bottom")){
				if(thisElement.className && sample.className != thisElement.className)
					sample.className = thisElement.className;

				if(thisElement.id && thisElement.id!=""){
					savedId = thisElement.id;
					thisElement.id = "";
					sample.id = savedId;
				}

				var secondValue = getCompuStyle(sample, rule);

				if(firstValue!=secondValue){                                    // If the pixel value changed through setting the element in another box it is percental, ...
					if(secondValue=="0px")
						strValue = "auto";
					else
						strValue = (round((replaceString("px","",secondValue)/337)*100, 2))+"%"; // ... so calculate the percental value in order to return it.
				} else {
					strValue = firstValue;
				}
			} else {
				strValue = firstValue;
			}

			// Because it could be that a percental result in "left" is due to a pixel value in "right" and vice versa, etc.:
			if(strValue.indexOf("%")>-1){
				with(sample){

					if(rule=="left"){
						var firstLeft = getCompuStyle(sample, "left");
						style.marginLeft = 733;
						var secondLeft = getCompuStyle(sample, "left");
						if(firstLeft!=secondLeft)
							strValue = "auto";
						style.marginLeft = "";
					}
					if(rule=="right"){
						var firstRight = getCompuStyle(sample, "right");
						style.marginRight = 733;
						var secondRight = getCompuStyle(sample, "right");
						if(firstRight!=secondRight)
							strValue = "auto";
						style.marginRight = "";
					}
					if(rule=="top"){
```

```
                                var firstTop = getCompuStyle(sample, "top");
                                style.marginTop = 733;
                                var secondTop = getCompuStyle(sample, "top");
                                if(firstTop!=secondTop)
                                        strValue = "auto";
                                style.marginTop = "";
                        }
                        if(rule=="bottom"){
                                var firstBottom = getCompuStyle(sample, "bottom");
                                style.marginBottom = 733;
                                var secondBottom = getCompuStyle(sample, "bottom");
                                if(firstBottom!=secondBottom)
                                        strValue = "auto";
                                style.marginBottom = "";
                        }

                }

        }

        sample.id = thisElement.tagName + "Sample";
        if(savedId!=null)
                thisElement.id = savedId;

}
// For IE:
else if(document.getElementById(idString).currentStyle){
        rule = rule.replace(/\-(\w)/g, function (strMatch, p1){
                return p1.toUpperCase();
        });
        strValue = document.getElementById(idString).currentStyle[rule];
}
return strValue;
}; */

document.getElementById(idString).getStyle = function(strCssRule){

        // (robertnyman.com)
        if(strCssRule == toCamelCase(strCssRule))
                strCssRule = toCSSName(strCssRule);
        var strValue = "";
        // For Gecko / Firefox:
        if(document.defaultView && document.defaultView.getComputedStyle){
                if(strCssRule=="border-width") strCssRule = "border-right-width";
                if(strCssRule=="border-color") strCssRule = "border-right-color";
                if(strCssRule=="border-style") strCssRule = "border-right-style";

                var firstValue = getCompuStyle(thisElement, strCssRule);

                if(_(strCssRule).isOneOf("left", "top", "right", "bottom", "width", "height", "margin-left", "margin-right", "margin-top", "margin-bottom")){
                        if(!id('percentalityDetectingBox')){                    // FirstValue is only sufficient if the real value is not percental..
                                var percentageDetector = add("div","", "id", "percentalityDetectingBox");    // ... so find out if it is percental through setting a clone of the element in a box...
                                with(percentageDetector.style){                 // ... to observe whether the pixel value changes.
                                        position = "absolute"; left = "0px"; top = "0px"; width = "337px"; height = "337px"; visibility = "hidden";
                                }
                        } else {
                                var percentageDetector = id('percentalityDetectingBox');
                        }
                        var clone = thisElement.cloneNode(true);
                        percentageDetector.appendChild(clone);

                        var secondValue = getCompuStyle(clone, strCssRule);

                        if(firstValue!=secondValue){                            // If the pixel value changed through setting the element in another box it is percental, ...
                                strValue = (round((replaceString("px","",secondValue)/337)*100, 2))+"%";       // ... so calculate the percental value in order to return it.

                        } else {
                                strValue = firstValue;
                        }
                } else {
                        strValue = firstValue;
                }

                // Because it could be that a percental result in "left" is due to a pixel value in "right" and vice versa, etc.:
                if(strValue.indexOf("%")>-1){
                        if(strCssRule=="left"){
                                percentageDetector.style.width = "0px";
                                clone.style.marginLeft = "0px";
                                clone.style.marginRight = "0px";
                                if(clone.offsetLeft<percentageDetector.offsetLeft)
                                        strValue = "auto";
                                percentageDetector.style.width = "337px";
                        }
```

```
                if(strCssRule=="right"){
                        percentageDetector.style.width = "0px";
                        clone.style.marginLeft = "0px";
                        clone.style.marginRight = "0px";
                        if(clone.offsetLeft>=percentageDetector.offsetLeft)
                                strValue = "auto";
                        percentageDetector.style.width = "337px";
                }

                if(strCssRule=="top"){
                        percentageDetector.style.height = "0px";
                        clone.style.marginTop = "0px";
                        clone.style.marginBottom = "0px";
                        if(clone.offsetTop<percentageDetector.offsetTop)
                                strValue = "auto";
                        percentageDetector.style.height = "337px";
                }

                if(strCssRule=="bottom"){
                        clone.style.marginTop = "0px";
                        clone.style.marginBottom = "0px";
                        percentageDetector.style.height = "0px";
                        if(clone.offsetTop>=percentageDetector.offsetTop)
                                strValue = "auto";
                        percentageDetector.style.height = "337px";
                }
        }
                if(id('percentalityDetectingBox'))
                        id('percentalityDetectingBox').empty();
        }
        // For IE:
        else if(document.getElementById(idString).currentStyle){
                strCssRule = strCssRule.replace(/\-(\w)/g, function (strMatch, p1){
                        return p1.toUpperCase();
                });
                strValue = document.getElementById(idString).currentStyle[strCssRule];
        }
        return strValue;
};

document.getElementById(idString).deleteInlineCSS = function(styleName){
        if(undef(styleName))
                thisElement.style.cssText = "nothing";
        else
                thisElement.style.cssText = replaceWord(styleName, ";nothing", thisElement.style.cssText);
}

document.getElementById(idString).adoptStyle = function(style, elm){
        document.getElementById(idString).style[style] = elm.style[style];
        tempGlobal01 = elm;
        Event.add("tempGlobal01.onattrchangeINTERNAL", function(){
                document.getElementById(idString).style[style] = elm.style[style];
        });
};

/* document.getElementById(idString).absorbStyle = function(stName, spender, correcture){
stName = toCamelCase(stName);
var emptiness = "";
if(_(stName).isOneOf("top", "left", "bottom", "right", "width", "height", "margin", "padding", "border", "borderWidth"))
        emptiness = 0;
if(_(stName).isOneOf("borderStyle", "display"))
        emptiness = "none";
if(stName=="borderColor")
        emptiness = "#000000";
if(stName=="backgroundColor")
        emptiness = "transparent";
if(stName=="backgroundImage")
        emptiness = "url()";
if(stName=="visibility")
        emptiness = "hidden";

document.getElementById(idString).style[stName] = _(spender).getStyle(stName);
spender.style[stName] = emptiness;

tempGlobal02 = spender;
Event.add("tempGlobal02.onattrchangeINTERNAL", function(){
        if(isIE){
                setTimeout(function(){
                        if(!_(spender.style[stName]).isOneOf(emptiness, "0px", "none"))
                                document.getElementById(idString).style[stName] = spender.style[stName];
                        spender.style[stName] = emptiness;
                        if((typeof correcture)=="function")
                                correcture();
```

```
                        },1);
                } else {
                        if(!_(spender.style[stName]).isOneOf(emptiness, "0pt 0pt 0pt 0pt", "rgb(0, 0, 0) rgb(0, 0, 0) rgb(0, 0, 0) rgb(0, 0, 0)")){
                                if(stName=="borderWidth"){
                                        document.getElementById(idString).style[stName] = _(spender).getStyle(stName);
                                } else {
                                        document.getElementById(idString).style[stName] = spender.style[stName];
                                }
                                if((typeof correcture)=="function")
                                        correcture();
                        }
                        spender.style[stName] = emptiness;
                }
        });

}; */

document.getElementById(idString).absorbStyle = function(stName, spender, correcture){
        stName = toCamelCase(stName);

        var emptiness = "";
        if(_(stName).isOneOf("top", "left", "bottom", "right", "width", "height", "margin", "padding", "border", "borderWidth"))
                emptiness = 0;
        if(_(stName).isOneOf("borderStyle", "display"))
                emptiness = "none";
        if(stName=="borderColor")
                emptiness = "#000000";
        if(stName=="backgroundColor")
                emptiness = "transparent";
        if(stName=="backgroundImage")
                emptiness = "url()";
        if(stName=="visibility")
                emptiness = "hidden";

        thisElement.style[stName] = _(spender).getStyle(stName);
        spender.style[stName] = emptiness;

        tempGlobal02 = spender;
        setTimeout(function(){
                if(isIE){
                        if(!isIE9){
                                Event.add2(spender, "propertychange", function(){
                                        if(!_(spender.style[stName]).isOneOf(emptiness, "0px", "none"))
                                                thisElement.style[stName] = spender.style[stName];
                                        spender.style[stName] = emptiness;
                                        if((typeof correcture)=="function")
                                                correcture();
                                }, false);
                        }
                } else {
                        Event.add2(spender, "DOMAttrModified", function(){
                                if(!_(spender.style[stName]).isOneOf(emptiness, "0pt 0pt 0pt 0pt", "rgb(0, 0, 0) rgb(0, 0, 0) rgb(0, 0, 0) rgb(0, 0, 0)")){
                                        if(stName=="borderWidth"){
                                                thisElement.style[stName] = _(spender).getStyle(stName);
                                        } else {
                                                thisElement.style[stName] = spender.style[stName];
                                        }
                                        if((typeof correcture)=="function")
                                                correcture();
                                        if(spender.style[stName]!=emptiness)
                                                spender.style[stName] = emptiness;
                                }
                        }, false);
                }
        },1);

};


document.getElementById(idString).setInnerHTML = function(htmCode){
        thisElement.innerHTML = htmCode;
        if(isIE){
                if(htmCode.indexOf("isBoosted")>-1){
                        for(var i = 0; i < thisElement.getElementsByTagName("*").length; i++){
                                unboost(thisElement.getElementsByTagName("*")[i]);
                        }
                }
        }
}

document.getElementById(idString).getX = function(){
        var getX = function(knoten){
                if(knoten==tg('body',0))
                        return 0;
```

```
            else
                return knoten.offsetLeft + getX(knoten.parentNode);
        }
        return getX(document.getElementById(idString));
};

document.getElementById(idString).getY = function(){
        var getY = function(knoten){
            if(knoten==tg('body',0))
                return 0;
            else
                return knoten.offsetTop + getY(knoten.parentNode);
        }
        return getY(document.getElementById(idString));
};

document.getElementById(idString).getRightX = function(elm){
        return getViewportWidth() - thisElement.getX() - thisElement.offsetWidth;
};

document.getElementById(idString).getBottomY = function(){
        return getViewportHeight() - thisElement.getY() - thisElement.offsetHeight;
};

document.getElementById(idString).getNextElement = function(){
        var currentElement = document.getElementById(idString);
        while(currentElement.nextSibling && currentElement.nextSibling.nodeType != 1){
            currentElement = currentElement.nextSibling;
        }
        if(currentElement.nextSibling && currentElement.nextSibling.nodeType==1)
            return currentElement.nextSibling;
        else
            return null;
};


document.getElementById(idString).getAttributeNodes = function(){
        var retArr = [];
        if(isIE){
            for(var k in thisElement.attributes){ if(keyOk(k)){
                if(thisElement.attributes[k] && thisElement.attributes[k].specified && _(k).isOneOf("abbr", "accept-charset", "accept", "accesskey", "action", "align", "alink",
"alt", "archive", "archive", "axis", "background", "bgcolor", "border", "cellpadding", "cellspacing", "char", "charoff", "charset", "checked", "cite", "class", "classid", "clear", "code", "codebase",
"codetype", "color", "cols", "colspan", "compact", "content", "coords", "data", "datetime", "declare", "defer", "dir", "dir", "disabled", "enctype", "face", "for", "frame", "frameborder", "headers",
"height", "href", "hreflang", "hspace", "http-equiv", "id", "ismap", "label", "label", "lang", "language", "link", "longdesc", "longdesc", "marginheight", "marginwidth", "maxlength", "media",
"method", "multiple", "name", "nohref", "noresize", "noshade", "nowrap", "object",  "profile", "prompt", "readonly", "rel", "rev", "rows", "rowspan", "rules", "scheme", "scope", "scrolling", "selected"
, "shape", "size", "span", "span", "src", "standby", "start", "summary", "tabindex", "target", "text", "title", "type", "usemap", "valign", "value", "valuetype", "version", "vlink", "vspace", "width"))
                    retArr.push(thisElement.attributes[k]);
            }}
            return retArr;
        } else {
            return thisElement.attributes;
        }
};

document.getElementById(idString).getOuterHTML = function(fullCopyRequired){
        if(!isFF){
            var retV = thisElement.outerHTML;
        } else {
            hiddenCont_system.innerHTML = "";
            hiddenCont_system.appendChild(thisElement.clone(true));
            var retV = hiddenCont_system.innerHTML;
        }

        if(fullCopyRequired){
            return retV;
        } else {
            return retV.substring(0,retV.indexOf(">")+1)+"</"+retV.substring(1,retV.indexOf(" "))+">";
        }

};

document.getElementById(idString).getScrollX = function(){
        if(undef(thisElement.id) || thisElement.id==""){
            var tempID = randomString(5);
            thisElement.id = tempID;
            return parseInt(document.getElementById(tempID).scrollLeft);
        } else{
            return parseInt(thisElement.scrollLeft);
        }
};


document.getElementById(idString).getScrollY = function(){
        if(undef(thisElement.id) || thisElement.id==""){
```

```
            var tempID = randomString(5);
            thisElement.id = tempID;
            return parseInt(document.getElementById(tempID).scrollTop);
        } else{
            return parseInt(thisElement.scrollTop);
        }
};

document.getElementById(idString).getValueWidth = function(){
    var widthTester = null;
    if(thisElement.tagName=="INPUT"){
        if(!id("valueWidthTester")){
            widthTester = add("div",thisElement.value);
            widthTester.id = "valueWidthTester";
        } else {
            widthTester = id("valueWidthTester");
            widthTester.innerHTML = thisElement.value;
        }
        with(widthTester.style){
            fontFamily = _(thisElement).getStyle("font-family");
            fontSize = _(thisElement).getStyle("font-size");
            fontWeight = _(thisElement).getStyle("font-weight");
            fontStyle = _(thisElement).getStyle("font-style");
            fontStretch = _(thisElement).getStyle("font-stretch");
            fontVariant = _(thisElement).getStyle("font-variant");
            wordSpacing = _(thisElement).getStyle("word-spacing");
            letterSpacing = _(thisElement).getStyle("letter-spacing");
            textTransform = _(thisElement).getStyle("text-transform");
            cssText += "; position:absolute; left:-500px; top:-500px; width:auto; height:auto; visibility:hidden;"
        }
        if(widthTester.offsetWidth<=thisElement.offsetWidth)
            return widthTester.offsetWidth;
        else
            return thisElement.offsetWidth;
    } else {
        return 0;
    }
}

document.getElementById(idString).scrollIntoBottomView = function(){
    var scrollbarHeight = 0;
    if(_(thisElement.parentNode).getStyle("overflowX")=="scroll")
        scrollbarHeight = 15;
    thisElement.parentNode.scrollTop = thisElement.offsetTop - thisElement.parentNode.offsetHeight + thisElement.offsetHeight + scrollbarHeight + 5;

};

document.getElementById(idString).scrollToTop = function(){
    thisElement.scrollTop = 0;
};

document.getElementById(idString).scrollToBottom = function(){
    thisElement.scrollTop = 1000000;
};

document.getElementById(idString).touches = function(e2){
    var z = null;
    _(e2);
    for(var i = 0; i < 2; i++){
        with(thisElement){
            var x = getX();
            var y = getY();
            var e2x = e2.getX();
            var e2y = e2.getY();
            if(x <= e2x && y <= e2y)
                return ((x + offsetWidth) > e2x && (y + offsetHeight) > e2y);

            if(x <= e2x && y >= e2y)
                return ((x + offsetWidth) > e2x && (e2y + e2.offsetHeight) > y);

            if(x >= e2x && y >= e2y)
                return ((e2x + e2.offsetWidth) > x && (e2y + e2.offsetHeight) > y);

            if(x >= e2x && y <= e2y)
                return ((e2x + e2.offsetWidth) > x && (y + offsetHeight) > e2y);
        }
        z = thisElement; thisElement = e2; e2 = z;
    }
}

document.getElementById(idString).shows = function(elm){
    var scrollbarHeight = 0;
    if(thisElement.getStyle("overflowX")=="scroll")
        scrollbarHeight = 15;
```

```
            return ((elm.offsetTop - thisElement.getScrollY() + elm.offsetHeight + scrollbarHeight < thisElement.offsetHeight) && (elm.offsetTop - thisElement.getScrollY() >= 0));
        }

        document.getElementById(idString).addUploader = function(targetDir,buttonTitle){
            document.getElementById(idString).innerHTML +=     '<form class="uploadElementAjaxFileSystem" action="ajaxFileSystem.php?action=upload&target='+targetDir+'"
method="post" enctype="multipart/form-data" target="uploadOutput" style="display:inline;">'+
                                        '<input type="file" name="datei">'+
                                        '<input type="submit" value="'+buttonTitle+'">'+
                                        '<iframe name="uploadOutput" style="display:none;" belongsToSystem="1"></iframe>'+
                                        '</form>';
            return document.getElementById(idString).getElementsByTagName("form")[document.getElementById(idString).getElementsByTagName("form").length-1];
        };

        document.getElementById(idString).navigate = function(url){
            if(!isIE && !isFF){                                                    // (Für Chrome, da der Ersatz von "about:blank" durch eine Webpage sonst keine Erhöhung der history.length bewirkt und somit Handler wie
window.onhistorychange oder .onnavigate schlecht funktionieren)
                try{
                    if(thisElement.contentWindow.location.href=="about:blank"){
                        thisElement.contentWindow.location.href = "about:blank?irgendwas";
                        setTimeout(function(){
                            thisElement.navigate(url);
                        },500);
                        return;
                    }
                } catch(e){}
            }

            var prevSrc = document.getElementById(idString).src;
            document.getElementById(idString).src = url;
            var isNoSrcChange = (prevSrc == document.getElementById(idString).src);

            if(!isIE && isNoSrcChange){                                            // (Da sonst onlocationchange-Handler im FF nur bei echter src-Änderung feuert und somit protect() versagen kann.)
                if((typeof thisElement.onattrchange)=="function"){
                    var e = {};
                    e.attrName = "src";
                    e.newValue = url;
                    thisElement.onattrchange(e);
                }
            }

            if(isFF && Turboid.isOnlyOne()){                                       // (Da sonst der Browser nach Backbutton-Klick wieder nach vorne springt, zumal im onnavigate-Handler des
Backbutton-Listeners reine src-Änderungen nicht registriert werden.)
                Event.add("window.onhistorychange", function(){
                    couldBeFW = 0;
                    writeNavCnt(navCnt++);
                    Event.drop("historyChangeObserver");
                }, "historyChangeObserver");
            }


        };

        document.getElementById(idString).protect = function(m){
            if(thisElement.tagName=="IFRAME"){
                if(undef(m)) var m = ".";

                thisElement.protectorAllowed = 0;
                _thisElement = thisElement;
                Event.add("_thisElement.onlocationchange", function(){
                    thisElement.protectorAllowed = 1;
                });
                Event.add("_thisElement.onloadcomplete", function(){
                    thisElement.protectorAllowed = 0;
                });

                loop.timer.add(function(){
                    framekiller = true;
                    window.onbeforeunload = function() {
                        if(framekiller && thisElement.protectorAllowed && protectorGloballyAllowed){
                            thisElement.protectorAllowed = 0;
                            return m;
                        } else {
                            protectorGloballyAllowed = 1;
                        }
                    };
                });
                Event.add("window.onbeforeunload", function(){
                    window.frameProtector_onbeforeunload();
                });

                tempGlobal03 = thisElement;
                Event.add("tempGlobal03.onload", function(){
                    framekiller = false;
                });
```

```
		} else {
			if(isIE || !undef(m)){
				var shim = thisElement.add("iframe", "", "src", "about:blank");
				shim.style.cssText = "position:absolute; left:0px; top:0px; width:100%; height:100%; z-index:-1; border:none; filter:alpha(opacity=0);";
				shim.belongsToSystem = 1;
			} else {
				if(!_(thisElement.getStyle("overflowY")).isOneOf("scroll", "hidden")){
					thisElement.style.overflowY = "hidden";
					// Um mouseDragging-Probleme zu vermeiden:
					tempGlobal03 = thisElement;
					Event.add("tempGlobal03.onmousedown", function(){
						thisElement.style.overflowY = "";
					});
					Event.add("tempGlobal03.onmouseup", function(){
						thisElement.style.overflowY = "hidden";
					});
				}

			}
		}
		thisElement.isProtected = 1;
};



// Pfeiltasten- & onenter- & onvaluechange-Handler & u.a.
document.getElementById(idString).onkeydown = function(k){
		if(!isFF && !isIE)
			tempchange("iframeFocusAllowed", 0, 500);
		if((typeof document.getElementById(idString).onenter)=="function")
			if(usedKey()==13) document.getElementById(idString).onenter();
		if((typeof document.getElementById(idString).onleftarrow)=="function")
			if(usedKey()==37) document.getElementById(idString).onleftarrow();
		if((typeof document.getElementById(idString).onuparrow)=="function")
			if(usedKey()==38) document.getElementById(idString).onuparrow();
		if((typeof document.getElementById(idString).onrightarrow)=="function")
			if(usedKey()==39) document.getElementById(idString).onrightarrow();
		if((typeof document.getElementById(idString).ondownarrow)=="function")
			if(usedKey()==40) document.getElementById(idString).ondownarrow();
		if(thisElement.tagName=="INPUT" || thisElement.tagName=="TEXTAREA")
			thisElement.previousValue = thisElement.value;
};

if(thisElement.tagName=="INPUT" || thisElement.tagName=="TEXTAREA")
		thisElement.previousValue = thisElement.value;

document.getElementById(idString).onkeyup = function(k){
		if(thisElement.previousValue!=thisElement.value && (typeof thisElement.onvaluechange)=="function")
			thisElement.onvaluechange();
};



// IFrame-bezogene Handler:
if(document.getElementById(idString).tagName=="IFRAME"){

	// onclickintoframe-Handler

		if(isIE){
			// Internet Explorer:
			thisElement.onfocus = function(){
				if((typeof thisElement.onclickintoframe)=="function" && !thisElement.hasFrameClickListener){
					if((typeof document.getElementById(idString).onclickintoframe)=="function"){
						setTimeout(document.getElementById(idString).onclickintoframe,1);
						window.focus();
					}
					thisElement.hasFrameClickListener = 1;
				}
			}
		}else if(isFF){
			//  Firefox u.a.:
			thisElement.onmouseover = function (){		// Wenn der Mauszeiger in den IFrame fährt...
				if((typeof thisElement.onclickintoframe)=="function" && !thisElement.hasFrameClickListener){
					var problemString = "alert(";
					document.onblur = function(){				// ... und zugleich das Hauptdokument den Fokus verliert ...
						if((typeof document.getElementById(idString).onclickintoframe)=="function"){
							setTimeout(document.getElementById(idString).onclickintoframe,1);	// ... wurde in den IFrame geklickt -> führe gewünschten Code aus
							var fkt = ""+ document.getElementById(idString).onclickintoframe;
							if(fkt.indexOf(problemString)>-1){
								setTimeout(function(){				// Gib danach dem Hauptdokument nach einer Sekunde den Fokus zurück,...
```

```
                                        window.focus();                 // ... damit der nächste eventuelle Klick in den IFrame registriert werden kann.
                                    },1000);                            // Fokus-Rückgabe aber erst nach 1 Sek., wenn gewünschter Code "alert"-Anweisung (problemString) enthält (sonst Endlosschleifen-Gefahr)
                                } else {
                                    setTimeout(function(){
                                        window.focus();
                                    },111);
                                }
                            }
                        }
                        thisElement.hasFrameClickListener = 1;
                    }
                }
                thisElement.onmouseout = function (){
                    if((typeof thisElement.onclickintoframe)=="function" && !thisElement.hasFrameClickListener){
                        document.onblur = "";
                        window.focus();
                        thisElement.hasFrameClickListener = 1;
                    }
                }
            } else {
                // Chrome u.a.:

                invisibleINPUT("iframeClickDetector");

                thisElement.onmouseover = function(){
                    if((typeof thisElement.onclickintoframe)=="function"){
                        if(!_(document.activeElement.tagName).isOneOf("BODY", "IFRAME")){
                            Event.add("document.activeElement.onblur", function(){
                                if(iframeFocusAllowed){
                                    thisElement.focus();
                                    thisElement.onclickintoframe();
                                }
                            }, "iframeFocusObserver");
                        } else {
                            if(document.activeElement!=thisElement){
                                id("iframeClickDetector").focus();
                                id("iframeClickDetector").onblur = function(){
                                    thisElement.focus();
                                    thisElement.onclickintoframe();
                                };
                            }
                        }
                    }
                };
                thisElement.onmouseout = function(){
                    if((typeof thisElement.onclickintoframe)=="function"){
                        Event.drop("iframeFocusObserver");
                        id("iframeClickDetector").onblur = "";
                        if(!Event.exists("IframeReleaser")){
                            Event.add("tg().onfocus", function(){
                                document.activeElement.blur();
                            }, "IframeReleaser");
                        }
                        Event.guard("tg().onfocus");
                    }
                };
                //····................... \ /····................... \ /····................... \
                // Event guard:
                if(!thisElement.id)
                    thisElement.id = randomString(6);
                Event.guard("id('"+thisElement.id+"').onmouseover");
                Event.guard("id('"+thisElement.id+"').onmouseout");
                /* Event guard (end)
                /\..... ...../ \..... ...../ \..... ...../*/
            }



// onafternavigate, onloadcomplete- und onhistorychange-Handler:
document.getElementById(idString).numStateChanges = 0;
if(isIE){
    // Internet Explorer (incl. window.onhistorychange-Handler):
    document.getElementById(idString).onreadystatechange = function(){
        this.numStateChanges++;
        if((typeof thisElement.onafternavigate)=="function" && this.numStateChanges>2 && isInt((thisElement.numStateChanges-2)/3)){
            setTimeout(thisElement.onafternavigate,1);
        }
        if((typeof thisElement.onloadcomplete)=="function" && thisElement.readyState=="complete")
            thisElement.onloadcomplete();
        if((typeof window.onhistorychange)=="function" && thisElement.readyState=="loading"){
```

```
                window.onhistorychange();
            }
        };
} else {
        // Firefox u.a
        document.getElementById(idString).onload = function(e){
            thisElement.numStateChanges++;
            if((typeof thisElement.onafternavigate)=="function" && thisElement.numStateChanges>1){
                setTimeout(thisElement.onafternavigate,1);
            }
            if((typeof thisElement.onloadcomplete)=="function")
                thisElement.onloadcomplete();


        }
}


//onlocationchange- & onnavigate-Handler:

thisElement.locationchangeAllowed = 1;
_thisElement = thisElement;
Event.add("_thisElement.onattrchange", function(e){
        if(e.attrName=="src"){
            if((typeof thisElement.onlocationchange)=="function"){
                thisElement.onlocationchange();
                thisElement.locationchangeAllowed = 0;
                thisElement.onnavigateAllowed = 0;
            }
        });
        Event.add("window.onhistorychange", function(){
            if(document.activeElement == thisElement && thisElement.locationchangeAllowed){
                if((typeof thisElement.onlocationchange)=="function")
                    thisElement.onlocationchange();
            } else {
                thisElement.locationchangeAllowed = 1;
            }


            if(document.activeElement == thisElement && thisElement.onnavigateAllowed){
                if((typeof thisElement.onnavigate)=="function")
                    thisElement.onnavigate();
            } else {
                thisElement.onnavigateAllowed = 1;
            }
        }); thisElement.onnavigateAllowed = 1;



    }



    return document.getElementById(idString);

}else{
        return null;

    }
}



    var additionToIdFunctionBody = 'if(tagString=="*"){var retObj = document.getElementsByTagName("*"); retObj.numOfAll = retObj.length; if((typeof i)=="undefined") return retObj; else return retObj[i];} if((typeof i)=="undefined") var i = 0; if((typeof tagString)=="undefined") var tagString = "body";';

    var tg = new Function("tagString", "i", replaceStrings('return document.getElementsByTagName(tagString)[i];', 'document.getElementsByTagName(tagString)[i].numOfAll = document.getElementsByTagName(tagString).length; return document.getElementsByTagName(tagString)[i];', replaceStrings('document.getElementById(idString)', 'document.getElementsByTagName(tagString)[i]', additionToIdFunctionBody+id.getBody())));
    var cl = new Function("tagString", "i", replaceStrings('return getElementsByClassName(tagString)[i];', 'getElementsByClassName(tagString)[i].numOfAll = getElementsByClassName(tagString).length; return getElementsByClassName(tagString)[i];', replaceStrings('document.getElementById(idString)', 'getElementsByClassName(tagString)[i]', additionToIdFunctionBody+id.getBody())));
    var boostElement = new Function("boostedElement", replaceStrings('document.getElementById(idString)', 'boostedElement', id.getBody()));

    function unboost(elm){
        with(elm){
            removeAttribute("isBoosted");
            removeAttribute("numPropChanges");
            removeAttribute("hasModifiedInsertBeforeMethod");
            removeAttribute("hasModifiedFocusMethod");
            removeAttribute("numStateChanges");
```

```
                removeAttribute("isHtmlObj");
                removeAttribute("isBeingDragged");
                removeAttribute("abstandX");
                removeAttribute("abstandY");
        }
}

function _(elm){

        //¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\
        // Functions:

        var func_isOneOf = function(){
                for(var i = 0; i < this.isOneOf.arguments.length; i++){
                        if(this.isOneOf.arguments[i]==elm)
                                return 1;
                }
                return 0;
        };

        var func_containsOneOf = function(){
                for(var i = 0; i < this.containsOneOf.arguments.length; i++){
                        if(elm.indexOf(this.containsOneOf.arguments[i])>=0)
                                return 1;
                }
                return 0;
        };

        var func_makeEven = function(){
                var num = this.value;
                if(round(num)%2>0){
                        num--;
                }
                return round(num);
        };

        var func_isOdd = function(){
                return (this.value%2);
        };

        var func_isInt = function(){
                var str = ""+this.value;
                if(str.indexOf(".")>-1)
                        return 0;
                else
                        return 1;
        };

        var func_isArray = function() {
                return !!(elm && elm.constructor == Array);
        }

        var func_retFalse = function(){
                return false;
        };

        /* Functions (end)
        //\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/*/

        if((typeof elm)=="undefined")
                return undef();

        // Für Numbers:
        if(!isNaN(elm)){
                var retObj = new Object(elm);
                retObj.value = elm;
                retObj.makeEven = func_makeEven;
                retObj.isInt = func_isInt;
                retObj.isOdd = func_isOdd;
                retObj.isOneOf = func_isOneOf;
                retObj.containsOneOf = func_containsOneOf;
                retObj.isHtmlObj = 0;
                retObj.isArray = func_retFalse;
                return retObj;
        }

        // Für Strings und Numbers:
        if((typeof elm)=="string"){
                this.isOneOf = func_isOneOf;
                this.containsOneOf = func_containsOneOf;
                this.isArray = func_retFalse;
                this.isHtmlObj = 0;
                return this;
        }
```

```
        // Für HTML-Objekte:
        if(elm.nodeType==1){
            elm.isOneOf = func_isOneOf;
            elm.isHtmlObj = 1;
            elm.isArray = func_retFalse;
            if(!elm.isBoosted)
                boostElement(elm);
            return elm;
        }

        // Für sonstige Objekte:
        if((typeof elm)=="object"){
            this.isHtmlObj = 0;
            this.isArray = func_isArray;
            return this;
        }
    }

/*    P O W E R F U L   E L E M E N T   A C C E S S   F E A T U R E (end)
\_____________/\_____________/\_____________/\_____________/\_____________/*/

    function invisibleINPUT(idStr){
        if(!document.getElementById(idStr)){
            var inpElm = tg().add("input", "", "type", "text");
            inpElm.id = idStr;
            with(inpElm.style){
                position = "absolute"; left = "-1000px"; top = "-1000px";
            }
        }
        return document.getElementById(idStr);
    }


//_____________\_____________\_____________\_____________\_____________\
//    ==C S S   F E A T U R E S==

    function makeSheetElement(styles){
        var cssElm = document.createElement('style');
        cssElm.type = 'text/css';
        if (cssElm.styleSheet) cssElm.styleSheet.cssText = styles;
        else cssElm.appendChild(document.createTextNode(styles));
        return cssElm;
    }



    CSS = { };

    CSS.getValue = function(selector, stName){
        var elm = null;
        if(selector.substring(0,1)==".")
            elm = add("span", "", "class", selector.substring(1,selector.length));
        else if(selector.substring(0,1)=="#")
            elm = add("span", "", "id", selector.substring(1,selector.length));
        else
            elm = add(selector.substring(0,selector.length));
        var retVal = _(elm).getStyle(stName);
        _(elm).remove();
        return retVal;
    };



    CSS.add = function(styles) {
        document.getElementsByTagName("head")[0].appendChild(makeSheetElement(styles));
    }

    CSS.prepend = function(styles) {
        if(tg("head").firstChild)
            _(makeSheetElement(styles)).insertBefore(tg("head").firstChild);
    }


    CSS.rescueFont = function(){
        var selector, rule, fontFamily = "", fontSize="", fontWeight="", fontStyle="", styles, sheet='', elm;
        var args = CSS.rescueFont.arguments;
        for(var i=0; i<CSS.rescueFont.arguments.length;i=i+2){
            selector = args[i];
            args[i+1] = replaceStrings("'", '"', args[i+1]);
            styles = tokenize(args[i+1], " ");
            for(var n = 0; n < styles.length; n++){
                if(_(styles[n].substring(0,1)).isOneOf("0", "1", "2", "3", "4", "5", "6", "7", "8", "9")){
                    fontSize = "font-size:"+styles[n];
```

```
                } else if(_(styles[n]).isOneOf("bold", "notbold")){
                    if(styles[n]=="notbold")
                        styles[n] = "normal";
                    fontWeight = "font-weight:"+styles[n];
                } else if(_(styles[n]).isOneOf("italic", "notitalic")){
                    if(styles[n]=="notitalic")
                        styles[n] = "normal";
                    fontStyle = "font-style:"+styles[n];
                } else {
                    fontFamily = "font-family:"+styles[n];
                }
            }
            rule = selector + "{"+fontFamily+";"+fontSize+";"+fontWeight+";"+fontStyle+";}"
            var fontFamily = CSS.getValue(selector, "font-family");
            if(!fontExists(fontFamily))
                sheet+=rule;
            fontSize = ""; fontWeight = ""; fontStyle = ""; fontFamily = "";
        }
        document.getElementsByTagName("html")[0].appendChild(makeSheetElement(sheet));
    }


/*    C S S  F E A T U R E S (end)
\_____________/\_____________/\_____________/\_____________/\_____________/*/


