



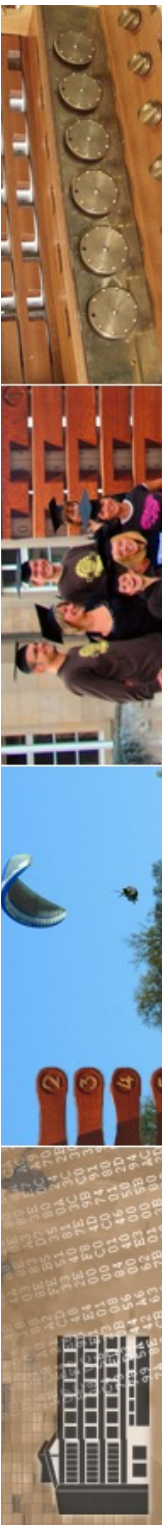
# Grundlagen Internet-Technologien

## INF3171

More JavaScript: Sicherheit und DOM

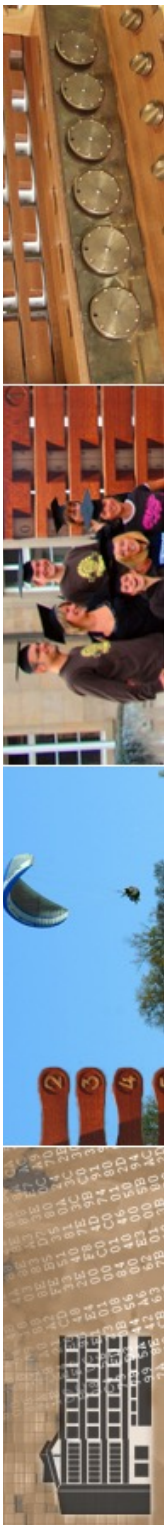
Version 1.0

16.05.2024





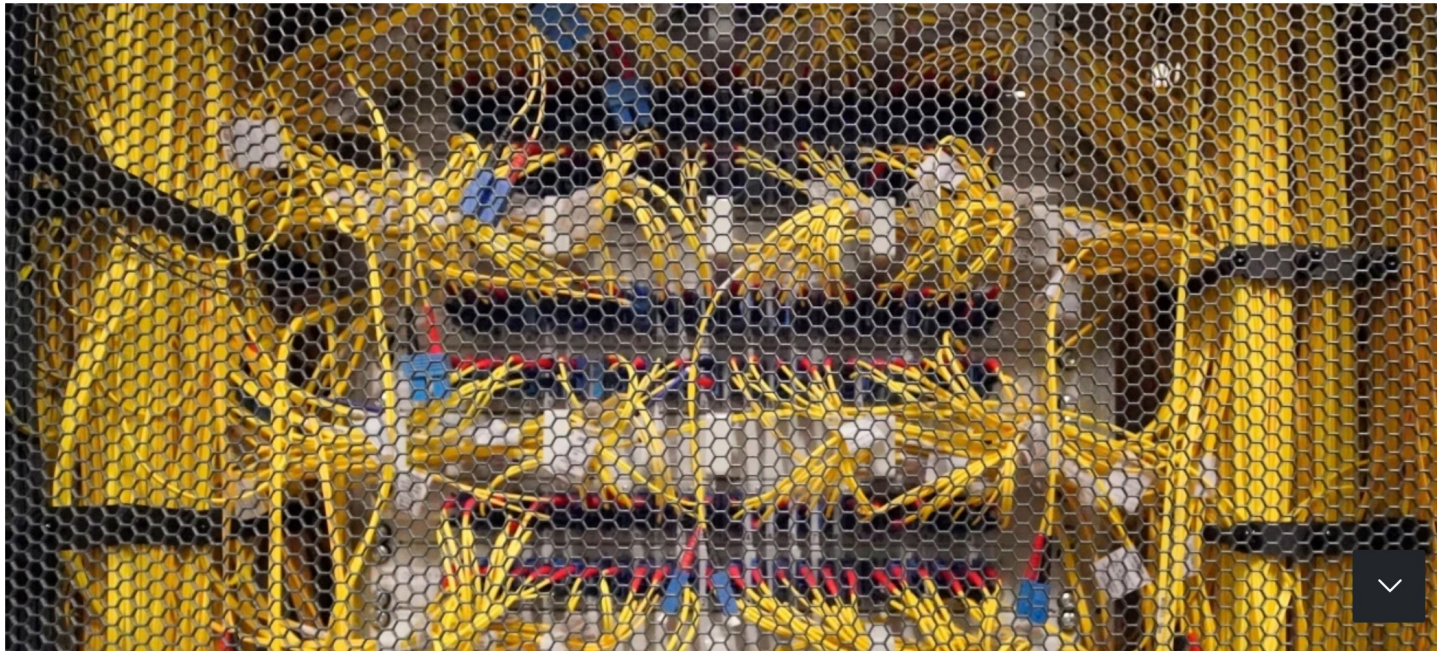
Neuer Datendurchsatz-Rekord an Europas größtem Internetknoten, dem DE-CIX Frankfurt. Parallel zu den Viertelfinal-Rückspielen der UEFA Champions League am 16. April um 21:21 Uhr (MESZ) flossen erstmals über **17 Terabit pro Sekunde (Tbit/s) (genau 17,09 Tbit/s)**. 17.04.2024





