

window.popup Objekt des Internet Explorer

Objekt für Popup-Fenster z.B. für Dialog, Meldungen, Tooltip

ab IE 5.5

Diese Fensterform ist eine stark reduzierte Instanz des window Objektes, wobei der Inhalt per Script kodiert werden muss (z.B. HTML-Tags).

Ein Popup-Fenster

wird aktuell, wenn es angezeigt wird

wird automatisch geschlossen, sobald ein anderes Objekt den Fokus erhält, also aktiv wird

z.B. durch User-Klick außerhalb des Popup-Fensters

Microsoft ändert fortlaufend Active-X-Eigenschaften von Windows und somit des IE

Diese fortlaufenden Änderungen muss der Programmierer in Erfahrung bringen.

Der Programmierer kann sich definitiv nicht auf Verfügbarkeit von Active-X-Controls verlassen und muss damit rechnen, dass seine Webseiten schlagartig nicht mehr komplett laufen weil u.a. Programmcode noch nicht angepasst ist. Ebenfalls muss der Programmierer Varianten von Windows und Patchzustände beachten, die prinzipiell Kostenprobleme verursachen können.

Mit anderen Worten: Wer Microsoft-Komponenten nutzt, muss wissen, was ihm blüht ... siehe nachfolgende Beispiel für Risiken.

Prinzipielle Lizenzprobleme für den Programmierer

Microsoft verlangt Lizenzierung von Windows. Bezüglich Windows-Versionen gibt es die Updatestufen z.B. per Servicepacks

Ein Windows mit Servicepack fällt unter die Lizenz des geupdateten Windows.

Ein Windows mit Vorversion zum Servicepack bedarf einer anderen Lizenz.

Will man z.B. den Internet Explorer 7 und 6 parallel testen, benötigt man 2 Windowslizenzen, da beide Versionen nicht parallel installierbar. Dazu kommt, dass es den IE 6 in 2 Versionen gibt: Win SP1 und SP2 (IE 7 nur ab Win SP2).

Für 3 Browserversionen benötigt man 3 Windowslizenzen, will man parallel testen.

Ein Blick auf Browser-Konkurrenzprodukte klärt die Sachlage unschlagbar: Opera ist z.B. parallel installierbar.

Hinweis: Man suche doch mal im Internet nach einem kostenlosen HTTP-Server vom Microsoft, um IE-Seite testen zu können, die JScript nutzen (inklusive Debugger). Denn sollte kein kostenloses Angebot findbar sein, kommen die Kosten von Entwicklungssoftware zum IE hinzu. Ein Blick auf Konkurrenz-HTTP-Server klärt die Sachlage: Apache-HTTP-Server ist kostenlos, allerdings nicht einfach einzurichten (Hinweis: Der HTTP-Server sollte virtuelle Hosts einrichten können und korrekt mit der Firewall des Users zusammenarbeiten können).

Abänderungen wegen Sicherheitspatches der jeweiligen Windows-Versionen

Abschaltungen von Active-X-Controls erfolgen auch im Rahmen der Sicherheitspatches zu Windows-Versionen.

Es ist auch möglich, dass wegen Sicherheitslücken abgeschaltet wird und somit Komponenten einer Webseite je nach Windowsversion nicht mehr laufen.

Im Rahmen der Sicherheitspatches ist es Microsoft sogar gelungen, Webseiten, die den MS-Encoder zur Komprimierung von

HTML- und JScript-Code nutzen, schlagartig unnutzbar zu machen: Ein Bug in einem Patch zu Windows XP - Q918899

Das Patch verursacht IE-Browser-Absturz bei per MS ScriptEncoder gepacktem JScript unter SP1 und 2 wenn HTTP 1.1 mit

Kompression genutzt wird z.B. bei

onclick-Handler auf IMG

klick ins Fenster per aktivem Popup

Der Absturz ist "read" -Fehler von immer ein und derselben Speicherstelle.

User, die dieses Patch installiert haben, können ab sofort keine IE-Seiten mit codiertem Script mehr ansehen.

Microsoft stellt Abhilfe nach geraumer Zeit zur Verfügung, jedoch spezifisch nach Windows XP-Version:

Patch Q918899 für

Windows XP SP1 Download für jedermann bereitgestellt

SP2 nur auf kostenpflichtige telefonische Anfrage des Users per Downloadlink bereitgestellt, da

Microsoft explizit die User registriert haben will, bei denen das

Patchproblem auftritt (User muss sich Telefonnummer besorgen)

Solange also das Patch zum fehlerhaften Patch vom User nicht installiert wird,

z.B. weil der User keine Ahnung hat, dass und wo er sich die Telefonnummer

von Microsoft besorgen muss bzw. zu besorgen hat, wird der User

IE-Seiten mit komprimierten Code dauerhaft nicht nutzen können.