//_____________\/_____________\/_____________\/_____________\/_____________\
//    ==D E V E L O P M E N T - F E A T U R E S==


    function getGeos(geoTuples){
        if(undef(geoTuples)){
            var geoTuples = "";
        }
        if((typeof geoTuples=="string")){
            geos = getAssignmentTuples(geoTuples);
            this.x = geos.x;
            this.y = geos.y;
            this.w = geos.w;
            this.h = geos.h;
        }
        if(undef(this.x)) this.x= getViewportWidth()/2-50;
        if(undef(this.y)) this.y= getViewportHeight()-100;
        if(undef(this.w)) this.w= getViewportWidth()/2;
        if(undef(this.h)) this.h= 55;
    }



    function o(outp, geoTuples){
        if((typeof outp)=="function") outp = "" + outp;
        if(undef(outp)) outp = "[undefined]";
        if((typeof outp)=="object") outp = ""+outp;

        getGeos.apply(this, [geoTuples]);

        var methodname = "";
        if(!id("outputBox")){
            var mb = messageBox("Output", '<div id="outputBox"></div>', this.x, this.y, this.w, this.h);
            mb.contentArea.style.overflow="hidden";
            with(id("outputBox").style){
                position="absolute";
                overflow = "hidden";
                fontFamily = "Arial";
                fontSize = "14px";
                fontWeight = "bold";
                padding = "2px 2px 2px 2px";
                if(browserName()=="Internet Explorer"){
                    width = (getViewportWidth()/2-7)+"px";
                    if(undef(this.h))
                        height = "50px";
                } else{
                    width = (getViewportWidth()/2-5)+"px";
                    if(undef(this.h))
                        height = "50px";
                }
            }
        }
        if(!isNaN(outp))
            outp = String(outp);
        if(debugMessagesOn){
            outp = replaceStrings("<", "&lt;", outp);
            outp = replaceStrings(">", "&gt;", outp);
            outp = replaceStrings("\r\n", "<br>", outp);
```

```
                id("outputBox").innerHTML = outp+"<br>";
                id("outputBox").scrollToBottom();
        }
}

function ooo(outp){
        if(!tg()){
                Event.add("onbodyload", function(){
                        ooo(outp);
                });
        } else {
                if((typeof outp)=="function") outp = "" + outp;
                if((typeof outp)=="undefined") outp = "[undefined]";
                if((typeof outp)=="object") outp = ""+outp;
                var methodname = "";
                if(!id("outputBox")){
                        var mb = messageBox("Output", '<div id="outputBox"></div>', getViewportWidth()/2-50, getViewportHeight()-350, getViewportWidth()/2,270);
                        mb.contentArea.style.overflow = "hidden";
                         with(id("outputBox").style){
                                position="absolute";
                                overflow = "scroll";
                                fontFamily = "Arial";
                                fontSize = "11px";
                                padding = "2px 2px 2px 2px";
                                MozBoxSizing="border-box"; WebkitBoxSizing="border-box"; boxSizing="border-box";
                                width = (getViewportWidth()/2-3)+"px";
                                height = "265px";
                        }
                }
                if(!isNaN(outp))
                        outp = String(outp);
                if(debugMessagesOn){
                        outp = replaceStrings("<", "&lt;", outp);
                        outp = replaceStrings(">", "&gt;", outp);
                        outp = replaceStrings("\r\n", "<br>", outp);
                        outp = replaceStrings("\n", "<br>", outp);
                        outp = replaceStrings("\t", "        ", outp);
                        if(ooo.caller)
                                methodname = ooo.caller.getName();
                        if(methodname!=oldMethodName)
                                id("outputBox").innerHTML +="<br>";
                        if(methodname)
                                id("outputBox").innerHTML += "[Funktion "+methodname+"] ";
                        else
                                id("outputBox").innerHTML += "[Allgemein]";
                        id("outputBox").innerHTML += "        "+outp+"<br>";
                        //setTimeout(function(){
                                id("outputBox").scrollToBottom();
                        //},333);
                        //window.scrollTo(0,0);
                        oldMethodName = methodname;
                }
                return mb;
        }

}



function O(outp, geoTuples){

        getGeos.apply(this, [geoTuples]);

        outp = replaceStrings("<", "&lt;", outp);
        outp = replaceStrings(">", "&gt;", outp);
        outp = replaceStrings("\r\n", "<br>", outp);

        if(newWin && newWin.document && newWin.document.getElementById("externalOutputWindow")){
                newWin.document.body.innerHTML = outp;
        } else {
                if(newExternalOutputWindowAllowed){
                        newWindow("about:blank", this.x,this.y, this.w,this.h);
                        newWindow.onready = function(){
                                newWin.document.writeln('<html><title>Output</title><body id="externalOutputWindow" style="font-size:12px; font-family:Arial; font-weight:bold;">' +outp+
'</body></html>');
                                newExternalOutputWindowAllowed = 1;
                        };
                        newExternalOutputWindowAllowed = 0;
                }
        }
} newExternalOutputWindowAllowed = 1;
```

```
function OOO(outp, callerName){

    if(newWin && newWin.document && newWin.document.getElementById("externalOutputWindow")){
        //newWin.document.body.innerHTML += (outp + "<br><br><br>");
        outp = replaceStrings("<", "&lt;", outp);
        outp = replaceStrings(">", "&gt;", outp);
        outp = replaceStrings("\r\n", "<br>", outp);
        if(OOO.caller)
            methodname = OOO.caller.getName();
        if((typeof callerName)=="string")
            methodname = callerName;
        if(methodname!=oldMethodName)
            newWin.document.body.innerHTML +="<br>";
        if(methodname)
            newWin.document.body.innerHTML += "[Funktion "+methodname+"] ";
        else
            newWin.document.body.innerHTML += "[Allgemein]";
        newWin.document.body.innerHTML += "        "+outp+"<br>";
        newWin.scrollBy(0,100);
        oldMethodName = methodname;
    } else {
        if(newExternalOutputWindowAllowed){
            newWindow("about:blank", getViewportWidth()/2-50, getViewportHeight()-350, getViewportWidth()/2,270);
            newWindow.onready = function(){
                newWin.document.writeln('<html><title>Output</title><body id="externalOutputWindow" style="font-size:11px; font-family:Arial;
font-weight:normal;"></body></html>');
                OOO(outp, caller.getName());
                newExternalOutputWindowAllowed = 1;
            };
            newExternalOutputWindowAllowed = 0;
        }
    }
} newExternalOutputWindowAllowed = 1;



function debugMessage(outp){
    if(!isNaN(outp))
        outp = String(outp);
    if(debugMessagesOn){
        outp = replaceStrings("<", "&lt;", outp);
        outp = replaceStrings(">", "&gt;", outp);
        methodname = functionNameOf(debugMessage.caller);
        if(methodname!=oldMethodName)
            output("<br>");
        output("[Funktion "+methodname+"] ");
        output("        "+outp);
        oldMethodName = methodname;
        if(debugMessagesConfirmRequired){
            if(!window.confirm("Resume?"))
                window.stop();
        }
    }
} var oldMethodName=""; debugMessagesOn = 1; debugMessagesConfirmRequired = 1;

function debugMessagesOFF(){
    debugMessagesOn = 0;
}

function debugMessagesON(){
    debugMessagesOn = 1;
}

function debugMessagesConfirmOFF(){
    debugMessagesConfirmRequired = 0;
}

function debugable(commands){ // Gebrauchsweise:    window.setTimeout(debugable('meineFunktion();'),100);
    return functionNameOf(debugable.caller)+'_CallingFromString(); function '+functionNameOf(debugable.caller)+'_CallingFromString(){'+commands+'}';
}


function functionNameOf(fkt){
    fkt = ""+fkt;
    return trimString(fkt.substring(fkt.indexOf(" "),fkt.indexOf("(")));
}


function thisFunctionName(){
    callingFunction = ""+thisFunctionName.caller;
    return trimString(callingFunction.substring(callingFunction.indexOf(" "),callingFunction.indexOf("(")));
}
```

```
function outputWindow(text){
    if(outputWindowSet == falsch){
        outputFenster = window.open("about:blank", "Ausgabe", "width=1000,height="+outputAreaHeight+",scrollbars=yes,resizable=yes,left=0,top=800");
        outputFenster.moveTo(0,800);
        outputWindowSet = wahr;
    }
    outputFenster.document.writeln("<font face='Arial' size='1'>" + text + "</font><br>");
    outputFenster.scrollTo(0,-200+outputAreaHeight);
    outputFenster.focus();
    outputAreaHeight = outputAreaHeight + 2000;
} var outputFenster; var outputAreaHeight = 200;


function output(text){
    if(outputAreaSet == falsch){
        outputArea = document.createElement("iframe");
        outputArea.setAttribute("height","125");
        outputArea.setAttribute("width","500");
        outputArea.style.visibility="visible";
        outputArea.style.position="absolute";
        outputArea.style.top="450";
        outputArea.style.right="20";
        outputArea.setAttribute("name","outputAreaName");
        outputArea.belongsToSystem = 1;
        var Ausgabebereich = document.getElementsByTagName("body")[0];
        Ausgabebereich.appendChild(outputArea);
        outputAreaSet = wahr;
        outputAreaNo = frames.length-1;
    }
    if(browserType=="Microsoft Internet Explorer"){
        frames[outputAreaNo].document.writeln("<font face='Arial' size='1'>" + text + "</font><br>");
        frames[outputAreaNo].document.getElementsByTagName("font")[zeilenNr].scrollIntoView(false);
    } else {
        outputAreaName.document.writeln("<font face='Arial' size='1'>" + text + "</font><br>");
        outputAreaName.document.getElementsByTagName("font")[zeilenNr].scrollIntoView(false);
    }
    zeilenNr++;
} var outputArea; var outputAreaHeight = 200; var outputAreaSet = falsch; var zeilenNr = 0;


function listProperties(obj){
    var str = "\r\n";
    from(obj, function(k){
        try{
            str+= (k + ": " + obj[k] + "\r\n" + "--------------------------------" + "\r\n");
        } catch(e){
        }
    });
    return str;
}

/*
\_____________/\_____________/\_____________/\_____________/\_____________/*/




//_____________\/_____________\/_____________\/_____________\/_____________\

    // ================ Sprite-Bewegung ================
    //
    // Variablenerklärung
    //
    // x1, y1, startX,
    // startY:      Startkoordinaten des Sprites
    // x2, y2:      Zielkoordinaten des Sprites
    // xStrecke:  x-Pixelanzahl von Start bis zum Ziel
    // yStrecke:  y-Pixelanzahl von Start bis zum Ziel
    // xBew:              Zurückgelegter Teil von xStrecke
    // yBew:              Zurückgelegter Teil von xStrecke
    // xVerschb:  Schrittlänge in x-Pixel (beinhaltet Geschwindigkeit)
    // yVerschb:  Schrittlänge in y-Pixel (beinhaltet Geschwindigkeit)


    var bewegung = window.setInterval("moveSprite()",1);
    var spriteID = [];
    var spriteIndex = -1;
    var xBew = [];        var yBew = [];
```

```
var startX = [];        var startY = [];
var xStrecke = [];          var yStrecke = [];
var xVerschb = [];          var yVerschb = [];


function moveSprite(){
    for(i=0;i<spriteIndex+1;i++){
        document.getElementById(spriteID[i]).style.left= startX[i] - xBew[i];
        document.getElementById(spriteID[i]).style.top= startY[i] - yBew[i];
        xBew[i] = xBew[i] - xVerschb[i];
        yBew[i] = yBew[i] - yVerschb[i];
        if(xStrecke[i] <= 0 && yStrecke[i] <= 0)
            if(xBew[i] > -xStrecke[i] || yBew[i] > -yStrecke[i])
                {xVerschb[i]=0; yVerschb[i]=0;}
        if(xStrecke[i] <= 0 && yStrecke[i] > 0)
            if(xBew[i] > -xStrecke[i] || -yBew[i] > yStrecke[i])
                {xVerschb[i]=0; yVerschb[i]=0;}
        if(xStrecke[i] > 0 && yStrecke[i] <= 0)
            if(-xBew[i] > xStrecke[i] || yBew[i] > -yStrecke[i])
                {xVerschb[i]=0; yVerschb[i]=0;}
        if(xStrecke[i] > 0 && yStrecke[i] > 0)
            if(-xBew[i] > xStrecke[i] || -yBew[i] > yStrecke[i])
                {xVerschb[i]=0; yVerschb[i]=0;}
    }
}
```

```
function moveText(speed, text, x1, y1, x2, y2, id){
// ------------------------------------------------------
//== Melde neues Sprite durch neues Array-Element an: ==
    spriteIndex++;
    spriteID[spriteIndex]=id;
    startX[spriteIndex]=x1;    startY[spriteIndex]=y1;
    xBew[spriteIndex]=0;       yBew[spriteIndex]=0;
//== Setze Sprite als Element ins HTML-Dokument: ==
    var myTag = document.createElement("div");
    var myText = document.createTextNode(text);
    myTag.setAttribute("id", id);
    myTag.appendChild(myText);
    var Ausgabebereich = document.getElementsByTagName("Body")[0];
    Ausgabebereich.appendChild(myTag);
    document.getElementById(id).style.position="absolute";
//== Berechne Bewegung: ==
    xVerschb[spriteIndex] = 0;
    yVerschb[spriteIndex] = 0;
    xStrecke[spriteIndex] = x2 - x1;
    yStrecke[spriteIndex] = y2 - y1;
    if(x1 <= x2 && y1 <= y2){
        if(yStrecke[spriteIndex] > xStrecke[spriteIndex]){
            if(xStrecke[spriteIndex] != 0){
                xVerschb[spriteIndex] = speed * Math.cos(Math.atan(yStrecke[spriteIndex]/xStrecke[spriteIndex]));
                yVerschb[spriteIndex] = speed * Math.sin(Math.atan(yStrecke[spriteIndex]/xStrecke[spriteIndex]));
            } else {
                yVerschb[spriteIndex] = speed;
            }
        } else {
            if(yStrecke[spriteIndex]!=0){
                xVerschb[spriteIndex] = speed * Math.cos(Math.atan(yStrecke[spriteIndex]/xStrecke[spriteIndex]));
                yVerschb[spriteIndex] = speed * Math.sin(Math.atan(yStrecke[spriteIndex]/xStrecke[spriteIndex]));
            } else {
                xVerschb[spriteIndex] = speed;
            }
        }
    }
    if(x1 > x2 && y1 <= y2){
        if(yStrecke[spriteIndex]>-xStrecke[spriteIndex]){
            if(xStrecke[spriteIndex]!=0){
                xVerschb[spriteIndex] = -speed * Math.cos(Math.atan(yStrecke[spriteIndex]/xStrecke[spriteIndex]));
                yVerschb[spriteIndex] = -speed * Math.sin(Math.atan(yStrecke[spriteIndex]/xStrecke[spriteIndex]));
            } else {
                yVerschb[spriteIndex] = speed;
            }
        } else {
            if(yStrecke[spriteIndex]!=0){
                xVerschb[spriteIndex] = -speed * Math.cos(Math.atan(yStrecke[spriteIndex]/xStrecke[spriteIndex]));
                yVerschb[spriteIndex] = -speed * Math.sin(Math.atan(yStrecke[spriteIndex]/xStrecke[spriteIndex]));
            } else {
                xVerschb[spriteIndex] = -speed;
            }
        }
    }
    if(x1 > x2 && y1 > y2){
        if(yStrecke[spriteIndex]<xStrecke[spriteIndex]){
```

```
            if(xStrecke[spriteIndex]!=0){
                xVerschb[spriteIndex] = -speed * Math.cos(Math.atan(yStrecke[spriteIndex]/xStrecke[spriteIndex]));
                yVerschb[spriteIndex] = -speed * Math.sin(Math.atan(yStrecke[spriteIndex]/xStrecke[spriteIndex]));
            } else {
                yVerschb[spriteIndex] = -speed;
            }
        } else {
            if(yStrecke[spriteIndex]!=0){
                xVerschb[spriteIndex] = -speed * Math.cos(Math.atan(yStrecke[spriteIndex]/xStrecke[spriteIndex]));
                yVerschb[spriteIndex] = -speed * Math.sin(Math.atan(yStrecke[spriteIndex]/xStrecke[spriteIndex]));
            } else {
                xVerschb[spriteIndex] = -speed;
            }
        }
    }
    if(x1 <= x2 && y1 > y2){
        if(-yStrecke[spriteIndex]>xStrecke[spriteIndex]){
            if(xStrecke[spriteIndex]!=0){
                xVerschb[spriteIndex] = speed * Math.cos(Math.atan(yStrecke[spriteIndex]/xStrecke[spriteIndex]));
                yVerschb[spriteIndex] = speed * Math.sin(Math.atan(yStrecke[spriteIndex]/xStrecke[spriteIndex]));
            } else {
                yVerschb[spriteIndex] = -speed;
            }
        } else {
            if(yStrecke[spriteIndex]!=0){
                xVerschb[spriteIndex] = speed * Math.cos(Math.atan(yStrecke[spriteIndex]/xStrecke[spriteIndex]));
                yVerschb[spriteIndex] = speed * Math.sin(Math.atan(yStrecke[spriteIndex]/xStrecke[spriteIndex]));
            } else {
                xVerschb[spriteIndex] = speed;
            }
        }
    }

}




function movePicture(speed, url, x1, y1, x2, y2, id){
// -----------------------------------------------------
//== Melde neues Sprite durch neues Array-Element an: ==
    spriteIndex++;
    spriteID[spriteIndex]=id;
    startX[spriteIndex]=x1;    startY[spriteIndex]=y1;
    xBew[spriteIndex]=0;       yBew[spriteIndex]=0;
//== Setze Sprite als Element ins HTML-Dokument: ==
    var myTag = document.createElement("div");
    var myPic = document.createElement("img");
    myTag.setAttribute("id", id);
    myPic.setAttribute("src", url);
    myTag.appendChild(myPic);
    var Ausgabebereich = document.getElementsByTagName("Body")[0];
    Ausgabebereich.appendChild(myTag);
    document.getElementById(id).style.position="absolute";
//== Berechne Bewegung: ==
    xVerschb[spriteIndex] = 0;
    yVerschb[spriteIndex] = 0;
    xStrecke[spriteIndex] = x2 - x1;
    yStrecke[spriteIndex] = y2 - y1;
    if(x1 <= x2 && y1 <= y2){
        if(yStrecke[spriteIndex] > xStrecke[spriteIndex]){
            if(xStrecke[spriteIndex] != 0){
                xVerschb[spriteIndex] = speed * Math.cos(Math.atan(yStrecke[spriteIndex]/xStrecke[spriteIndex]));
                yVerschb[spriteIndex] = speed * Math.sin(Math.atan(yStrecke[spriteIndex]/xStrecke[spriteIndex]));
            } else {
                yVerschb[spriteIndex] = speed;
            }
        } else {
            if(yStrecke[spriteIndex]!=0){
                xVerschb[spriteIndex] = speed * Math.cos(Math.atan(yStrecke[spriteIndex]/xStrecke[spriteIndex]));
                yVerschb[spriteIndex] = speed * Math.sin(Math.atan(yStrecke[spriteIndex]/xStrecke[spriteIndex]));
            } else {
                xVerschb[spriteIndex] = speed;
            }
        }
    }
    if(x1 > x2 && y1 <= y2){
        if(yStrecke[spriteIndex]>-xStrecke[spriteIndex]){
            if(xStrecke[spriteIndex]!=0){
                xVerschb[spriteIndex] = -speed * Math.cos(Math.atan(yStrecke[spriteIndex]/xStrecke[spriteIndex]));
                yVerschb[spriteIndex] = -speed * Math.sin(Math.atan(yStrecke[spriteIndex]/xStrecke[spriteIndex]));
            } else {
                yVerschb[spriteIndex] = speed;
```

```
                }
            } else {
                if(yStrecke[spriteIndex]!=0){
                    xVerschb[spriteIndex] = -speed * Math.cos(Math.atan(yStrecke[spriteIndex]/xStrecke[spriteIndex]));
                    yVerschb[spriteIndex] = -speed * Math.sin(Math.atan(yStrecke[spriteIndex]/xStrecke[spriteIndex]));
                } else {
                    xVerschb[spriteIndex] = -speed;
                }
            }
        }
        if(x1 > x2 && y1 > y2){
            if(yStrecke[spriteIndex]<xStrecke[spriteIndex]){
                if(xStrecke[spriteIndex]!=0){
                    xVerschb[spriteIndex] = -speed * Math.cos(Math.atan(yStrecke[spriteIndex]/xStrecke[spriteIndex]));
                    yVerschb[spriteIndex] = -speed * Math.sin(Math.atan(yStrecke[spriteIndex]/xStrecke[spriteIndex]));
                } else {
                    yVerschb[spriteIndex] = -speed;
                }
            } else {
                if(yStrecke[spriteIndex]!=0){
                    xVerschb[spriteIndex] = -speed * Math.cos(Math.atan(yStrecke[spriteIndex]/xStrecke[spriteIndex]));
                    yVerschb[spriteIndex] = -speed * Math.sin(Math.atan(yStrecke[spriteIndex]/xStrecke[spriteIndex]));
                } else {
                    xVerschb[spriteIndex] = -speed;
                }
            }
        }
        if(x1 <= x2 && y1 > y2){
            if(-yStrecke[spriteIndex]>xStrecke[spriteIndex]){
                if(xStrecke[spriteIndex]!=0){
                    xVerschb[spriteIndex] = speed * Math.cos(Math.atan(yStrecke[spriteIndex]/xStrecke[spriteIndex]));
                    yVerschb[spriteIndex] = speed * Math.sin(Math.atan(yStrecke[spriteIndex]/xStrecke[spriteIndex]));
                } else {
                    yVerschb[spriteIndex] = -speed;
                }
            } else {
                if(yStrecke[spriteIndex]!=0){
                    xVerschb[spriteIndex] = speed * Math.cos(Math.atan(yStrecke[spriteIndex]/xStrecke[spriteIndex]));
                    yVerschb[spriteIndex] = speed * Math.sin(Math.atan(yStrecke[spriteIndex]/xStrecke[spriteIndex]));
                } else {
                    xVerschb[spriteIndex] = speed;
                }
            }
        }

    }
/*
\_____________/\_____________/\_____________/\_____________/\_____________/*/

//_____________\/_____________\/_____________\/_____________\/_____________\
//    == W A I T I N G  S C R E E N  F E A T U R E ==

/*    Waiting = {};

    Waiting.display = function(){
        if(!wCover){
            wCover = makeBox("white");
            _(wCover).add("img", "", "src", "_images/wait.gif").style.cssText = "position:absolute; left:50%; top:50%; margin-left:-31px; margin-top:-70px;";
        } else {
            wCover.style.display = "block";
        }
    }; wCover = null;

    Waiting.hide = function(){
        wCover.style.display = "none";
    }; */

/*    W A I T I N G  S C R E E N  F E A T U R E (end)
\_____________/\_____________/\_____________/\_____________/\_____________/*/


//_____________\/_____________\/_____________\/_____________\/_____________\
//    == P R O G R E S S - B A R - F E A T U R E ==

    function ProgressBar(){
        ProgressBarSelf = this;
        this.set = function(x,y,barwidth){
            // Füge Roh-Leerkasten und -Füllflüssigkeit dem HTML-Dokument hinzu:
            var myBar = document.createElement("div");
            myBar.setAttribute("id", "progressbar");
            var barFilling = document.createElement("div");
            barFilling.setAttribute("id", "fillingFluid");
            myBar.appendChild(barFilling);
            var Ausgabebereich = document.getElementsByTagName("Body")[0];
```

```
        Ausgabebereich.appendChild(myBar);
        // Setze Eigenschaften des leeren Kastens "progressbar":
        document.getElementById("progressbar").style.position="absolute";
        document.getElementById("progressbar").style.left=x+"px";
        document.getElementById("progressbar").style.top=y+"px";
        document.getElementById("progressbar").style.width=barwidth;
        document.getElementById("progressbar").style.height="20px";
        document.getElementById("progressbar").style.border="1px solid blue";
        document.getElementById("progressbar").style.background="rgb(200,200,255)";
        // Setze Füllflüssigkeit-Anfangseigenschaften (noch ohne Volumen)
        document.getElementById("fillingFluid").style.height="19px";
        document.getElementById("fillingFluid").style.width="0px";
        document.getElementById("fillingFluid").style.background="rgb(255,0,255)";
}

var targetFillExtent = 0; var fillExtent = 0; var fillingMovement; var stopuhr;

this.filler = function(seconds){
        var schrittgroesseInPixeln = 5;                                         // Bewegungsflüssigkeit des Fortschrittsbalkens
        if(stopuhr.stop()>((seconds/targetFillExtent)*1000)*schrittgroesseInPixeln-1){ // Erweitere Balken alle xyz Sek.
                stopuhr.start();
                fillExtent = fillExtent + schrittgroesseInPixeln;                       // Ändere Füllkastenbreite im Speicher
        }
        document.getElementById("fillingFluid").style.width = fillExtent;       // Ändere Füllkastenbreite visuell gemäß aktuellem Stand
        if(fillExtent>=targetFillExtent)                                        // Wenn Balken voll, dann Ende
                window.clearInterval(fillingMovement);
}

        this.fill = function(seconds){
                stopuhr = new Clock();
                stopuhr.start();
                targetFillExtent = document.getElementById("progressbar").style.width;          // Maximalausfüllung soll der Breite des leeren Kastens entsprechen
                targetFillExtent = targetFillExtent.substring(0,targetFillExtent.length-2);     // Zum Rechnen muss das 'px' am Ende entfernt werden
                fillingMovement = window.setInterval("ProgressBarSelf.filler("+seconds+")", 1);                 // Start der Füllbewegung
        }

        this.stop = function (){
                window.clearInterval(fillingMovement);
        }
    }

/*    P R O G R E S S - B A R - F E A T U R E (end)
\_____________/\_____________/\_____________/\_____________/\_____________/*/









//_____________\/_____________\/_____________\/_____________\/_____________\
//    == S L I D E S H O W - F E A T U R E ==

    slideshowPicList = [];
    slideshowRahmenBreite = 0;          // die breite des rahmens ist gleich die breite des bildes
    slideshowRahmenHoehe = 0;           // die höhe des rahmens ist gleich die höhe des bildes

function slideshow_addPic(picFile){
    slideshowPicList.push(picFile);
}


function slideshow_visualize(breite, hoehe){

    slideshowRahmenBreite = breite;
    slideshowRahmenHoehe = hoehe;

    rahmenRand          = 2;
    rahmenStil      = "solid";
    rahmenFarbe         = "000000";
    rahmenHintergrundFarbe  = "999999";

    schriftArt      = "verdana";
    schriftFarbe        = "000000";
    schriftGroesse      = 1;

    trennZeichen        = "-";          // trennzeichen zwischen den links

    srollVarianten      = 1;        // 0 = von unten nach oben, 1 = von rechts nach links

    /*******************************************************/

    showFrame = document.createElement("iFrame");
```

```
        showFrame.setAttribute("width",breite+80);
        showFrame.setAttribute("height",hoehe+80);
        showFrame.setAttribute("scrolling","no");
        showFrame.setAttribute("frameBorder","0");
        showFrame.setAttribute("name","showFrameName");
        showFrame.belongsToSystem = 1;
        Ausgabebereich = document.getElementsByTagName("Body")[0];
        Ausgabebereich.appendChild(showFrame);

        showFrameNo = frames.length - 1;


        frames[showFrameNo].document.write("<div id='rahmen' style='width:"
        +(slideshowRahmenBreite+40)+"; height:"
        +(slideshowRahmenHoehe+40)+"; border:"
        +rahmenRand+"px "
        +rahmenStil+" #"
        +rahmenFarbe+"; background-color:#"
        +rahmenHintergrundFarbe+";overflow:hidden'>");


        (srollVarianten==0) ?
        frames[showFrameNo].document.write("<div id='jumper' style='position:relative;top:18;left:0'>") :
        frames[showFrameNo].document.write("<div id='jumper' style='position:relative;left:18;top:18'>");

        (srollVarianten==0) ?
        frames[showFrameNo].document.write("<table width="
        +slideshowRahmenBreite+" cellspacing=0 cellpadding=0>") :
        frames[showFrameNo].document.write("<table height="
        +slideshowRahmenHoehe+" cellspacing=0 cellpadding=0>");

        for (picIndex=0;picIndex<slideshowPicList.length;picIndex++)
        {
                (srollVarianten==0) ?
                frames[showFrameNo].document.write("<tr><td width="
                +slideshowRahmenBreite+" height="
                +rahmenHoehe+">") :
                frames[showFrameNo].document.write("<td width="
                +slideshowRahmenBreite+" height="
                +slideshowRahmenHoehe+">");

                (srollVarianten==0) ?
                frames[showFrameNo].document.write("<img src="+slideshowPicList[picIndex]+"></td></tr>"
                +"<tr><td height=25></td></tr>") :
                frames[showFrameNo].document.write("<img src="+slideshowPicList[picIndex]+"></td>"
                +"<td><div style='width:25'></div></td>");
        }

        frames[showFrameNo].document.write("</table></div></div><br>");


}


function slideshow_switch()
{
        posOfSlideElement+=(slideshowRahmenBreite+25);

        thePos      = posOfSlideElement-(posOfSlideElement*2)+18;

        browser     = navigator.appName;
        usAgent = navigator.userAgent;

        if (browser == "Microsoft Internet Explorer" || usAgent.indexOf("Opera") >- 1)
        clearTimeout(time);

        dmx();

} var posOfSlideElement = 0; var thePos=dimx=Dimx=0,time;


function slideshow_jump(pos)
{
        thePos      = pos-(pos*2)+18;

        browser     = navigator.appName;
        usAgent = navigator.userAgent;

        if (browser == "Microsoft Internet Explorer" || usAgent.indexOf("Opera") >- 1)
        clearTimeout(time);

        dmx();
}
```

```
function dmx()
{
	time  = setTimeout("dmx()",0);

	dimx	= Math.round(Dimx+=((thePos)-Dimx)*4/20);

	if (frames[showFrameNo].document.getElementById)
	{
		(srollVarianten==0) ?
		frames[showFrameNo].document.getElementById("jumper").style.top  = dimx :
		frames[showFrameNo].document.getElementById("jumper").style.left = dimx;
	}
	else if (frames[showFrameNo].document.all)
	{
		(srollVarianten==0) ?
		frames[showFrameNo].document.all["jumper"].style.top  = dimx :
		frames[showFrameNo].document.all["jumper"].style.left = dimx;
	}

	if (thePos == dimx) clearTimeout(time);
}

/*
\_____________/\_____________/\_____________/\_____________/\_____________/*/



//_____________\/_____________\/_____________\/_____________\/_____________\
//    == D R A W I N G - F E A T U R E ==

	function DrawingKit(){
		if(browserType=="Microsoft Internet Explorer"){

			//¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\
			// MS Public Functions:

			this.setColor = function(color){
				currentColor = color;
			}

			this.setFillColor = function(color){
				currentFillColor = color;
				if(color=="none")
					filled = "False";
				else
					filled = "True";
			}

			this.setStrokeWidth = function(width){
				currentStrokeWidth = width;
			}

			this.drawLine = function(x1,y1,x2,y2){
				id("figureContainer").innerHTML += '<v:line style="behavior: url(#default#VML);position:absolute;" from="'+x1+','+y1+'" to="'+x2+','+y2+'" strokeweight="'+
currentStrokeWidth+'" strokecolor="'+currentColor+'" ></v:line>';
			}

			this.drawCurve = function(x1,y1,x2,y2,x3,y3,x4,y4){
				id("figureContainer").innerHTML += '<v:curve style="behavior: url(#default#VML);position:absolute;" from="'+x1+','+y1+'" control1="'+x2+','+y2+'" control2="'+x3+','
+y3+'" to="'+x4+','+y4+'" strokeweight="'+currentStrokeWidth+'" strokecolor="'+currentColor+'"  filled="'+filled+'" fillcolor="'+currentFillColor+'" ></v:curve>';
			}

			this.drawEllipse = function(x,y,rx,ry){
				// Ankerpunkt in die Mitte statt links oben:
				x = x - rx;
				y = y - ry;
				// Zeichnen
				id("figureContainer").innerHTML += '<v:oval style="behavior: url(#default#VML);position:absolute;top:'+y+';left:'+x+';width:'+rx*2+'px;height:'+ry*2+'px"
strokeweight="'+currentStrokeWidth+'" strokecolor="'+currentColor+'" filled="'+filled+'" fillcolor="'+currentFillColor+'"> </v:oval>';
			}

			this.drawCircle = function(x,y,r){
				// Ankerpunkt in die Mitte statt links oben:
				x = x - r;
				y = y - r;
				// Zeichnen:
				id("figureContainer").innerHTML += '<v:oval style="behavior: url(#default#VML);position:absolute;top:'+y+';left:'+x+';width:'+r*2+'px;height:'+r*2+'px"
strokeweight="'+currentStrokeWidth+'" strokecolor="'+currentColor+'" filled="'+filled+'" fillcolor="'+currentFillColor+'"> </v:oval>';
			}

			this.drawRect = function(x,y,w,h,rx,ry){
				id("figureContainer").innerHTML += '<v:rect style="behavior: url(#default#VML);display:inline-block; position:absolute; top:'+y+';left:'+x+';width:'+w+'px;height:'+h+
```

```
'px" strokeweight="'+currentStrokeWidth+'" strokecolor="'+currentColor+'" filled="'+filled+'" fillcolor="'+currentFillColor+'"> </v:rect>';
		}

		this.drawPolyline = function(points){
			var pointsString = "";
			//Falls Koordinaten als String übergeben, dann behalte Parameter wie er ist
			if((typeof points)=="string"){
				pointsString = points;
			}else{
			// Sonst ist es wohl ein Array, das in einen String gepackt wird (Schema: "123,21 32,34 23,11 23,21 ...")
				for(var i=0;i<points.length;i=i+2){
					pointsString += points[i] + ",";
					pointsString += points[i+1] + " ";
				}
			}
			id("figureContainer").innerHTML += '<v:polyline style="behavior: url(#default#VML);display:inline-block;" points="'+pointsString+'" strokeweight="'+currentStrokeWidth+
'" strokecolor="'+currentColor+'" filled="'+filled+'" fillcolor="'+currentFillColor+'"> </v:polyline>';
		}

		this.clear = function(){
			id("figureContainer").innerHTML = "";
		}

		this.hideCanvas = function(){
			id('figureContainer').style.visibility = "hidden";
		}

		this.showCanvas = function(){
			id('figureContainer').style.visibility = "visible";
		}

		this.setViewBox = function(x1,y1,w,h){
			id("viewBox").style.left = x1;
			id("viewBox").style.top = y1;
			id("viewBox").style.width = w;
			id("viewBox").style.height = h;
			id("figureContainer").style.left = -x1;
			id("figureContainer").style.top = -y1;
			id("figureContainer").style.width = w+x1;
			id("figureContainer").style.height = h+y1;
		}

		/*
		//\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/*/

	} else {
		//¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\
		// FF & Co. Public Functions:

		this.setColor = function(color){
			currentColor = color;
		}

		this.setFillColor = function(color){
			currentFillColor = color;
		}

		this.setStrokeWidth = function(width){
			currentStrokeWidth = width;
		}

		this.drawLine = function(x1,y1,x2,y2){
			var line = svgDocument.createElementNS(svgns, "line");
			line.setAttributeNS(null, "x1", x1);
			line.setAttributeNS(null, "y1", y1);
			line.setAttributeNS(null, "x2", x2);
			line.setAttributeNS(null, "y2", y2);
			line.setAttributeNS(null, "stroke", currentColor);
			line.setAttributeNS(null, "stroke-width", currentStrokeWidth);
			svgDocument.documentElement.appendChild(line);
		}

		this.drawCurve = function(x1,y1,x2,y2,x3,y3,x4,y4){
			var curve = svgDocument.createElementNS(svgns, "path");
			curve.setAttributeNS(null, "d", "M "+x1+","+y1+" C "+x2+","+y2+" "+x3+","+y3+" "+x4+","+y4);
			curve.setAttributeNS(null, "stroke", currentColor);
			curve.setAttributeNS(null, "stroke-width", currentStrokeWidth);
			curve.setAttributeNS(null, "fill", currentFillColor);
			svgDocument.documentElement.appendChild(curve);
		}

		this.drawEllipse = function(x,y,rx,ry){
			var ellipse = svgDocument.createElementNS(svgns, "ellipse");
			ellipse.setAttributeNS(null, "cx", x);
```