# Champions League trägt zu Datenrekord bei

Von Inga Janović 17.04.2024, 13:19 Lesezeit: 2 Min.



**Fußballbegeisterung lässt sich an den Datenmengen, die durch den Frankfurter Internetknoten fließen, ablesen: Dort wurde während der ersten Viertelfinals der Champions League ein neuer Rekord erzielt.**

 Teilen  Merken  Drucken

Fünzfzigmal der Mount Everest, 420 Kilometer: So hoch wäre der Papierstapel, schriebe man all die Informationen auf, die am Dienstag um 21.21 Uhr in nur einer Sekunde durch den Internetknoten in Frankfurt geflossen sind. Das ist ein neuer Rekord an Europas größtem



# JavaScript und Sicherheit

- JavaScript ist aktiver Code, der im Browser ausgeführt wird:  
potentielles Sicherheitsrisiko
- Grundprinzip der Sicherheit von JavaScript:  
**Sandbox**
  - JavaScript-Programm „eingesperrt“ in Sandkasten,  
etwa ohne Zugriff auf das Dateisystem





# Same-Origin

- JavaScript hat zunächst nur Zugriff auf Dokumente vom *gleicher* Web-Quelle
  - Ausnahmen gibt es, etwa bei Ajax
  - bei Subdomains ggf. Ausnahme in der Art `document.domain = "informatik.uni-tuebingen.de"`