(Microsoft-Support ist z.T. nur in Englisch).

Abänderungen wegen Browser-Inkompatibilität

Popupblocker-Fehler

Die Microsoft Browser-Version IE 7 ist nicht abwärtskompatibel bezüglich Popup per window.createPopup()

Popup per window-Objekt ist ein Markenzeichen des IE, das im IE 7 nicht mehr fehlerfrei nutzbar ist.

Der Fehler liegt in der Popup-Blockerverwaltung des IE und wurde mit dem IE 7 implementiert.

Der Fehler tritt nicht auf, wenn ein Fenster per window.open() erzeugt wurde.

Bedingung:

Scriptfehleranzeige ist erlaubt im IE 7

Popupblocker ist im IE abgeschaltet



ein aktives Fenster (Register) mit Dokument, dass fortlaufend (rekursiv) genau 1 window.popup per .show() erzeugt.
 ein weiteres Fenster (Register) z.B. leere Seite (about:blank)

beide (Register) liegen in einer gemeinsamen IE-Instanz

Ablauf: Wird Focus auf Register der leeren Seite gehalten und wird parallel das Popup per .show() erzeugt, bricht der Browser das Dokument mit .show() ab (Scriptfehler).

Der Popublocker für die leere Seite verursacht den Programmfehler im Dokument mit .show(). Es wird folgende Meldung angezeigt (in der Informationsleiste):

'Ein Popup wurde geblockt. Klicken Sie hier, um das Popup bzw. weitere Optionen anzuzeigen.'

Die Bedeutung der Meldung laut Microsoft-Hilfe im IE 7:

Der Popublocker hat ein Popufenster geblockt. Sie können den Popublocker deaktivieren oder Popups temporär zulassen, indem Sie auf die Informationsleiste klicken.

Die Realität zur obigen Meldung ist völlig anders:

Linke oder rechte Maus auf die Meldung liefert z.B. Einstellungen darunter

Popublocker einschalten
 weitere Informationen

jedoch keine Möglichkeit wie laut Bedeutung

Damit gilt: Der abgeschaltete Popublocker ist in Wirklichkeit aktiv.

Pikant: Ein Popup erscheint normalerweise auch über fremde Fenster, die nicht das Popup erzeugt haben (z.B. Fenster einer Windowsanwendung z.B. einer anderen IE-Instanz)

Der Popublocker des IE bemeckert aber NUR Webseite, die das Popup erzeugt.

Durch das Abwürgen von Popup wird das Popup natürlich auf und für anderen Seiten nicht relevant; im Falle einer anderen IE-Instanz also auch für diese nicht relevant, obwohl diese Instanz per Popublocker verwaltet wird.

Der Popublocker beschneidet die Popup-Reichweite an der Wurzel, ist aber nicht objektorientiert zu den anderen Webseiten (die nicht das Popup erzeugt haben).

Der Popublocker ist nicht als Filter aufgesetzt sondern reingestrickt worden.

Der Popublockerfehler verändert die Eventverwaltung:

Es werden u.a. ignoriert

onfocus
 onblur
 onfocusin
 onfocusout

und viele andere, so dass trotz Events z.B. des Body der Popublockerfehler entsteht.

// nachfolgender Code setzt focus nicht neu: Fenstereintrag in Taskleiste blinkt eventuell

```

window.focus();
window.document.focus();
if(document.body!=null)
{if(document.body.style!='hidden') // wenn hidden so focus() nicht möglich (Scriptfehler erzeugt)
 {document.body.focus();}
}
// wenn paralleles Fenster offen (on oder offline), so Scriptfehler erzeugt
popupzeiger.show(...);

```

Hinweis: Der Popufehler ist so elementar, dass die vielen Beta-Testphasen des IE mehr als fragwürdig erscheinen, wie die Angabe von Microsoft, dass Code neu programmiert wurde, um den IE sicherer zu machen.

focus-Methode beim IE 7

windows.focus() document.focus() und body.focus() funktionieren NICHT
 zwischen Register in einem IE-Fenster
 zwischen Fensters z.B. in Taskleiste

Hinweis:

.focus() setzt Element aktiv, gibt dem Element den Focus und feuert dann onfocus
 .setActive() ist Teilmenge von .focus(): nur das aktiv setzen
 funktioniert nicht mit allen Elementen, mit denen .focus() funktioniert

animierte Gif (mit Timer)

Animierte Gifs (mit Timer), die unter IE 6 korrekt laufen, müssen unter IE 7 im Timer nicht mehr laufen:

z.B. garnicht mehr sichtbar, oder Timer nicht verwendet.

Dann müssen animierte Gif-Bilder nach IE-Version bereitgestellt werden.