```
        ellipse.setAttributeNS(null, "cy", y);
        ellipse.setAttributeNS(null, "rx", rx);
        ellipse.setAttributeNS(null, "ry", ry);
        ellipse.setAttributeNS(null, "stroke", currentColor);
        ellipse.setAttributeNS(null, "stroke-width", currentStrokeWidth);
        ellipse.setAttributeNS(null, "fill", currentFillColor);
        svgDocument.documentElement.appendChild(ellipse);
    }

    this.drawCircle = function(x,y,r){
        var circle = svgDocument.createElementNS(svgns, "circle");
        circle.setAttributeNS(null, "cx", x);
        circle.setAttributeNS(null, "cy", y);
        circle.setAttributeNS(null, "r",  r);
        circle.setAttributeNS(null, "stroke", currentColor);
        circle.setAttributeNS(null, "stroke-width", currentStrokeWidth);
        circle.setAttributeNS(null, "fill", currentFillColor);
        svgDocument.documentElement.appendChild(circle);
    }

    this.drawRect = function(x,y,w,h,rx,ry){
        var rect = svgDocument.createElementNS(svgns, "rect");
        rect.setAttributeNS(null, "x", x);
        rect.setAttributeNS(null, "y", y);
        rect.setAttributeNS(null, "width",  w);
        rect.setAttributeNS(null, "height", h);
        if((typeof rx)=="number") { // Falls der Nutzer gerundete Ecken wünscht
            rect.setAttributeNS(null, "rx",  rx);
            rect.setAttributeNS(null, "ry", ry);
        }

        rect.setAttributeNS(null, "stroke", currentColor);
        rect.setAttributeNS(null, "stroke-width", currentStrokeWidth);
        rect.setAttributeNS(null, "fill", currentFillColor);
        svgDocument.documentElement.appendChild(rect);
    }


    this.drawPolyline = function(points){
        var pointsString = "";
        //Falls Koordinaten als String übergeben, dann behalte Parameter wie er ist
        if((typeof points)=="string"){
            pointsString = points;
        }else{
        // Sonst ist es wohl ein Array, das in einen String gepackt wird (Schema: "123,21 32,34 23,11 23,21 ...")
            for(var i=0;i<points.length;i=i+2){
                pointsString += points[i] + ",";
                pointsString += points[i+1] + " ";
            }
        }
        var polyline = svgDocument.createElementNS(svgns, "polyline");
        polyline.setAttributeNS(null, "points", pointsString);
        polyline.setAttributeNS(null, "stroke", currentColor);
        polyline.setAttributeNS(null, "stroke-width", currentStrokeWidth);
        polyline.setAttributeNS(null, "fill", currentFillColor);
        svgDocument.documentElement.appendChild(polyline);
    }



        this.clear = function(){
            removeAllChildNodes(svgDocument.documentElement);
        }

        this.hideCanvas = function(){
            id('svgFrame').style.visibility = "hidden";
        }

        this.showCanvas = function(){
            id('svgFrame').style.visibility = "visible";
        }

        this.setViewBox = function(x1,y1,w,h){
            id("viewBox").style.left = x1;
            id("viewBox").style.top = y1;
            id("viewBox").style.width = w;
            id("viewBox").style.height = h;
            id("svgFrame").style.left = -x1;
            id("svgFrame").style.top = -y1;
            id("svgFrame").style.width = w+x1;
            id("svgFrame").style.height = h+y1;
        }
}
/*
//\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/*/
```

```
        //¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\
        // Private Function(s):

        function init(){
            if(browserType=="Microsoft Internet Explorer"){
                // Füge VML-Container hinzu (DIV) und füge dem Dokument VML-Fähigkeiten hinzu:
                document.getElementsByTagName("body")[0].innerHTML += '<div id="viewBox"
style="position:absolute;left:0px;top:0px;width:100%;height:100%;overflow:hidden;"><div id="figureContainer"
style="position:absolute;left:0px;top:0px;width:100%;height:100%"></div></div>';
                var htmlDoc = document.getElementsByTagName("html")[0];
                htmlDoc.setAttribute("xmlns:v","urn:schemas-microsoft-com:vml");
                // Führe onready-Code des Nutzers aus:
                var onreadyExecuter = setInterval(function(){
                    if((typeof thisDrawingKit.onready)=="function" || (typeof thisDrawingKit.onready)=="string"){
                        clearInterval(onreadyExecuter);
                        setTimeout(thisDrawingKit.onready,1);
                        thisDrawingKit.onready = 0;
                    }
                },1);
            } else {
                // Füge SVG-Bett (Iframe mit Datei mit SVG-Gerüst) hinzu:
                document.getElementsByTagName("body")[0].innerHTML += '<div id="viewBox"
style="position:absolute;left:0px;top:0px;width:100%;height:100%;overflow:hidden;"><iframe id="svgFrame" width="100%" height="100%" src="svgFrame.svg" belongsToSystem="1"
style="position:absolute;top:0px;left:0px;overflow:hidden" frameborder="0"></iframe></div>';
                var svgFrame = frames[frames.length-1];
                // Warte bis SVG-Gerüst im IFrame gefunden,
                // dann mache SVG-Gerüst den anderen Funktionen bekannt ("svgDocument")
                // und führe den onready-Code des Nutzers, sobald vorhanden, aus:
                var readinessChecker = setInterval(function(){
                    if(svgFrame.document.documentElement){
                        svgDocument = svgFrame.document;
                        if((typeof thisDrawingKit.onready)=="function" || (typeof thisDrawingKit.onready)=="string"){
                            clearInterval(readinessChecker);
                            setTimeout(thisDrawingKit.onready,1);
                            thisDrawingKit.onready = 0;
                        }
                    }
                },1);
            }
        }
        /*
        //\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/*/


        //¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\
        // Properties and initializing Code:
        var drawingKitReady = 0;
        var currentColor = "black";
        var currentFillColor = "none";
        var filled = "False";
        var currentStrokeWidth = 1;
        var thisDrawingKit = this;
        var svgDocument; // (für Firefox u.a.)
        var svgns = "http://www.w3.org/2000/svg"; // // (für Firefox u.a.)
        this.onready;
        init();
        /*
        //\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/*/
    }


/*
\_____________/\_____________/\_____________/\_____________/\_____________/*/









//_____________\/_____________\/_____________\/_____________\/_____________\
//    == S O U N D - F E A T U R E ==

    function playSound(soundFilename) {

        var Ausgabebereich = document.getElementsByTagName("body")[0];

        var absoluteContainer = document.createElement("div");
        absoluteContainer.style.position = "absolute";
        absoluteContainer.style.left = "0";
        absoluteContainer.style.top = "0";
        Ausgabebereich.appendChild(absoluteContainer);
```

```
        var embedElement = document.createElement("embed");
        embedElement.setAttribute("type","application/x-mplayer2");
        embedElement.setAttribute("src",soundFilename);
        embedElement.setAttribute("autostart","1");
        embedElement.setAttribute("","false");
        embedElement.setAttribute("width","0");
        embedElement.setAttribute("height","0");
        embedElement.setAttribute("ShowStatusBar","0");
        embedElement.setAttribute("ShowControls","0");
        if(navigator.appName=="Microsoft Internet Explorer")
                embedElement.setAttribute("hidden","true");
        absoluteContainer.appendChild(embedElement);
}


function playPreloadedSound(volume){
        if(navigator.appName=="Microsoft Internet Explorer")
                document.getElementById('sound').stop();                // IE spielt den Sound im Vorladestatus, jetzt kann Abspielen abgebrochen werden
        var soundFrameAusgabebereich = soundFrame.document.getElementsByTagName("Body")[0];
        var embedElement = soundFrame.document.createElement("embed");
        embedElement.setAttribute("type","application/x-mplayer2");
        embedElement.setAttribute("src", preloadedSoundFilename);
        embedElement.setAttribute("autostart","1");
        embedElement.setAttribute("loop","false");
        embedElement.setAttribute("width","0");
        embedElement.setAttribute("height","0");
        embedElement.setAttribute("ShowStatusBar","0");
        embedElement.setAttribute("ShowControls","0");
        embedElement.setAttribute("volume",-(10-volume)*1000);
        soundFrameAusgabebereich.appendChild(embedElement);
}

function preloadSound(soundFilename){

        var Ausgabebereich = document.getElementsByTagName("body")[0];
        preloadedSoundFilename = soundFilename;

        // Sound vorladen:
        var absoluteContainer = document.createElement("div");
        absoluteContainer.style.position = "absolute";
        absoluteContainer.style.left = "0";
        absoluteContainer.style.top = "0";
        Ausgabebereich.appendChild(absoluteContainer);

        preloadElement = document.createElement("embed");
        preloadElement.setAttribute("type","application/x-mplayer2");
        preloadElement.setAttribute("src",soundFilename);
        preloadElement.setAttribute("id","sound");
        preloadElement.setAttribute("autostart","0");
        preloadElement.setAttribute("loop","false");
        preloadElement.setAttribute("width","0");
        preloadElement.setAttribute("height","0");
        preloadElement.setAttribute("ShowStatusBar","0");
        preloadElement.setAttribute("ShowControls","0");
        if(navigator.appName=="Microsoft Internet Explorer"){       // IE muss den Sound wirklich abspielen, um ihn vorzuladen
                preloadElement.setAttribute("hidden","true");
                preloadElement.setAttribute("autostart","true");
                preloadElement.setAttribute("volume","-10000");
        }
        absoluteContainer.appendChild(preloadElement);

        // Versteckten iFrame erzeugen, um darin mit playPreloadedSound( ) den vorgeladenen Ton zu plazieren:
        soundFrame = document.createElement("iframe");
        soundFrame.setAttribute("src","about:blank");
        soundFrame.setAttribute("id","soundFrame");
        soundFrame.style.visibility = "hidden";
        soundFrame.style.position = "absolute";
        soundFrame.belongsToSystem = 1;
        Ausgabebereich.appendChild(soundFrame);
        soundFrame = frames[frames.length-1];

    } var preloadedSoundFilename;


/*
\_____________/\_____________/\_____________/\_____________/\_____________/*/


        var Ajax = {};

        var ajaxKey = randomString(10);
        var ajaxUserKey = randomString(10);
```

```
function prepareRequestObjects(){
    var key = ajaxKey;
    var userKey = ajaxUserKey;


    if(XMLHttpRequest){
        var nativeXMLHttpRequest = XMLHttpRequest;
        XMLHttpRequest = function(triedKey){
            if(triedKey==key || triedKey==userKey){
                return new nativeXMLHttpRequest();
            }else{
                // alert("prepareRequestObjects(): Wrong or missing key.");
                var open_old = nativeXMLHttpRequest.prototype.open;
                nativeXMLHttpRequest.prototype.open = function(){
                   if(arguments[1].indexOf("ajaxFileSystem.php")==-1)
                      return open_old.apply(this,arguments);
                };
                return new nativeXMLHttpRequest();
            }
        };
    }

    if(isIE){
        if(ActiveXObject){
            var nativeActiveXObject = ActiveXObject;
            ActiveXObject = function(progID, triedKey){
                if(progID=="Msxml2.Xmlhttp" || progID=="Microsoft.XMLHTTP"){
                    if(triedKey==key || triedKey==userKey)
                        return new nativeActiveXObject(progID);
                    else
                        alert("prepareRequestObjects(): Wrong or missing key.");
                } else {
                    return new nativeActiveXObject(progID);
                }
            };
        }
    }

    prepareRequestObjects = null;
} prepareRequestObjects();



function getRequestObjectFnc(){

    var key = ajaxKey;
    var userKey = ajaxUserKey;

    Ajax.getObject = function(triedKey) {

        if(triedKey==key || triedKey==userKey){
            var ro;
            // Mozilla-kompatibel?
            if (window.XMLHttpRequest){
            // Ja, ein Mozilla-kompatibler Browser
                try {
                    // Objekt ableiten
                    ro = new XMLHttpRequest(key);
                } catch(e) {
                    // Objekt konnte nicht abgeleitet werden
                    ro = null;
                }
            } else {
                // Teil fuer den IE
                try {
                    // Neue Methode versuchen
                    ro = new ActiveXObject("Msxml2.Xmlhttp", key);
                } catch(e) {
                    try {
                        // Wenn die neue nicht klappt,
                        //dann vielleicht die alte?
                        ro = new ActiveXObject("Microsoft.XMLHTTP", key);
                    } catch (e) {
                        // ActiveX ausgeschaltet oder kein JS aktiviert
                        ro = null;
                    }
                }
            }
            // Haben wir ein Objekt oder nicht?
            if (ro == null) {
                // Fehlermeldung
                alert ('Ihr Browser unterstützt leider kein AJAX');
            }
```

```
                return ro;
            } else {
                alert("Ajax.getObject(): Wrong or missing key");
            }
        }

        getRequestObjectFnc = null;

    } getRequestObjectFnc();


    function secureRequestObjectINTERN(){
        var arr = secureRequestObjectINTERN.arguments;
        arr[arr.length] = FileSystemClass;
        arr[arr.length+1] = ServerInfo;
        hideGlobal("ajaxKey").allow(arr);

    }


    function setupAjaxGetKeyFnc(){
        var userKey = ajaxUserKey;
        ajaxUserKey = null;

        Ajax.getKey = function(){
            if(Ajax.getKey.caller!=null){
                Ajax.getKey = null;
                return userKey;
            } else {
                alert("Ajax.getKey() cannot be used outside a function.");
            }
        };

    } setupAjaxGetKeyFnc();


/*      function secureRequestObjectFnc(){
            var userKey = ajaxUserKey;
            ajaxUserKey = null;

            Ajax.allow = function(){
                var arr = [];

                for(var i = 0; i < Ajax.allow.arguments.length; i++){
                    arr.push(Ajax.allow.arguments[i]);
                }

                ajaxUserKey = userKey;
                userKey = null;
                hideGlobal("ajaxUserKey").allow(arr);
            }

            secureRequestObjectFnc = null;

        } secureRequestObjectFnc(); */


        setTimeout(function(){
            loop(function(){
                for(var i in window){ if(keyOk(i)){
                    try{
                        if(window[i] && window[i].getAllResponseHeaders)
                            alert("Warning: AjaxObject declared global. (Security risk)");
                    } catch (e) {}
                }}
            },1000);
        },1000);


        Event.add("onbodyload", function(){
            fs.getKey = function(){
                alert("FileSystem.getKey() has to be used immediately after document loading or earlier.");
            };
            Ajax.getKey = function(){
                alert("Ajax.getKey() has to be used immediately after document loading or earlier.");
            };
        });



//_____________\/_____________\/_____________\/_____________\/_____________\
//    ==FILESYSTEM-FEATURE==
```

```
var fileSystemKey = randomString(10);

function FileSystemClass(key){
	//¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\
	// Public Functions:
	this.getKey = function(){
		if(this.getKey.caller==null){
			alert("FileSystem.getKey() cannot be used outside a function.");
		} else {
			var retTempKey = tempKey;
			tempKey = null;
			auth_tempKey = null;
			return retTempKey;
		}
	};
	//-------------------------------------------------------------------------------------------------------------------------------------------------
	this.auth_getKey = function(){			// wie .getKey(), jedoch extra für das Auth-Feature, da der User sonst keinen Schlüssel bekommen würde
		if(this.auth_getKey.caller==null){
			alert("FileSystem.getKey() cannot be used outside a function.");
		} else {
			var retTempKey = auth_tempKey;
			auth_tempKey = null;
			return retTempKey;
		}
	};
	//-------------------------------------------------------------------------------------------------------------------------------------------------
	this.makeDir = function(dirname, triedKey){
		var retObj = {};
		sendRequest(	"actualizer="+String(Math.round(Math.random()*10000))+"&"+
				"action=mkDir&"+
				"filename="+dirname,
				triedKey, retObj);
		return retObj;
	};
	//-------------------------------------------------------------------------------------------------------------------------------------------------
	this.deleteDir = function(dirname, triedKey){
		var retObj = {};
		sendRequest(	"actualizer="+String(Math.round(Math.random()*10000))+"&"+
				"action=delDir&"+
				"filename="+dirname,
				triedKey, retObj);
		return retObj;
	};
	//-------------------------------------------------------------------------------------------------------------------------------------------------
	this.emptyDir = function(dirname, triedKey){
		var retObj = {};
		sendRequest(	"actualizer="+String(Math.round(Math.random()*10000))+"&"+
				"action=emptyDir&"+
				"filename="+dirname,
				triedKey, retObj);
		return retObj;
	};
	//-------------------------------------------------------------------------------------------------------------------------------------------------
	this.makeFile = function(filename, data, triedKey){
		var retObj = {};
		sendRequest(	"actualizer="+String(Math.round(Math.random()*10000))+"&"+
				"action=makeFile&"+
				"filename="+filename+ "&" +
				"content="+encodeURIComponent(data),
				triedKey, retObj);
		return retObj;
	};
	//-------------------------------------------------------------------------------------------------------------------------------------------------
	this.appendTo = function(filename, data, triedKey){
		var retObj = {};
		sendRequest(	"actualizer="+String(Math.round(Math.random()*10000))+"&"+
				"action=appendTo&"+
				"filename="+filename+ "&" +
				"content="+encodeURIComponent(data),
				triedKey, retObj);
		return retObj;
	};
	//-------------------------------------------------------------------------------------------------------------------------------------------------
	this.write = function(filename, data, triedKey){
		var retObj = {};
		sendRequest(	"actualizer="+String(Math.round(Math.random()*10000))+"&"+
				"action=write&"+
				"filename="+filename+ "&" +
				"content="+encodeURIComponent(data),
				triedKey, retObj);
		return retObj;
	};
	//-------------------------------------------------------------------------------------------------------------------------------------------------
```

```
this.rename = function(filename, newFilename, triedKey){
	var retObj = {};
	sendRequest(	"actualizer="+String(Math.round(Math.random()*10000))+"&"+
			"action=renameFile&"+
			"filename="+filename+ "&" +
			"newFilename="+newFilename,
			triedKey, retObj);
	return retObj;
};
//-----------------------------------------------------------------------------------------------------------------------------
this.deleteFile = function(filename, triedKey){
	var retObj = {};
	sendRequest(	"actualizer="+String(Math.round(Math.random()*10000))+"&"+
			"action=deleteFile&"+
			"filename="+filename,
			triedKey, retObj);
	return retObj;
};
//-----------------------------------------------------------------------------------------------------------------------------
this.execute = function(phpCode, triedKey){
	var retObj = {};
	sendRequest(	"actualizer="+String(Math.round(Math.random()*10000))+"&"+
			"action=execute&"+
			"content="+phpCode,
			triedKey, retObj);
	return retObj;
};
//-----------------------------------------------------------------------------------------------------------------------------
this.knock = function(filename, urlParameters, triedKey){
	if(triedKey==key){
		var requestObject = Ajax.getObject(ajaxKey);
		try {
			requestObject.open('POST',filename);
			requestObject.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
			requestObject.send(urlParameters);
		} catch(e) { }
	} else {
		alert("FileSystem.request(): Wrong or missing key.");
		alert("FileSystem.request(): Wrong or missing key in: \r\n"+this.request.caller);
	}
};
//-----------------------------------------------------------------------------------------------------------------------------
this.request = function(filename, urlParameters, triedKey){
	if(triedKey==key){
		var retObj = {};
		if(undef(urlParameters))
			var urlParameters = null;
		var requestObject = Ajax.getObject(ajaxKey);
		requestObject.onreadystatechange = function(){statehandler(requestObject, retObj)};
		try {
			requestObject.open('POST',filename);
			requestObject.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
			requestObject.send(urlParameters);
		} catch(e) { }
		return retObj;
	} else {
		alert("FileSystem.request(): Wrong or missing key.");
		alert("FileSystem.request(): Wrong or missing key in: \r\n"+this.request.caller);
	}
};
//-----------------------------------------------------------------------------------------------------------------------------
this.requestFileDate = function(filename){
	var retObj = {};
	var requestObject = Ajax.getObject(ajaxKey);
	requestObject.onreadystatechange = function(){
		if(requestObject.readyState == 4){
			response = requestObject.getResponseHeader("Last-Modified");
			if(retObj && (typeof retObj.onready)=="function")
				retObj.onready();
		}
	};
	try {
		requestObject.open('HEAD',filename+"?actualizer="+String(Math.round(Math.random()*10000)));
		requestObject.send(null);
	} catch(e) { }
	return retObj;
};
//-----------------------------------------------------------------------------------------------------------------------------
this.search = function(filename){
	var retObj = {};
	var requestObject = Ajax.getObject(ajaxKey);
	requestObject.onreadystatechange = function(){
		if(requestObject.readyState == 4){
			response = requestObject.status;
```

```
                    if(retObj && (typeof retObj.onready)=="function")
                        retObj.onready();
                }
        };
        try {
                requestObject.open('HEAD',filename+"?actualizer="+String(Math.round(Math.random()*10000)));
                requestObject.send(null);
        } catch(e) { }
        return retObj;
};
//---------------------------------------------------------------------------------------------------------------------------------
this.exists = function(){
        if(response=="200")
                return 1;
        else
                return 0;
};
//---------------------------------------------------------------------------------------------------------------------------------
this.error = function(){
        if(response=="FileSystemError")
                return 1;
        else
                return 0;
};
//---------------------------------------------------------------------------------------------------------------------------------
this.getXMLResponse = function(tagName) {
        if(response!="writingJustDone"){
                if(xmlResponse.getElementsByTagName(tagName)[0].childNodes.length>0)
                        return xmlResponse.getElementsByTagName(tagName)[0].childNodes[0].nodeValue;
                else
                        return "";
        }else{
                return null;
        }
};
//---------------------------------------------------------------------------------------------------------------------------------
this.getResponse = function() {
        return response;
};
//---------------------------------------------------------------------------------------------------------------------------------
this.deleteResponse = function() {
        response=window[window[randomString(10)]]; // Setze auf undefined;
        xmlResponse=response; // Setze auf undefined;
};
/*
\__/\__/\__/\__/\__/\__/\__/\__//\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/*/

//¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\
// Private Functions:
//---------------------------------------------------------------------------------------------------------------------------------
var statehandler = function(requestObject, retObj){
        if(requestObject.readyState == 4){
                response = requestObject.responseText;
                if(response.indexOf("<!--separatorXYZ//-->")>-1){
                        response = response.split("<!--separatorXYZ//-->")[1];
                }
                if(requestObject.status=="404")
                        response="FileSystemError";
                xmlResponse = requestObject.responseXML;
                if(retObj && (typeof retObj.onready)=="function")
                        retObj.onready();
        }
};
//---------------------------------------------------------------------------------------------------------------------------------
var sendRequest = function(urlParameters, triedKey, retObj){
        if(triedKey==key){
                var requestObject = Ajax.getObject(ajaxKey);
                requestObject.onreadystatechange = function(){statehandler(requestObject, retObj)};
                try {
                        requestObject.open('POST','ajaxFileSystem.php?actualizer='+String(Math.round(Math.random()*10000)));
                        requestObject.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
                        requestObject.send(urlParameters);
                } catch(e) {alert(e);}
        } else {
                alert("File System: Wrong or missing key");
                alert("FileSystem.request(): Wrong or missing key in: \r\n"+sendRequest.caller);
        }
};
//---------------------------------------------------------------------------------------------------------------------------------
var checkFileSystemAvailability = function(){
        var requestObject = Ajax.getObject(ajaxKey);
        requestObject.onreadystatechange = function(){
                if(requestObject.readyState == 4){
                        if(requestObject.status!="200"){
```

```
                        alert("File \"ajaxFileSystem.php\" or PHP-Server missing.\r\n" +
                        "They are necessary to make JavaScript able to handle files.");
                    }
                }
            };
            try {
                requestObject.open('HEAD',"ajaxFileSystem.php?actualizer="+String(Math.round(Math.random()*10000)));
                requestObject.send(null);
            } catch(e) {  }
        }
        /*
        \__/\__/\__/\__/\__/\__/\__/\__//\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/*/

        // Properties and initializing Code:
        var tempKey = auth_tempKey = key; fileSystemKey = null;
        var response;
        var xmlResponse;
        var thisFileSystem = this;
        var ajaxKey = get_ajaxKey();
        this.onready;
        checkFileSystemAvailability();
}




secureRequestObjectINTERN();
        var fs = new FileSystemClass(fileSystemKey);


        var FileSystem = fs;
        FileSystemClass = null;




/*
\_____________/\_____________/\_____________/\_____________/\_____________/*/




//_____________\/_____________\/_____________\/_____________\/_____________\
//    ==EVENTS - FEATURE==

    function AjaxEventSystem(){

        //¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\
        // Public Functions:

        this.setListeningInterval = function(value){
            listeningInterval = value;
        }

        this.getEventType = function(){
            return eventType;
        }

        this.getEventContent = function(){
            return eventContent;
        }

        this.getEventSource = function(){
            return eventSource;
        }

        this.fire = function(eventContent, eventType, eventSource){
            debugMessage(stempel()+"Will Event vom Typ \""+eventType+"\" feuern..." );
            var xmlCode =  "<r><t>"
                         + eventType     + "</t><c>"
                         + eventContent + "</c><s>"
                         + eventSource  + "</s></r>";
            firingDS.request(eventsIndexFilename);
            firingDS.onready = function(){
                // Falls neu gefundener Index in begInd.dat wirklich größer geworden (und nicht wegen Cache geringer scheint), dann speichere Event,
                // sonst wiederhole Feuerungsversuch
                if(parseInt(firingDS.getResponse())>=officialEventIndex){
                    officialEventIndex = parseInt(firingDS.getResponse());
                    debugMessage(stempel()+"Ermittelter offizieller Index: "+officialEventIndex);
```

```
                debugMessage(stempel()+"Versuche begInd.dat mit Index "+String(officialEventIndex+1)+" zu überschreiben (um e"+String(officialEventIndex)+" zu feuern)...");
                var firingSteps = function(){
                    firingDS.write(eventsIndexFilename, String(officialEventIndex+1));
                    firingDS.onready = function(){
                        debugMessage(stempel()+"Habe begInd.dat mit Index "+String(officialEventIndex+1)+" überschrieben und feuere e"+String(officialEventIndex)+".xml");
                        firingDS.makeFile("evDir/e"+String(officialEventIndex)+".xml", xmlCode);
                        // Zur Sicherheit checken, ob in begInd.dat beabsichtigter Wert gelandet - wenn nicht, dann nochmal
                        firingDS.onready = function(){
                            firingDS.request(eventsIndexFilename);
                            firingDS.onready = function(){
                                if((parseInt(firingDS.getResponse())<officialEventIndex) || !(parseInt(firingDS.getResponse())>0)){
                                    debugMessage(stempel()+"Etwas stimmt mit begInd.dat nicht. (Variable officialEventIndex derzeit: "+officialEventIndex+")");
                                    firingSteps();
                                }
                            };
                        };
                        ownFiredEvents.push(officialEventIndex);
                    };
                };
                firingSteps();
            } else {
                debugMessage("Gefundener offizieller Index zu klein. Wiederhole Versuch.");
                thisObj.fire(eventContent, eventType, eventSource);
            }
        };
    }

/*
//\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/*/


//¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\
// Private Functions:

function startEventListening(){  // (setTimeout-Schleifenfunktion)
    /* Funktionsweise:
    System wartet auf evtl. Entstehung einer Eventdatei "e[index].xml".
    Falls entdeckt: Inhalte lesen und erwarteten Index erhöhen.
    Falls erwarteter Index selbst schon mal gefeuert: Erwaretetn Index nochmal erhöhen (um nicht die selbstgefeuerten Events auszulesen)
    */
    var listen = function(){
        if((typeof thisObj.onevent)=="function" || (typeof thisObj.onevent)=="string"){
            var listeningSteps = function(){
                if(!ownFiredEvents.contains(internalEventIndex)){
                    datSystem.search(eventsDirName + "/e"+String(internalEventIndex)+".xml");
                    datSystem.onready = function(){
                        if(datSystem.exists()){
                            datSystem.request(eventsDirName + "/e"+String(internalEventIndex)+".xml");
                            datSystem.onready = function(){
                                internalEventIndex++;
                                setTimeout(thisObj.onevent,1);
                                eventType = datSystem.getXMLResponse('t');
                                eventContent = datSystem.getXMLResponse('c');
                                eventSource = datSystem.getXMLResponse('s');
                                listeningSteps();
                            };
                        } else {  // Lücken vermeiden, darum auch nach übernächstem Index suchen:
//**************************************************************************************************************************************************************************************
                            if(!ownFiredEvents.contains(internalEventIndex+1)){
                                datSystem.search(eventsDirName + "/e"+String(internalEventIndex+1)+".xml");
                                datSystem.onready = function(){
                                    if(datSystem.exists()){
                                        datSystem.request(eventsDirName + "/e"+String(internalEventIndex+1)+".xml");
                                        datSystem.onready = function(){
                                            internalEventIndex = internalEventIndex + 2;
                                            setTimeout(thisObj.onevent,1);
                                            eventType = datSystem.getXMLResponse('t');
                                            eventContent = datSystem.getXMLResponse('c');
                                            eventSource = datSystem.getXMLResponse('s');
                                        };
                                    }
                                };
                            } else {
                                internalEventIndex = internalEventIndex + 2;
                            }
//**************************************************************************************************************************************************************************************
                        }
                    };
                } else {
                    internalEventIndex++;
                }
            };
            listeningSteps();
        }
```

```
            setTimeout(listen,listeningInterval);
        }; listen();
}

function completeInit(){
        datSystem.makeFile(freshnessFilename,"");
        datSystem.onready = function(){
            datSystem.makeFile(eventsIndexFilename,"0");
            internalEventIndex = officialEventIndex = 0;
            startEventListening();
        };
}

function init(){
/* Erklärungen:
Da für jedes Ereignis eine indexversehene Ereignis-Datei "e0.xml", "e1.xml" usw. im Verz. "evDir" erstellt wird,
sollte zu Beginn jeder Sitzung von Usern "aufgeräumt" werden, was sich zuletzt an Ereignissdateien angesammelt hat.
Um sicherzugehen, dass dies wirklich eine neue Sitzung ist, wird anhand einer "freshnessMark"-Datei festgestellt, wie lange
die letzte Sitzung her ist. Ist die Marke älter als 60 Minuten, muss es sich um eine neue Sitzung handeln, da die Marke während einer
laufenden Sitzung (d.h. während mindestens 1 AjaxEventSystem-User online ist) alle paar Minuten oder häufiger aktualisiert wird
(Intervall "updatingFreshnessMark").
Im folgenden Code wird sichergestellt, dass der erste eintretende User aufräumt und ein neues Verz. "evDir" anlegt mit
freshnessMark-Datei und einer Event-Index-Datei darin, und dass wenn er nicht der erste ist, den derzeitigen offiziellen
Event-Index der Gruppe übernimmt. Pseudo-Code:

FALLS evDir da, DANN
        FALLS "freshness.dat" vorhanden, DANN
                FALLS "freshness.dat" frisch, DANN
                        FALLS evIndex da, DANN
                                interner Index = Inhalt von evIndex
                                offizieller Index = Inhalt von evIndex
                        SONST
                                erzeuge evIndex
                SONST
                        Leere evDir
                        CALL PrepareEvDIr()
        SONST
                CALL PrepareEvDIr()
SONST
        erzeuge evDir
        CALL PrepareEvDIr()

PROC PrepareEvDir()
        erzeuge "fressness.dat"
        erzeuge "evIndex mit Inhalt 0
        interner Index = 0
        offizieller Index = 0
*/

        var serverData = new ServerInfo();
        serverData.onready = function(){
            serverData.requestInfo("Date");
            serverData.onready = function(){
                var nowDate = new Date(serverData.getInfo());
                datSystem.search(eventsDirName);
                datSystem.onready = function(){
                    if(datSystem.exists()){
                        datSystem.search(freshnessFilename);
                        datSystem.onready = function(){
                            if(datSystem.exists()){
                                datSystem.requestFileDate(freshnessFilename);
                                datSystem.onready = function(){
                                    var fileDate = new Date(datSystem.getResponse());
                                    if(nowDate.getTime()-fileDate.getTime() < 3600000){
                                        datSystem.search(eventsIndexFilename);
                                        datSystem.onready = function(){
                                            if(datSystem.exists()){
                                                datSystem.request(eventsIndexFilename);
                                                datSystem.onready = function(){
                                                    internalEventIndex = officialEventIndex = parseInt(datSystem.getResponse());
                                                    startEventListening();
                                                };
                                            } else {
                                                datSystem.makeFile(eventsIndexFilename, "0");
                                                datSystem.onready = function(){
                                                    startEventListening();
                                                }
                                            }
                                        };
                                    } else {
                                        datSystem.emptyDir(eventsDirName);
                                        datSystem.onready = function(){
                                            completeInit();
                                        };
```

```
                                        }
                                    };
                                } else {
                                    completeInit();
                                }
                            };
                        } else {
                            datSystem.makeDir(eventsDirName);
                            datSystem.onready = function(){
                                completeInit();
                            }
                        }
                    };

            };
        };

        var updatingFreshnessMark = setInterval(function(){
            datSystem.onready = function(){
                datSystem.search(freshnessFilename);
                datSystem.onready = function(){
                    if(datSystem.exists())
                        datSystem.write(freshnessFilename,"");
                    else
                        datSystem.makeFile(freshnessFilename,"");
                };
            };
            },randomTo(300000)+300000);

        }
        /*
        //\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/*/


        //¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\/¯\
        // Properties and initializing Code:
        var officialEventIndex;
        var internalEventIndex;
        var ownFiredEvents = [];
        var eventType = "";
        var eventContent = "";
        var eventSource = "";
        var eventsDirName = "evDir";
        var eventsIndexFilename = eventsDirName + "/begInd.dat";
        var freshnessFilename = eventsDirName + "/freshnessMark.dat";
        var listeningInterval = 1000;
        var thisObj = this;
        this.onevent;
        var datSystem = new FileSystem();
        var firingDS = new FileSystem();
        datSystem.onready = init;
        /*
        //\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/\_/*/
    }

/*
\_____________/\_____________/\_____________/\_____________/\_____________/*/
//_____________\/_____________\/_____________\/_____________\/_____________\
//    == S E R V E R - F E A T U R E ==

    function ServerInfo(){

        this.requestInfo = function(headerName){
            var requestObject = Ajax.getObject(ajaxKey);
            requestObject.onreadystatechange = function(){
                if(requestObject.readyState == 4){
                    ready = 1;
                    response = requestObject.getResponseHeader(headerName);
                }
            };
            try {
                requestObject.open('HEAD','x');
                requestObject.send(null);
            } catch(e) { }
        }

        this.getInfo = function() {
```

```
        return response;
    }

    function continueIfReady(){
        var readinessChecker = setInterval(function(){
            if((typeof thisObj.onready)=="function" || (typeof thisObj.onready)=="string"){
                // Vom Benutzer spezifizierte Reaktion ausführen, wenn bereit:
                if(ready){
                    ready = 0;
                    setTimeout(thisObj.onready,1);
                    thisObj.onready = 0;
                }
            }
        },1);
    }

    // Properties and initializing Code:
    var serverTime;
    var serverName;
    var ready=1;
    var response;
    var thisObj = this;
    this.onready;
    var ajaxKey = get_ajaxKey();
    continueIfReady();
}


/*
\_____________/\_____________/\_____________/\_____________/\_____________/*/



//_____________\/_____________\/_____________\/_____________\/_____________\
//    ==SPEEDTESTER-FEATURE==

    function SpeedTester(){

        this.getSpeed = function(){
            return loadTimeFrame.document.getElementsByTagName("p")[1].innerHTML;
            loadTimeFrame = 0;
        };

        this.requestSpeed = function(){
            id("speedTestingArea").empty();
            id("speedTestingArea").add("iframe","","src","datSystem.php?action=getLoadtime").belongsToSystem = 1;
            loadTimeFrame = frames[frames.length-1];
        };

        var loadTimeFrame = 0;
        add("div","","id","speedTestingArea").style.display = "none";
        var thisObj = this;
        this.onready = 0;

        var readinessChecker = function(){
            if(loadTimeFrame){
                if(loadTimeFrame.document){
                    if(loadTimeFrame.document.getElementsByTagName("p")[1]){
                        if((typeof thisObj.onready)=="function"){
                            setTimeout(thisObj.onready,1);
                            thisObj.onready = 0;
                        }
                    }
                }
            }
            setTimeout(readinessChecker,111);
        }; readinessChecker();

    }

/*
\_____________/\_____________/\_____________/\_____________/\_____________/*/



//_____________\/_____________\/_____________\/_____________\/_____________\
//    ==PERFORMANCE-CHECK-FEATURE==

    var Performance = {};

    Performance.rateSpeed = function(){
        var cl = new Clock();
```

```
        cl.start();
        var x = "a";

        for(var i = 0; i < 50000; i++){
                x+="a";
                x="a";
                if(cl.stop()>3000)
                        break;
        }
        cl = cl.stop();
        if(cl >= 3000)
                return "very slow";
        if(cl >= 2250)
                return "slow";
        if(cl >= 1500)
                return "middle";
        if(cl >= 750)
                return "fast";
        if(cl >= 0)
                return "very fast";
};