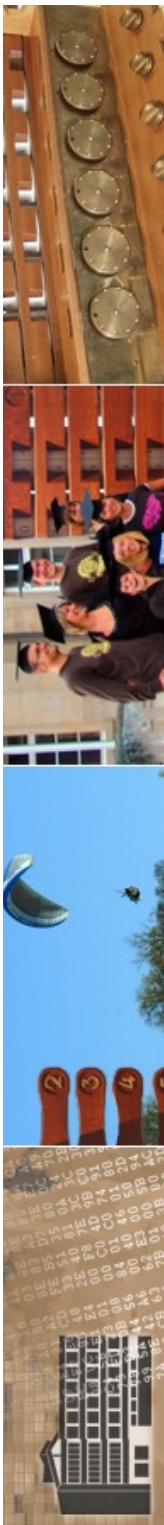
# JavaScript und Browser-Erscheinung

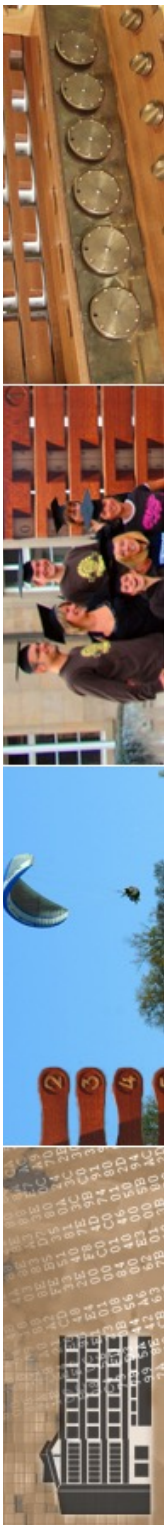
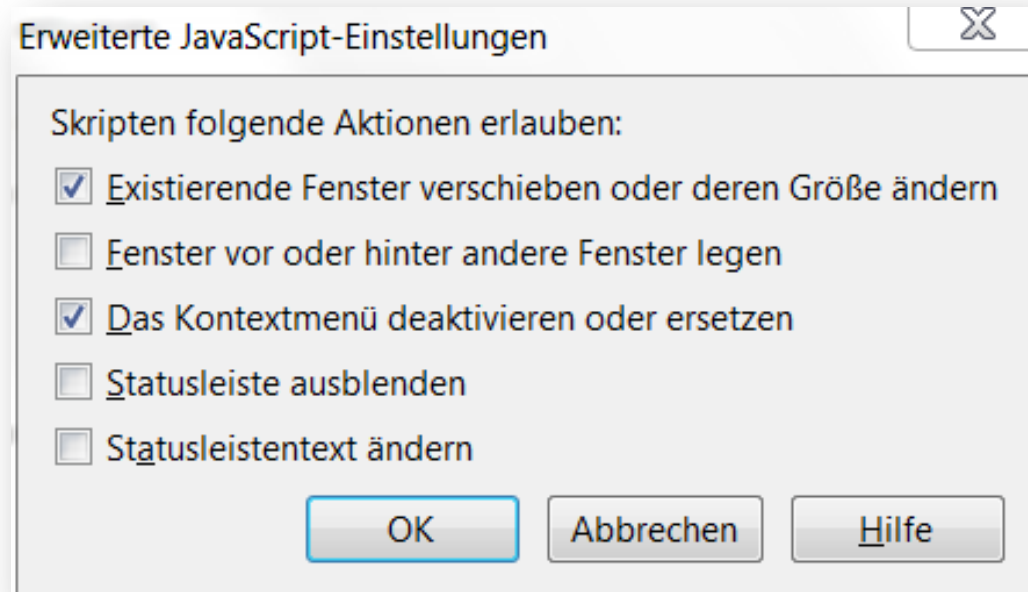
- von Anfang an kritisch:

## JavaScript kann Browser-Erscheinung verändern

Gefahr, dass der Nutzer getäuscht wird, etwa durch überschriebenen Titel/Statuszeile

Browser schränken deshalb diese Möglichkeiten stark, teilweise konfigurierbar, ein







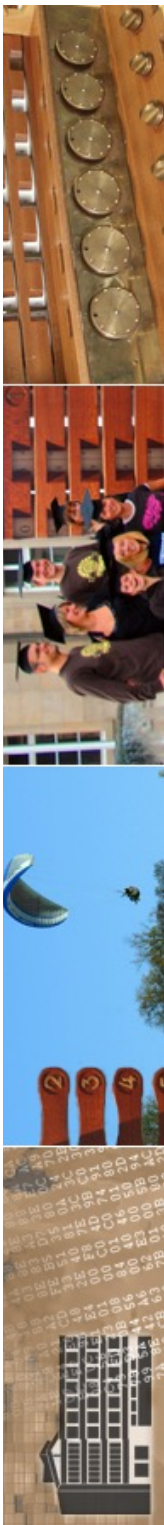
# destruktives JavaScript

- JavaScript kann PopUps öffnen - damit kann man ggf. den Nutzer „fluten“

einfaches

`window.open(...)`

wird deshalb von Popup-Blocker blockiert







# XSS

- Cross-Site-Scripting: Einbinden von fremden (schädlichen) JavaScript in eine andere, an sich harmlose und vertrauenswürdige Seite

**Cross-Site Scripting (XSS)**, engl. für *Webseitenübergreifendes Skripting*) ist das Einschleusen von fremden, möglicherweise schädlichen JavaScripten in eine Website. Es handelt sich weniger um ein Sicherheitsproblem innerhalb von JavaScript, sondern um eine Sicherheitslücke in fehlerhaften Webanwendungen, die Daten aus nicht vertrauenswürdigen Quellen (z.B. aus Formulareingaben oder HTTP-Parametern) ungefiltert ins HTML einbauen.