Abänderungen wegen Rechtsstreitigkeiten von Microsoft mit Fremdanbietern

Ein sehr bekanntes Beispiel ist die nachträglich eingeführte Einschränkung von Active-X-Controls wegen Patentwahrung durch Microsoft, wobei für den JScript-Programmierer massive Änderungen eintreten.

Wegen Patentwahrung hat Microsoft ein zunächst freiwilliges Patch herausgegeben, dass bei ActiveX-Control per APPLET, EMBED oder OBJECT, die auf dem Bildschirm rendern (mit oder ohne Userschnittstelle), dafür sorgt, dass bei mouseover über das Control eine Sprechblase erscheint, die darauf hinweist, dass das Objekt als ActiveX-Control klickbar ist. Diese Sprechblase erscheint auch, wenn das Control keine Userschnittstelle hat, also diese gar nicht klickbar ist.

Es wurde das Eventmodell gleichzeitig geändert:



Es werden alle Events solange unterdrückt, bis der User die Sprechblase geklickt hat.
 Das Klicken muss auf das Objekt im Sprechblasenrahmen erfolgen, der so groß ist, wie die Dimension, in der gerendert wurde.
 Es muss also ERST per Mausclick das Control aktiviert werden, ehe das Control klickbar und damit die Eventsteuerung aktiviert ist.
 Ein Control, dass programmtechnisch zwar was rendert, aber ansonsten ohne sichtbare programmtechnisch startet, muss ebenfalls geklickt werden, obwohl es bereits läuft und es nichts zu klicken gäbe (wenn keine Eventsteuerung eingebaut wurde).
 Wegen blockierter Eventsteuerung ist also die Sprechblase z.B. nicht automatisch klickbar.
 Die Eventauslösung per nicht-objekteigenen Eventhandler, der für das Objekt per fireEvent() ein Event auslöst, ist solange blockiert, bis der User die Sprechblase geklickt hat.

style.visibility='hidden' wird ignoriert

Die Sprechblase erscheint auch dann, wenn das Control mit style.visibility='hidden' belegt ist, also sich unsichtbar rendert:

Der Sprechblasenrahmen hat genau die Dimension wie die des unsichtbaren Controls. Der Sprechblasenrahmen erscheint also Zusammenhangslos, und der User weiß nicht, warum er klicken soll, wenn er nichts sieht. Vor allem weiß er nicht, WAS er klickt ... ideale Basis für Schadsoftware per Script.

Diese Sprechblase erscheint nur DANN NICHT, wenn die Userschnittstelle mit Breite == Höhe == 0 gerendert wird. Sollte die Userschnittstelle in einem Container liegen, z.B. DIV, dann wird der Container, wenn er in der Dimension kleiner ist, also die Userschnittstelle, angepasst. Daher muss der Container ebenfalls mit Breite == Höhe == 0 gerendert werden. Wegen Dimensionierung auf 0 sollte style.visibility="hidden" sein. Im Falle eines Containers reicht es, den style des Containers zu ändern, da visibility normalerweise vererbt wird an Kinder, also auch an das Control.

Abänderung wegen Abschaltungen

DirectX ist wegen Abschaltung von Active-X--Controls nicht mehr abwärtskompatibel:

Z.B. wurde bei Win XP SP2 Direct Animation aus DirectX schlagartig durch Abschaltung von Bibliotheken dezimiert, die es bei Win XP SP1 aber noch gibt.

Hier ein Beispiel aus dem Jahr 2004: Abschaltungen von Active-X-Controls

ActiveX-Controls und Unterstützung/Verbot 20041215

erlaubt sind noch

Tabular Data-Steuerelement {333C7BC4-460F-11D0-BC04-0080C7055A83} Das TDC (Tabular Data-Steuerelement) ermöglicht die Weiterverarbeitung von Daten, die nur im Textformat vorliegen, beispielsweise durch Darstellung in einer Tabelle oder Sortierung. Weitere Informationen:•

http://msdn.microsoft.com/workshop/database/tdc/tabular_data_control_node_entry.asp(http://msdn.microsoft.com/workshop/database/tdc/tabular_data_control_node_entry.asp)

Microsoft Agent Control - Version 2.0 {D45FD31B-5C6E-11D1-9EC1-00C04FD7081F} Microsoft Agent repräsentiert die neue Generation des ursprünglichen Office-Assistenten. Anstatt den Assistenten jedoch innerhalb eines Rahmens darzustellen wird hier lediglich der Charakter bzw. Agent selbst dargestellt und kann auch in Webseiten verwendet werden. Weitere Informationen:•

<http://msdn.microsoft.com/library/partbook/egvb6/introducingmicrosoftagent.htm>(<http://msdn.microsoft.com/library/partbook/egvb6/introducingmicrosoftagent.htm>)

Microsoft MSChat-Steuerelement-Objekt 2.0 - 2.5 {D6526FE0-E651-11CF-99CB-00C04FD64497} Dieses Steuerelement wird von Webautoren verwendet, um text- und graphisch basierte Chatgemeinden für Echtzeitkonversationen im Web zu erstellen.