Performance.rateMemory = function(){
        var cl = new Clock();
        cl.start();
        var x = "a";

        for(var i = 0; i < 30000; i++){
                x+="a";
                if(cl.stop()>3000)
                        break;
        }
        cl = cl.stop();
        if(cl >= 3000)
                return "very low";
        if(cl >= 2250)
                return "low";
        if(cl >= 1500)
                return "middle";
        if(cl >= 750)
                return "high";
        if(cl >= 0)
                return "very high";
};

Performance.rateGraphics = function(col){
        if(undef(col))
                var col = "white";

        var retObj = {  };
        var boxes = [];

        for(var i = 0; i < 8; i++){
                boxes[i] = makeBox(cX(500),cY(500),500,500, col);
                _(boxes[i]).hide();
        }


for(var i = 0; i < 8; i++){
        _(boxes[i]).show(2.5);
}


var graphicsTester = function(){
        var t = 500;
        if(isIE)        var op = parseInt(boxes[0].filters.alpha.opacity) / 100;
        else            var op = parseFloat(boxes[0].style.opacity);

        if(op==0.1)
                graphicsClock.start();
        if(op==0.12){
                var t = graphicsClock.stop();
                if(!isIE)
                        loop.timer.drop("graphicsTester");
                else
                        id("gt").onattrchange = function(){ };

                for(var i = 0; i < 8; i++){
                        _(boxes[i]).remove();
                }
                if((typeof retObj.onready) == "function"){

                        if(t>=500) retObj.graphicSpeed = "very slow";
                        if(t>=375 && t<500) retObj.graphicSpeed = "slow";
```

```
                    if(t>=225 && t<375) retObj.graphicSpeed = "middle";
                    if(t>=100 && t<225) retObj.graphicSpeed = "fast";
                    if(t>=0 && t<100) retObj.graphicSpeed = "very fast";

                    retObj.onready();
                }
            }
        };
        boxes[0].id = "gt";
        if(isIE)
            id("gt").onattrchange = graphicsTester;
        else
            loop.timer.add(graphicsTester, "graphicsTester");
        return retObj;
};  graphicsClock = new Clock();

/* P E R F O R M A N C E - C H E C K - F E A T U R E (end)
\_____________/\_____________/\_____________/\_____________/\_____________/*/



//_____________\_____________\_____________\_____________\_____________\
//    == M O U S E - F E A T U R E ==

        function mouseOnoveriframeFnc(e){
            d = 0;
            if(isIE){
                var e = window.event;
                var d = 2;
            }
            if(typeof Mouse.onoveriframe == "function"){
                for(var i = 0; i < tg("iframe").numOfAll; i++){
                    if(e.clientX > tg("iframe", i).offsetLeft && e.clientX < tg("iframe", i).offsetLeft + tg("iframe", i).offsetWidth +d && e.clientY > tg("iframe", i).offsetTop && e.clientY < tg("iframe", i).offsetTop + tg("iframe", i).offsetHeight + d)
                        Mouse.onoveriframe(e);
                }
            }
        }
        if(isIE)
            Event.add("document.onmouseout", mouseOnoveriframeFnc);
        else
            Event.add("window.onmouseout", mouseOnoveriframeFnc);


        function MouseListener(){
            //..... \ /..... \ /..... \
            // Public Functions:

            this.getX = function(){
                return this.x;
            };
            this.getY = function(){
                return this.y;
            };
            this.getScrX = function(){
                return this.scrX;
            };
            this.getScrY = function(){
                return this.scrY;
            };
            this.isAbove = function(elm){
                if(this.getX()>elm.offsetLeft && this.getY()>elm.offsetTop && this.getX()<elm.offsetLeft+elm.offsetWidth && this.getY()<elm.offsetTop+elm.offsetHeight)
                    return 1;
                else
                    return 0;
            }
            this.getElement = function(){
                return this.srcElm;
            };
            this.catchAll = function(){
                if(isIE){
                    var thisObj = this;
                    eventCatcher = add("ins");
                    eventCatcher.style.cssText = "position:absolute; left:-100px; top:-100px; display:none";
                    eventCatcher.xy = -1;
                    eventCatcher.onpropertychange = function(){
                        var e = window.event;
                        thisObj.x = e.clientX;
                        thisObj.y = e.clientY;
                        thisObj.scrX = e.screenX;
                        thisObj.scrY = e.screenY;
```

```
		}
		loop.timer.add(function(){
			eventCatcher.xy = -eventCatcher.xy;
		});
	} else {
		var topBox = makeBox(    0,0,        300,75, "green");
		var bottomBox = makeBox(     0,100,     300,75, "blue");
		var leftBox = makeBox(    0,75,       125,25, "yellow");
		var rightBox = makeBox(  175,75,    125,25, "red");

		topBox.onmouseover = function(e){
			topBox.style.top = (e.clientY-10-topBox.offsetHeight)+"px";

			bottomBox.style.top = (topBox.offsetTop+100)+"px";
			leftBox.style.top = (topBox.offsetTop+75)+"px";
			rightBox.style.top = (topBox.offsetTop+75)+"px";
		};
		bottomBox.onmouseover = function(e){
			bottomBox.style.top = (e.clientY+10)+"px";

			topBox.style.top = (bottomBox.offsetTop-100)+"px";
			leftBox.style.top = (bottomBox.offsetTop-25)+"px";
			rightBox.style.top = (bottomBox.offsetTop-25)+"px";
		};
		leftBox.onmouseover = function(e){
			leftBox.style.left = (e.clientX-10-leftBox.offsetWidth)+"px";

			topBox.style.left = leftBox.offsetLeft;
			bottomBox.style.left = leftBox.offsetLeft;
			rightBox.style.left = (leftBox.offsetLeft+175)+"px";
		};
		rightBox.onmouseover = function(e){
			rightBox.style.left = (e.clientX+10)+"px";

			topBox.style.left = (rightBox.offsetLeft-175)+"px";
			bottomBox.style.left = (rightBox.offsetLeft-175)+"px";
			leftBox.style.left = (rightBox.offsetLeft-175)+"px";
		};
	}
};

/* Public Functions (end)
/\.............. ...............\ \.............. ...............\ \.............. ...............*/

var clickhandler = function(e, thisObj){
	if (isIE) {
		var e = window.event;
		if(e.button == 1 && (typeof thisObj.onleftclick)=="function")
			thisObj.onleftclick();
		if(e.button == 2 && (typeof thisObj.onrightclick)=="function")
			thisObj.onrightclick();
		if(e.button == 4 && (typeof thisObj.onmiddleclick)=="function")
			thisObj.onmiddleclick();
	} else {
		if(e.which == 1 && (typeof thisObj.onleftclick)=="function")
			thisObj.onleftclick();
		if(e.which == 3 && (typeof thisObj.onrightclick)=="function")
			thisObj.onrightclick();
		if(e.which == 2 && (typeof thisObj.onmiddleclick)=="function")
			thisObj.onmiddleclick();
	}
};


var handler = function(e, thisObj){
	if(!e) var e = window.event;
	thisObj.x = e.clientX;
	thisObj.y = e.clientY;
	thisObj.scrX = e.screenX;
	thisObj.scrY = e.screenY;

	if(isIE)
		thisObj.srcElm = e.srcElement;
	else
		thisObj.srcElm = e.target;


};

var init = function(thisObj){
	Event.add("onbodyload", function(){
		if(isIE){
			Event.add("tg().onmousemove", function(e){          // für IE
				handler(e, thisObj);
```

```
                        });
                        Event.add("tg().onmousedown", function(e){          // für IE
                            clickhandler(e, thisObj);
                        });
                    } else {
                        Event.add("document.onmousemove", function(e){      // für FF
                            handler(e, thisObj);
                        });
                        Event.add("document.onmousedown", function(e){      // für FF
                            clickhandler(e, thisObj);
                        });
                    }
                });
            };

            this.x = null;
            this.y = null;
            this.scrX = null;
            this.scrY = null;
            this.srcElm = null;
            var eventCatcher = null;
            init(this);

        }; var Mouse = new MouseListener();

/*
\_____________/\_____________/\_____________/\_____________/\_____________/*/



//_____________\/_____________\/_____________\/_____________\/_____________\
//    == S T R I N G - F E A T U R E S ==

        function trimString(kette){
            if(kette.indexOf(" ")==0)
                kette = kette.substring(1,kette.length);
            if(kette.lastIndexOf(" ")==kette.length-1)
                kette = kette.substring(0,kette.length-1);
            return kette;
        }

        function randomString(c){
            var str = "";
            for(var i = 0; i<c; i++){
                str += String.fromCharCode(randomTo(25)+97);
            }
            return str;
        }

        function replaceString(suchwort, ersatzwort, satz){
            if(satz.indexOf(suchwort)>-1){
                satz = satz.substring(0,satz.indexOf(suchwort)) + ersatzwort + satz.substring(satz.indexOf(suchwort)+suchwort.length,satz.length);
                return satz;
            } else {
                return satz;
            }
        }

        function replaceStrings(suchstr, ersatzstr, satz){
            if(satz==null)
                return null;
            if(!isNaN(satz))
                satz = "" + satz;
            if((typeof satz)=="object")
                satz = String(satz);
            var teile = satz.split(suchstr);
            return teile.join(ersatzstr);
        }


        function replaceWord(suchwort, ersatzwort, satz){
            if(satz.toLowerCase().indexOf(suchwort.toLowerCase())>-1){
                satz = satz.substring(0,satz.toLowerCase().indexOf(suchwort.toLowerCase())) + ersatzwort + satz.substring(satz.toLowerCase().indexOf(suchwort.toLowerCase())+
suchwort.length,satz.length);
                return satz;
            } else {
                return satz;
            }
        }

        function replaceWords(suchwort, ersatzwort, satz){
            while(satz.toLowerCase().indexOf(suchwort.toLowerCase())>-1){
```

```
                satz = satz.substring(0,satz.toLowerCase().indexOf(suchwort.toLowerCase())) + ersatzwort + satz.substring(satz.toLowerCase().indexOf(suchwort.toLowerCase())+
suchwort.length,satz.length);
            }
            return satz;
        }



        function replaceAt(i, ersatzwort, satz){
            if(ersatzwort.length > 0)
                var span = ersatzwort.length;
            else
                var span = 1;
            var firstPart = satz.substring(0,i);
            var secondPart = satz.substring(i+span);
            return firstPart + ersatzwort + secondPart;

        }

        function shortenSpaces(str){
            var rStr = str;
            while(rStr.indexOf("  ") > -1){
                rStr = replaceStrings("  ", " ", rStr);
            }
            return trimString(rStr);
        }


function count(sw, str){
    return (str.split(sw)).length-1;
}

function tokenize(str){
    var arr = [];
    var replacements = [];
    var buchst = "";
    var open = -1;

    // Jedes gewünschte Trennzeichen bekommt ein repräsentatives Trennzeichen
    for(var i = 1; i < tokenize.arguments.length; i++){
        replacements[i] = "§"+randomString(3)+"§";
    }

    // Zähle Anf.-Zeichen
    // Falls ungerade Zahl, lösche letztes Anf.-Zeichen
    if(isOdd(count('"',str))){
        str = replaceAt(str.lastIndexOf('"'),' ',str);
    }

    // Schütze zwischen Anf.-Zeichen liegenden Text durch Verbergung der darin liegenden Trennzeichen, und dies durch Ersetzung mit Pseudotrennzeichen (replacements);
    for(var i=0; i<str.length; i++){
        buchst = str.substring(i,i+1);
        if(buchst=='"')
            open = open * (-1);
        for(var n = 1; n < tokenize.arguments.length; n++){
            if(open==1 && buchst==tokenize.arguments[n]){
                str = str.substring(0,i) + replacements[n] + str.substring(i+1,str.length);
            }
        }
    }
    str = replaceStrings('"', "", str);

    // ersetze alle übrigen Anf.-Zeichen, Punkte, Semikolons und doppelten Leerzeichen mit "[TRENNZE]":
    for(var i = 1; i < tokenize.arguments.length; i++){
        str = replaceStrings(tokenize.arguments[i],"[TRENNZE]",str);
    }

    // Splitte in Array mit Leerzeichen als Trennzeichenangabe:
    arr = str.split("[TRENNZE]");

    // Gehe zurück und verwandele §LZ§ etc. zurück
    for(var i=0; i<arr.length; i++){
        for(var n = 1; n < replacements.length; n++){
            arr[i] = replaceStrings(replacements[n], tokenize.arguments[n], arr[i]);
        }
        if(arr[i]==""){         // Nebenbei beseitige durch Trennzeichenkonzentrationen entstandene leere Elemente
            arr.drop(i); i--;
        }

    }

    return arr;
}
```

```
function capitalize(str){
	return str.substring(0,1).toUpperCase() + str.substring(1,str.length);
}

function isCapitalized(str){
	return (str.substring(0,1)!=str.substring(0,1).toLowerCase());
}

function toCSSName(s){
 for(var exp=/([a-z])([A-Z])/; exp.test(s); s=s.replace(exp,RegExp.$1.toLowerCase()+"-"+RegExp.$2.toLowerCase()));
 return s;
};

function toCamelCase(str){
	var pieces;
	if(str.indexOf("-")>-1){
		pieces = str.split("-");
		for(var i = 1; i < pieces.length; i++){
			pieces[i] = replaceAt(0, pieces[i].substring(0,1).toUpperCase(), pieces[i]);
		}
		return pieces.join("");
	}
	return str;
}
```

```
/*
\_____________/\_____________/\_____________/\_____________/\_____________/*/




//_____________\/_____________\/_____________\/_____________\/_____________\
//   ==CURSOR- AND MARKING-FEATURES==


	var Cursor = {};

	Cursor.paste = function(str1, str2) {
		if(!undef(document.activeElement.value)){
			if(undef(str2)) str2="";
			/* für Internet Explorer */
			if(typeof document.selection != 'undefined') {
				/* Einfügen des Formatierungscodes */
				var range = document.selection.createRange();
				var insText = range.text;
				range.text = str1 + insText + str2;
				/* Anpassen der Cursorposition */
				range = document.selection.createRange();
				if (insText.length == 0) {
					range.move('character', -str2.length);
				} else {
					range.moveStart('character', str1.length + insText.length + str2.length);
				}
				range.select();
			}
			/* für neuere auf Gecko basierende Browser */
			else  {
				var target = document.activeElement;
				/* Einfügen des Formatierungscodes */
				var start = target.selectionStart;
				var end = target.selectionEnd;
				var insText = target.value.substring(start, end);
				target.value = target.value.substr(0, start) + str1 + insText + str2 + target.value.substr(end);
				/* Anpassen der Cursorposition */
				var pos;
				if (insText.length == 0) {
					pos = start + str1.length;
				} else {
					pos = start + str1.length + insText.length + str2.length;
				}
				target.selectionStart = pos;
				target.selectionEnd = pos;
			}
		}
	};


	Cursor.getPos = function(borderType){
		if(!undef(document.activeElement.value)){
			if(!undef(document.selection)) {
				// [Internet Explorer:]
```

```
                var range = document.selection.createRange();
                if(document.activeElement.tagName == "INPUT"){ // (nicht für Textareas)
                    if(undef(borderType) || borderType==0 || borderType=="start")
                        return Math.abs(range.moveStart("character", -1000000));
                    if(borderType==1 || borderType=="end")
                        return Math.abs(range.moveEnd("character", -1000000));
                } else { // (auch für Textareas)
                    var element = document.activeElement;
                    var stored_range = range.duplicate(); // We'll use this as a 'dummy'
                    stored_range.moveToElementText( element ); // Select all text
                    stored_range.setEndPoint( 'EndToEnd', range ); // Now move 'dummy' end point to end point of original range
                    // Now we can calculate start and end points:
                    if(undef(borderType) || borderType==0 || borderType=="start")
                        return stored_range.text.length - range.text.length;
                    if(borderType==1 || borderType=="end")
                        return (stored_range.text.length - range.text.length) + range.text.length;
                }
            } else {
                // [Firefox:]
                if(undef(borderType) || borderType==0 || borderType=="start"){
                    return document.activeElement.selectionStart;
                }
                if(borderType==1 || borderType=="end"){
                    return document.activeElement.selectionEnd;
                }
            }
        } else {
            return null;
        }
    };

    Cursor.toEnd = function(){
        if((typeof document.activeElement.value) != 'undefined'){
            if(typeof document.selection != 'undefined'){
                var field = document.activeElement;
                var buffer = field.value;
                field.focus();

                var range = document.selection.createRange();
                range.move('character', -document.activeElement.value.length)
                range.moveEnd('character', 0)
                range.select();

                field.value = "";
                field.focus();
                field.value = buffer;
            } else {
                Cursor.to(document.activeElement.value.length);
            }
        }
    };


    Cursor.to = function(pos1, pos2){
        if(typeof document.activeElement.value != 'undefined'){
            if(typeof document.selection != 'undefined'){
                Cursor.toEnd();
                if((typeof pos2)=='undefined') pos2 = pos1;
                var range = document.selection.createRange();
                range.move('character', -document.activeElement.value.length+pos1)
                range.moveEnd('character', pos2-pos1)
                range.select();
            } else {
                if((typeof pos2)=='undefined') pos2 = pos1;
                document.activeElement.selectionStart = pos1;
                document.activeElement.selectionEnd = pos2;
            }
        }
    };

    Cursor.unselect = function(){
        if(browserName()=="Internet Explorer"){
            document.selection.empty();
        } else {
            if(!id("deselector")){
                var deselectorElement = add("input", "", "id", "deselector", "type", "text");
                deselectorElement.style.display = "none";
                id("deselector").focus();
            } else {
                id("deselector").focus();
            }
        }
    };
```

```
//..... \ /..... \ /..... \
// Save- & Restore-Feature:

    function GSel(){
        var d=document;
        if(d.selection)
            return d.selection.type=="Text" ? d.selection : null;
        if(window.getSelection)
            return window.getSelection();
        return null;
    }
    function CRng(){
        var sel=GSel();
        if(sel){
            if(sel.createRange) return sel.createRange();
            if(sel.rangeCount && sel.getRangeAt) return sel.getRangeAt(0);
        }
        return null;
    }
    function Sel(rng){
        if(rng.select) rng.select();
        else {
            var s=GSel();
            if(s.removeAllRanges && s.addRange){
                s.removeAllRanges();
                s.addRange(rng);
            }
        }
    }

    Cursor.save = function(){
        RNG = CRng();
    };

    Cursor.restore = function(){
        if(RNG) Sel(RNG);
    };

    var RNG=null;

/*Save- & Restore-Feature (end)
/\..... ...../ \..... ...../ \..... ...../*/

/*
\_____/\_____/\_____/\_____/\_____/*/




//_____\_____\_____\_____\_____\
//    == W I N D O W - F E A T U R E ==

function newWindow(url, x,y,w,h,winName) {
    if(newWindow.caller)
        var e = newWindow.caller.arguments[0];
    if(isIE){
        if(!window.event || window.event.type!="click") e = 0; else e = "[object MouseEvent]";
    }
    if(!e || e!="[object MouseEvent]"){
        var m = message('<center>Es wird ein neues Browserfenster geöffnet.<br></center><button id="winOpenerButton">Open</button><br><center>Bitte Eingabetaste drücken oder hier klicken.</center>', 10);
        id("winOpenerButton").focus();
        id("winOpenerButton").onclick = function(){
            newWindow(url, x,y,w,h, winName);
            if(!undef(winName))
                newWin.name = winName;
            m.close();
        };
    } else {
        var options ;
        options = "width=" + w + ",height=" + h + ",";
        options += "left=" + x + ",top=" + y + ",";
        options +="resizable=yes,scrollbars=yes,status=no,";
        options += "menubar=no,toolbar=no,location=no,directories=no";
        if(newWin)                    // falls schon ein Neufenster existiert, dann schließen, um neue Fenstergröße zu erzielen
            newWin.close();
        newWin = window.open(url, 'id'+Math.round(Math.random()*10000), options);
        newWin.focus();
```

```
            if(!undef(winName))
                newWin.name = winName;
            if(newWindow.onready && (typeof newWindow.onready)=="function")
                newWindow.onready();
        }
    } var newWin;


    function codeToNewWindow(code,x,y,w,h){
        var options ;
        options = "width=" + w + ",height=" + h + ",";
        options += "left=" + x + ",top=" + y + ",";
        options +="resizable=yes,scrollbars=yes,status=no,";
        options += "menubar=no,toolbar=no,location=no,directories=no";
        var targetWin = window.open("about:blank", 'id'+Math.round(Math.random()*10000), options);
        targetWin.document.writeln('<html><body leftmargin="0" topmargin="0">'+code+'</body></html>');
        window.setTimeout(function(){
            targetWin.stop();
        },10000);
    }


    function codeToChildWindow(code,x,y,w,h){
        var localaddress = location.href;
        var options ;
        var targetWin;
        options = "width=" + w + ",height=" + h + ",";
        options += "left=" + x + ",top=" + y + ",";
        options +="resizable=yes,scrollbars=yes,status=no,";
        options += "menubar=no,toolbar=no,location=no,directories=no";
        if(navigator.appName=="Microsoft Internet Explorer"){
            targetWin = window.open(location.href, 'id'+Math.round(Math.random()*10000), options);
            var schleife = function(){
                if(targetWin.location.href==localaddress  && targetWin.document.getElementsByTagName("body")[0]){
                        targetWin.document.getElementsByTagName("body")[0].innerHTML = code;
                } else {
                    setTimeout(schleife,1);
                }
            }; schleife();
        } else {
            targetWin = window.open('about:blank', 'id'+Math.round(Math.random()*10000), options);
            var schleife = function(){
                if(targetWin.location.href=="about:blank" && targetWin.document.getElementsByTagName("body")[0]){
                    targetWin.stop();
                    targetWin.document.getElementsByTagName('head')[0].innerHTML = tg('head',0).innerHTML;
                    targetWin.document.getElementsByTagName('body')[0].innerHTML = code;
                } else {
                    setTimeout(schleife,1);
                }
            }; schleife();
        }
    }


/*
\_____________/\_____________/\_____________/\_____________/\_____________/*/



function getViewportWidth(win) {
    if (win == undefined)
        win = window;
    if (win.innerWidth) {
        return win.innerWidth;
    } else {
        if (win.document.documentElement && win.document.documentElement.clientWidth) {
            return win.document.documentElement.clientWidth;
        }
        return win.document.documentElement.offsetWidth;
    }
}

function getViewportHeight(win) {
    if (win == undefined)
        win = self;
    if (win.innerHeight) {
        return win.innerHeight;
    } else {
        if (win.document.documentElement && win.document.documentElement.clientHeight) {
            return win.document.documentElement.clientHeight;
        }
        return win.document.documentElement.offsetHeight;
    }
```

```
}


function maximizeWindow() {
	if (navigator.appName=="Microsoft Internet Explorer") {
		window.moveTo(0,0);
		top.window.resizeTo(screen.availWidth,screen.availHeight);
	} else {
		if (top.window.outerHeight<screen.availHeight-20||top.window.outerWidth<screen.availWidth-20) {
			window.moveTo(0,0);
			window.resizeTo(screen.availWidth,screen.availHeight);
		}
	}
}


function setMinimumWindowInnerSize(x,y) {
	if(getViewportWidth(self)<x||getViewportHeight(self)<y){
		if (navigator.appName=="Microsoft Internet Explorer") {
			window.moveTo(0,0);
			top.window.resizeTo(screen.availWidth,screen.availHeight);
		} else {
			window.moveTo(0,0);
			window.resizeTo(screen.availWidth,screen.availHeight);
		}
	}
}


var viewportWidth = getViewportWidth();
var viewportHeight = getViewportHeight();

function getBrowserHeight(){
	if(browserName("Internet Explorer")){
		var h_n = getViewportHeight()+120;

		window.resizeTo(viewportWidth+9, h_n);

		var h2 = getViewportHeight();

		var origHeight = (h_n - h2) + viewportHeight;

		window.resizeTo(viewportWidth+8, origHeight);

		return origHeight;
	} else {
		return window.outerHeight;
	}
}

function hasMailFormat(mail) {
	var regexist = false;
	var res = false;
	if(typeof(RegExp) == 'function') {
		var testregex = new RegExp('abc');
		if(testregex.test('abc') == true) {
			regexist = true;
		}
	}

	if(regexist == true) {
		reg = new RegExp('^([a-zA-Z0-9\\-\\.\\_]+)(\\@)([a-zA-Z0-9\\-\\.]+)(\\.)([a-zA-Z]{2,4})$');
		res = (reg.test(mail));
	} else {
		res = (mail.search('@') >= 1 && mail.lastIndexOf('.') > mail.search('@') && mail.lastIndexOf('.') >= mail.length-5)
	}
	return(res);
}




function filenameFromPath(path){
	if(path.indexOf("\\")>-1)
		return path.substring(path.lastIndexOf("\\")+1);
	else if(path.indexOf("/")>-1)
		return path.substring(path.lastIndexOf("/")+1);
	else return path;
}


function dirFromPath(path){
	if(path.indexOf("\\")>-1)
```

```
        return path.substring(0,path.lastIndexOf("\\")+1);
    else if(path.indexOf("/")>-1)
        return path.substring(0,path.lastIndexOf("/")+1);
    else return path;
}




//_____________\/_____________\/_____________\/_____________\/_____________\
//    ==SUBMENU - FEATURE==

        function useSubmenus(){
            if(getElementsByClassName("submenu")) {
                firstSubmenuEntryHappened = []; var j=0;
                for(j=0; j < getElementsByClassName("submenu").length; j++)
                    firstSubmenuEntryHappened[j] = false;
                if(navigator.appName=="Microsoft Internet Explorer"){
                    setTimeout(
                        'for(submenuNo=0; submenuNo < getElementsByClassName("submenu").length; submenuNo++){                                    '+
                        '    getElementsByClassName("submenu")[submenuNo].childNodes[0].childNodes[0].nextSibling.nextSibling.style.visibility = "hidden"; '+
                        '    getElementsByClassName("submenu")[submenuNo].childNodes[0].childNodes[0].nextSibling.nextSibling.style.position = "absolute";    '+
                        '    getElementsByClassName("submenu")[submenuNo].childNodes[0].childNodes[0].nextSibling.nextSibling.style.left = "0px";         '+
                        '    getElementsByClassName("submenu")[submenuNo].outerHTML = replaceString("<LI ","<LI onmouseover=\\"javascript:enterSubmenu("+submenuNo+")\\" ",
getElementsByClassName("submenu")[submenuNo].outerHTML);  '+
                        '}'
                    ,1);
                } else {
                    setTimeout(
                        'for(submenuNo=0; submenuNo < getElementsByClassName("submenu").length; submenuNo++){                                    '+
                        '    getElementsByClassName("submenu")[submenuNo].childNodes[0].childNodes[0].nextSibling.nextSibling.style.visibility = "hidden"; '+
                        '    getElementsByClassName("submenu")[submenuNo].childNodes[0].childNodes[0].nextSibling.nextSibling.style.position = "absolute";    '+
                        '    getElementsByClassName("submenu")[submenuNo].childNodes[0].childNodes[0].nextSibling.nextSibling.style.left = "0px";         '+
                        '    getElementsByClassName("submenu")[submenuNo].setAttribute("onmouseover","javascript:enterSubmenu("+submenuNo+")");         '+
                        '    getElementsByClassName("submenu")[submenuNo].setAttribute("onmouseout","javascript:exitSubmenu("+submenuNo+")");        '+
                        '}'
                    ,1);
                }
            }
        }

        function enterSubmenu(submenuNo){

            if(navigator.appName=="Microsoft Internet Explorer" && !firstSubmenuEntryHappened[submenuNo]){
                getElementsByClassName("submenu")[submenuNo].childNodes[0].childNodes[0].nextSibling.nextSibling.innerHTML = "<br>" + getElementsByClassName("submenu")[
submenuNo].childNodes[0].childNodes[0].nextSibling.nextSibling.innerHTML;
                firstSubmenuEntryHappened[submenuNo] = true;
            }
            getElementsByClassName("submenu")[submenuNo].childNodes[0].childNodes[0].nextSibling.nextSibling.style.visibility = "visible";
        }

        function exitSubmenu(submenuNo){
            getElementsByClassName("submenu")[submenuNo].childNodes[0].childNodes[0].nextSibling.nextSibling.style.visibility = "hidden";
        }

/*
\_____________/\_____________/\_____________/\_____________/\_____________/*/




//_____________\/_____________\/_____________\/_____________\/_____________\
//    ==MATHEMATICAL FUNCTIONS==

    function randomTo(max){
        var digits = getDigits(max);
        var result = 0;
        for(var i=0; i<digits.length; i++){
            for(var k=0; k<digits[i]; k++){
                result += Math.round(Math.random()*Math.pow(10,i));
            }
        }
        return result;
    }


    function getDigits(value){
        var digits = [];
```

```
        value = String(value);
        for(var i=0; i<value.length; i++)
                digits.push(parseInt(value.charAt(i)));
        return digits.reverse();
}

function countDigits(value){
        var nod = 0;
        while(value>=1){
                nod++;
                value = value / 10;
        }
        return nod;
}

function round(val, numOfDigits){
        if(undef(numOfDigits))
                numOfDigits = 0;
        val = val * (Math.pow(10,numOfDigits));
        val = Math.round(val);
        return val / (Math.pow(10,numOfDigits));
}

function makeEven(num){
        if(round(num)%2>0){
                num--;
        }
        return round(num);
}

function numIn(str){
        if(!isNaN(parseInt(str)))
                return parseInt(str);
        return parseFloat(str.match(/[.\-\d]/g, "").join(""));
}

function isOdd(n){
        return (n%2);
}

function isInt(num){
        var str = ""+num;
        if(str.indexOf(".")>-1)
                return 0;
        else
                return 1;
}

function forceDigits(val, digNum){
        var neededZeros = digNum - countDigits(val);
        var val = "" + val;
        var additionalDigits = "";
        if(neededZeros>0){
                for(var i = 0; i < neededZeros; i++){
                        additionalDigits+="0";
                }
        }
        return additionalDigits+val;
}

function ignoreLeadingZeros(str){
        while(str.indexOf("0")==0){
                str = str.substr(1,str.length);
        }
        if(str=="")
                return 0;
        else
                return parseInt(str);
}

/*
\_____________/\_____________/\_____________/\_____________/\_____________/*/



//_____________\/_____________\/_____________\/_____________\/_____________\
//    ==CLOCK-FEATURE==


function Clock(){
        this.stopzeit = 0;
        this.start = function() {
                var zeit = new Date();
                this.stopzeit = Date.UTC(zeit.getYear(), zeit.getMonth(), zeit.getDay(), zeit.getHours(), zeit.getMinutes(), zeit.getSeconds()) + zeit.getMilliseconds();
```

```
        }
        this.stop = function() {
            var zeit = new Date();
            return Date.UTC(zeit.getYear(), zeit.getMonth(), zeit.getDay(), zeit.getHours(), zeit.getMinutes(), zeit.getSeconds()) + zeit.getMilliseconds() - this.stopzeit;
        }
    }


/*
\_____________/\_____________/\_____________/\_____________/\_____________/*/




//_____________\/_____________\/_____________\/_____________\/_____________\
//    ==SLEEPING-FEATURE==

    function sleep(gewZeit){
        var sleepClock = new Clock();
        sleepClock.start();
        do{
        }while(sleepClock.stop()<gewZeit);
    }

/*
\_____________/\_____________/\_____________/\_____________/\_____________/*/



function keysort(toBeSorted){
    var tempArray = [];
    var returnArray = [];
    for(var i in toBeSorted){
        if(keyOk(i))
            tempArray.push(i);
    }
    tempArray.sort();
    for(var j = 0; j < tempArray.length; j++){
        returnArray[tempArray[j]] = toBeSorted[tempArray[j]];
    }
    return returnArray;
}

function getKeyOfPeak(obj){
    var highestVal = null;
    var peakKey = null;
    for(var i in obj){if(keyOk(i)){
        if(highestVal!=null){
            if(obj[i] > highestVal){
                highestVal = obj[i];
                peakKey = i;
            }
        } else {
            highestVal = obj[i];
            peakKey = i;
        }
    }}
    return peakKey;
}

function getKeyOfLowest(obj){
    var lowestVal = null;
    var peakKey = null;
    for(var i in obj){if(keyOk(i)){
        if(lowestVal!=null){
            if(obj[i] < lowestVal){
                lowestVal = obj[i];
                peakKey = i;
            }
        } else {
            lowestVal = obj[i];
            peakKey = i;
        }
    }}
    return peakKey;
}


function valuesort(asArr, ASC){
    var newArr = {};
    var key = null;
```

```
	if(undef(ASC)) var ASC = 1;

	while(asArr.hasProperties()){
		if(ASC<0)
			key = getKeyOfPeak(asArr);
		if(ASC>0)
			key = getKeyOfLowest(asArr);

		if(keyOk(key))
			newArr[key] = asArr[key];
		delete(asArr[key]);
	}
	return newArr;

}


function hasArrayElements(v){
	var x = null;
	try {
		if((typeof c)!="string"){
			x = v.length;
			if(x>0)
				return 1;
			else
				return 0;
		} else {
			return 0;
		}
	} catch(e){
		return 0;
	}
}


function skypeInstalled(){
	var activex = ((navigator.userAgent.indexOf('Win')  != -1) && (navigator.userAgent.indexOf('MSIE') != -1) && (parseInt(navigator.appVersion) >= 4 ));
	if(activex){	// Falls es ein AcitiveX-Browser ist (also IE auf Windows-PCs)
		document.write('<script language="VBScript">Function vbsFoundSkype : on error resume next : Set oSkype = CreateObject("Skype.Detection") : vbsFoundSkype = IsObject(oSkype)
: Set oSkype = nothing : End Function<\/script>');
		if(vbsFoundSkype())
			return 1;
	}else if(preciseBrowserName()=="Mozilla Firefox"){  // Falls es Firefox ist
		if(typeof(navigator.mimeTypes["application/x-skype"])=="object")
			return 1;
		else
			return 0;
	} else {
		return 0;
	}
}


function javaActive(codebase){
	if(isIE){
		try {
		  var jws = new ActiveXObject('JavaWebStart.isInstalled');
		  return true;
		}
		catch (e) {
		  return false;
		}
	} else {
		if(undef(codebase)) codebase=dirFromPath(location.href);
		if(!id('javaDetecting')) tg().add("div", "", "id", "javaDetecting");
		id('javaDetecting').style.css = "position:absolute; visibility:hidden; width:0; height:0;";
		id('javaDetecting').innerHTML='<applet codebase="'+codebase+'" code="JavaDetecting" width="0" height="0" mayscript="mayscript" name="detectingApplet"></applet>';
		try{
			detectingApplet.testMethod("xy");
			return 1;
		}catch(e){
			return 0;
		}
	}
}


//_____________\/_____________\/_____________\/_____________\/_____________\
//   ==URL PARAMETER FEATURE==
```

```
var URLParams = {};


function urlParameter(varName, url, sign){
	if(undef(sign) && (typeof url)!="string")
		var sign = "?";
	if(undef(sign) && (typeof url)=="string" && url.length==1){
		var sign = url;
	}
	if(undef(sign))
		var sign = "?";
	if((typeof url)=="string" && url.length>1){	// Falls komplette URL dabei
		if(url.indexOf(sign)==-1)
			return null;
		else
			var paramListString = (url.split(sign))[1];
	} else if((typeof url)=="string" && url.length==0){ // Falls keine URL, aber Parameterzeichen dabei
			var url = window.location.href;
			if(url.indexOf(sign)==-1)
				return null;
			else
				var paramListString = (url.split(sign))[1];
	} else {	// Falls nur Var-Name ohne URL und Parameterzeichen angegeben
		var url = window.location.href;
		if(url.indexOf(sign)==-1)
			return null;
		else
			var paramListString = (url.split(sign))[1];
	}


	var parameters = [];
	if(paramListString.indexOf("&")>-1){
		var expressions = paramListString.split("&");
		for(var i=0; i<expressions.length;i++){
			var varAndVal = expressions[i].split("=");
			parameters[varAndVal[0]] = varAndVal[1];
		}
	} else {
		var varAndVal = paramListString.split("=");
		parameters[varAndVal[0]] = varAndVal[1];
	}
	return parameters[varName];
}

URLParams.get = urlParameter;