- [https://wiki.selfhtml.org/wiki/JavaScript/Tutorials/Cross\\_Site\\_Scripting](https://wiki.selfhtml.org/wiki/JavaScript/Tutorials/Cross_Site_Scripting)
- Cookie-Diebstahl
- Keylogging
- Phishing





# XSS

- Um XSS zu verhindern, muss verhindert werden, dass eingegebene Daten Skripte enthalten können
  - das ist teilweise schwieriger als es zunächst erscheint





# ganz alte Browser

Windows XP

google chrome download - Bing

Datei Bearbeiten Ansicht Favoriten Extras ?

Zurück Zurück Suchen Favoriten Medien

Adresse <http://www.bing.com/search?srch=106&FORM=AS6&q=google+chrome+download> Wechseln zu

ALLE BILDER VIDEOS NEWS

Anmelden Prämien

Bing google chrome download

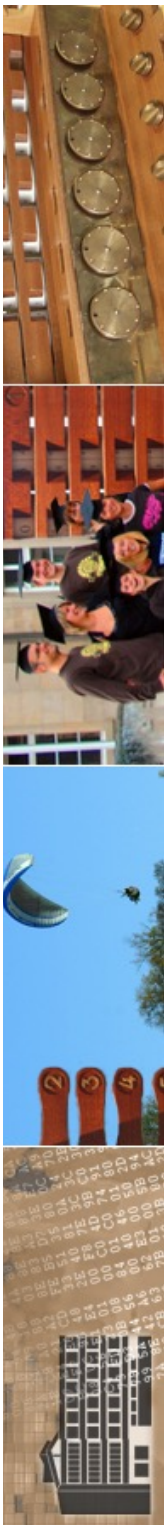
Ihr Browser ist veraltet und funktioniert mit bestimmten Websites möglicherweise nicht.

Sie haben nicht den neuesten von Microsoft empfohle  
Laden Sie den neuen Microsoft Edge herunter, um das Beste aus dem Web herauszuholen.

Jetzt herunterladen

Google Chrome – den schnellen und sicheren Browser von ...

©2024 Tübing Start Server nich... update - Bing Netzwerkve... google chro... Shell DE 11:24





# Verbindung JavaScript und HTML

- bisher: JavaScript “nette Scriptsprache“ ,  
die der Browser interpretiert
- nun:
  - interessant, weil mit HTML verbindbar
  - **Erweiterung** von HTML/CSS

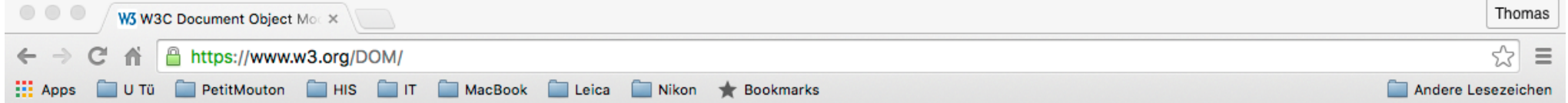




# DOM

- **DOM: Document Object Model**
  - <http://www.w3.org/DOM/>
  - <https://dom.spec.whatwg.org/>
- **Modell, um auf Webdokumente zuzugreifen**
  - zentral: die zugrundeliegende Baumstruktur
  - *The Document Object Model is a platform- and language-neutral interface that will allow programs and scripts to dynamically access and update the content, structure and style of documents. The document can be further processed and the results of that processing can be incorporated back into the presented page. This is an overview of DOM-related materials here at W3C and around the web.*