Microsoft ActiveX Upload-Steuerelement, Version 1.5 {886e7bf0-c867-11cf-b1ae-00aa00a3f2c3} Dieses Steuerelement kann auf vielerlei Art genutzt werden, um auf einfache Weise Webinhalte via Drag and Drop zu veröffentlichen. Weitere Informationen:• 230298 (<http://support.microsoft.com/kb/230298/DE/>) - Posting Acceptor Release Notes

• http://msdn.microsoft.com/workshop/management/tools/reference/file_upload_control.asp

verboten sind

Datenbindung RDS {BD96C556-65A3-11D0-983A-00C04FC29E36} {BD96C556-65A3-11D0-983A-00C04FC29E33} Die RDS (Remote Data Service) Steuerelemente ermöglichen dem Browser, client-basierte SQL



Abfragen an einen Webserver zu stellen. Inzwischen wurde RDS jedoch durch neuere Standards wie SOAP abgelöst, von einer weiteren Verwendung von RDS wird daher abgeraten. Weitere Informationen:• 184375 (<http://support.microsoft.com/kb/184375/DE/>) - Sicherheitsaspekte bei RDS 1.5, IIS 3.0 oder 4.0 und ODBC

<http://msdn.microsoft.com/library/en-us/iissdk/iis/remotedatabindingwithremotedataservice.asp>
(<http://msdn.microsoft.com/library/en-us/iissdk/iis/remotedatabindingwithremotedataservice.asp>)
http://msdn.microsoft.com/library/en-us/dnmdac/html/data_mdacroadmap.asp
(http://msdn.microsoft.com/library/en-us/dnmdac/html/data_mdacroadmap.asp)

XMLDSO, XMLDocument, DOMDocument, und XMLIslandPeer {550dda30-0541-11d2-9ca9-0060b0ec3d39} {CFC399AF-D876-11d0-9C10-00C04FC99C8E} {e54941b2-7756-11d1-bc2a-00c04fb925f3} {7108ECB4-AFDC-11D1-ADC1-00805FC752D8} XMLDSO, XMLDocument, DOMDocument, und XMLIslandPeer ermöglichen die Verarbeitung von XML Daten, etwa die Bindung von HTML Elementen an einen XML Datensatz, oder das Einlesen, Manipulieren, und Zurückschreiben von XML Daten.

Die Steuerelemente DOMDocument und XMLIslandPeer bzw. die dazugehörigen ClassIDs sind nicht mehr aktuell, so dass von einer generellen Freigabe dieser Steuerelementgruppe abgeraten wird. Weitere Informationen:• http://msdn.microsoft.com/library/en-us/xmlsdk/htm/xml_concepts2_7ook.asp (http://msdn.microsoft.com/library/en-us/xmlsdk/htm/xml_concepts2_7ook.asp)

Internet Explorer

Active Setup / IE Active Setup-Steuerelement {F72A7B0E-0DD8-11D1-BD6E-00AA00B92AF1} Dieses Steuerelement enthält die in Microsoft Security Bulletin MS99-037 beschriebene Sicherheitsanfälligkeit. Um eine weitere Ausführung zu verhindern wurde im Rahmen dieses Security Bulletins ein Kill-Bit gesetzt, so dass selbst bei einer Freigabe dieses Controls eine Ausführung blockiert wird. Weitere Informationen:•

<http://www.microsoft.com/technet/security/bulletin/ms99-037.msp> (<http://www.microsoft.com/technet/security/bulletin/ms99-037.msp>)
<http://www.microsoft.com/technet/security/bulletin/fq99-037.msp>
(<http://www.microsoft.com/technet/security/bulletin/fq99-037.msp>)

240797 (<http://support.microsoft.com/kb/240797/DE/>) - So verhindern Sie die Ausführung von ActiveX-Steuerelementen in Internet Explorer

Media Player / Active Movie Runtime {A4001DE0-7075-11d0-89AB-00A0C9054129} Die Funktionalität dieses Steuerelements wird nun durch das Windows Media Player ActiveX Steuerelement abgedeckt. Das Active Movie Runtime Steuerelement wird daher nicht mehr unterstützt, von einer Freigabe wird abgeraten.