URLParams.set = function (varName, val, url){
	if(url.indexOf("?") == -1){				// Falls es keine typische Parameter-URL ist, dann Fragezeichen und Parameter-Term hinten dran.
		url+="?"+varName+"="+val;
	} else if(url.indexOf("?"+varName+"=") == -1 && url.indexOf("&"+varName+"=") == -1){	// Falls doch, aber Parametername nicht drin, dann nur Parameter-Term einsetzen.
		if(url.indexOf("?") == url.length-1)
			var concat = "";
		else
			var concat = "&";
		url = replaceStrings("?", "?"+varName+"="+val+concat, url);
	} else {					// Falls Parametername drin, dann Wert austauschen
		var paramListString = (url.split("?"))[1];
		var paramTerms = paramListString.split("&");
		for(var i = 0; i < paramTerms.length; i++){
			if(paramTerms[i].indexOf(varName+"=")==0)
				paramTerms[i] = (varName+"="+val);
		}
		paramListString = paramTerms.join("&");
		url = (url.split("?"))[0]+"?"+paramListString;


	}

	return url;
};


URLParams.extend = function (varName, val, url){
	var newValString = URLParams.get(varName, url) + "," + val;
	return URLParams.set(varName, newValString, url);

};

URLParams.retract = function (varName, val, url){
	var valString = URLParams.get(varName, url);
	var arr = valString.split(",");
```

```
        arr.dropByValue(val);
        valString = arr.join(",");

        return URLParams.set(varName, valString, url);
    };

    URLParams.drop = function (varName, url){
        var val = URLParams.get(varName, url);
        var paramListString = url.split("?")[1];
        var paramTerms = paramListString.split("&");
        for(var i = 0; i < paramTerms.length; i++){
            if(paramTerms[i]==varName+"="+val)
                paramTerms.drop(i);
        }

        return url.split("?")[0] + "?" + paramTerms.join("&");
    };




/*    URL PARAMETER FEATURE(end)
\_____________/\_____________/\_____________/\_____________________/\___________________/*/


function getTarget(e){
    if(isIE)
        return window.event.srcElement;
    else
        return getTarget.caller.arguments[0].target;
};

function usedKey1(k, a, b){

    if(undef(a)){
        if(k){
            return k.which;
        } else if(window.event.keyCode){
            if(window.event.keyCode){
                return window.event.keyCode;
            }
        }
        return 0;

    } else if(undef(b)) {
        if(k){
            if(k.which == a)
                return 1;
        } else if(window.event.keyCode){
            if(window.event.keyCode){
                if(window.event.keyCode==a)
                    return 1;
            }
        }
        return 0;

    } else {
        // Funktionsweise dieses Funtionsabschnitts: bei Drücken einer Taste wird diese in
        // der globalen Variable 'currentlyPressedKey' gespeichert. Bei Loslassen
        // der Taste wird sie aus der globalen Variable gelöscht.
        // Wird beim Drücken einer Taste festgestellt, dass die globale Variable
        // noch einen Tastendruck gespeichert hat, wird dieser alte Tastendruck und
        // der neue mit den gesuchten Tasten verglichen und entsprechendes zurückgegeben.

        if((document.onkeyup + "").indexOf("currentlyPressedKeyDeletingFunction")==-1){
            document.onkeyup = function(k){
                var currentlyPressedKeyDeletingFunction;
                currentlyPressedKey = 0;
            }
        }

        var firstKey = currentlyPressedKey;
        currentlyPressedKey = usedKey1(k);
        if((firstKey == a && currentlyPressedKey == b) || (firstKey == b && currentlyPressedKey == a))
            return 1;
        else
            return 0;

    }

} currentlyPressedKey = 0;
```

```
function usedKey(a, b, c, d){

    var k= usedKey.caller.arguments[0];

    if(undef(a)){
        if(k){
            return k.which;
        } else if(window.event && window.event.keyCode){
            return window.event.keyCode;
        }
        return 0;
    } else if(undef(b)) {
        if(k){
            if(k.which == a)
                return 1;
        } else if(window.event.keyCode){
            if(window.event.keyCode){
                if(window.event.keyCode==a)
                    return 1;
            }
        }
        return 0;

    } else {

        if((document.onkeyup + "").indexOf("currentlyPressedKeyDeletingFunction")==-1){
            document.onkeyup = function(k){
                var currentlyPressedKeyDeletingFunction;
                delete(currentlyPressedKeys[String(usedKey())]);
            }
        }

        if(k){
            newKey = k.which;
        } else if(window.event.keyCode){
            if(window.event.keyCode){
                newKey = window.event.keyCode;
            }
        }

        currentlyPressedKeys[String(newKey)] = newKey;

        if(!undef(d)){
            if(currentlyPressedKeys[String(a)] && currentlyPressedKeys[String(b)] && currentlyPressedKeys[String(c)] && currentlyPressedKeys[String(d)]) return 1;  else return 0;
        } else if(!undef(c)){
            if(currentlyPressedKeys[String(a)] && currentlyPressedKeys[String(b)] && currentlyPressedKeys[String(c)]) return 1;  else return 0;
        } else {
            if(currentlyPressedKeys[String(a)] && currentlyPressedKeys[String(b)]) return 1;  else return 0;
        }

    }

} currentlyPressedKeys = [];


function getKeyName(code) {
    Output=document.getElementById('show');
    var text;

    switch (code) {
        case 6: text = "mac"; // backspace
        break;
        case 8: text = "backspace"; // backspace
        break;
        case 9: text = "tab"; // tab
        break;
        case 13: text = "enter"; // enter
        break;
        case 16: text = "shift"; // shift
        break;
        case 17: text = "ctrl"; // ctrl
        break;
        case 18: text = "alt"; // alt
        break;
        case 19: text = "break"; // pause/break
        break;
        case 20: text = "caps lock"; // caps lock
        break;
        case 27: text = "esc"; // escape
        break;
        case 32: text = "space"; // Blank
        break;
        case 33: text = "page up"; // page up, to avoid displaying alternate character and confusing people
        break;
```

```
        case 34: text = "page down"; // page down
        break;
        case 35: text = "end"; // end
        break;
        case 36: text = "home"; // home
        break;
        case 37: text = "left arrow"; // left arrow
        break;
        case 38: text = "up arrow"; // up arrow
        break;
        case 39: text = "right arrow"; // right arrow
        break;
        case 40: text = "down arrow"; // down arrow
        break;
        case 43: text = "numpad plus";
        break;
        case 45: text = "insert"; // insert
        break;
        case 46: text = "delete"; // delete
        break;
        case 91: text = "left windows key"; // left window
        break;
        case 92: text = "right windows key"; // right window
        break;
        case 93: text = "select"; // select key
        break;
        case 96: text = "numpad 0"; // numpad 0
        break;
        case 97: text = "numpad 1"; // numpad 1
        break;
        case 98: text = "numpad 2"; // numpad 2
        break;
        case 99: text = "numpad 3"; // numpad 3
        break;
        case 100: text = "numpad 4"; // numpad 4
        break;
        case 101: text = "numpad 5"; // numpad 5
        break;
        case 102: text = "numpad 6"; // numpad 6
        break;
        case 103: text = "numpad 7"; // numpad 7
        break;
        case 104: text = "numpad 8"; // numpad 8
        break;
        case 105: text = "numpad 9"; // numpad 9
        break;
        case 106: text = "multiply"; // multiply
        break;
        case 107: text = "add"; // add
        break;
        case 109: text = "subtract"; // subtract
        break;
        case 110: text = "decimal point"; // decimal point
        break;
        case 111: text = "devide"; // divide
        break;
        case 112: text = "F1"; // F1
        break;
        case 113: text = "F2"; // F2
        break;
        case 114: text = "F3"; // F3
        break;
        case 115: text = "F4"; // F4
        break;
        case 116: text = "F5"; // F5
        break;
        case 117: text = "F6"; // F6
        break;
        case 118: text = "F7"; // F7
        break;
        case 119: text = "F8"; // F8
        break;
        case 120: text = "F9"; // F9
        break;
        case 121: text = "F10"; // F10
        break;
        case 122: text = "F11"; // F11
        break;
        case 123: text = "F12"; // F12
        break;
        case 144: text = "num lock"; // num lock
        break;
        case 145: text = "scroll lock"; // scroll lock
        break;
```

```
            case 186: text = "Ü"; // semi-colon
            break;
            case 187: text = "+"; // equal-sign
            break;
            case 188: text = ","; // comma
            break;
            case 189: text = "-"; // dash
            break;
            case 190: text = "."; // period
            break;
            case 191: text = "#"; // forward slash
            break;
            case 192: text = "Ö"; // grave accent
            break;
            case 219: text = "ß"; // open bracket
            break;
            case 220: text = "^"; // back slash
            break;
            case 221: text = "´"; // close bracket
            break;
            case 222: text = "Ä"; // single quote
            break;
    }

    if(!text) text = String.fromCharCode(code);
    return text;
}


function getChar(code) {
    Output=document.getElementById('show');
    var text = null;

    switch (code) {
        case 6: text = ""; // backspace
        break;
        case 8: text = ""; // backspace
        break;
        case 9: text = " "; // tab
        break;
        case 13: text = ""; // enter
        break;
        case 16: text = ""; // shift
        break;
        case 17: text = ""; // ctrl
        break;
        case 18: text = ""; // alt
        break;
        case 19: text = ""; // pause/break
        break;
        case 20: text = ""; // caps lock
        break;
        case 27: text = "" // escape
        break;
        case 32: text = " "; // Blank
        break;
        case 33: text = ""; // page up, to avoid displaying alternate character and confusing people
        break;
        case 34: text = ""; // page down
        break;
        case 35: text = ""; // end
        break;
        case 36: text = ""; // home
        break;
        case 37: text = ""; // left arrow
        break;
        case 38: text = ""; // up arrow
        break;
        case 39: text = ""; // right arrow
        break;
        case 40: text = ""; // down arrow
        break;
        case 43: text = "+";
        break;
        case 45: text = ""; // insert
        break;
        case 46: text = ""; // delete
        break;
        case 59: text = "Ü"; // delete
        break;
        case 91: text = ""; // left window
        break;
        case 92: text = ""; // right window
        break;
```

```
        case 93: text = ""; // select key
        break;
        case 96: text = "0"; // numpad 0
        break;
        case 97: text = "1"; // numpad 1
        break;
        case 98: text = "2"; // numpad 2
        break;
        case 99: text = "3"; // numpad 3
        break;
        case 100: text = "4"; // numpad 4
        break;
        case 101: text = "5"; // numpad 5
        break;
        case 102: text = "6"; // numpad 6
        break;
        case 103: text = "7"; // numpad 7
        break;
        case 104: text = "8"; // numpad 8
        break;
        case 105: text = "9"; // numpad 9
        break;
        case 106: text = "*"; // multiply
        break;
        case 107: text = "+"; // add
        break;
        case 109: text = "-"; // subtract
        break;
        case 110: text = ","; // decimal point
        break;
        case 111: text = "/"; // divide
        break;
        case 112: text = ""; // F1
        break;
        case 113: text = ""; // F2
        break;
        case 114: text = ""; // F3
        break;
        case 115: text = ""; // F4
        break;
        case 116: text = ""; // F5
        break;
        case 117: text = ""; // F6
        break;
        case 118: text = ""; // F7
        break;
        case 119: text = ""; // F8
        break;
        case 120: text = ""; // F9
        break;
        case 121: text = ""; // F10
        break;
        case 122: text = ""; // F11
        break;
        case 123: text = ""; // F12
        break;
        case 144: text = ""; // num lock
        break;
        case 145: text = ""; // scroll lock
        break;
        case 186: text = "Ü"; // semi-colon
        break;
        case 187: text = "+"; // equal-sign
        break;
        case 188: text = ","; // comma
        break;
        case 189: text = "-"; // dash
        break;
        case 190: text = "."; // period
        break;
        case 191: text = "#"; // forward slash
        break;
        case 192: text = "Ö"; // grave accent
        break;
        case 219: text = "ß"; // open bracket
        break;
        case 220: text = "^"; // back slash
        break;
        case 221: text = "´"; // close bracket
        break;
        case 222: text = "Ä"; // single quote
        break;
    }
```

```
	if(text==null) text = String.fromCharCode(code);
	return text;
}


function usedButton(){
	var b = usedButton.caller.arguments[0];
	if(browserName()=="Internet Explorer" || browserName()=="Konqueror"){
		if(window.event.button==1) return "left";
		if(window.event.button==4) return "middle";
		if(window.event.button==2) return "right";
	} else if(browserName()=="Firefox") {
		if(b.which==1) return "left";
		if(b.which==2) return "middle";
		if(b.which==3) return "right";
	} else if(browserName()=="Safari" || browserName()=="Chrome") {
		if(window.event.button==0) return "left";
		if(window.event.button==1) return "middle";
		if(window.event.button==2) return "right";
	}
}


function onAlt(key, fkt){
	var hiddenBox = tg().add("div","");
	with(hiddenBox.style){ position = "absolute"; left = "-1000px"; top = "-1000px"; }
	hiddenBox.innerHTML+='<a href="javascript:" accesskey="'+key+'" id="hiddenAnchor"></a>';
	hiddenBox.getElementsByTagName("a")[0].onfocus = function(){
		id("hiddenAnchor").blur();
		fkt();
	};
}



function add(tagName, content, attrName1, attrVal1, attrName2, attrVal2, attrName3, attrVal3, attrName4, attrVal4, attrName5, attrVal5){
	if(undef(content))
		content = "";
	var tag = document.createElement(tagName);
	if((typeof attrName1)=="string"){
		tag.setAttribute(attrName1, attrVal1);
		if(attrName1=="class") tag.className = attrVal1;
		if((typeof attrName2)=="string"){
			tag.setAttribute(attrName2, attrVal2);
			if(attrName2=="class") tag.className = attrVal2;
		if((typeof attrName3)=="string"){
			tag.setAttribute(attrName3, attrVal3);
			if(attrName3=="class") tag.className = attrVal3;
			if((typeof attrName4)=="string"){
				tag.setAttribute(attrName4, attrVal4);
				if(attrName4=="class") tag.className = attrVal4;
				if((typeof attrName5)=="string")
					tag.setAttribute(attrName5, attrVal5);
					if(attrName5=="class") tag.className = attrVal5;
			}
		}
		}
	}
	if(content!="")
		tag.appendChild(document.createTextNode(content));
	tg('body',0).appendChild(tag);
	return tag;
}



function coverDisplay(col){
	var cover = add("div","");
	if(undef(col)) col = "green";
	cover.style.cssText = "position:fixed; left:0px; top:0px; width:100%; height:100%; z-index:10000; background-color:"+col+";";
	cover.remove = function(){
		cover.parentNode.removeChild(cover);
	}
	return cover;
}

function getX(knoten){
	if(knoten==tg('body',0))
		return 0;
	else
		return knoten.offsetLeft + getX(knoten.parentNode);
}
```

```
function getY(knoten){
	if(knoten==tg('body',0))
		return 0;
	else
		return knoten.offsetTop + getY(knoten.parentNode) - knoten.parentNode.scrollTop;
}




function getClickTarget(e){
	if(navigator.appName=="Microsoft Internet Explorer")
		return window.event.srcElement;
	else
		return e.target;
}



Event.add("document.onmouseout", function(e){
	if(!e)
		e = window.event;
	if((typeof document.onmouseleftall)=="function" && (e.clientX<0 || e.clientY<0 || e.clientX >= getViewportWidth() || e.clientY >= getViewportHeight())){
		document.onmouseleftall(e);
	}
},"mouseleftall");



function keyListener(){
	document.onkeydown = function(k){
		if((typeof document.onenter)=="function")
			if(usedKey()==13)
				setTimeout(document.onenter,1);
		if((typeof document.onuparrow)=="function")
			if(usedKey()==38)
				setTimeout(document.onuparrow,1);
		if((typeof document.ondownarrow)=="function")
			if(usedKey()==40)
				setTimeout(document.ondownarrow,1);
		if((typeof document.onleftarrow)=="function")
			if(usedKey()==37)
				setTimeout(document.onleftarrow,1);
		if((typeof document.onrightarrow)=="function")
			if(usedKey()==39)
				setTimeout(document.onrightarrow,1);
		if((typeof document.onescape)=="function")
			if(usedKey()==27)
				setTimeout(document.onescape,1);
		function onkeydownEnd() {}; // This is only for marking purposes.
	};

	// Overwriting-Observer:
	Event.guard("document.onkeydown");

} keyListener();



//_____________\_____________\_____________\_____________\_____________\
//    ==ONDEACTIVATE  FEATURE==

	Event.add("onbodyload", function(){
		previousActiveElement = document.activeElement;

		loop.timer.add(function(){
			if(previousActiveElement && document.activeElement!=previousActiveElement && (typeof previousActiveElement.ondeactivate)=="function"){
				previousActiveElement.ondeactivate();
				previousActiveElement = document.activeElement;
			}
			previousActiveElement = document.activeElement;
		});
	}); var previousActiveElement = null;


/*    ONDEACTIVATE  FEATURE(end)
\_____________/\_____________/\_____________/\_____________/\_____________/*/



function googleAdsTo(x,y, i){
	if((typeof i)=="String")
		i = parseInt(i);
	var el = document.getElementsByName("google_ads_frame")[i];
	el.style.position = "absolute";
```

```
    el.style.left = x+"px";
    el.style.top = y+"px";
}



//_____________\_____________\_____________\_____________\_____________\
//    ==GET NUMBER OF PRESSED KEYS==

    var pressedKeys = [];

    function getPressedKeysNum(){
        return pressedKeys.length;
    }


    Event.add("document.onkeydown", function(e){
        if(!e) var e = window.event;
        if(!pressedKeys.contains(e.keyCode))
            pressedKeys.push(e.keyCode);
    });


    Event.add("document.onkeyup", function(e){
        if(!e) var e = window.event;
        pressedKeys.dropByValue(e.keyCode);
    });

/*    GET NUMBER OF PRESSED KEYS(end)
\_____________/\_____________/\_____________/\_____________/\_____________/*/


//_____________\_____________\_____________\_____________\_____________\
//    ==MESSAGE-BOX - FEATURE==


        function messageBox(capTitle,htmContent,x,y, w, h){

            if(undef(w)) w = 300;
            if(undef(h)) h = 300;
            if(undef(x)) x = cX(w);
            if(undef(y)) y = cY(h)-50;

            var mb = makeBox(x,y,w,h,"green");

            mb.caption = makeBox(x,y,w,20);
            mb.contentArea = makeBox(x, y+20, w, h-20);
            mb.closeButton = makeBox(x+w-18,y+2,16,16);
            mb.onclose = "";

            // Fixing:
            var mustBeFixed = (!isIE || document.hasDoctype())?true:false;
            mb.style.position = mustBeFixed?"fixed":"absolute";
            mb.caption.style.position = mustBeFixed?"fixed":"absolute";
            mb.contentArea.style.position = mustBeFixed?"fixed":"absolute";
            mb.closeButton.style.position = mustBeFixed?"fixed":"absolute";


            with(mb){
                caption.className = "mbCaption";
                contentArea.className = "mbContentArea";
                closeButton.className = "mbCloseButton";

                if(!mbStylesAreSet){
                    CSS.prepend('.mbCaption {  z-index:10010; background-color:#444444; color:white; font-family:Arial; font-weight:bold; font-size:16px;
padding-left:3px;-moz-box-sizing:border-box; -webkit-box-sizing:border-box; box-sizing:border-box; }  .mbContentArea {  z-index:10000; overflow:auto; background-color:#AAAAAA;
color:black;  font-family:Arial; font-size:12px; padding:3px 3px 3px 3px; -moz-box-sizing:border-box; -webkit-box-sizing:border-box; box-sizing:border-box; }  .mbCloseButton {
z-index:10020; color:white; background-color:#550000; line-height:15px; font-family:Arial; font-size:17px; font-weight:bold; border:outset 1px gray; text-align:center; overflow:hidden;
cursor:default; -moz-box-sizing:border-box; -webkit-box-sizing:border-box; box-sizing:border-box; }');
                    mbStylesAreSet = 1;
                }

                caption.innerHTML = capTitle;
                contentArea.innerHTML = htmContent;
                closeButton.innerHTML = "<b>&times;</b>";

                mb.resizeTo = function(w,h){
                    mb.style.width = w+"px";
                    mb.style.height = h+"px";
                    caption.style.width = w+"px";
                    contentArea.style.width = w+"px";
                    contentArea.style.height = h+"px";
                    closeButton.style.left = (offsetLeft + w - 18)+"px";
                };
```

```
mb.moveTo = function(x,y){
	mb.style.left = x+"px";
	mb.style.top = y+"px";
	caption.style.left = x+"px";
	caption.style.top = y+"px";
	contentArea.style.left= x+"px";
	contentArea.style.top = (y+20)+"px";
	closeButton.style.left= (x + offsetWidth - 18)+"px";
	closeButton.style.top = (y+2)+"px";
};

mb.close = function(){
	_(caption).remove();
	_(contentArea).remove();
	_(closeButton).remove();
	_(mb).remove();
};

//..... \ /..... \ /..... \
// Events:
closeButton.onselectstart = function() {
	return false;
};
closeButton.onmousedown = function() {
	closeButton.style.borderStyle = "inset";
	return false;
	};
	closeButton.onmouseout = function(){
		closeButton.style.borderStyle = "outset";
	};
	closeButton.onmouseup = function(){
		mb.close();
		setTimeout(mb.onclose,1);
	};
	_(caption).makeDraggable(contentArea, closeButton, mb);
	/* Events (end)
	/\..... ...../ \..... ...../ \..... ...../  */
}

return mb;
} mbStylesAreSet = 0;


function iWindow(url, title, x,y, w,h){

	var htmContent = '<iframe src="'+url+'" width="100%" height="100%" frameborder="0"></iframe>';
	var mb = messageBox(title, htmContent, x, y, w, h);

	//..... \ /..... \ /..... \
	// Styles:
	mb.style.overflow = "visible";
	mb.style.border = "none";
	mb.contentArea.style.borderWidth = "1px";
	/* Styles (end)
	/\..... ...../ \..... ...../ \..... ...../  */

	//..... \ /..... \ /..... \
	// Make embedded site able to control this window:
	var thisExternal = frames[frames.length-1];
	var giveControl = function(){
		try{
			thisExternal.moveTo = function(x,y){
				mb.moveTo(x,y);
			};
			thisExternal.resizeTo = function(w,h){
				mb.resizeTo(w,h);
			};
			thisExternal.close = function(w,h){
				mb.close();
			};
			thisExternal.onunload = function(){
				setTimeout(giveControl,1);
			};
		} catch(e) {

		}
	};
	setTimeout(giveControl,333);
	/* Make embedded site able to control this window (end)
	/\..... ...../ \..... ...../ \..... ...../  */


	return mb;
```

```
	}


	function letConfirm(text, yesFnc, noFnc){
		var mb = messageBox(	"Bitte bestätigen",
					"<center><br>"+text+
					'<br><br><form class="arrowKeyFeature"><button id="confirmBox_YesButton">Ja</button>  <button id="confirmBox_NoButton">Nein</button></form></center><br>');
		mb.style.height = "auto";
		mb.contentArea.style.height = "auto";
		id("confirmBox_YesButton").style.width = 50+"px";
		id("confirmBox_NoButton").style.width = 50+"px";
		setTimeout(function(){ id("confirmBox_NoButton").focus(); }, 1);
		id("confirmBox_YesButton").onclick = function(){
			yesFnc();
			mb.close();
		};
		id("confirmBox_NoButton").onclick = function(){
			if((typeof noFnc)=="function")
				noFnc();
			mb.close();
		};
		return mb;
	}
```

```
/*
\_____________/\_____________/\_____________/\_____________/\_____________/*/


//_____________\/_____________\/_____________\/_____________\/_____________\
//    ==MESSAGE-FEATURE==


	var currentAlertBox = null;
	var currentContent = null;

	function message(content, sec){
		if(currentContent == content){
			return currentAlertBox;
		} else {
			var box = add("div"); currentAlertBox = box; currentContent = content;
			box.innerHTML = content;
			box.isOpen = 1;
			box.style.cssText = "position:absolute;";
			box.style.cssText += "; left:"+cX(box.offsetWidth)+"px; top:"+cY(box.offsetHeight)+"px; background-color:#DDDDDD; color:black; border:double 4px black; z-index:30000;";
			box.style.cssText += "; font-size:12px; font-family:Arial; font-weight:bold; padding:12px 6px 12px 6px; text-align:center;";
			var closeButton = _(box).add("div","x");
			box.closeButton = closeButton;
			closeButton.style.cssText = "position:absolute; right:0px; top:0px; width:10px; height:12px; font-size:12px; color:white; background-color:gray; font-weight:bold; line-height:11px; padding-left:1px; cursor:default;"
			closeButton.onmouseover = function(){
				closeButton.style.backgroundColor = "#990000";
			};
			closeButton.onmouseout = function(){
				closeButton.style.backgroundColor = "gray";
			};
			closeButton.onmousedown = function(){
				closeButton.style.backgroundColor = "red";
			};
			closeButton.onmouseup = function(){
				if((typeof box.onclose)=="function")
					box.onclose();
				_(box).remove();
				box.isOpen = 0;
				currentAlertBox = null;
				currentContent = null;
			};
			var closingProcess = function(){
				if(_(box).isIn(document)){			// Falls die Box noch drin ist und noch nicht durch Schließknopf-Klick removed worden ist,
					if((typeof box.onclose)=="function")// feure evtl. onclose-Ereignis ab etc.
						box.onclose();
					closeButton.onmouseup = function(){
						box.style.display = "none";
					};
					_(box).hide(1);
					setTimeout(function(){
						_(box).remove();
					}, 10000);
					currentAlertBox = null;
					currentContent = null;
					box.isOpen = 0;
				}
```

```
            };

            if(!undef(sec))
                setTimeout(closingProcess,sec*1000);

            box.close = closeButton.onmouseup;

            return box;
        }
    }

    function quickMessage(content, sec){
        if(undef(sec))
            var sec = 0.5;
        var box = message(content);
        box.style.cssText += "; font-size:12px; padding:12px 6px 12px 6px; color:black; border:solid 2px gray;";
        _(box.closeButton).remove();
        setTimeout(function(){
            _(box).hide(1);
        },sec*1000);
        setTimeout(function(){
            _(box).close();
        },sec*1000*2);
        return box;
    }

        /*
    \_____________/\_____________/\_____________/\_____________/\_____________/*/



//_____________\/_____________\/_____________\/_____________\/_____________\
//    ==CONTEXT-MENU-FEATURE==

        function ContextMenu(){


            function Option(label){
                this.scrollIntoView = function(){
                    if(isIE){
                        var prevScrollX = tg().getScrollX();
                        var prevScrollY = tg().getScrollY();
                    }
                    this.html.scrollIntoView();
                    if(isIE && (prevScrollX != tg().getScrollX() || prevScrollY != tg().getScrollY()))
                        window.scrollTo(prevScrollX, prevScrollY); // ... da sich IE-Fenster in manchen Fällen hier "verscrollt".
                };
                this.bindToClipboard = function(text){
                    this.html.style.position = "relative";
                    var clpb = _(this.html).add("div");
                    clpb.style.cssText = "position:absolute; left:2px; top:3px; width:90%; height:80%; z-index:1000;font-size:80%;";
                    if(isIE) clpb.style.cssText+= "; background-color:green; filter:alpha(opacity=0);";
                    clpb.onmousedown = this.html.onmousedown;
                    _(clpb).bindToClipboard(text);
                };
                this.separate = function(color1, color2){
                    if(undef(color1))
                        var color1 = "gray";
                    if(undef(color2))
                        var color2 = "#DDDDDD";
                    this.html.innerHTML = '<div class="separator" style="font-size:1px; border-top:solid 1px '+color1+'; border-bottom:solid 1px '+color2+'; border-right:none;
border-left:none;"></div>' + this.html.innerHTML;
                    this.isSeparated = 1;
                };
                this.separateUp = function(color1, color2){
                    if(undef(color1))
                        var color1 = "gray";
                    if(undef(color2))
                        var color2 = "#DDDDDD";
                    this.html.innerHTML = this.html.innerHTML+'<div class="separator" style="font-size:1px; border-top:solid 1px '+color1+'; border-bottom:solid 1px '+color2+';
border-right:none; border-left:none;"></div>';
                    this.isSeparatedUp = 1;
                };
                this.makeTickable = function(){
                    this.isTickable = 1;
                    if(isIE)
                        this.html.innerHTML = '<span class="tickBox" style="color:black; visibility:hidden; font-family:Wingdings">ü</span>' + this.html.innerHTML;
                    else
                        this.html.innerHTML = '<span class="tickBox" style="color:black; visibility:hidden;"> • </span>' + this.html.innerHTML;
                };
                this.tick = function(){
                    this.isTicked = 1;
                    _(this.html).cl("tickBox").style.visibility = "visible";
```

```
	}
	this.untick = function(){
		this.isTicked = 0;
		_(this.html).cl("tickBox").style.visibility = "hidden";
	}
	this.unticks = function(){
		this.complementLabels = [];
		for(var i = 0; i < this.unticks.arguments.length; i++){
			this.complementLabels[i] = this.unticks.arguments[i];
		}
	};
	this.getIndex = function(){
		for(var i = 0; i < this.parentMenu.options.length; i++){
			if(this.parentMenu.options[i]==this)
				return i;
		}
		return null;
	};
	this.addSubmenu = function(){
		this.submenu = new ContextMenu();
		this.submenu.parentOption = this;
		this.submenu.rootMenu = this.parentMenu.rootMenu;
		this.submenu.html.rootMenuID = this.submenu.rootMenu.id;
		if(this.parentMenu.submenus==null)
			this.parentMenu.submenus = [];
		this.parentMenu.submenus.push(this.submenu);
		if(this.parentMenu.rootMenu.allSubmenus==null)
			this.parentMenu.rootMenu.allSubmenus = [];
		this.parentMenu.rootMenu.allSubmenus.push(this.submenu);
		return this.submenu;
	};
	this.openSubmenu = function(){
		this.parentMenu.getRootMenu().display(); // Angezeigtheit des kompletten Menüs wird abgesichert, falls User kurz zuvor auf Enter gedrückt oder auf einen Menüpunkt geklickt hat
		if(this.parentMenu.parentOption)
			this.parentMenu.parentOption.openSubmenu(); // So können auch tiefere Submenüs direkt mit dem gesamten Menüpfad geöffnet werden
		this.activate();

		if(!_(this.parentMenu.rootMenu.dir).isOneOf(-1, "left"))
			this.submenu.display(getX(this.parentMenu.activeOption.html)+this.parentMenu.activeOption.html.offsetWidth, getY(this.parentMenu.activeOption.html));
		else
			this.submenu.displayRightAligned(getX(this.parentMenu.activeOption.html), getY(this.parentMenu.activeOption.html));

		//this.submenu.display(getX(this.parentMenu.activeOption.html)+this.parentMenu.activeOption.html.offsetWidth,getY(this.parentMenu.activeOption.html));
		this.deactivate();
	};
	this.deactivate = function(message){
/*		if(this.label=="Fremdwörter")
			ooo(this.deactivate.caller);  */
		if(!this.isDisabled)
			this.html.className = this.defaultClassName;
		this.parentMenu.activeOption = null;
		if(message!="notGlobally")
			this.parentMenu.rootMenu.activeOptionGlobal = null;
		this.isActive = 0;
		onactionHandler(this, "deactivate");
	};
	this.activate = function(){
		if(!this.isDisabled)
			this.html.className = this.activatedClassName;
		if(this.parentMenu.activeOption && this.parentMenu.activeOption!=this)
			this.parentMenu.activeOption.deactivate();
		this.parentMenu.activeOption = this;
		this.parentMenu.rootMenu.activeOptionGlobal = this; // Hier muss noch eine Regelung zur Deaktivierung gefunden werden.
		this.isActive = 1;
		if(!this.isDisabled)
			onactionHandler(this, "activate");
	};
	this.disable = function(){
		this.isDisabled = 1;
	};
	this.label			= label;
	this.id				= null;
	this.parentMenu			= null;
	this.precOption		= null;
	this.nextOption			= null;
	this.isActive			= null;
	this.isSeparated			= null;
	this.isSeparatedUp		= null;
	this.isTickable		= null;
	this.isTicked			= null;
	this.complementLabels		= null;
	this.submenu			= null;
	this.html			= null;
	this.activatedHtml		= null;
```

```
        this.isDisabled             = 0;
        this.defaultClassName           = "contextMenuOption";
        this.activatedClassName     = "activatedOption";
};



function onactionHandler(obj, type){
        var e = { }; e.src = obj; e.type = type;
        var obj = obj.parentMenu?obj.parentMenu:obj;

        if((typeof obj.rootMenu.onaction)=="function")
                obj.rootMenu.onaction(e);
}

var enableCharControls = function(menu){
        Event.add("document.onkeydown", function(){
                var pressedChar = getKeyName(usedKey()).toLowerCase();
                if(menu.activeOption && menu.activeOption!=menu.options[menu.options.length-1]){       // Wenn eine Option aktiv ist, ...
                        var begin = menu.options.getIndex(menu.activeOption)+1;                         // beginne mit der Suche ab dieser Option.
                } else {                                                                                // Wenn nicht, oder wenn nur die allerletzte Option aktiv ist,
                        var begin = 0;                                                                  // beginne ganz oben.
                }
                if(pressedChar.length==1){
                        for(var i=begin; i<menu.options.length; i++){                                   // Suche ab eben festgelegten Beginn
                                if(menu.options[i].label.toLowerCase().indexOf(pressedChar)==0 && !menu.options[i].isDisabled){
                                                                                                        // nach Übereinstimmung des Optionsanfgsbchstb. & Tastendruck,
                                        menu.options[i].activate();                                     // aktiviere die Option in dem Fall und beende die Suche.
                                        if(!_(menu.html).shows(menu.options[i].html))
                                                menu.options[i].scrollIntoView();
                                        break;
                                }
                                if(i==menu.options.length-1){                                           // Falls Suche erfolglos an Menü-Ende angekommen,
                                        for(var j=0; j<begin; j++){                                     // suche von ganz oben bis zur anfangs festgelegten Startoption
                                                if(menu.options[j].label.toLowerCase().indexOf(pressedChar)==0 && !menu.options[j].isDisabled){
                                                                                                        // nach Übereinstimmung des Optionsanfgsbchstb. & Tastendruck,
                                                        menu.options[j].activate();                     // aktiviere die Option in dem Fall und beende die Suche.
                                                        if(!_(menu.html).shows(menu.options[j].html))
                                                                menu.options[j].scrollIntoView();
                                                        break;
                                                }
                                        }
                                }
                        }
                }
        }, menu.id+"_CharContr");
};

this.enableCharControls = function(fullActivationRequested){
        if(fullActivationRequested){
                this.charControlsDesired = 1;
                enableCharControls(this);
                function enableChars(menu){
                        for(var i = 0; i < menu.options.length; i++){
                                if(menu.options[i].submenu){
                                        menu.options[i].submenu.charControlsDesired = 1;
                                        enableCharControls(menu.options[i].submenu);
                                        enableChars(menu.options[i].submenu);
                                }
                        }
                }
                enableChars(this);
        } else {
                this.charControlsDesired = 1;
                enableCharControls(this);
        }
};


var disableCharControls = function(menu){
        if(Event.exists("document.onkeydown", menu.id+"_CharContr"))
                Event.drop("document.onkeydown", menu.id+"_CharContr");
};


this.disableCharControls = function(fullDeactivationRequested){
        if(fullDeactivationRequested){
                this.charControlsDesired = 0;
                disableCharControls(this);
                function disableChars(menu){
                        for(var i = 0; i < menu.options.length; i++){
                                if(menu.options[i].submenu){
                                        menu.options[i].submenu.charControlsDesired = 0;
                                        disableCharControls(menu.options[i].submenu);
```