About DOM . [DOM Activity statement](#)  
[Technical Reports](#) . [Technical Materials](#)  
[Test Suites](#) . [FAQ](#) . [Mailing List](#)  
 Members only resource: [DOM Interest Group](#)

# Document Object Model (DOM)

## Table of contents

1. [What's new?](#)
2. [What is the Document Object Model?](#)
3. [Why the Document Object Model?](#)
4. [W3C Activity Statement on the Document Object Model](#)
5. [Public Release of Specifications](#)
6. [Questions, comments, and suggestions about the DOM](#)
7. [DOM Conformance Test Suites](#)
8. [Related Resources](#)

## What's new?

- ▶ **20090106:** The WebApps WG Drives DOM Specifications. The [W3C Web Applications Working Group](#) has taken over responsibility for the Document Object Model specifications, including a new revision of DOM Level 3 Events, a new DOM Core specification, and potentially any errata on older DOM specifications. Discussion can be directed to either the [public-webapps@w3.org](mailto:public-webapps@w3.org) or the [www-dom@w3.org](mailto:www-dom@w3.org) mailing lists.
- ▶ **20080122:** The Document Object Model Activity is closed. The Document Object Model Working Group was closed in the Spring of 2004, after the completion of the DOM Level 3 Recommendations. Several W3C Working Groups have since taken the lead in maintaining and continuing to develop standard APIs for the Web since then; HTML, SVG, CSS, or WebAPI being among them. W3C will continue to develop APIs in various Working Groups. Learn more about the achievements from the [DOM Activity page](#). Got an idea on how to use this page? Send an email to [Philippe](mailto:Philippe).

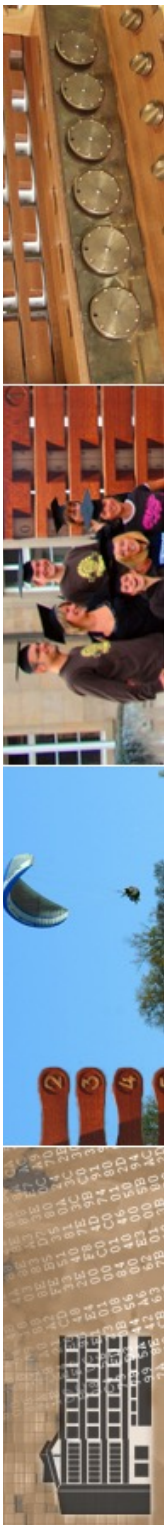
W3C's DOM news are also available as a [RSS feed](#) .

## What is the Document Object Model?

The Document Object Model is a platform- and language-neutral interface that will allow programs and scripts to dynamically access and update the content, structure and style of documents. The document can be further processed and the results of that processing can be incorporated back into the presented page. This is an overview of DOM-related materials here at W3C and around the web.

## Why the Document Object Model?

"Dynamic HTML" is a term used by some vendors to describe the combination of HTML, style sheets and scripts that allows documents to be



# DOM

Living Standard — Last Updated 31 March 2023



## Participate:

[GitHub whatwg/dom](#) (new issue, open issues)

[Chat on Matrix](#)

## Commits:

[GitHub whatwg/dom/commits](#)

[Snapshot as of this commit](#)

[@thedomstandard](#)

## Tests:

[web-platform-tests dom/](#) (ongoing work)

## Translations (non-normative):

[日本語](#)

## Abstract

DOM defines a platform-neutral model for events, aborting activities, and node trees.

## Table of Contents

- 1 Infrastructure
  - 1.1 Trees
  - 1.2 Ordered sets
  - 1.3 Selectors
  - 1.4 Namespaces
- 2 Events





# What is the Document Object Model?

*Editors*

Jonathan Robie, Texcel Research

## Introduction

The Document Object Model (DOM) is a programming API for HTML and XML documents. It defines the logical structure of documents and the way a document is accessed and manipulated. In the DOM specification, the term "document" is used in the broad sense - increasingly, XML is being used as a way of representing many different kinds of information that may be stored in diverse systems, and much of this would traditionally be seen as data rather than as documents. Nevertheless, XML presents this data as documents, and the DOM may be used to manage this data.