Media Player / ActiveMovie-Steuerelement {05589FA1-C356-11CE-BF01-00AA0055595A} Die Funktionalität dieses Steuerelements wird nun durch das Windows Media Player ActiveX Steuerelement abgedeckt. Das Active Movie Steuerelement wird daher nicht mehr unterstützt, von einer Freigabe wird abgeraten.

Media Player / Microsoft NetShow Player {2179C5D3-EBFF-11CF-B6FD-00AA00B4E220} Die Funktionalität dieses Steuerelements wird nun durch das Windows Media Player ActiveX Steuerelement abgedeckt. Das NetShow Player Steuerelement wird daher nicht mehr unterstützt, von einer Freigabe wird abgeraten.

Media Player / Windows Media Player {22D6F312-B0F6-11D0-94AB-0080C74C7E95} Dies ist das Steuerelement für Windows Media Player version 6.4 und war Installationsbestandteil bis einschließlich Windows Media Player Version 8. Ab Windows Media Player 9 wurde diese ClassID durch die neue ClassID {6BF52A52-394A-11D3-B153-00C04F79FAA6} abgelöst, deren Verwendung stattdessen empfohlen wird. Ab Windows Media Player Version 9 wird ferner die alte ClassID anhand eines Wrappers automatisch auf die neue ClassID umgeleitet. Die ClassID für Windows Media Player Version 9 ist jedoch nicht in der Liste der vom Administrator genehmigten Steuerelemente enthalten, und muss bei Bedarf manuell hinzugefügt werden.

Animierte Schaltflächen {0482B100-739C-11CF-A3A9-00A0C9034920} Dieses Steuerelement erlaubte in frühen Versionen des Internet Explorer die Verwendung animierter Schaltflächen auf Webseiten. Das Steuerelement wird



nicht mehr unterstützt und dürfte nur noch vereinzelt im Einsatz sein. Von der Freigabe des Steuerelements wird daher abgeraten.

IE Label-Steuerelement

{99B42120-6EC7-11CF-A6C7-00AA00A47DD2} Dieses Steuerelement ist nicht mehr aktuell und seit Internet Explorer Version 5 auch kein Bestandteil der Installation mehr. Das Steuerelement wird nicht mehr unterstützt und dürfte nur noch vereinzelt im Einsatz sein. Von einer Freigabe des Steuerelements wird daher abgeraten. Weitere Informationen: • 190045 (<http://support.microsoft.com/kb/190045/DE/>) - INFO: ActiveX Controls That Are Removed from Internet Explorer 5

IE Menu-Steuerelement {74701400-9DD9-11CF-A662-00AA00C066D2} Dieses Steuerelement ermöglicht die Handhabung von Menüstrukturen in Webseiten, wird jedoch nicht mehr unterstützt und dürfte nur noch selten Verwendung finden. Von einer Freigabe des Steuerelements wird daher abgeraten.

IE Preloader-Steuerelement {16E349E0-702C-11CF-A3A9-00A0C9034920} Dieses Steuerelement ermöglichte das Vorladen von Webseiten, ist jedoch inzwischen nicht mehr aktuell, wird nicht mehr unterstützt und dürfte nicht mehr im Einsatz sein. Aufgrund einer potentiellen Sicherheitsanfälligkeit in diesem Steuerelement wird von einer Freigabe abgeraten. Weitere Informationen: • 231452 (<http://support.microsoft.com/kb/231452/DE/>) - Update Available for "Legacy ActiveX Control" Issue

IE Timer-Steuerelement {59CCB4A0-727D-11CF-AC36-00AA00A47DD2} Dieses Steuerelement ist nicht mehr aktuell und seit Internet Explorer Version 5 kein Bestandteil der Installation mehr. Das Steuerelement wird nicht mehr unterstützt und dürfte nur noch vereinzelt im Einsatz sein. Von einer Freigabe des Steuerelements wird daher abgeraten. Weitere Informationen: • 190045 (<http://support.microsoft.com/kb/190045/DE/>) - INFO: ActiveX Controls That Are Removed from Internet Explorer 5

MCSiMenü {275E2FE0-7486-11D0-89D6-00A0C90C9B67} Dieses Steuerelement dient der Anpassung von Popupmenüs, ist jedoch nicht mehr aktuell und wurde nach Windows 98 nicht mehr ausgeliefert. Das Steuerelement wird nicht mehr unterstützt und dürfte nur noch vereinzelt im Einsatz sein. Von einer Freigabe des Steuerelements wird daher abgeraten.