```
                                disableChars(menu.options[i].submenu);
                            }
                        }
                    }
                    disableChars(this);
                } else {
                    this.charControlsDesired = 0;
                    disableCharControls(this);
                }
            };

            var setKeyEvents = function(menu){

                var reverseOpening = _(menu.rootMenu.dir).isOneOf(-1, "left");

                var selectOption = function(menu){
                    if(menu.keyControlsDesired){
                        var activeOption = menu.activeOption;
                        menu.getRootMenu().hide("noDeactivate");          // Schließt Wurzelmenü, also das ganze Menü.
                        if(activeOption && activeOption.onselect){
                            onactionHandler(activeOption, "select");
                            activeOption.onselect();
                            activeOption.deactivate();
                        }
                    }
                }; Event.set("document.onenter", function(){  selectOption(menu); }, menu.id);

                var leaveAll = function(menu){
                    if(menu.keyControlsDesired){
                        menu.getRootMenu().hide();          // Schließt Wurzelmenü, also das ganze Menü
                    }
                }; Event.set("document.onescape", function(){ leaveAll(menu) }, menu.id);

                var moveup = function(menu){
                    if(menu.keyControlsDesired && menu.activeOption && menu.isOpen){
                        if(menu.activeOption.precOption.isDisabled){          // Falls die Option als unaktivierbar festgelegt worden ist,
                            menu.activeOption.precOption.activate();  // dann überspringe sie.(d.h. erst recht aktivieren, damit
                            moveup(menu);                             // durch einen neuen Methodenaufruf die nächste aktiviert werden kann)
                            return;
                        }
                        menu.activeOption.precOption.activate();
                        // Bei Überhöhe & Scrollbar:
                        if(!_(menu.html).shows(menu.activeOption.html))
                            menu.activeOption.scrollIntoView();
                        menu.rootMenu.automaticHidingAllowed = 0;
                    }
                }; Event.set("document.onuparrow", function(){ menu.rootMenu.keyPressed = 1; moveup(menu); },"ContextMenuOnuparrow");

                var movedown = function(menu){
                    if(menu.keyControlsDesired && menu.activeOption && menu.isOpen){       // Falls die Option als unaktivierbar festgelegt worden ist,
                        if(menu.activeOption.nextOption.isDisabled){                       // dann überspringe sie.(d.h. erst recht aktivieren, damit
                            menu.activeOption.nextOption.activate();                       // durch einen neuen Methodenaufruf die nächste aktiviert werden kann)
                            movedown(menu);
                            return;
                        }
                        menu.activeOption.nextOption.activate();
                        // Bei Überhöhe & Scrollbar:
                        if(!_(menu.html).shows(menu.activeOption.html))
                            _(menu.activeOption.html).scrollIntoBottomView();

                        menu.rootMenu.automaticHidingAllowed = 0;
                    }
                }; Event.set("document.ondownarrow", function(){ menu.rootMenu.keyPressed = 1; movedown(menu); }, "ContextMenuOndownarrow");

                var moveright = function(menu){
                    if(menu.keyControlsDesired){
                        if(menu.activeOption && menu.activeOption.submenu && menu.isOpen){
                            // Öffne Submenü an den Koordinaten seiner Mutteroption:
                            var x = getX(menu.activeOption.html);
                            var y = getY(menu.activeOption.html);
                            if(!reverseOpening)
                                menu.activeOption.submenu.display(x+menu.activeOption.html.offsetWidth, y);
                            if(reverseOpening){
                                menu.activeOption.submenu.displayRightAligned(x, y);
                            }
                            // Deaktiviere die im Muttermenü evtl. noch aktive Option:
                            if(menu.activeOption)
                                menu.activeOption.deactivate("notGlobally");
                            disableCharControls(menu);
                        }
                        menu.rootMenu.automaticHidingAllowed = 0;
                    }
                }; Event.set("document.onrightarrow", (reverseOpening)? function(){ moveleft(menu); } : function(){ moveright(menu); }, "ContextMenuOnrightarrow");
```

```
                var moveleft = function(menu){
                    if(menu.keyControlsDesired && menu.isOpen){
                        if(menu.parentOption){
                            if(menu.activeOption)
                                menu.activeOption.deactivate();
                            menu.parentOption.activate();
                            if(menu.parentOption.parentMenu.charControlsDesired)
                                enableCharControls(menu.parentOption.parentMenu);
                            menu.hide();
                            // Da mit Links-Taste Menü geschlossen wird:

                            Event.set("document.onenter", function(){ selectOption(menu.parentOption.parentMenu) }, "ContextMenuOnenter");
                            Event.set("document.onuparrow", function(){ moveup(menu.parentOption.parentMenu) }, "ContextMenuOnuparrow");
                            Event.set("document.ondownarrow", function(){ movedown(menu.parentOption.parentMenu) }, "ContextMenuOndownarrow");
                            Event.set("document.onleftarrow", (reverseOpening)? function(){ moveright(menu.parentOption.parentMenu) } : function(){ moveleft(menu.
parentOption.parentMenu) }, "ContextMenuOnleftarrow");
                            Event.set("document.onrightarrow", (reverseOpening)? function(){ moveleft(menu.parentOption.parentMenu) } : function(){ moveright(menu.
parentOption.parentMenu) }, "ContextMenuOnrightarrow");
                        }
                        menu.rootMenu.automaticHidingAllowed = 0;
                    }
                }; Event.set("document.onleftarrow", (reverseOpening)? function(){ moveright(menu); } : function(){ moveleft(menu); }, "ContextMenuOnleftarrow");

                Event.add("document.onkeyup",function(){
                    menu.rootMenu.keyPressed = 0;
                },"contextMenuOnKeyUp");

            };


            this.enableKeyControls = function(){
                this.keyControlsDesired = 1;
                setKeyEvents(this);
            };

            var disableKeyControls = function(menu){
                setTimeout(function(){
                    if(Event.exists("document.onenter", menu.id))
                        Event.drop("document.onenter", menu.id);
                    if(Event.exists("document.onescape", menu.id))
                        Event.drop("document.onescape", menu.id);
                },111);
            };

            var disableCharControls = function(menu){
                if(Event.exists("document.onkeydown", menu.id+"_CharContr"))
                    Event.drop("document.onkeydown", menu.id+"_CharContr");
            };


            this.disableKeyControls = function(fullDeactivationRequested){
                this.keyControlsDesired = 0;
                disableKeyControls(this);
                if(fullDeactivationRequested){
                    function disableKeys(menu){
                        for(var i = 0; i < menu.options.length; i++){
                            if(menu.options[i].submenu){
                                menu.options[i].submenu.keyControlsDesired = 0;
                                disableKeyControls(menu.options[i].submenu);
                                disableKeys(menu.options[i].submenu);
                            }
                        }
                    }
                    disableKeys(this);
                }
            };

            var activateMouseControls = function(thisOption, thisMenu){

                var hoverReaction = function(thisOption, thisMenu){
                    thisMenu.rootMenu.mouseIsIn = 1;
                    if(!thisMenu.rootMenu.keyPressed && !thisOption.isDisabled){
                        // Falls eine andere Option noch aktiv ist und ein Submenü hat, schließe letzteres zuerst:
                        if(thisOption.parentMenu.openedSubmenu)
                            thisOption.parentMenu.openedSubmenu.hide();
                        // Jetzt aktiviere die angesteuerte Option:
                        thisOption.activate();
                        // Bei Per-Maus-Rückkehr in ein Elternmenü soll dieses wieder per Tastatur steuerbar sein:
                        if(thisOption!=thisMenu.options[0] && thisMenu.keyControlsDesired)
                            setKeyEvents(thisMenu);
                        // Falls die Option ein Submenü besitzt, öffne es:
                        if(thisOption.submenu){
                            // Öffne Submenü an den Koordinaten seiner Mutteroption:
                            setTimeout(function(){
```

```
                        if(thisOption.isActive){
                            var x = getX(thisOption.html);
                            var y = getY(thisOption.html);
                            if(!_(thisMenu.rootMenu.dir).isOneOf(-1, "left"))
                                thisOption.submenu.display(x+thisOption.html.offsetWidth, y);
                            else
                                thisOption.submenu.displayRightAligned(x, y);
                            thisOption.submenu.activeOption.deactivate();
                        }
                    },thisOption.submenu.delay);
                }
            }
        }; thisOption.html.onmouseover = function(){ hoverReaction(thisOption, thisMenu); };

        var clickReaction = function(thisOption){
            // thisOption.html.onmouseover = "";                    // Sonst lässt der IE manche Submenüs stehen.
            if(!thisOption.isDisabled){
                thisOption.parentMenu.getRootMenu().hide();     // Schließt Wurzelmenü, also das ganze Menü.
                if(thisOption.onselect){
                    onactionHandler(thisOption, "select");
                    thisOption.onselect();
                }
            }
        };
        thisOption.html.onmousedown = function(){
            if(!thisOption.isTickable){
                clickReaction(thisOption);
            } else {
                if(!thisOption.isTicked){
                    thisOption.tick();
                    if(thisOption.complementLabels!=null){
                        for(var i = 0; i < thisOption.complementLabels.length; i++){
                            thisMenu.options[thisOption.complementLabels[i]].untick();
                        }
                    }
                } else {
                    thisOption.untick();
                }
            }
        };
        thisOption.html.onmouseup = function(){ if(thisOption.isTickable) clickReaction(thisOption); };
    };

    this.getRootMenu = function(){
        if(!this.parentOption)
            return this;
        else
            return this.parentOption.parentMenu.getRootMenu();
    };

    this.getActiveOption = function(){                  // Gibt derzeit aktive Option zurück.
        var activeOption;
        for(var i = 0; i < this.options.length; i++){
            if(this.options[i].submenu){
                activeOption = this.options[i].submenu.getActiveOption();
                if(activeOption)
                    break;
            }
            if(this.options[i].isActive){
                activeOption = this.options[i];
                break;
            }
        }
        return activeOption;
    };

    this.listOptions = function(){          // Gibt Liste aller Optionen auch aus den Submenüs zurück
        function listOptions(menu){
            var str = "";
            for(var i = 0; i < menu.options.length; i++){
                str += (" " + menu.options[i].label);
                if(menu.options[i].submenu)
                    str += listOptions(menu.options[i].submenu);
            }
            return str;
        }
        return listOptions(this);
    };

    this.clone = function(){
        var clonedMenu = new ContextMenu();
        //clonedMenu.html.style.cssText = this.html.style.cssText;
        clonedMenu.activatedHtml.style.cssText = this.activatedHtml.style.cssText;
        var cloneMenu = function(originalMenu, newMenu){
```

```
        var activeOption = null;
        for(var i = 0; i < originalMenu.options.length; i++){
            newMenu.addOptions(originalMenu.options[i].label);
            if(originalMenu.options[i].isActive){
                activeOption = originalMenu.options[i];
                originalMenu.options[i].deactivate();
            }
            newMenu.options[i].html.className = originalMenu.options[i].html.className;
            newMenu.options[i].defaultClassName = originalMenu.options[i].defaultClassName;
            newMenu.options[i].activatedClassName = originalMenu.options[i].activatedClassName;
            //newMenu.options[i].html.style.cssText = originalMenu.options[i].html.style.cssText;
            if(activeOption){
                originalMenu.options[i].activate();
                activeOption = null;
            }
            newMenu.options[i].onselect = originalMenu.options[i].onselect;
            if(originalMenu.options[i].submenu){
                newMenu.options[i].addSubmenu();
                newMenu.options[i].submenu = cloneMenu(originalMenu.options[i].submenu, newMenu.options[i].submenu);
                //newMenu.options[i].submenu.html.style.cssText = originalMenu.options[i].submenu.html.style.cssText;
            }
            if(originalMenu.options[i].isSeparated)
                newMenu.options[i].separate();
            if(originalMenu.options[i].isSeparatedUp)
                newMenu.options[i].separateUp();
        }
        newMenu.html.className = originalMenu.html.className;
        return newMenu;
    }
    return cloneMenu(this, clonedMenu);
};


this.remove = function(){
    var removeAllSubmenus = function(menu){
        for(var i = 0; i < menu.options.length; i++){
            if(menu.options[i].submenu){
                removeAllSubmenus(menu.options[i].submenu);
            }
        }
        menu.options = null;
        menu.rootMenu = null;
        _(menu.html).remove();
        _(menu.activatedHtml).remove();
        _(menu.stylebufferElement).remove();
        _(menu.corona).remove();
        if(Event.exists("document.onkeydown", menu.id+"_CharContr"))
            Event.drop("document.onkeydown", menu.id+"_CharContr");
        if(Event.exists("document.onenter", menu.id))
            Event.drop("document.onenter", menu.id);
        if(Event.exists("document.onescape", menu.id))
            Event.drop("document.onescape", menu.id);
        if(Event.exists("window.onmousedown", menu.id+"_onOutsideMDown"))
            Event.drop("window.onmousedown", menu.id+"_onOutsideMDown");
        if(Event.exists("document.getElementsByTagName('body')[0].onmousedown", menu.id+"_onOutsideMDown"))
            Event.drop("document.getElementsByTagName('body')[0].onmousedown", menu.id+"_onOutsideMDown");
        if(Event.exists("window.onmouseup", menu.id+"_onOutsideMUp"))
            Event.drop("window.onmouseup", menu.id+"_onOutsideMUp");
        if(Event.exists("document.getElementsByTagName('body')[0].onmouseup", menu.id+"_onOutsideMUp"))
            Event.drop("document.getElementsByTagName('body')[0].onmouseup", menu.id+"_onOutsideMUp");

    };
    removeAllSubmenus(this);
    this.setByRef(0);
};


this.addShadow = function(opc, ab, cd){
    var addShadow = function(menu, opc, ab, cd){
        var i=0; loop(function(){
            if(menu.options && menu.options[i].submenu)
                addShadow(menu.options[i].submenu, opc, ab, cd);
        i++; }, 1, menu.options.length);
        _(menu.html).addShadow(opc, ab, cd);
    }
    addShadow(this, opc, ab, cd);
};

this.displayRightAligned = function(x, y){
    with(this.html.style){
        this.display(x,y);
        left = x - this.html.offsetWidth;
    }
}
```

```
var setOptionInDOM = function(opt){
	if(!_(opt.html).isIn(document)){
		opt.html.appendChild(document.createTextNode(opt.label));
		opt.parentMenu.html.appendChild(opt.html);
		opt.activatedHtml = document.createElement("div"); // Nur ein "Aufnehmer"-Element für Stildefinitionen, die später von diesem "Aufnehmer" gelesen werden können
		//opt.html.className = "contextMenuOption";
		// Hack, um Mouseover im IE auch an Options-Stellen wirken zu lassen, an denen kein Text ist:
		if(isIE){
			var sp = document.createElement("span");
			opt.html.appendChild(sp);
			sp.style.width = "0px";
		}
	}
};

this.display = function(x,y, positionVal){

	if(!undef(x)) x+="px";
	if(!undef(y)) y+="px";

	var htm = this.html;
	var activeHtm = this.activatedHtml;

	// Falls irgendwelche Options erst mit dem Display ins Dokument eingestezt werden sollen
	for(var i = 0; i < this.options.length; i++){
		setOptionInDOM(this.options[i]);
	}


// Grafische Anzeige:
 with(htm.style){
	position = "absolute";
	if(!undef(x)){
		if(!this.parentOption && this.rightAxis){
			right = x;
		} else {
			left = x;
		}
	}
	if(!undef(y)){
		if(!this.parentOption && this.bottomAxis){
			bottom = y; top = "auto";
		} else {
			top = y; bottom = "auto";
		}
	}
	if(_(_(htm).getStyle("z-index")).isOneOf(0, "auto"))
		zIndex = "999";
	display = "block"; cursor = "pointer";
	if(positionVal=="fixed" || (this.parentOption && this.parentOption.parentMenu.html.style.position=="fixed"))
		htm.style.position = "fixed";
	/* if(htm.shadow)
		htm.shadow.style.zIndex = ; */
}

// Übernehme Grafikstile des evtl. vorhandenen Elternmenüs außer speziell für dieses Submenü definierte Stile:
if(this.parentOption){
	var pm = this.parentOption.parentMenu;

	// Stile des Menüs allgemein
	var pmStyle = pm.html.style;
	with(htm.style){
		/* if(color == "")			color = pmStyle.color;
		if(fontFamily == "")		fontFamily = pmStyle.fontFamily;
		if(fontSize == "")		fontSize = pmStyle.fontSize;
		if(fontWeight == "")		fontWeight = pmStyle.fontWeight;
		if(backgroundColor == "")	backgroundColor = pmStyle.backgroundColor;
		if(borderStyle == "")		borderStyle = pmStyle.borderStyle;
		if(borderWidth == "")		borderWidth = pmStyle.borderWidth;
		if(borderColor == "")		borderColor = pmStyle.borderColor;
		if(cursor == "")			cursor = pmStyle.cursor; */
		if(!this.htmlPaddingStyle) this.htmlPaddingStyle = pm.htmlPaddingStyle;

		// Falls das Elternmenü einen synthetisierten Hintergrund hat, übernehme diesen und seine Eigenschaften:
		if(pm.html.hasSyntheticBg){
			setTimeout(function(){
				if(!htm.hasSyntheticBg){
					synthesizeBg(htm);
				}
				with(htm.bg.style){
					MozOpacity = pm.html.bg.style.MozOpacity;
					opacity = pm.html.bg.style.opacity;
```

```
                        filter = pm.html.bg.style.filter;
                    }
                },1);
        }
}

// Stil aktivierter Optionen:
var pmActiveStyle = pm.activatedHtml.style;
with(activeHtm.style){
        if(color == "")             color = pmActiveStyle.color;
        if(fontFamily == "")        fontFamily = pmActiveStyle.fontFamily;
        if(fontSize == "")          fontSize = pmActiveStyle.fontSize;
        if(fontWeight == "")        fontWeight = pmActiveStyle.fontWeight;
        if(backgroundColor == "") backgroundColor = pmActiveStyle.backgroundColor;
        if(borderStyle == "")       borderStyle = pmActiveStyle.borderStyle;
        if(borderWidth == "")           borderWidth = pmActiveStyle.borderWidth;
        if(borderColor == "")       borderColor = pmActiveStyle.borderColor;
}

// Klassennamen und IDs:
var className1 = this.rootMenu.html.className;
var className2 = "sub_"+this.rootMenu.html.className;
if(htm.className.indexOf("submenu_")==-1){
        if(this.parentOption.parentMenu == this.rootMenu)
                var className3 = "submenu_"+this.parentOption.id;
        else
                var className3 = this.parentOption.parentMenu.html.className + "_" + this.parentOption.id;
        } else {
                var className3 = "submenu_"+this.parentOption.id;
        }

        htm.className = className1 + " " + className2 + " " +className3;
}


// Padding soll anders als die anderen Stile über eine Extra-Eigenschaft abgefragt werden
// (damit Optionen-Aktivmarkierung ganze Breite einnehmen kann):
if(this.htmlPaddingStyle){
        for(var i=0; i< this.options.length; i++){
                this.options[i].html.style.padding = this.htmlPaddingStyle;
        }
}


// Falls für das gesamte Menü kein Aktiviert-Stil definiert worden ist:
/* if (activeHtm.style.fontWeight == "")
        activeHtm.style.fontWeight = "bold";
if(activeHtm.style.backgroundColor == "")
        activeHtm.style.backgroundColor = "yellow"; */


// Überhöhenbehandlung:
if(htm.offsetHeight>getViewportHeight()-30){
        if(this.managesOverheight){
                with(htm.style){
                        if(isIE && (width=="" || width=="auto"))
                                width = (htm.offsetWidth + 20) + "px";
                        left = x; top = "0px"; height = "0px"; overflowY = "scroll";
                        setTimeout(function(){ top="0"; top="0px"; var savedVal = width; width = "10000px"; width=savedVal;  height = (getViewportHeight()-20)+"px"; },80);
                }
                htm.scrollTop = 0;
        } else {
                htm.style.height = "auto";
        }
}
if(htm.offsetHeight>getViewportHeight()-65){
        htm.style.top = "0px";
}
// Durch unteren Bildschirmrand abgeschnittene Submenüs vermeiden:
if(this.parentOption && htm.offsetHeight+htm.offsetTop>getViewportHeight()){
        var t = htm.offsetTop;
        t -= htm.offsetHeight + htm.offsetTop - getViewportHeight();
        htm.style.top = (t-20)+"px";
}
// Durch rechten Bildschirmrand abgeschnittene Submenüs vermeiden:
if(this.parentOption && htm.offsetWidth+htm.offsetLeft>getViewportWidth()){
        htm.style.left = (pm.html.offsetLeft - this.html.offsetWidth) + "px";
}
// Objekttechnik (unabh. von Grafikausgabe):
this.isOpen = 1;
if(this.parentOption)
        this.parentOption.parentMenu.openedSubmenu = this;
for(var i = 0; i < this.options.length; i++){       // Aktiviere erste nicht-disabled-Option
        if(!this.options[i].isDisabled){
```

```
                this.options[i].activate();
                break;
        }
    }
    if(this.keyControlsDesired)
        setKeyEvents(this);
    if(this.charControlsDesired)
        enableCharControls(this);
    // Da IE Submenüs manchmal stehen lässt bzw. unerwartet wieder anzeigt:
    if(this.parentOption){
        if(!this.parentOption.parentMenu.isOpen)
            this.hide();
    }

    // Für onmouseleftall, damit es auch über IFrames u.ä. funktioniert:
    var thisCorona = this.corona;
    setTimeout(function(){
        with(thisCorona.style){
            left = (htm.offsetLeft-30)+"px";        top = (htm.offsetTop-30)+"px";          // hier vorher 10-er-Werte (30er jetzt zuviel?)
            width = (htm.offsetWidth + 60)+"px";    height = (htm.offsetHeight + 60)+"px";  // hier vorher 20-er-Werte (60er jetzt zuviel?)
            display = "block";
            var this_zIndex = _(htm).getStyle("z-index");
            if(_(this_zIndex).isOneOf(0, "auto")){
                this_zIndex = 999;
                htm.style.zIndex = this_zIndex;
            }
            zIndex = this_zIndex - 1;
            if(isIE){
                backgroundColor = "black";
                filter = "alpha(opacity=0)";
            }
            if(Mouse.isAbove(thisCorona))
                thisCorona.style.display = "none";
        }
    },111);
    // Breite fixieren:
    if(this.isOpen && htm.style.width=="" && htm.style.overflowY != "scroll"){
        var blw = numIn(_(htm).getStyle("borderLeftWidth"));
        var brw = numIn(_(htm).getStyle("borderRightWidth"))
        if(isNaN(blw)) blw = 0;
        if(isNaN(brw)) brw = 0;
        htm.style.width=(htm.offsetWidth-blw-brw+5)+"px";
        if(htm.bg)
            htm.bg.style.width=(htm.offsetWidth)+"px";
    }
    /* if(this.isOpen && htm.style.width=="" && htm.style.overflowY != "scroll"){
        htm.style.width=htm.offsetWidth-numIn(_(htm).getStyle("borderLeftWidth"))-numIn(_(htm).getStyle("borderRightWidth"))+5;
        if(htm.bg)
            htm.bg.style.width=htm.offsetWidth;
    } */

    // Events:
    if((typeof this.ondisplay)=="function")
        this.ondisplay();
    onactionHandler(this, "display");
};


this.displayPartially = function(x,y){
    if(this.parentOption)
        this.parentOption.parentMenu.isOpen = 1;
    if(!undef(y))
        this.display(x,y);
    else
        this.display();
}



this.hide = function(message){
    // this.html.style.overflow = "visible";
    // Schließe dieses (evtl. Sub-)Menü:

    if(this.activeOption && message!="noDeactivate")
        this.activeOption.deactivate();
    if(this.isClosable){
        this.isOpen = 0;
        if(this.parentOption)
            this.parentOption.parentMenu.openedSubmenu = null;
        this.html.style.display = "none";
        this.corona.style.display = "none";
        disableKeyControls(this);
         if(Event.exists("document.onkeydown", this.id+"_CharContr"))
            Event.drop("document.onkeydown", this.id+"_CharContr");
```

```
        }
        // Schließe auch alle seine Submenüs:
        if(this.options){
            for(var i = 0; i < this.options.length; i++){
                if(this.options[i].submenu && this.options[i].submenu.isOpen) {
                    this.options[i].submenu.hide(message);
                }
            }
        }
        // Sonstiges:
        /* if(Event.exists("document.onkeyup","contextMenuOnKeyUp")){
            Event.drop("document.onkeyup", "contextMenuOnKeyUp");
        } */

        // Events:
        if((typeof this.onhide)=="function")
            this.onhide();
        onactionHandler(this, "hide");
};

this.softHide = function(){

};


this.getOption = function(str){
        var path = str.split("|");
        var option = this.options[path[0]];
        for(var i = 1; i<path.length; i++){
            option = option.submenu.options[path[i]];
        }
        return option;
};

this.getOptionByArray = function(path, m){      // m damit für die Geschwindigkeit in addOption() auf arr.pop() verzichtet werden kann
        var option = this.options[path[0]];
        for(var i = 1; i<path.length-m; i++){
            option = option.submenu.options[path[i]];
        }
        return option;
};

this.deleteOption = function(sign){
        var index = null;
        with(this){
            // Grafische Löschung:
            _(options[sign].html).remove();
            // Interne Löschung:
            if((typeof sign)=="string"){
                index = options[sign].getIndex();
                options.drop(index);
                delete(options[sign]);
            }
            if(!isNaN(sign)){
                index = sign;
                delete(options[options[sign].label]);
                options.drop(index);
            }
            // Verkettung aktualisieren:
            if(options.length>=1){
                if(index==0){
                    options[index].precOption = options[options.length-1];
                    options[options.length-1].nextOption = options[index];
                } else if(index==options.length) {
                    options[0].precOption = options[options.length-1];
                    options[options.length-1].nextOption = options[0];
                } else {
                    options[index-1].nextOption = options[index];
                    options[index].precOption = options[index-1];
                }
            }
        }
};

this.insertOption = function(label, i){
        with(this){
            options[label] = new Option(label);
            options[label].id = (this.rootMenu.lastOptionId++);
            options[label].parentMenu = this;
            options.insertAt(i, options[label]);
            // Nachbarschaftsregelung (Verkettete Liste; s. Erklärungen in addOptions-Methode):
            if(i==0){
                options[label].precOption = options[options.length-1];
                options[label].nextOption = options[i+1];
```

```
                options[i+1].precOption = options[0];
                options[options.length-1].nextOption = options[0];
            } else {
                options[label].precOption = options[i-1];
                options[label].nextOption = options[i+1];
                options[i-1].nextOption = options[label];
                options[i+1].precOption = options[label];
            }
            // Für die grafische Darstellung:
            options[label].html  = add("div", label);
            //html.insertBefore(options[label].html , options[i+1].html);
            _(options[label].html).insertBefore(options[i+1].html);
            options[label].activatedHtml = document.createElement("div");
            // Maustechnik:
            activateMouseControls(options[label], this);
            // Hack, um Mouseover im IE auch an Options-Stellen wirken zu lassen, an denen kein Text ist:
            if(isIE){
                var sp = document.createElement("span");
                options[label].html.appendChild(sp);
                sp.style.width = "0px";
            }
        }
        return this.options[label];
};

this.addOption = function(label){
        if(label.indexOf("|")>-1){
            var arr = label.split("|");
            var label = arr[arr.length-1];
            var option = this.getOptionByArray(arr,1);
            if(!option.submenu)
                option.addSubmenu();
            if(this.addOption.get("noDOM"))
                option.submenu.addOptions.noDOM = 1;
            return option.submenu.addOptions(label);
        } else {
            if(this.addOption.get("noDOM"))
                this.addOptions.noDOM = 1;
            return this.addOptions(label);
        }

};



this.addOptions = function(){
        var desiredOptionLabels = null;
        if((typeof this.addOptions.arguments[0])=="object")
            desiredOptionLabels = this.addOptions.arguments[0];
        else
            desiredOptionLabels = this.addOptions.arguments;
        for(var i=0; i<desiredOptionLabels.length; i++){
            var label = desiredOptionLabels[i];
            with(this){
                if(!options)
                    options = new Array();
                options[label] = new Option(label);
                options[label].parentMenu = this;
                options[label].id = (this.rootMenu.lastOptionId++);
                if(options.length==0){                          // Falls es in dem Menü noch keine andere Option gibt,
                    options[label].precOption = options[label];     // bestimme diese Option zur Vor-
                    options[label].nextOption = options[label];     // und Nachoption ihrer selbst.
                } else {
                    options[label].precOption = options[options.length-1];//Falls schon mindestens eine Option vorhanden ist, wird die letzte
                    options[options.length-1].nextOption = options[label];//zum Vornachbarn der neuen Option
                    options[0].precOption = options[label];             // und bekommt die neue als Nachoption,
                    options[label].nextOption = options[0];             // während die oberste Option die neue als Voroption bekommt
                }                                                   // und die neue die oberste als Nachoption.
                options.push(options[label]);
                // Für die graphische Darstellung:
                options[label].html = document.createElement("div");
                options[label].html.className = "contextMenuOption";
                if(!this.addOptions.get("noDOM"))
                    setOptionInDOM(options[label]);
                var marker = 1; // Für eval-Bearbeitung (Herstellung von this.addOptions() )
                // Maustechnik:
                activateMouseControls(options[label], this);
            }
        }
        return this.options[this.addOptions.arguments[0]];
};
```

```
this.knows = function(obj){
	if(obj.html && obj.html.rootMenuID == this.rootMenu.id || obj.parentMenu && obj.parentMenu.html && obj.parentMenu.html.rootMenuID == this.rootMenu.id)
		return 1;
	while(obj.parentNode){
		if((obj.rootMenuID && obj.rootMenuID == this.rootMenu.id))
			return 1;
		obj = obj.parentNode;
	}
	return 0;
};

function init(menu){
	menu.id = randomString(5);

	// Erzeuge DIV-Behälter für die Menüpunkte und verstecke ihn zunächst:
	menu.html = add("div","","class","contextMenu ");
	menu.html.style.display = "none";
	menu.html.isContextMenu = 1;
	menu.html.rootMenuID = menu.id;
	// Lege Standard-Voreinstellung für aktivierte Menüpunkte Fest:
	menu.activatedHtml = add("div","");		// Nur ein "Aufnehmer"-Element für Stildefinitionen, die später von diesem "Aufnehmer" gelesen werden
	menu.activatedHtml.style.display = "none";
	menu.stylebufferElement = add("div","");		// Nur ein "Aufnehmer"-Element für Stildefinitionen, die später von diesem "Aufnehmer" gelesen werden
	menu.stylebufferElement.style.display = "none";

	// Mouseover- & Mouseout-Management:

menu.corona = add("div", ""); // Für onmouseleftall, damit es auch über IFrames u.ä. funktioniert:
menu.corona.style.display = "none";
menu.corona.style.position = "absolute";
menu.corona.onmouseover = function(){
	menu.corona.style.display = "none";
	/* setTimeout(function(){
		if(Mouse.getElement()!=menu.corona && menu.isOpen)
			menu.corona.style.display = "block";
	},1000); */
}

setTimeout(function(){
	if(menu.parentOption)
		menu.onmouseused = menu.parentOption.parentMenu.onmouseused;
	menu.html.onmouseover = function(){
		menu.corona.style.display="block";
		menu.rootMenu.mouseIsIn = 1;
		if((typeof menu.onmouseused)=="function"){
			setTimeout(menu.onmouseused,1);
		}
	};
	menu.html.onmouseout = function(){
		setTimeout(function(){
			if(!menu.rootMenu.mouseIsIn){
				if((typeof menu.rootMenu.onmouseleftall)=="function"){
					setTimeout(menu.rootMenu.onmouseleftall,1);
				}
				menu.rootMenu.automaticHidingAllowed = 1;
				setTimeout(function(){
					if(menu.rootMenu.automaticHidingAllowed && !menu.rootMenu.mouseIsIn && menu.rootMenu.isClosable)
						menu.rootMenu.hide();
				},menu.rootMenu.hidingDelay);
			}

		},111);

		menu.rootMenu.mouseIsIn = 0;
	};
},333);

// Immer schließen, wenn außerhalb geklickt:
var outsideClickReaction = function(e, thisMenu){
	if(thisMenu.hidingAllowed && thisMenu.rootMenu && !thisMenu.rootMenu.knows(getClickTarget(e)) && getClickTarget(e).className.indexOf(thisMenu.rootMenu.html
.className+" ")==-1){
		thisMenu.hidingAllowed = 0;
		thisMenu.rootMenu.hide();
	}
}; var thisMenu = this; var origDocumentOnMousedown = document.onmousedown;
setTimeout(function(){
	menu.hidingAllowed = 1;
	if(isIE){
		Event.add("document.getElementsByTagName('body')[0].onmousedown", function(e) { outsideClickReaction(e, menu);  }, menu.id+"_onOutsideMDown");
		Event.add("document.getElementsByTagName('body')[0].onmouseup", function(e) { menu.hidingAllowed = 1;  }, menu.id+"_onOutsideMUp");
	}else{
		Event.add("window.onmousedown", function(e) { outsideClickReaction(e, menu);  }, menu.id+"_onOutsideMDown");
		Event.add("window.onmouseup", function(e) { menu.hidingAllowed = 1;  }, menu.id+"_onOutsideMUp");
	}
```

```
		},333);


		// Standard-Stile:
		setTimeout(function(){
			if(menu == menu.rootMenu)
				CSS.prepend(".contextMenu { border:solid 1px black; background-color:white; } .activatedOption { background-color:gray; }");
		},1);

	}

	this.id				= null;
	this.isOpen			= null;
	this.options			= null;
	this.parentOption		= null;
	this.activeOption			= null;
	this.activeOptionGlobal			= null;
	this.openedSubmenu		= null;
	this.html			= null;
	this.htmlPaddingStyle			= null;
	this.activatedHtml		= null;
	this.stylebufferElement			= null;
	this.corona				= null;
	this.onmouseused		= null;
	this.onmouseleftall		= null;
	this.mouseIsIn			= 0;
	this.keyPressed			= null;
	this.keyControlsDesired			= 1;
	this.charControlsDesired	= 1;
	this.isClosable			= 1;
	this.automaticHidingAllowed		= 1;
	this.managesOverheight		= 1;
	this.rightAxis			= 0;
	this.bottomAxis			= 0;
	this.delay			= 1;
	this.dir			= 1;
	this.hidingDelay			= 1000;
	this.allSubmenus			= null;
	this.submenus			= null;
	this.lastOptionId			= -1;
	this.rootMenu			= this;
	init(this);

}


ContextMenu.knows = function(obj){
	if(obj instanceof ContextMenu || (obj.parentMenu || obj.parentMenu instanceof ContextMenu))
		return 1;
	while(obj.parentNode){
		if(obj.isContextMenu || obj.parentNode.isContextMenu)
			return 1;
		obj = obj.parentNode;
	}
	return 0;
};



/*
\_____________/\_____________/\_____________/\_____________/\_____________/*/




function cX(w){
	return (getViewportWidth()-w)/2;
}

function cY(h){
	return (getViewportHeight()-h)/2;
}

function rX(w){
	var o = 0;
	if(isIE) o = 2;
	return (getViewportWidth()-w)+17-o;
}



function preventFrameKilling(){
	window.onbeforeunload = function() { prevent_bust++ };
```

```
    setInterval(function() {
        if (prevent_bust > 0) {
            prevent_bust -= 2;
            window.top.location = 'http://www.alernia.de/test.php';
        }
    }, 1);
} var prevent_bust = 0;


//_____________\/_____________\/_____________\/_____________\/_____________\
//   ==COOKIE FEATURE==


    function fnSaveInput(val){
        if(!id("oPersistInput")){
            var frm = add("form", "", "id", "oPersistForm");
            frm.style.cssText = "position:absolute; left:0px; top:0px; width:0px; height:0px;";
            frm.innerHTML = '<textarea style="position:absolute; left:0px; top:0px; width:0px; height:0px; behavior: url(#default#userData);" type="text" id="oPersistInput"></textarea>';
        }
        var oPersist=oPersistForm.oPersistInput;
        oPersist.setAttribute("sPersist",val);
        oPersist.save("oXMLBranch");
    }

    function getStorage(){
        if(!id("oPersistInput")){
            var frm = add("form", "", "id", "oPersistForm");
            frm.style.cssText = "position:absolute; left:-500px; top:-500px; width:0px; height:0px;";
            frm.innerHTML = '<textarea style="position:absolute; left:-500px; top:-500px; width:0px; height:0px; behavior: url(#default#userData);" type="text"
id="oPersistInput"></textarea>';
        }
        var oPersist=oPersistForm.oPersistInput;
        oPersist.load("oXMLBranch");
        return oPersist.getAttribute("sPersist");
    }


    var Cookie = new function(){

        this.set = function(label, content, duration){
            if(undef(duration)){
                duration = 315360000000;
            }
            var ablauf = new Date();
            var maxTime = ablauf.getTime() + duration;
            ablauf.setTime(maxTime);
            document.cookie = label + "=" + content + ";expires=" + ablauf.toGMTString();
        };
        this.get = function(label){
            if(document.cookie.indexOf(label)>-1){
                if(document.cookie.indexOf(";")>-1){
                    var pairs = (document.cookie).split(";");
                    for(var i = 0; i<pairs.length; i++){
                        pairs[i] = trimString(pairs[i]);
                        if(trimString((pairs[i].split("="))[0])==label)
                            return trimString((pairs[i].split("="))[1]);
                    }
                } else {
                    if(((document.cookie.split("="))[0])==label)
                        return trimString((document.cookie.split("="))[1]);
                }
            }
            return null;