Popupmenüobjekt {7823A620-9DD9-11CF-A662-00AA00C066D2} Dieses Steuerelement ist nicht mehr aktuell und seit Internet Explorer Version 5 kein Bestandteil der Installation mehr. Das Steuerelement wird nicht mehr unterstützt und dürfte nur noch vereinzelt im Einsatz sein. Von einer Freigabe des Steuerelements wird daher abgeraten. Weitere Informationen: • 190045 (<http://support.microsoft.com/kb/190045/DE/>) - INFO: ActiveX Controls That Are Removed from Internet Explorer 5

Microsoft Agent Control - Version 1.5 {F5BE8BD2-7DE6-11D0-91FE-00C04FD701A5} Microsoft Agent repräsentiert die neue Generation des ursprünglichen Office-Assistenten. Anstatt den Assistenten jedoch innerhalb eines Rahmens darzustellen wird hier lediglich der Charakter bzw. Agent selbst dargestellt und kann auch in Webseiten verwendet werden. Diese Version des Steuerelements ist jedoch nicht mehr aktuell und wird nicht mehr unterstützt. Von einer Freigabe des Steuerelements wird daher abgeraten. Weitere Informationen: • <http://msdn.microsoft.com/library/partbook/egvb6/introducingmicrosoftagent.htm> (<http://msdn.microsoft.com/library/partbook/egvb6/introducingmicrosoftagent.htm>)

Aktive Inhalte im Internet Explorer

Ab IE 6.0 ist das Blockieren aktiver Inhalte möglich, z.B. als Standardeinstellung. Es wird also dem IE verboten, JScript zu nutzen. Daher muss mit Start der Webseite auf das Blockieren von Inhalten der Webseite, die auf JScript basieren, aufmerksam gemacht werden.



Bleibt die Blockierung aktiv, so muss die Webseite ALLE Elemente, die per Script angesteuert werden, inaktiv machen: Am besten garnicht erst anzeigen. Oder es wird eine scriptfreie Version der Webseite per <NOSCRIPT> aktiviert, wobei dann Browser vorzuziehbar sind, die z.B. CSS exakter rendern als der IE (will man keine IE-spezifischen HTML-Elemente verwenden).

Wenn der IE 6.x aktive Inhalte blockiert, wird NOSCRIPT-Tag aktiviert, Ausnahme: Frameset

FRAMESET ist ein aktiver Inhalt:

Da der Frameset anstelle <BODY> kodiert sein muss, gilt:

Alle Tags, die für BODY zulässig sind, werden ignoriert, auch NOSCRIPT.

Wird neben Frameset noch BODY kodiert, so wird Frameset ignoriert.

Die Freigabe der Scriptblockierung erzeugt Ausführung aller Script-Teile inklusive der Eventauslösungen

Bsp.: Folgendes funktioniert vom Dokument, das window.open() hat im geöffneten Dokument (Quelltext im Dokument das window.open() verwendet):

```
function Y_unload(X00){X85[X00].close();}
var X85=new Array();var X86=new Array();
X85[0]=window.open(...);
var X87='parent.Y_unload(0);'; X86[0]=new Function(",X87);
X85[0].document.body.onunload=X86[0];
```

Wird die Scriptblockierung im geöffneten Fenster abgeschaltet, so wird das Fenster geschlossen, weil onunload ausgelöst wird.

Achtung: document.body.onunload funktioniert ev. nicht mehr wenn z.B. mit attachevent() aktiviert wurde

Folgende Metatags sind für den IE 6.x aktiver Inhalt:

```
<META HTTP-EQUIV="imagetoolbar" CONTENT="no">
unterdrückt NICHT IE-Kontextmenü rechte Maus auf Bild
<META HTTP-EQUIV="site-enter" CONTENT="revealtrans(duration=0.3, transition=12) ">
<META HTTP-EQUIV="site-exit" CONTENT="revealtrans(duration=0.3, transition=12) ">
```

Achtung: Für das Hinzufügen von Elementen in den BODY (document.body) per DOM-Funktion createElement() MUSS der Body komplett geparkt sein (document.body.readyState == 'complete').

Grund: Es wird standargemäß immer am Ende des BODY angefügt.

Für das Hinzufügen nicht an das Ende des BODY muss im HTML-Code ein Platzhalter z.B. DIV kodiert sein, innerhalb dessen dann die neuen HTML-Elemente erzeugt werden.

Objektbeschreibung

Erzeugung:

.createPopup()

Popupfenster ohne irgendwelche Fensterelemente öffnen z.B. für Anzeige von Tooltips Fenster wird mit der Anzeige per Methode .show() zum aktuellen Fenster erst geschlossen, wenn **anderes Fenster** aktuell wird z.B. durch Klicken außerhalb des Popupfensters

ab IE 5.5

Syntax:

```
[ var Zeiger = ] logischer_window_name.createPopup()
```

Zeiger

Referenz auf das Popupfenster (Zeiger entspricht ID), wird für die Verwendung der Eigenschaften und Methoden benutzt per Zeiger.eigenschaft Zeiger.methode()

logischer_window_name

Zeiger laut open()