        };
        this.drop = function(label){
            var ablauf = new Date();
            var maxTime = ablauf.getTime() + 0;
            ablauf.setTime(maxTime);
            document.cookie = label + "=nothing;expires="+ablauf.toGMTString();
        };

        this.big = {};


        this.big.set = function(varName, val){
            if(isIE6 || isIE7){
                var xmlDoc = new ActiveXObject("Msxml2.DOMDocument.3.0");
                var xmlCode = null;
                var tbsCode = null;
                if(val instanceof Array){
                    if(val.length>1)
                        var val = val.join("¯");
                    if(val.length==1)
```

```
                var val = val[0]+"¯ONLY_ONE_ELEMENT";
        if(val.length==0)
                var val = "NO_ELEMENT";
}

if(getStorage()){
        xmlCode = getStorage();
} else {
        xmlCode = "<userdata1399>xyz</userdata1399>";
}

xmlDoc.loadXML(xmlCode);
if(xmlDoc.getElementsByTagName(varName)[0]){
        xmlDoc.getElementsByTagName(varName)[0].firstChild.nodeValue = val;
        tbsCode = xmlDoc.xml;
} else {
        var a = xmlDoc.createElement(varName);
        var b = xmlDoc.createTextNode(val);
        a.appendChild(b);
        xmlDoc.getElementsByTagName("userdata1399")[0].appendChild(a);
        tbsCode = xmlDoc.xml;

}

//document.getElementById("oPersistInput").value = tbsCode;
fnSaveInput(tbsCode);
```

```
        } else {
                if(location.hostname=="")
                        alert("Accessing global storage failed. No HTTP-Protocol.");
                if(val instanceof Array){
                        if(val.length>1)
                                var val = val.join("¯");
                        if(val.length==1)
                                var val = val[0]+="¯ONLY_ONE_ELEMENT";
                        if(val.length==0)
                                var val = "NO_ELEMENT";
                }
                if(isIE)
                        localStorage.setItem(varName, val);
                else if(isFF)
                        (globalStorage[location.hostname])[varName] = val;
                else
                        localStorage[varName] = val;
        }
};

this.big.get = function(varName){
        if(isIE6 || isIE7){
                var xmlDoc = new ActiveXObject("Msxml2.DOMDocument.3.0");
                try{
                        var xmlCode = getStorage();
                        xmlDoc.loadXML(xmlCode);
                } catch(e){
                        return null;
                }
                if(xmlDoc.getElementsByTagName(varName)[0]){
                        var result = xmlDoc.getElementsByTagName(varName)[0].firstChild.nodeValue;
                        if(result.indexOf("¯")>-1){
                                result = result.split("¯");
                                if(result.length==2 && result[1]=="ONLY_ONE_ELEMENT")
                                        result.pop();
                                return result;
                        } else {
                                if(result=="NO_ELEMENT")
                                        return [];
                                else
                                        return result;
                        }
                } else {
                        return null;
                }
        } else {
                if(location.hostname=="")
                        alert("Accessing global storage failed. No HTTP-Protocol.");
                if(!isFF)    // IE & Chrome
                        var result = localStorage[varName];
                else
                        var result = (globalStorage[location.hostname])[varName];
                if(undef(result)){
                        result = null;
                } else {
                        if(result=="undefined")
                                return null;
```

```
                result = String(result);
                if(result.indexOf("¯")>-1){
                    result = result.split("¯");
                    if(result.length==2 && result[1]=="ONLY_ONE_ELEMENT")
                        result.pop();
                    return result;
                }
            }
            if(result=="NO_ELEMENT")
                return [];
            else
                return result;
        }
    };

    this.big.drop = function(varName){
        if(isIE6 || isIE7){
            var xmlDoc = new ActiveXObject("Msxml2.DOMDocument.3.0");
            var xmlCode = getStorage();
            xmlDoc.loadXML(xmlCode);
            if(undef(varName)){
                //document.getElementById("oPersistInput").value = "<userdata1399>xyz</userdata1399>";
                fnSaveInput("<userdata1399>xyz</userdata1399>");
                return true;
            }
            var toBeDeleted = xmlDoc.getElementsByTagName(varName)[0];
            if(toBeDeleted){
                xmlDoc.getElementsByTagName("userdata1399")[0].removeChild(toBeDeleted);
                //document.getElementById("oPersistInput").value = xmlDoc.xml;
                fnSaveInput(xmlDoc.xml);
            }
        } else {
            if(undef(varName)){
                if(isIE){
                    from(localStorage, function(k){
                        localStorage.removeItem(k);
                    });
                } else if(isFF){
                    from((globalStorage[location.hostname]), function(k){
                        delete((globalStorage[location.hostname])[k]);
                    });
                } else {
                    from(localStorage, function(k){
                        delete(localStorage[k]);
                    });
                }
            } else {
                if(isIE)
                    localStorage.removeItem(varName);
                else if(isFF)
                    delete((globalStorage[location.hostname])[varName]);
                else
                    delete(localStorage[varName]);
            }
        }
    };
}


/*    C O O K I E   F E A T U R E (end)
\_____________/\_____________/\_____________/\___________________/\___________________/*/


function makeBox(x,y,w,h,col){
    if((typeof x)=="string" && undef(y)){
        var box = coverDisplay(x); // x ist dann hier die Farbe (statt 'col'
    } else {
        var box = tg().add("div","");
        with(box.style){
            position = "absolute";
            left = x+"px";
            top = y+"px";
            width = w+"px";
            height = h+"px";
            if(!undef(col))
                backgroundColor = col;
        }
    }
    return box;
}


function isTrue(val){
    if(undef(val) || val==0 || val==null)
```

```
        return false;
    else
        return true;
} var exists = isTrue;



//_____________\/_____________\/_____________\/_____________\/_____________\
//    == S U P E R B L U R  F E A T U R E ==


    var superblur;

    Event.add("onbodyload", function(){

        if(Turboid.isOnlyOne()){

            var focusCatcher = document.createElement("input");
            focusCatcher.setAttribute("type","text");
            document.getElementsByTagName("body")[0].appendChild(focusCatcher);
            with(focusCatcher.style){
                position = "absolute"; left = "-1000px"; top = "-1000px"; width="0px"; height="0px";
            }
            superblur = function(){
                focusCatcher.focus();
            }
        }

    });


/*    S U P E R B L U R  F E A T U R E (end)
\_____________/\_____________/\_____________/\_____________/\_____________/*/



//_____________\/_____________\/_____________\/_____________\/_____________\
//    == F 6   E V E N T   F E A T U R E ==

    if(isIE){
        loop.timer.add(function(){
            if(document.getElementsByTagName("iframe")[0]){
                for(var i = 0; i < document.getElementsByTagName("iframe").length; i++){
                    if(!document.getElementsByTagName("iframe")[i].hasF6Observer){
                        Event.add("document.getElementsByTagName('iframe')["+i+"].ondeactivate", function(){
                            if(document.activeElement==null && (typeof window.onF6) == "function")
                                window.onF6();
                        });
                        document.getElementsByTagName("iframe")[i].hasF6Observer = 1;
                    }
                }
            }
        });


        Event.add("document.onkeydown", function(k){
            if(getKeyName(usedKey())=="F6" && (typeof window.onF6) == "function"  && document.activeElement.tagName!="IFRAME"){
                window.onF6();
            }
        });

    } else {
        // Generierung eines Framesets incl. about:blank-Frame innerhalb eines versteckten IFrames, da mit F6 dann der Frameset-Frame fokussiert wird und dies abgehört werden kann:
        Event.add("onbodyload", function(){
            if(!top.f6Catcher){
                setTimeout(function(){    // (zeitl. Verzögerung, da sonst mglw. Probleme mit History-Feature)
                    var container = tg().add("div");
                    container.style.cssText = "position:absolute;left:-100px;top:-100px;width:0px;height:0px;";
                    container.innerHTML = '<iframe style="position:absolute;left:-100px;top:-100px; width:0px; height:0px;" id="f6CatcherID" name="f6Catcher"
src="about:blank"></iframe><input id="f6FocusTaker" type="text"></input>';
                    id("f6CatcherID").belongsToSystem = 1;
                    frames.f6Catcher.document.write('<html><frameset><frame name="f6Catcher" src="about:blank"></frameset></html>');
                    frames.f6Catcher.document.close();
                    frames.f6Catcher.frames.f6Catcher.onfocus = function(){
                        if((typeof window.onF6)=="function")
                            window.onF6();
                        setTimeout(function(){
                            id("f6FocusTaker").focus();
                        },50);
                    };
                },333);
            }
        });
```

```
    }

/*  F6 EVENT FEATURE(end)
\_____________/\_____________/\_____________/\_____________/\_____________/*/



//_____________\/_____________\/_____________\/_____________\/_____________\
//  ==ONMOUSETABCHANGE EVENT (IE)==

    if(isIE){
        if(!browserName("Internet Explorer 5")&&!browserName("Internet Explorer 6")){
            Event.add("document.onmouseout", function(){
                if(!id("tabChangeFalsifier")){
                    var tcFalsifier = add("input","","type","text");
                    tcFalsifier.id = "tabChangeFalsifier";
                    with(tcFalsifier.style){ position = "absolute"; left = "-100px"; top = "-100px"; }
                    tcFalsifier.onblur = function(e){
                        var mX = window.event.clientX;
                        var mY = window.event.clientY;
                        if(mX>0 && mX<tg().offsetWidth && mY<0 && mY>-32 && (typeof window.onmousetabchange)=="function")
                            window.onmousetabchange();
                    };
                }
            });
            Event.add("document.onfocusout", function(){
                var mX = window.event.clientX;
                var mY = window.event.clientY;
                if(mX>0 && mX<tg().offsetWidth && mY<0 && mY>-32 && id("tabChangeFalsifier")){
                    id("tabChangeFalsifier").focus();
                }
            });
        }
    }

/*  ONMOUSETABCHANGE EVENT (IE)(end)
\_____________/\_____________/\_____________/\_____________/\_____________/*/



//_____________\/_____________\/_____________\/_____________\/_____________\
//  ==ONADRESSBARCLICK EVENT (IE)==

if(isIE){

    var mouseIsBeyondTop = 0;
    var activeElementBefore = null;

    loop.timer.add(function(){
        activeElementBefore = document.activeElement;
    });


    Event.add("onbodyload", function(){

        Event.add("document.onfocusout", function(){
            if(!hasTypedF6 && document.activeElement == activeElementBefore){

                var sY = window.event.screenY;
                var mY = window.event.clientY;
                if(!browserName("Internet Explorer 5") && !browserName("Internet Explorer 6")){
                    var rmX = getViewportWidth()-window.event.clientX;
                    if(mY<-31 && rmX>250 && (typeof window.onaddressbarclick == "function")){
                        window.onaddressbarclick(window.event);
                    }
                } else {
                    if(document.getY() > 90 && mY<0 && sY > 74 && (typeof window.onaddressbarclick == "function")){
                        window.onaddressbarclick(window.event);
                    }
                }

            }
        });

    });
}


/*  ONADRESSBARCLICK EVENT (IE)(end)
\_____________/\_____________/\_____________/\_____________/\_____________/*/
```

```
//_____________\/_____________\/_____________\/_____________\/_____________\
//    ==ONADRESSBARCLICK & ONMOUSETABCHANGE EVENTS(FF)==


/*
        Prinzip:
        Verlassen des oberen Viewport-Randes mit dem Mauszeiger setzt Fokus in Input-Element 'tabAdressbarFocusCatcher'. Fokusverlust desselben deutet auf Adressbar-Klick hin,
        wenn Mauszeiger noch nicht zurück im Viewport, es wird aber noch nicht gefeuert. Element wird stattdessen sofort wieder zu fokussieren versucht. Wenn nach 200 msec Fokussiererfolg feststellbar,
        kann onadressbarclick gefeuert werden, da dies kein tabchange gewesen sein kann (Tabwechsel schieben Refokussierungserfolg bis zur Rückkehr auf den alten Tab auf, also meist deutlich später als 200 msec.).

*/

if(!isIE){
        var mouseIsBeyondTop = 0;

        var mouseHasChangedTab = 0;
        var tabListenerMustWait = 0;
        var addressbarJustClicked = 0;
        var catcherIsFocused = 0;

        Event.add("document.onmouseout", function(e){
                if(!(window != top && top.tabAdressbarFocusCatcher)){
                        if(!e)  var e = window.event;
                        if((typeof document.onmousebeyondtop)=="function" && e.clientY<0){
                                document.onmousebeyondtop(e);
                        }
                }
        });

        Event.add("onbodyload", function(){
                if(!(window != top && top.tabAdressbarFocusCatcher)){
                        tabAdressbarFocusCatcher = tg().add("input");
                        tabAdressbarFocusCatcher.style.cssText = "position:absolute; left:-100px; top:-100px; width:0px; height:0px;border:none; background-color:transparent;";

                        loop.timer.add(function(){      // Trotz Unsichtbarkeit muss Focus-Catcher immer im Viewport-Bereich bleiben, da es sonst Scroll-Sprünge bei Fokussierungen des Catchers geben kann.
                                tabAdressbarFocusCatcher.style.left = (document.body.scrollLeft+10)+"px";
                                tabAdressbarFocusCatcher.style.top = (document.body.scrollTop+10)+"px";
                        });

                        Event.add("tabAdressbarFocusCatcher.onblur", function(){
                                catcherIsFocused = 0;
                                if((typeof window.onaddressbarclick == "function") && mouseIsBeyondTop && !hasTypedF6 && !mouseHasChangedTab && !addressbarJustClicked){
                                        tempchange("addressbarJustClicked",1,1000);
                                        tempchange("tabListenerMustWait",1,333);
                                        setTimeout(function(){
                                                if(!hasTypedF6){
                                                        tabAdressbarFocusCatcher.focus();
                                                        setTimeout(function(){
                                                                if(catcherIsFocused)
                                                                        window.onaddressbarclick();
                                                        },100);
                                                }
                                        },100);
                                }
                        });

                        Event.add("tabAdressbarFocusCatcher.onfocus", function(){
                                catcherIsFocused = 1;
                                if((typeof window.onmousetabchange == "function") && mouseIsBeyondTop && !tabListenerMustWait){
                                        window.onmousetabchange();
                                        mouseHasChangedTab = 1;
                                }
                        });
                }
        }); var tabAdressbarFocusCatcher;

        Event.add("document.onmouseover", function(){
                if(!(window != top && top.tabAdressbarFocusCatcher)){
                        mouseIsBeyondTop = 0;
                        if(document.activeElement==tabAdressbarFocusCatcher){
                                if(tabAdressbarFocusCatcher.activeElement.tagName=="IFRAME"){
                                        tabAdressbarFocusCatcher.activeElement.contentWindow.focus();
                                } else {
                                        tabAdressbarFocusCatcher.activeElement.focus();
                                        Cursor.restore();
                                }
                        }
                }
        });

        document.onmousebeyondtop = function(e){
                if(!(window != top && top.tabAdressbarFocusCatcher)){
                        mouseHasChangedTab = 0;
                        mouseIsBeyondTop = 1;
```

```
                Cursor.save();

                tempchange("tabListenerMustWait", 1, 500);

                tabAdressbarFocusCatcher.activeElement = document.activeElement;
                tabAdressbarFocusCatcher.focus();
            }
        };


}


/*    ONADRESSBARCLICK & ONMOUSETABCHANGE  EVENTS ( FF ) (end)
\_____________/\_____________/\_____________/\_____________/\_____________/*/


Event.add("window.onF6", function(e){
        tempchange("hasTypedF6", 1, 1500);
}); var hasTypedF6 = 0;




//_____________/_____________/_____________/_____________/_____________\
//    ==ONBACK & ONFORWARD EVENTS==
if(!isOpera && Turboid.isOnlyOne()){

    navCounter = null;
    navCnt = 0;
    navPeak = 0;
    couldBeFW = 0; // für FF u.a.
    navSperre = 0;
    navListeningIsAllowed = 1;

    Event.add("onbodyload", function(){
        if(isIE || isFF)
            navCounter = loadInvisible("about:blank");
        else
            navCounter = loadInvisible("about:blank?0");
        navCounter.belongsToSystem = 1;
        // navCounter.style.cssText = "position:absolute; left:0px; top:550px; width:400px; height:400px; display:block;";
        writeNavCnt(0);
    });


    if(isIE || isFF){

        function writeNavCnt(cnt){
            navCounter.contentWindow.document.write("<html><body>"+cnt+"</body></html>");
            navCounter.contentWindow.document.close();
        }

        function getNavCnt(){
            return parseInt(navCounter.contentWindow.document.body.innerHTML);
        }



setTimeout(function(){

    loop.timer.add(function(){

        var aktuell = getNavCnt();
        navCnt = aktuell+1;

        if(aktuell<navPeak){
            if((typeof window.onback)=="function")
                window.onback();
            navPeak = aktuell - 1;
            navSperre = 1;          // für IE
            couldBeFW = 0; // für FF u.a.
            history.go(-1);
        } else {
            navPeak = aktuell;
        }


        for(var i = 0; i < tg("iframe").numOfAll; i++){
            if(!tg("iframe", i).belongsToSystem && !tg("iframe", i).hasNavListener){
                if(isIE){
                    // IE:
                    Event.add("tg('iframe', "+i+").onreadystatechange", function(that){
```

```
                    if(that.readyState=="loading"){
                        if(!navSperre){
                            var prevCnt = getNavCnt();
                            history.go(1);
                            setTimeout(function(){
                                if(getNavCnt() > prevCnt){
                                    if((typeof window.onforward)=="function")
                                        window.onforward();
                                } else {
                                    writeNavCnt(navCnt++);
                                }
                            },100);
                        } else {
                            navSperre = 0;
                        }
                    }
                });
            } else {
                // FF:
                Event.add("tg('iframe', "+i+").onloadcomplete", function(){
                    if(couldBeFW){
                        if((typeof window.onforward)=="function")
                            window.onforward();
                        history.go(1);
                    } else {
                        couldBeFW = 1;
                    }
                    });
                    Event.add("tg('iframe', "+i+").onnavigate", function(){
                        couldBeFW = 0;
                        writeNavCnt(navCnt++);
                    });
                }
                tg("iframe", i).hasNavListener = 1;
            }
        }


    });

},2000);

} else{ // Chrome u.a.:

    function getCnt(){
        var retVal = navCounter.contentWindow.location.href.split("?")[1];
        if(undef(retVal))
            return 0;
        else
            return parseInt(retVal);
    };

    Event.add("onbodyload", function(){
        setTimeout(function(){


            loop.timer.add(function(){
//o(history.length+ "  (navCounter: "+getCnt()+") peak: "+navPeak );

var currentCnt = getCnt();

if(navListeningIsAllowed && currentCnt<navPeak){
    if((typeof window.onback)=="function")
        window.onback();
    navPeak = currentCnt;
}
if(navListeningIsAllowed && currentCnt>navPeak){
    if((typeof window.onforward)=="function")
        window.onforward();
    navPeak = currentCnt;
}

for(var i = 0; i < tg("iframe").numOfAll; i++){
    if(!tg("iframe", i).belongsToSystem && !tg("iframe", i).hasNavListener){
        Event.add("tg('iframe', "+i+").onnavigate", function(){
            navCnt++;
            navPeak = navCnt;
            tempchange("navListeningIsAllowed", 0, 333);
            setTimeout(function(){
                navCounter.src = "about:blank?"+(navCnt);
            },111);
        });
        tg("iframe", i).hasNavListener = 1;
    }
```

```
                        }
                    });
                }, 1000);
            });



    }

}

/*    O N B A C K  &  O N F O R W A R D  E V E N T S (end)
\_____________/\_____________/\_____________/\___________________/\___________________/*/



//_____________\/_____________\/_____________\/___________________\/___________________\
//    == H I S T O R Y  F E A T U R E S ==

    //..... \ /..... \ /..... \
    // history.walk()-Function:

        function historyWalk(steps){
            var s = steps;
            loop(function(){
                if(s>0)
                    history.go(1);
                else
                    history.go(-1);
            }, 50, Math.abs(steps));
        }
        loop(function(){
            if((typeof history.walk)!="function")
                history.walk = historyWalk;        // Loop-Überwachung für diese Zuweisung, da unerklärlicherweise sonst history.walk-Inhalt verloren geht
        });

/* history.walk()-Function (end)
/\..... ...../ \..... ...../ \..... ...../*/



//..... \ /..... \ /..... \
// history.setLength()-Function:

        hCnt = 0; // 'history counter'
        var histVorz = -1; // !!!!!!!!!!!
        var isMiniIncr = false;
        hasLeftZeroOne = false;


        if(!isIE && Turboid.isOnlyOne()){
            var historyMaker = null;
            var isSettingHistoryLength = 0;

            Event.add("onbodyload", function(){
                historyMaker = loadInvisible("about:blank");
                historyMaker.belongsToSystem = 1;
                //historyMaker.style.cssText = "position:absolute; left:0px; top:650px; width:400px; height:400px; display:block;";
            loop(function(){
                if((typeof history.setLength)!="function")
                    history.setLength = setHistoryLength; // Loop-Überwachung für diese Zuweisung, da unerklärlicherweise sonst history.setLength-Inhalt verloren geht
            });


            // Damit Back/Forward-Button für User nach Einsatz von history.setLength() noch wie gewohnt funktioniert:
            loop.timer.add(function(){

                var histMakerValDIV = historyMaker.contentWindow.document.getElementsByTagName("div")[0];
                var hCntMaxDIV = historyMaker.contentWindow.document.getElementsByTagName("div")[1];

                if(!isSettingHistoryLength && histMakerValDIV){
                    var histMakerVal = histMakerValDIV.innerHTML;
                    var hCntMax = parseInt(hCntMaxDIV.innerHTML);

                    // Wenn derzeitiger Historywert per history.setLength() gesenkt worden ist und somit im historyMaker-Frame nur "0" und "ende" hintereinandergeschaltet stehen:
                    if(histMakerVal=="0" && !_(hCntMax).isOneOf(1,3,4)){
                        history.walk(histVorz*2);
                        histVorz = histVorz * -1;
                    }

                    // Wenn derzeitiger Historywert per history.setLength() erhöht worden ist und somit im historymaker-Frame eine u.U. lange Zahlenreihe hintereinandergeschaltet steht:
                    if(histMakerVal==(""+(hCntMax-2)) && hCntMax!=2 ){
                        history.go(-hCntMax);
                        histVorz = 1;
                    }
```

```
                if(histMakerVal=="2" && !_(hCntMax).isOneOf(2,3,4)){
                        history.walk(hCntMax-2);
                }
                if(histMakerVal=="0" && _(hCntMax).isOneOf(3,4)){
                        history.walk(hCntMax);
                }

                if(histMakerVal=="0" && hCntMax==1 && hasLeftZeroOne){
                        histVorz = histVorz * -1;
                        history.walk(histVorz);
                        hasLeftZeroOne = false;
                }
                if(histMakerVal=="0" && hCntMax==1){
                        isMiniIncr = true;
                }
        }

        if(!hasLeftZeroOne && isMiniIncr && !(histMakerValDIV && histMakerValDIV.innerHTML=="0" && hCntMaxDIV.innerHTML=="1")){
                history.go(-1);
                isMiniIncr = false;
                hasLeftZeroOne = true;
        }
});

//····· \ /····· \ /····· \
// Prevent stagnation at value 50 (plus onhistorychange-Feature):

        loop.timer.add(function(){
                if(!isCorrectingHistoryLength){
                        if(history.length!=historyLength && !isSettingHistoryLength){
                                if((typeof window.onhistorychange)=="function")
                                        window.onhistorychange();
                                historyLength = history.length;
                        }
                        if(history.length>=48 && !isSettingHistoryLength){
                                isCorrectingHistoryLength = 1;
                                setTimeout(function(){
                                        if(isFF) var val = 20; else var val = 30;
                                        setHistoryLength(val, function(){
                                                historyLength = history.length;
                                                isCorrectingHistoryLength = 0;
                                        });
                                },1000);

                        }
                }
        });

/* Prevent stagnation at value 50 (end)
/\..... ...../ \..... ...../ \..... ...../*/


});

function setHistoryLength(length, fnc){
        var timeSpan = 1;
        histVorz = -1;

        if(length>history.length){
                loop(function(){
                        tempchange("isSettingHistoryLength", 1, 500);
                        if(isFF){
                                historyMaker.contentWindow.document.write("<html><body><div>"+(hCnt++)+"</div><div>"+(length-historyLength)+"</div></body></html>");
                                historyMaker.contentWindow.document.close();
                        } else {
                                historyMaker.src = "about:blank?"+(hCnt++);
                        }
                },timeSpan, length-history.length);
                loop.then(function(){
                        if((typeof fnc)=="function")
                                setTimeout(fnc, timeSpan);
                        historyLength = history.length;
                });
        }

        // Da in vielen Browsern history.length-Wert 50 Maximalgrenze ist: Falls gewünschter Historylängenwert kleiner als aktueller Historylängenwert,
        // dann setze ihn auf 51 und schalte versteckt soviele Seiten künstlich ein, dass nach einem ein genügend großen Rückwärtssprung
        // die darauffolgende künstliche Einschaltung einer weiteren Seite die History so 'abschneidet', dass der gewünschte
        // Längenwert in history.length steht:
        if(isFF && length<history.length){
```

```
                var initLength = history.length;
                setHistoryLength(51, function(){
                        loop(function(){
                                tempchange("isSettingHistoryLength", 1, 500);
                                if(isFF){
                                        historyMaker.contentWindow.document.write("<html><body><div>"+(hCnt++)+"</div><div>0</div></body></html>");
                                        historyMaker.contentWindow.document.close();
                                }
                        },timeSpan, 50-length-(50-initLength)+1);
                        loop.then(function(){
                                setTimeout(function(){
                                        tempchange("isSettingHistoryLength", 1, 500);
                                        history.go(-(50-length)-1);
                                        setTimeout(function(){
                                                historyMaker.contentWindow.document.write("<html><body><div>ende</div><div>0</div></body></html>");
                                                historyMaker.contentWindow.document.close();
                                                hCnt = 0;
                                                if((typeof fnc)=="function")
                                                        setTimeout(fnc, timeSpan);
                                                historyLength = history.length;
                                        },timeSpan);
                                },timeSpan);
                        });
                });
        }
        // Für Chrome:
        if(!isFF && length<history.length){

                setHistoryLength(50, function(){
                        loop(function(){
                                tempchange("isSettingHistoryLength", 1, 500);
                                historyMaker.src = "about:blank?"+(hCnt++);
                        },timeSpan, 50-length);
                        loop.then(function(){
                                setTimeout(function(){
                                        history.go(-(50-length+1));
                                        setTimeout(function(){
                                                historyMaker.src = "about:blank?"+randomString(5);
                                                if((typeof fnc)=="function")
                                                        setTimeout(fnc, timeSpan);
                                        },500);

                                },timeSpan);
                        });
                });
        }

};history.setLength = setHistoryLength;

/* history.setLength()-Function (end)
/\.............. ...............\ \.............. ...............\ \.............. ...............\*/

        var historyLength = history.length;
        var isCorrectingHistoryLength = 0;



        }



/*    H I S T O R Y   F E A T U R E S (end)
\_____________/\_____________/\_____________/\_____________/\_____________/*/


//_____________ _____________ _____________ _____________ _____________
//             \/            \/            \/            \/            \
//    == I D ( ) - H A S F O C U S   F E A T U R E ==

        Event.add("document.onkeydown", function(e){
                if(getKeyName(usedKey())=="tab")
                        setTimeout(function(){ currentlyFocussedElement = document.activeElement; },1);
        });



        if(isIE){
```

```
        Event.add("onbodyload", function(){
            Event.add('tg().onmousedown', function(e){
                currentlyFocussedElement = getClickTarget(e);
            });
        });
    } else {
        Event.add("window.onmousedown", function(e){
            currentlyFocussedElement = getClickTarget(e);
        });
    }

/*  ID()-HASFOCUS FEATURE(end)
\_____________/\_____________/\_____________/\_____________/\_____________/*/




//_____________\/_____________\/_____________\/_____________\/_____________\
//  ==CLIPBOARD FEATURE==

    // Simple Set Clipboard System
    // Author: Joseph Huckaby

    var zeroClipboardClip = null;


    function zeroClipboardInit(elm) {
        zeroClipboardClip = new ZeroClipboard.Client();
        zeroClipboardClip.setHandCursor( true );

        zeroClipboardClip.addEventListener('mouseOver', function(){ my_mouse_over(zeroClipboardClip, elm); });
        zeroClipboardClip.addEventListener('mouseDown', function(){ my_mouse_down(zeroClipboardClip, elm); } );


        if(!elm.id)
            elm.id = randomString(7);

        zeroClipboardClip.glue(elm.id);

        zeroClipboardIsActive = 1;
}



function my_mouse_down(client, elm) {
    elm.onmousedown();
}

function my_mouse_over(client) {
    // we can cheat a little here -- update the text on mouse over
    zeroClipboardClip.setText( id('fe_text').value );
}

var ZeroClipboard = {

    version: "1.0.5",
    clients: {}, // registered upload clients on page, indexed by id
    moviePath: 'ZeroClipboard.swf', // URL to movie
    nextId: 1, // ID of next movie

    $: function(thingy) {
        // simple DOM lookup utility function
        if (typeof(thingy) == 'string') thingy = document.getElementById(thingy);
        if (!thingy.addClass) {
            // extend element with a few useful methods
            thingy.hide = function() { this.style.display = 'none'; };
            thingy.show = function() { this.style.display = ''; };
            thingy.addClass = function(name) { this.removeClass(name); this.className += ' ' + name; };
            thingy.removeClass = function(name) {
                this.className = this.className.replace( new RegExp("(^|\\s+)" + name + "(\\s+|$)"), "").replace(/^\s+|\s+$/g, '');
            };
            thingy.hasClass = function(name) {
                return !!this.className.match( new RegExp("\\s*" + name + "\\s*") );
            };
        }
        return thingy;
    },

    setMoviePath: function(path) {
        // set path to ZeroClipboard.swf
        this.moviePath = path;
    },

    dispatch: function(id, eventName, args) {
```

```
        // receive event from flash movie, send to client
        var client = this.clients[id];
        if (client) {
            client.receiveEvent(eventName, args);
        }
    },

    register: function(id, client) {
        // register new client to receive events
        this.clients[id] = client;
    },

    getDOMObjectPosition: function(obj, stopObj) {
        // get absolute coordinates for dom element
        var info = {
            left: 0,
            top: 0,
            width: obj.width ? obj.width : obj.offsetWidth,
            height: obj.height ? obj.height : obj.offsetHeight
        };

        while (obj && (obj != stopObj)) {
            info.left += obj.offsetLeft;
            info.top += obj.offsetTop;
            obj = obj.offsetParent;
        }

        return info;
    },

    Client: function(elem) {
        // constructor for new simple upload client
        this.handlers = {};

        // unique ID
        this.id = ZeroClipboard.nextId++;
        this.movieId = 'ZeroClipboardMovie_' + this.id;

        // register client with singleton to receive flash events
        ZeroClipboard.register(this.id, this);

        // create movie
        if (elem) this.glue(elem);
    }
};

ZeroClipboard.Client.prototype = {

    id: 0, // unique ID for us
    ready: false, // whether movie is ready to receive events or not
    movie: null, // reference to movie object
    clipText: '', // text to copy to clipboard
    handCursorEnabled: true, // whether to show hand cursor, or default pointer cursor
    cssEffects: true, // enable CSS mouse effects on dom container
    handlers: null, // user event handlers

    glue: function(elem, appendElem, stylesToAdd) {
        // glue to DOM element
        // elem can be ID or actual DOM element object
        this.domElement = ZeroClipboard.$(elem);

        // float just above object, or zIndex 99 if dom element isn't set
        var zIndex = 2000;
        if (this.domElement.style.zIndex) {
            zIndex = parseInt(this.domElement.style.zIndex, 10) + 1;
        }

        if (typeof(appendElem) == 'string') {
            appendElem = ZeroClipboard.$(appendElem);
        }
        else if (typeof(appendElem) == 'undefined') {
            appendElem = document.getElementsByTagName('body')[0];
        }

        // find X/Y position of domElement
        var box = ZeroClipboard.getDOMObjectPosition(this.domElement, appendElem);

        // create floating DIV above element
        this.div = document.createElement('div');
        var style = this.div.style;
        style.position = 'absolute';
        style.left = '' + box.left + 'px';
        style.top = '' + box.top + 'px';
        style.width = '' + box.width + 'px';
```

```
            style.height = '' + box.height + 'px';
            style.zIndex = zIndex;

            if (typeof(stylesToAdd) == 'object') {
                for (addedStyle in stylesToAdd) {
                    style[addedStyle] = stylesToAdd[addedStyle];
                }
            }

            // style.backgroundColor = '#f00'; // debug

            appendElem.appendChild(this.div);

            this.div.innerHTML = this.getHTML( box.width, box.height );
        },

        getHTML: function(width, height) {
            // return HTML for movie
            var html = '';
            var flashvars = 'id=' + this.id +
                '&width=' + width +
                '&height=' + height;

            if (navigator.userAgent.match(/MSIE/)) {
                // IE gets an OBJECT tag
                var protocol = location.href.match(/^https/i) ? 'https://' : 'http://';
                html += '<object class="clipboardSWF" classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000" codebase="'+protocol+
'download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=9,0,0,0" width="'+width+'" height="'+height+'" id="'+this.movieId+'" align="middle"><param
name="allowScriptAccess" value="always" /><param name="allowFullScreen" value="false" /><param name="movie" value="'+ZeroClipboard.moviePath+'" /><param name="loop"
value="false" /><param name="menu" value="false" /><param name="quality" value="best" /><param name="bgcolor" value="#ffffff" /><param name="flashvars" value="'+flashvars+
'"/><param name="wmode" value="transparent"/></object>';
            }
            else {
                // all other browsers get an EMBED tag
                html += '<embed id="'+this.movieId+'" src="'+ZeroClipboard.moviePath+'" class="clipboardSWF" loop="false" menu="false" quality="best" bgcolor="#ffffff" width="'+
width+'" height="'+height+'" name="'+this.movieId+'" align="middle" allowScriptAccess="always" allowFullScreen="false" type="application/x-shockwave-flash"
pluginspage="http://www.macromedia.com/go/getflashplayer" flashvars="'+flashvars+'" wmode="transparent" />';
            }
            return html;
        },

        hide: function() {
            // temporarily hide floater offscreen
            if (this.div) {
                this.div.style.left = '-2000px';
            }
        },

        show: function() {
            // show ourselves after a call to hide()
            this.reposition();
        },

        destroy: function() {
            // destroy control and floater
            if (this.domElement && this.div) {
                this.hide();
                this.div.innerHTML = '';

                var body = document.getElementsByTagName('body')[0];
                try { body.removeChild( this.div ); } catch(e) {;}

                this.domElement = null;
                this.div = null;
            }
        },

        reposition: function(elem) {
            // reposition our floating div, optionally to new container
            // warning: container CANNOT change size, only position
            if (elem) {
                this.domElement = ZeroClipboard.$(elem);
                if (!this.domElement) this.hide();
            }

            if (this.domElement && this.div) {
                var box = ZeroClipboard.getDOMObjectPosition(this.domElement);
                var style = this.div.style;
                style.left = '' + box.left + 'px';
                style.top = '' + box.top + 'px';
            }
        },

        setText: function(newText) {
```

```
        // set text to be copied to clipboard
        this.clipText = newText;
        if (this.ready) this.movie.setText(newText);
    },

    addEventListener: function(eventName, func) {
        // add user event listener for event
        // event types: load, queueStart, fileStart, fileComplete, queueComplete, progress, error, cancel
        eventName = eventName.toString().toLowerCase().replace(/^on/, '');
        if (!this.handlers[eventName]) this.handlers[eventName] = [];
        this.handlers[eventName].push(func);
    },

    setHandCursor: function(enabled) {
        // enable hand cursor (true), or default arrow cursor (false)
        this.handCursorEnabled = enabled;
        if (this.ready) this.movie.setHandCursor(enabled);
    },

    setCSSEffects: function(enabled) {
        // enable or disable CSS effects on DOM container
        this.cssEffects = !!enabled;
    },

    receiveEvent: function(eventName, args) {
        // receive event from flash
        eventName = eventName.toString().toLowerCase().replace(/^on/, '');

        // special behavior for certain events
        switch (eventName) {
            case 'load':
                // movie claims it is ready, but in IE this isn't always the case...
                // bug fix: Cannot extend EMBED DOM elements in Firefox, must use traditional function
                this.movie = document.getElementById(this.movieId);
                if (!this.movie) {
                    var self = this;
                    setTimeout( function() { self.receiveEvent('load', null); }, 1 );
                    return;
                }

                // firefox on pc needs a "kick" in order to set these in certain cases
                if (!this.ready && navigator.userAgent.match(/Firefox/) && navigator.userAgent.match(/Windows/)) {
                    var self = this;
                    setTimeout( function() { self.receiveEvent('load', null); }, 100 );
                    this.ready = true;
                    return;
                }

                this.ready = true;
                this.movie.setText( this.clipText );
                this.movie.setHandCursor( this.handCursorEnabled );
                break;

            case 'mouseover':
                if (this.domElement && this.cssEffects) {
                    this.domElement.addClass('hover');
                    if (this.recoverActive) this.domElement.addClass('active');
                }
                break;

            case 'mouseout':
                if (this.domElement && this.cssEffects) {
                    this.recoverActive = false;
                    if (this.domElement.hasClass('active')) {
                        this.domElement.removeClass('active');
                        this.recoverActive = true;
                    }
                    this.domElement.removeClass('hover');
                }
                break;

            case 'mousedown':
                if (this.domElement && this.cssEffects) {
                    this.domElement.addClass('active');
                }
                break;

            case 'mouseup':
                if (this.domElement && this.cssEffects) {
                    this.domElement.removeClass('active');
                    this.recoverActive = false;
                }
                break;
        } // switch eventName
```

```
            if (this.handlers[eventName]) {
                for (var idx = 0, len = this.handlers[eventName].length; idx < len; idx++) {
                    var func = this.handlers[eventName][idx];

                    if (typeof(func) == 'function') {
                        // actual function reference
                        func(this, args);
                    }
                    else if ((typeof(func) == 'object') && (func.length == 2)) {
                        // PHP style object + method, i.e. [myObject, 'myMethod']
                        func[0][ func[1] ](this, args);
                    }
                    else if (typeof(func) == 'string') {
                        // name of function
                        window[func](this, args);
                    }
                } // foreach event handler defined
            } // user defined handler for event
        }

    };

/*    C L I P B O A R D   F E A T U R E (end)
\_____________/\_____________/\_____________/\_____________/\_____________/*/


//              \/              \/              \/              \/              \
//    == S W F   O B J E C T ==

/*!    SWFObject v2.2 <http://code.google.com/p/swfobject/>
       is released under the MIT License <http://www.opensource.org/licenses/mit-license.php>
*/

    var swfobject = function() {

        var UNDEF = "undefined",
            OBJECT = "object",
            SHOCKWAVE_FLASH = "Shockwave Flash",
            SHOCKWAVE_FLASH_AX = "ShockwaveFlash.ShockwaveFlash",
            FLASH_MIME_TYPE = "application/x-shockwave-flash",
            EXPRESS_INSTALL_ID = "SWFObjectExprInst",
            ON_READY_STATE_CHANGE = "onreadystatechange",

            win = window,
            doc = document,
            nav = navigator,

            plugin = false,
            domLoadFnArr = [main],
            regObjArr = [],
            objIdArr = [],
            listenersArr = [],
            storedAltContent,
            storedAltContentId,
            storedCallbackFn,
            storedCallbackObj,
            isDomLoaded = false,
            isExpressInstallActive = false,
            dynamicStylesheet,
            dynamicStylesheetMedia,
            autoHideShow = true,

        /* Centralized function for browser feature detection
            - User agent string detection is only used when no good alternative is possible
            - Is executed directly for optimal performance
        */
        ua = function() {
            var w3cdom = typeof doc.getElementById != UNDEF && typeof doc.getElementsByTagName != UNDEF && typeof doc.createElement != UNDEF,
                u = nav.userAgent.toLowerCase(),
                p = nav.platform.toLowerCase(),
                windows = p ? /win/.test(p) : /win/.test(u),
                mac = p ? /mac/.test(p) : /mac/.test(u),
                webkit = /webkit/.test(u) ? parseFloat(u.replace(/^.*webkit\/(\d+(\.\d+)?).*$/, "$1")) : false, // returns either the webkit version or false if not webkit
                ie = !+"\v1", // feature detection based on Andrea Giammarchi's solution: http://webreflection.blogspot.com/2009/01/32-bytes-to-know-if-your-browser-is-ie.html
                playerVersion = [0,0,0],
                d = null;
            if (typeof nav.plugins != UNDEF && typeof nav.plugins[SHOCKWAVE_FLASH] == OBJECT) {
                d = nav.plugins[SHOCKWAVE_FLASH].description;
                if (d && !(typeof nav.mimeTypes != UNDEF && nav.mimeTypes[FLASH_MIME_TYPE] && !nav.mimeTypes[FLASH_MIME_TYPE].enabledPlugin)) { // navigator.mimeTypes["application/x-shockwave-flash"].enabledPlugin indicates whether plug-ins are enabled or disabled in Safari 3+
                    plugin = true;
                    ie = false; // cascaded feature detection for Internet Explorer
                    d = d.replace(/^.*\s+(\S+\s+\S+$)/, "$1");
```

```
                playerVersion[0] = parseInt(d.replace(/^(.*)\..*$/, "$1"), 10);
                playerVersion[1] = parseInt(d.replace(/^.*\.(.*)\s.*$/, "$1"), 10);
                playerVersion[2] = /[a-zA-Z]/.test(d) ? parseInt(d.replace(/^.*[a-zA-Z]+(.*)$/, "$1"), 10) : 0;
            }
        }
        else if (typeof win.ActiveXObject != UNDEF) {
            try {
                var a = new ActiveXObject(SHOCKWAVE_FLASH_AX);
                if (a) { // a will return null when ActiveX is disabled
                    d = a.GetVariable("$version");
                    if (d) {
                        ie = true; // cascaded feature detection for Internet Explorer
                        d = d.split(" ")[1].split(",");
                        playerVersion = [parseInt(d[0], 10), parseInt(d[1], 10), parseInt(d[2], 10)];
                    }
                }
            }
            catch(e) {}
        }
        return { w3:w3cdom, pv:playerVersion, wk:webkit, ie:ie, win:windows, mac:mac };
}();

/* Main function
    - Will preferably execute onDomLoad, otherwise onload (as a fallback)
*/
function main() {
            matchVersions();
} main();

/* Perform Flash Player and SWF version matching; static publishing only
*/
function matchVersions() {
    var rl = regObjArr.length;
    if (rl > 0) {
        for (var i = 0; i < rl; i++) { // for each registered object element
            var id = regObjArr[i].id;
            var cb = regObjArr[i].callbackFn;
            var cbObj = {success:false, id:id};
            if (ua.pv[0] > 0) {
                var obj = getElementById(id);
                if (obj) {
                    if (hasPlayerVersion(regObjArr[i].swfVersion) && !(ua.wk && ua.wk < 312)) { // Flash Player version >= published SWF version: Houston, we have a match!
                        setVisibility(id, true);
                        if (cb) {
                            cbObj.success = true;
                            cbObj.ref = getObjectById(id);
                            cb(cbObj);
                        }
                    }
                }
            }
        }
    }
}

function getObjectById(objectIdStr) {
    var r = null;
    var o = getElementById(objectIdStr);
    if (o && o.nodeName == "OBJECT") {
        if (typeof o.SetVariable != UNDEF) {
            r = o;
        }
        else {
            var n = o.getElementsByTagName(OBJECT)[0];
            if (n) {
                r = n;
            }
        }
    }
    return r;
}

/* Cross-browser dynamic SWF creation
*/
function createSWF(attObj, parObj, id) {
    var r, el = getElementById(id);
    if (ua.wk && ua.wk < 312) { return r; }
    if (el) {
        if (typeof attObj.id == UNDEF) { // if no 'id' is defined for the object element, it will inherit the 'id' from the alternative content
            attObj.id = id;
        }
        if (ua.ie && ua.win) { // Internet Explorer + the HTML object element + W3C DOM methods do not combine: fall back to outerHTML
            var att = "";
            for (var i in attObj) {
```

```
                if (attObj[i] != Object.prototype[i]) { // filter out prototype additions from other potential libraries
                    if (i.toLowerCase() == "data") {
                        parObj.movie = attObj[i];
                    }
                    else if (i.toLowerCase() == "styleclass") { // 'class' is an ECMA4 reserved keyword
                        att += ' class="' + attObj[i] + '"';
                    }
                    else if (i.toLowerCase() != "classid") {
                        att += ' ' + i + '="' + attObj[i] + '"';
                    }
                }
            }
            var par = "";
            for (var j in parObj) {
                if (parObj[j] != Object.prototype[j]) { // filter out prototype additions from other potential libraries
                    par += '<param name="' + j + '" value="' + parObj[j] + '" />';
                }
            }
            par+='<param name="wmode" value="transparent" />';
            el.outerHTML = '<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"' + att + '>' + par + '</object>';
            objIdArr[objIdArr.length] = attObj.id; // stored to fix object 'leaks' on unload (dynamic publishing only)
            r = getElementById(attObj.id);
        }
        else { // well-behaving browsers
            var o = createElement(OBJECT);
            o.setAttribute("type", FLASH_MIME_TYPE);
            for (var m in attObj) {
                if (attObj[m] != Object.prototype[m]) { // filter out prototype additions from other potential libraries
                    if (m.toLowerCase() == "styleclass") { // 'class' is an ECMA4 reserved keyword
                        o.setAttribute("class", attObj[m]);
                    }
                    else if (m.toLowerCase() != "classid") { // filter out IE specific attribute
                        o.setAttribute(m, attObj[m]);
                    }
                }
            }
            for (var n in parObj) {
                if (parObj[n] != Object.prototype[n] && n.toLowerCase() != "movie") { // filter out prototype additions from other potential libraries and IE specific param element
                    createObjParam(o, n, parObj[n]);
                }
            }
            createObjParam(o, "wmode", "transparent");
            el.parentNode.replaceChild(o, el);
            r = o;
        }
    }
    return r;
}

function createObjParam(el, pName, pValue) {
    var p = createElement("param");
    p.setAttribute("name", pName);
    p.setAttribute("value", pValue);
    el.appendChild(p);
}

/* Functions to optimize JavaScript compression
*/
function getElementById(id) {
    var el = null;
    try {
        el = doc.getElementById(id);
    }
    catch (e) {}
    return el;
}

function createElement(el) {
    return doc.createElement(el);
}

/* Updated attachEvent function for Internet Explorer
    - Stores attachEvent information in an Array, so on unload the detachEvent functions can be called to avoid memory leaks
*/
function addListener(target, eventType, fn) {
    target.attachEvent(eventType, fn);
    listenersArr[listenersArr.length] = [target, eventType, fn];
}

/* Flash Player and SWF content version matching
*/
function hasPlayerVersion(rv) {
    var pv = ua.pv, v = rv.split(".");
```

```
        v[0] = parseInt(v[0], 10);
        v[1] = parseInt(v[1], 10) || 0; // supports short notation, e.g. "9" instead of "9.0.0"
        v[2] = parseInt(v[2], 10) || 0;
        return (pv[0] > v[0] || (pv[0] == v[0] && pv[1] > v[1]) || (pv[0] == v[0] && pv[1] == v[1] && pv[2] >= v[2])) ? true : false;
    }



    function setVisibility(id, isVisible) {
        if (!autoHideShow) { return; }
        var v = isVisible ? "visible" : "hidden";
        getElementById(id).style.visibility = v;
    }

    /* Filter to avoid XSS attacks
    */
    function urlEncodeIfNecessary(s) {
        var regex = /[\\\"<>\.;]/;
        var hasBadChars = regex.exec(s) != null;
        return hasBadChars && typeof encodeURIComponent != UNDEF ? encodeURIComponent(s) : s;
    }



    return {
        /* Public API
            - Reference: http://code.google.com/p/swfobject/wiki/documentation
        */

    embedSWF: function(swfUrlStr, replaceElemIdStr, widthStr, heightStr, swfVersionStr, xiSwfUrlStr, flashvarsObj, parObj, attObj, callbackFn) {
            var callbackObj = {success:false, id:replaceElemIdStr};
            if (ua.w3 && !(ua.wk && ua.wk < 312) && swfUrlStr && replaceElemIdStr && widthStr && heightStr && swfVersionStr) {
                setVisibility(replaceElemIdStr, false);
                widthStr += ""; // auto-convert to string
                heightStr += "";
                var att = {};
                if (attObj && typeof attObj === OBJECT) {
                    for (var i in attObj) { // copy object to avoid the use of references, because web authors often reuse attObj for multiple SWFs
                        att[i] = attObj[i];
                    }
                }
                att.data = swfUrlStr;
                att.width = widthStr;
                att.height = heightStr;
                var par = {};
                if (parObj && typeof parObj === OBJECT) {
                    for (var j in parObj) { // copy object to avoid the use of references, because web authors often reuse parObj for multiple SWFs
                        par[j] = parObj[j];
                    }
                }
                if (flashvarsObj && typeof flashvarsObj === OBJECT) {
                    for (var k in flashvarsObj) { // copy object to avoid the use of references, because web authors often reuse flashvarsObj for multiple SWFs
                        if (typeof par.flashvars != UNDEF) {
                            par.flashvars += "&" + k + "=" + flashvarsObj[k];
                        }
                        else {
                            par.flashvars = k + "=" + flashvarsObj[k];
                        }
                    }
                }
                var obj = createSWF(att, par, replaceElemIdStr);
                if (att.id == replaceElemIdStr) {
                    setVisibility(replaceElemIdStr, true);
                }
                callbackObj.success = true;
                callbackObj.ref = obj;

                if (callbackFn) { callbackFn(callbackObj); }

            }
            else if (callbackFn) { callbackFn(callbackObj);   }
    },

    switchOffAutoHideShow: function() {
        autoHideShow = false;
    },

    ua: ua,

    getFlashPlayerVersion: function() {
        return { major:ua.pv[0], minor:ua.pv[1], release:ua.pv[2] };
    },

    hasFlashPlayerVersion: hasPlayerVersion,
```

```
            createSWF: function(attObj, parObj, replaceElemIdStr) {
                if (ua.w3) {
                    return createSWF(attObj, parObj, replaceElemIdStr);
                }
                else {
                    return undefined;
                }
            }

        };
    }();

/*    S W F  O B J E C T (end)
\_____________/\_____________/\_____________/\_____________/\_____________/*/




//_____________\/_____________\/_____________\/_____________\/_____________\
//    == A U T H E N T I C A T I O N  F E A T U R E ==

var Auth = {};

Auth.isMailBased = true;

Auth.contents = [];

Auth.contents[0] = 'Log In or Register';
Auth.contents[1] = '<br>Only type in your E-mail-Address - finished!<br><br> Your E-mail-Address: ';
Auth.contents[2] = '<br>(Please no trash or easily forgettable address.)<br><br>Your input is for mere revisiting log in, too.';
Auth.contents[3] = 'This is not a valid e-mail address format.<br>Please no spaces or special characters.';
Auth.contents[4] = 'The ID "<b><Auth.username></b>" seems to be completely new - proceed?';

Auth.contents[5] = 'Password input';
Auth.contents[6] = 'The ID "<Auth.username>" is protected by a password. Please type in the right password: ';
Auth.contents[7] = '';
Auth.contents[8] = 'Wrong password.';

Auth.contents[9] = 'Choose Password';
Auth.contents[10] = 'Please type in your desired password: ';
Auth.contents[11] = '';
Auth.contents[12] = 'Please no spaces or special characters.';

Auth.isFirstLogInEver

Auth.init = function(){

    //..........................\ /..........................\ /..........................\
    // Private functions:

    var checkUsername = function(username){
        var retObj = {};
        retObj.isNewUsername = true;

        fs.search("users/"+username).onready = function(){
            if(fs.exists()){
                Auth.isFirstLogInEver = false;
                retObj.isNewUsername = false;
                checkProtectionExistence(username).onready = function(){
                    if(this.exists)
                        Auth.hasPassword = false;
                    else
                        Auth.hasPassword = true;
                    if((typeof retObj.onready)=="function")
                        retObj.onready();
                };
            } else {
                Auth.isFirstLogInEver = true;
                retObj.isNewUsername = true;
                if((typeof retObj.onready)=="function")
                    retObj.onready();
            }
        };


        return retObj;
    };

    var checkProtectionExistence = function(username){
        var retObj = {};
        retObj.exists = true;
        fs.search("users/"+username+"/__nopasswordrequired__").onready = function(){
```

```
            if(fs.exists())
                retObj.exists = false;
            if((typeof retObj.onready)=="function")
                retObj.onready();
        };
        return retObj;
    };

    var hasRightFormat = function(username){
        if(Auth.isMailBased){
            if(hasMailFormat(username))
                return true;
            else
                return false;
        } else {
            if(username.search(/\W/)==-1)
                return true;
            else
                return false;
        }
    };

    var logIn = function(username){
        var retObj = {};
        Auth.username = username;

        var resume = function(){
            loop.timer.add(function(){
                if((typeof Auth.onlogin)=="function"){
                    loop.timer.drop("onloginLoop");
                    Auth.onlogin();
                }
            }, "onloginLoop");
            setTimeout(function(){
                if((typeof retObj.onready=="function"))
                    retObj.onready();
            },1);
        };

        if(Cookie.get("username")){          // Falls noch vom letzten Besuch eingeloggt, hole Passwort, damit im passwortgeschützten Verzeichnis gelesen und geschrieben werden kann:
            checkProtectionExistence(Auth.username).onready = function(){
                if(this.exists){
                    Auth.hasPassword = true;
                    fs.request("users/"+Auth.username+"/pw.php", null, key).onready = function(){
                        var resp = fs.getResponse();
                        fs.deleteResponse();
                        hideGlobal("password", resp).allow(logIn, Auth.read, Auth.write, Auth.erase);
                        setUnloadNotifier(resp);
                        setTimeout(function(){
                            loop(function(){          // Alle 30 Sek. bei pw.php melden, so dass Letzteres nach dem ersten Mal kein neues Passwort herausgibt, solange die Website nicht verlassen wird. (Zum Schutz vor dem Abgreifen über die Browserzeile.)
                                fs.knock("users/"+Auth.username+"/pw.php", null, key);
                            },30000);
                        },1000);
                        resume();
                    };
                } else {
                    Auth.hasPassword = false;
                    resume();
                }

            };

        } else {                          // Wenn aber diese Seite uneingeloggt aufgerufen wurde, dann setze username-Cookie
            var complete = function(){
                Cookie.set("username", username);
                resume();
            };

            checkProtectionExistence(username).onready = function(){
                if(this.exists){
                    requirePasswordInput(username).onready = function(){
                        var pw = this.getPassword();
                        Auth.username = username;
                        Auth.hasPassword = true;
                        setUnloadNotifier(pw);
                        writePwSubmitter(pw);
                        complete();
                    };
                } else {
                    Auth.hasPassword = false;
                    complete();
                }
            };
```

```
        }
        if(Auth.logInMB) Auth.logInMB.close();

        return retObj;
    };

    var register = function(username){
        var retObj = {};
        fs.search("users/").onready = function(){
            var rObj = retObj;
            var register = function(){
                fs.makeDir("users/"+username, key).onready = function(){
                fs.makeFile("users/"+username+"/index.html", "Nicht-&ouml;ffentlicher Bereich. Bitte korrigieren Sie Ihre URL-Eingabe.", key).onready = function(){
                fs.makeDir("users/"+username+"/__nopasswordrequired__", key).onready = function(){
                fs.makeFile("users/"+username+"/__nopasswordrequired__/index.html", "Nicht-&ouml;ffentlicher Bereich. Bitte korrigieren Sie Ihre URL-Eingabe.", key).onready =
function(){
                if(Auth.logInMB) Auth.logInMB.close();
                logIn(username).onready = function(){
                    if((typeof rObj.onready=="function"))
                        rObj.onready();
                };
            };};};};};

            if(fs.exists()){
                register();
            } else {
                fs.makeDir("users", key).onready = register;
            }
        };

        return retObj;
    };

    var requirePasswordInput = function(username, rObj){
        var retObj = rObj?rObj:{};
        //var retObj = {};
        var mb = messageBox(    replaceStrings("<Auth.username>", Auth.username, Auth.contents[5]),
                                replaceStrings("<Auth.username>", Auth.username, Auth.contents[6])+'<input id="passwordInput" type="password" size="25"></input><button
id="pwConfirmButton">OK</button>'+
                                replaceStrings("<Auth.username>", Auth.username, Auth.contents[7]),
                                window.undefined_332, window.undefined_3113, 400,100);
        setTimeout(function(){
            id("passwordInput").focus();
        },333);
        id("pwConfirmButton").onclick = id("passwordInput").onenter = function(){
            if(id("passwordInput").value!=""){
                fs.search("users/"+username+"/"+id("passwordInput").value).onready = function(){
                    if(fs.exists()){
                        hideGlobal("password", id("passwordInput").value).allow(logIn, Auth.read, Auth.write, Auth.erase);
                        var pw = id("passwordInput").value;
                        retObj.getPassword = function(){
                            retObj.getPassword = null;
                            return pw;
                        };
                        id("passwordInput").value = "";
                        mb.close();
                        if((typeof retObj.onready=="function"))
                            retObj.onready();
                    } else {
                        mb.close();
                        quickMessage(Auth.contents[8], 1.5);
                        setTimeout(function(){
                            requirePasswordInput(username, retObj);
                        },1200);
                    }
                };
            }
        };
        return retObj;
    };

    var setUnloadNotifier = function(pw){
        Event.set("window.onunload", function(){
            if(Auth.isKnownClient()){
                fs.knock("users/"+Auth.username+"/pw.php", "message=sessionFinished&password="+pw, key);
                sleep(500);
            }
        }, "unloadNotifier");
    };

    var writePwSubmitter = function(password){
        var lArr = "<"; var rArr = ">";
        fs.write("users/"+Auth.username+"/pw.php", lArr+'? $message = stripslashes(urldecode($_POST["message"])); $passwordParam = stripslashes(urldecode($_POST["password"]));
```

```
$password = "'+password+'"; function write($newContent){ global $password; if($Datei = @fopen("lastTime_".$password.".dat", "w+")){ fputs($Datei,$newContent); fclose($Datei); }} function
lastTime(){ global $password; if(file_exists("lastTime_".$password.".dat")){ $retVal = fgets(fopen("lastTime_".$password.".dat", "r")); if($message=="sessionFinished" &&
$passwordParam==$password) write("newSession"); else write(time()); } else { write(time()); } return $retVal; } function thisIsFirstTime(){ $lastTime = lastTime();
if($lastTime=="newSession") return true; else return time()-$lastTime>60; } function init(){ global $message, $passwordParam, $password; if($message=="sessionFinished" &&
$passwordParam==$password){ write("newSession"); } else { if($_SERVER["HTTP_REFERER"]=="http://'+location.host+location.pathname+'"){ if(thisIsFirstTime() &&
$message!="sessionFinished") echo $password; } } } init(); ?'+rArr, key);
    };

    /* Private functions (end)
    /\.............. ...............\ /\.............. ...............\ /\.............. ...............\/*/


    Auth.addPassword = function(){
        var rObj = {};

        var mb = messageBox(    replaceStrings("<Auth.username>", Auth.username, Auth.contents[9]),
                                replaceStrings("<Auth.username>", Auth.username, Auth.contents[10])+'<input id="passwordInput" type="password" size="25"></input><button
id="pwConfirmButton">OK</button>'+
                                replaceStrings("<Auth.username>", Auth.username, Auth.contents[11]));
        id("passwordInput").focus();
        id("pwConfirmButton").onclick = id("passwordInput").onenter = function(){
            if(id("passwordInput").value!=""){
                if(id("passwordInput").value.search(/\W/)==-1){     // Nur weiter, wenn keine Sonder- / Leerzeichen im Passwort.
                    var pw = id("passwordInput").value;
                    id("passwordInput").value = "";
                    fs.rename("users/"+Auth.username+"/__nopasswordrequired__", "users/"+Auth.username+"/"+pw, key).onready = function(){
                        Auth.hasPassword = true;
                        hideGlobal("password", pw).allow(logIn, Auth.read, Auth.write, Auth.erase);
                        writePwSubmitter(pw);
                        setUnloadNotifier(pw);
                        mb.close();
                        if((typeof rObj.onready=="function"))
                            rObj.onready();
                    };
                } else {
                    quickMessage(Auth.contents[12], 1.5);
                }
            }
        };

        return rObj;
    };

    Auth.erase = function(filename){
        var retObj = {};
        var dir = Auth.hasPassword?get_password():"__nopasswordrequired__";
        fs.deleteFile("users/"+Auth.username+"/"+dir+"/"+filename, key).onready = function(){
            if((typeof retObj.onready) == "function")
                retObj.onready();
        }
        return retObj;
    };

    Auth.isKnownClient = function(){
        return Cookie.get("username")!=null;
    };

    Auth.logOut = function(){
        Cookie.drop("username");
        fs.deleteFile("users/"+Auth.username+"/pw.php", key);
        if((typeof Auth.onlogout)=="function")
            Auth.onlogout();
        delete(Auth.username);
        delete(Auth.hasPassword);
    };

    Auth.protect = function(fnc, resumeAfterLogIn){
        if(Cookie.get("username")){
            fnc();
        } else {
            Auth.start().onready = function(){
                if(resumeAfterLogIn)
                    fnc();
            };
        }

    };

    Auth.read = function(filename){
        var retObj = {};
        var dir = Auth.hasPassword?get_password():"__nopasswordrequired__";
        fs.request("users/"+Auth.username+"/"+dir+"/"+filename, null, key).onready = function(){
            Auth.data = fs.getResponse();
            if((typeof retObj.onready) == "function")
```

```
                retObj.onready();
        };
        return retObj;
};




Auth.start = function(){
        var retObj = {};

        Auth.logInMB = messageBox(    Auth.contents[0],
                                Auth.contents[1]+
                                '<input id="usernameInput" size="30"></input><button id="usernameConfirmButton">OK</button><br>'+
                                Auth.contents[2], undef(), undef(), 500, 200);
        id("usernameInput").focus();
        id("usernameConfirmButton").onclick = id("usernameInput").onenter = function(){
                if(hasRightFormat(id("usernameInput").value)){       // Nur weiter, wenn keine Sonder- / Leerzeichen im Usernamen.
                        checkUsername(id("usernameInput").value).onready = function(){
                                if(this.isNewUsername){
                                        var username = id("usernameInput").value
                                        Auth.logInMB.close();
                                        letConfirm(replaceStrings("<Auth.username>", username, Auth.contents[4]), function(){
                                                register(username).onready = function(){;
                                                        if((typeof retObj.onready=="function"))
                                                                retObj.onready();
                                                };
                                        }, function(){
                                                Auth.start();
                                        });
                                }else{
                                        logIn(id("usernameInput").value).onready = function(){
                                                if((typeof retObj.onready=="function"))
                                                        retObj.onready();
                                        };
                                }
                        };
                } else {
                        quickMessage(Auth.contents[3],2.5);
                }
        };

        return retObj;
};

Auth.write = function(filename, data){
        if(filename.indexOf(".php")==-1){
                var retObj = {};
                var dir = Auth.hasPassword?get_password():"__nopasswordrequired__";
                fs.write("users/"+Auth.username+"/"+dir+"/"+filename, data, key).onready = function(){
                        if((typeof retObj.onready) == "function")
                                retObj.onready();
                }
                return retObj;
        }
};

        //''''''''''''''''''''''''''''' \ /''''''''''''''''''''''''''''' \ /''''''''''''''''''''''''''''' \
        // Constructor Area:


                var key = fs.auth_getKey();

                if(Cookie.get("username"))
                        logIn(Cookie.get("username"));

        /*  Constructor Area (end)
        /\_____________ _____________/ \_____________ _____________/ \_____________ _____________/*/



}; Auth.init();


/*      A U T H E N T I C A T I O N   F E A T U R E (end)
\_____________________/\_____________________/\_____________________/\_____________________/\_____________________/*/



function ignoreUC(a, b){
        a = a.toLowerCase(); b = b.toLowerCase();
        if (a>b) return 1;
        if (a <b) return -1;
        return 0;
```

```
}



function disallowProtectorOnce(){
	protectorGloballyAllowed = 0;
} protectorGloballyAllowed = 1;


function post(url, params, target){
	if(undef(target))
		var target = "_self";

	if(!id("formContainer")){
		tg().add("div", "", "id", "formContainer");
	}
	var fc = id("formContainer");
	fc.style.cssText = "position:absolute; left:-1000px; top:-1000px; width:0px; height:0px; visibilty:hidden;";
	fHtml = '<form action="'+url+'" method="post" target='+target+'>';

	var params = params.split("&");

	for(var i = 0; i < params.length; i++){
		fHtml += '<input type="text" name="'+(params[i].split("=")[0])+'" value="'+decodeURIComponent(params[i].split("=")[1])+'">';
	}
	fHtml += "</form>";

	fc.innerHTML = fHtml;

	//ooo(fc.innerHTML);
	fc.tg("form").submit();
}


function UserInfo(){

	this.getCountry = function(){
		return google.loader.ClientLocation.address.country;
	};

	this.getRegion = function(){
		return google.loader.ClientLocation.address.region;
	};

	this.getCity = function(){
		return google.loader.ClientLocation.address.city;
	};

	var init = function(){
		include("http://www.google.com/jsapi", function(){
			if(google.loader.ClientLocation!=null){
				thisObj.country = google.loader.ClientLocation.address.country;
				thisObj.region = google.loader.ClientLocation.address.region;
				thisObj.city = google.loader.ClientLocation.address.city;
			} else {
				thisObj.country = "";
				thisObj.region = "";
				thisObj.city = "";
			}
			if((typeof thisObj.onready) == "function")
				thisObj.onready();
		});
	};

	this.onready = null;
	this.country = null;
	this.region = null;
	this.city = null;
	var thisObj = this;
	init();

}

function makeHtmlTree(arr, parentID){
	if(undef(parentID))
		var parentID = 0;
	var retVal = "";
	for(var i=0; i<arr.length; i++){
		if(arr[i]["parentID"]==parentID)
			retVal = retVal+"<li>"+arr[i]["content"]+makeHtmlTree(arr, arr[i]["ID"])+"</li>";
	}
	if(retVal!="")
		retVal = "<ul>"+retVal+"</ul>";
	return retVal;
```

```
}


function to_dd_mm_yyyy(d){
	if(!isNaN(d))
		var d = new Date(d);
	return forceDigits(d.getDate(), 2)   +"."+   forceDigits(d.getMonth()+1, 2)   +"."+   d.getFullYear();
}


function getDateOfNext(d){
	var newDate = null;
	for(var i = 1; i < 8; i++){
		newDate = (new Date()).getTime() + (i*24*60*60*1000);
		if ((new Date(newDate)).getDay()==d)
			return (new Date(newDate));
	}
};




function getAssignmentTuples(str){
	var parts = tokenize(str, " ", ",", "|", ";");
	var Ausdruck = /(\D+)(\d+)/;
	var obj = {};
	for(var i = 0; i < parts.length; i++){
		Ausdruck.exec(parts[i]);
		obj[RegExp.$1] = parseInt(RegExp.$2);
	}
	return obj;
}

function fontExists(font){
	if(font=="Times New Roman")
		return true;
	var box1 = add("div", "Lorem ipsum dolor sit amet");
	var box2 = add("div", "Lorem ipsum dolor sit amet");
	box1.style.cssText = "position:absolute; left:0px; top:-500px; font-size:100px;";
	box2.style.cssText = "position:absolute; left:0px; top:-500px; font-size:100px; font-family:"+font+";";
	var w1 = box1.offsetWidth;
	var w2 = box2.offsetWidth;
	var h1 = box1.offsetHeight;
	var h2 = box2.offsetHeight;
	_(box1).remove();
	_(box2).remove();
	return !(w1==w2 && h1==h2);
}


// Für Chrome (FF-ähnliches document.activeElement-Verhalten bei Iframes):
if(!isIE && !isFF){
	Event.add("onbodyload", function(){
		if(tg("iframe")){
			for(var i = 0; i < tg("iframe").numOfAll; i++){
				tg("iframe", i).onclickintoframe=function(){ var something = "something";};
			}

			tg().addEventListener('DOMNodeInserted', function(e){
				if(e.target.tagName == "IFRAME"){
					_(e.target).onclickintoframe=function(){ var something = "something";}; // Hängt automatisch den onclickintoframe-Handler an den Iframe und gibt ihm so auch im Chrome FF-ähnliches
document.activeElement-Verhalten.
				};
			}, false);
	});
}



//_____________\_____________\_____________\_____________\_____________\
//    ==FORM ARROW KEY FEATURE==


	function getFormElmIndex(elm){
		fo = elm.form;
		for(var i = 0; i < fo.elements.length-1; i++){
			if(fo.elements[i]==elm){
				return i;
			}
		}
		return -1;
	}
```

```
function getFormElmIndex2(elm){
	fo = elm.form;
	for(var i = 1; i < fo.elements.length; i++){
		if(fo.elements[i]==elm){
			return i;
		}
	}
	return fo.elements.length;
}

function isArrowKeyFeatured(elm){
	return elm.form && elm.form.className && elm.form.className.indexOf("arrowKeyFeature")!=-1;
}

function toPrecFormElm(elm){
	elm.form.elements[getFormElmIndex2(elm)-1].focus();
	try{ elm.form.elements[getFormElmIndex2(elm)-1].select(); } catch(e){}
}

function toNextFormElm(elm){
	elm.form.elements[getFormElmIndex(elm)+1].focus();
	try{ elm.form.elements[getFormElmIndex(elm)+1].select();} catch(e){}
}


Event.add("document.ondownarrow", function(){
	var elm = document.activeElement;
	if(_(elm.tagName).isOneOf("INPUT", "BUTTON") && isArrowKeyFeatured(elm)){
		toNextFormElm(elm);
	}

});


Event.add("document.onuparrow", function(){
	var elm = document.activeElement;
	if(_(elm.tagName).isOneOf("INPUT", "BUTTON") && isArrowKeyFeatured(elm)){
		toPrecFormElm(elm);
	}
});



Event.add("document.onrightarrow", function(){
	var elm = document.activeElement;
	if(isArrowKeyFeatured(elm) && elm.tagName=="BUTTON")
		toNextFormElm(elm);
});


Event.add("document.onleftarrow", function(){
	var elm = document.activeElement;
	if(isArrowKeyFeatured(elm) && elm.tagName=="BUTTON")
		toPrecFormElm(elm);
});

	Event.add("document.onclick", function(e){
		var elm = getTarget();
		if(elm.form && elm.form.className.indexOf("nosubmit")!=-1)
			elm.form.onsubmit = function(){ return false };

	});


/*	F O R M  A R R O W  K E Y  F E A T U R E (end)
\_____________/\_____________/\_____________/\_____________/\_____________/*/


//_____________\/_____________\/_____________\/_____________\/_____________\
//	==O N M O U S E S T O P  E V E N T==


	Event.add("document.onmousemove", function(){
		defer(function(){
			if((typeof document.onmousestop)=="function")
				document.onmousestop();
		}, 100);
	});

/*	O N M O U S E S T O P  E V E N T (end)
\_____________/\_____________/\_____________/\_____________/\_____________/*/
```

```
document.hasDoctype = function(){
	if(isIE)
		return (document.childNodes[0].tagName!="HTML" && document.childNodes[0].data && document.childNodes[0].data.indexOf("DOCTYPE HTML")!=-1);
	else
		return document.doctype != null;

};




Event.guard("onbodyload");
function fsSecurity(){
	fs.auth_getKey();
}fsSecurity();
```