Zugriff:

zeiger_auf_popup_fenster.eigenschaft
zeiger_auf_popup_fenster.methode

zeiger_auf_popup_fenster

laut Erzeugung

Beispiel 1:

```
<SCRIPT LANGUAGE="JScript">
// erzeugen
var PopupFenster = window.createPopup(); // windows referenziert das aktuelle und instanzierte Fenster

// Körper erzeugen also body Objekt
var PopupFensterKoerper = PopupFenster.document.body;
```



```
// Körper füllen
PopupFensterKoerper.innerHTML = "Das ist ein Popup-Fenster";

// alles anzeigen
PopupFenster.show(100, 100, 200, 50, document.body);
</SCRIPT>
```

Beispiel 2 für diverse Meldungsfenster per **jeweiligem** Popup-Fenster
(es kann immer nur genau 1 Popup-Fenster angezeigt werden):

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JScript">

    var PopupFensterFeld = new Array();

    // nachfolgende Felder beschreiben die Popup-Fenster und müssen
    // identische Anzahl der Feldelemente haben
    // Feld-Index ist die Nummer des Popup-Fensters

    var PopupFenster_RahmenStyle_Feld = new Array
    (
        "solid black 4px",
        "solid blue 3px",
        "solid green 2px"
    );

    var PopupFenster_HintergrundFarbe_Feld = new Array
    (
        "yellow",
        "gray",
        "orange"
    );

    var PopupFenster_Inhalt_Feld = new Array
    (
        "<B>Inhalt PopupFenster 0<B>",
        "Inhalt PopupFenster 1",
        "<TT>Inhalt PopupFenster 2<TT>"
    );

    var PopupFenster_X_Koordinate_Feld = new Array
    (
        100,
        150,
        200
    );

    var PopupFenster_Y_Koordinate_Feld = new Array
    (
        150,
        200,
        250
    );

    var PopupFenster_Breite_Koordinate_Feld = new Array
    (
        80,
        140,
        200
    );

    var PopupFenster_Hoehe_Koordinate_Feld = new Array
    (
        100,
        140,
        180
    );

    var AnzahlPopupFenster = PopupFenster_Hoehe_Koordinate_Feld.length;

    function PopupFensterInstanzieren(NummerDesPopupFensters, ZeigerAufElternFenster)
        // NummerDesPopupFensters ab 0
    {
```



```

        PopupFensterFeld[NummerDesPopUpFensters] =
            ZeigerAufElternFenster.createPopup();
    }

function PopupFensterFuellen(NummerDesPopUpFensters)
{
    // Body des Popup-Fensters gestalten
    var PopupFenster_Body =
        PopupFensterFeld[NummerDesPopUpFensters].document.body;

    PopupFenster_Body.style.backgroundColor =
        PopupFenster_HintergrundFarbe_Feld[NummerDesPopUpFensters];

    PopupFenster_Body.style.border =
        PopupFenster_RahmenStyle_Feld[NummerDesPopUpFensters];

    PopupFenster_Body.innerHTML =
        PopupFenster_Inhalt_Feld[NummerDesPopUpFensters];
}

function PopupFensterAnzeigen(NummerDesPopUpFensters,
    ObjektZuDemPopUpFensterRelativPositioniertIst
)
{
    // Popup-Fenster anzeigen und damit öffnen
    PopupFensterFeld[NummerDesPopUpFensters].show(
        PopupFenster_X_Koordinate_Feld[NummerDesPopUpFensters],
        PopupFenster_Y_Koordinate_Feld[NummerDesPopUpFensters],
        PopupFenster_Breite_Koordinate_Feld[NummerDesPopUpFensters],
        PopupFenster_Hoeh_Koordinate_Feld[NummerDesPopUpFensters],
        ObjektZuDemPopUpFensterRelativPositioniertIst
    );
}

function PopupFensterSchliessen(NummerDesPopUpFensters)
{
    var Zeiger = PopupFensterFeld[NummerDesPopUpFensters];

    if (Zeiger.isOpen)
    { Zeiger.hide(); }
}

function Init()
{
    // PopupFenster instanzieren im aktuellen Fenster (window)
    for (var i = 0 ; i < AnzahlPopUpFenster; i++)
    {
        PopupFensterInstanzieren(i, window);
        PopupFensterFuellen(i);
    }
}
</SCRIPT>
</HEAD>
<BODY onload="Init();">
    Durch Klick ausserhalb des Popup-Fensters wird dieses auch automatisch geschlossen.
    <BR>
    <INPUT TYPE=button
        VALUE="PopUpFenster 0 anzeigen"
        onclick=" PopupFensterAnzeigen(0, document.body);"
    >
    <INPUT TYPE=button
        VALUE="PopUpFenster 1 anzeigen"
        onclick=" PopupFensterAnzeigen(1, document.body);"
    >
    <INPUT TYPE=button
        VALUE="PopUpFenster 2 anzeigen"
        onclick=" PopupFensterAnzeigen(2, document.body);"
    >
    <BR>
    <INPUT TYPE=button
        VALUE="PopUpFenster 0 schliessen"
        onclick=" PopupFensterSchliessen(0);"
    >
    >

```




```

<INPUT TYPE=button
      VALUE="PopUpFenster 1 schliessen"
      onclick=" PopUpFensterSchliessen(1);"
>
<INPUT TYPE=button
      VALUE="PopUpFenster 2 schliessen"
      onclick=" PopUpFensterSchliessen(2);"
>
</BODY>
</HTML>

```

Hinweis: Es ist aufgrund mangelnder Eigenschaften nicht möglich, Daten von einem Popup-Fenster zu empfangen, die per Eventhandler übergeben werden sollen, der im Popupfenster aufgerufen wird, aber im Dokument kodiert ist, das das Popupfenster erzeugt. Leider existiert keine Eigenschaft .opener. Als Ersatz dient folgendes Beispiel, das ein normales Fenster benutzt:

```

function OnClickHandler()
{
    alert(FensterZeiger.ID_TextArea.value;
    FensterZeiger.close();
}

....

// Fenster öffnen
var FensterZeiger=window.open("", null, "height=250,width=300,status=no,toolbar=no,menubar=no,location=no");
var FensterDokumentZeiger = FensterZeiger.document;

var Kette= '<HTML>'
          + '<HEAD></HEAD>'
          + '<BODY >'
          + '<TEXTAREA ID="ID_TextArea" ROWS="5" COLS="20"></TEXTAREA>'
          + '<BR>'
          + '<INPUT TYPE="button" VALUE="Text an Aufrufer übergeben"'
          + ' onclick="opener.OnClickHandler();"'
          + '>'
          + '</BODY>'
          + '</HTML>';

FensterDokumentZeiger.open("text/html");
FensterDokumentZeiger.write(Kette);
FensterDokumentZeiger.close();

```

Eigenschaften:

.document	Zeiger auf das HTML-Dokument im Popup-Fenster, das per .show() instanziiert wird Es können damit alle Eigenschaften und Methoden des Objektes document referenziert werden siehe Objekt window.popup Syntax: [var Zeiger =] zeiger_auf_popup_fenster.document
.isOpen	nur lesen prüfen ob ein per .createPopup() instanziiertes Popup-Fenster angezeigt wird siehe Objekt window.popup Syntax: [var Wert =] zeiger_auf_popup_fenster.isOpen Wert true, so angezeigt false, so nicht angezeigt
	zeiger_auf_popup_fenster laut Erzeugung per .createPopup()

Methoden:

.hide()	ein angezeigtes Popup-Fenster schliessen Hinweis: Das Popup-Fenster wird automatisch geschlossen, wenn ein anderes Objekt den Focus erhält bzw. aktiv wird, z.B. durch Klick des Users außerhalb des Popupfensters. ändert Wert der Eigenschaft .isOpen siehe Objekt window.popup Syntax: zeiger_auf_popup_fenster.hide()
.show()	liefert nichts ein per .createPopup() instanziiertes Popup-Fenster anzeigen Hinweis: Es kann immer nur genau 1 Popup-Fenster angezeigt werden.
	zeiger_auf_popup_fenster laut Erzeugung per .createPopup()



Das Popup-Fenster wird automatisch geschlossen, wenn ein anderes Objekt den Focus erhält bzw. aktiv wird, z.B. durch Klick des Users außerhalb des Popupfensters
ändert Wert der Eigenschaft .isOpen
siehe Objekt window.popup
Syntax:

```
zeiger_auf_popup_fenster. .show(Wert, Wert2, Wert3, Wert4 [, Zeiger])
```

Wert1	X-Koordinate der linken oberen Fenstecke bezüglich eines Objektes oder des Bildschirms Koordinaten-Ursprung (0,0) des Bildschirms liegt in der linken oberen Ecke Integer, in Pixel, >= 0
Wert2	Y-Koordinate der linken oberen Fenstecke bezüglich eines Objektes oder des Bildschirms Koordinaten-Ursprung (0,0) des Bildschirms liegt in der linken oberen Ecke Integer, in Pixel, >= 0
Wert3	Breite des Fensters in Pixel Integer, > 0
Wert4	Höhe des Fensters in Pixel Integer, > 0
Zeiger	auf Objekt zu dem Wert1 und Wert2 relativ sind Standard: Bildschirm
zeiger_auf_popup_fenster	laut Erzeugung per .createPopup()

liefert nichts