With the Document Object Model, programmers can create and build documents, navigate their structure, and add, modify, or delete elements and content. Anything found in an HTML or XML document can be accessed, changed, deleted, or added using the Document Object Model, with a few exceptions - in particular, the DOM interfaces for the internal subset and external subset have not yet been specified.

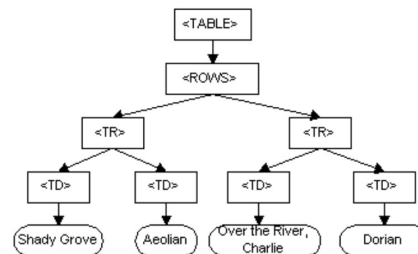
As a W3C specification, one important objective for the Document Object Model is to provide a standard programming interface that can be used in a wide variety of environments and applications. The Document Object Model can be used with any programming language. In order to provide precise, language-independent specification of the Document Object Model interfaces, we have chosen to define the specifications in OMG IDL, as defined in the [CORBA 2.2 specification](#). In addition to the OMG IDL specification, we provide language bindings for Java and ECMAScript (an industry-standard scripting language based on JavaScript and JScript). *Note: OMG IDL is used only as a language-independent and implementation-neutral way to specify interfaces. Various other IDLs could have been used; the use of OMG IDL does not imply a requirement to use a specific object binding runtime.*

## What the Document Object Model is

The Document Object Model is a programming API for documents. The object model itself closely resembles the structure of the documents it models. For instance, consider this table, taken from an HTML document:

```
<TABLE>
<ROWS>
<TR>
<TD>Shady Grove</TD>
<TD>Aeolian</TD>
</TR>
<TR>
<TD>Over the River, Charlie</TD>
<TD>Dorian</TD>
</TR>
</ROWS>
</TABLE>
```

The Document Object Model represents this table like this:





## Einführung in das Document Object Model (DOM)

**Autor:** U. Altmann, Institut für Medizinische Informatik, Justus-Liebig-Universität Gießen  
Heinrich-Buff-Ring 44, 35392 Gießen  
e-Mail: [Udo.Altmann@informatik.med.uni-giessen.de](mailto:Udo.Altmann@informatik.med.uni-giessen.de)

**Quellenhinweis:** Mit dieser Einführung soll ein Überblick über die vorhandenen Standards im Bereich DOM sowie deren Anwendung gegeben werden. Die Textpassagen sind teilweise Übersetzungen aus den W3C-Standards. Aus diesen wurden auch die Abbildungen entnommen. Die Standards wurden, soweit sie in der Einführung benötigt werden als Kopie heruntergeladen und werden lokal referenziert. Verbindlich sind in jedem Fall die englische Originale unter <http://www.w3.org/DOM>

### Was ist das DOM

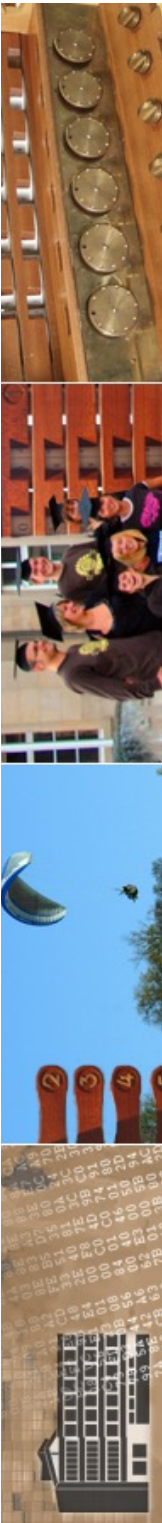
- Application Programming Interface (API) für HTML **und** XML
- definiert die **logische** Struktur von Dokumenten
- Begriff "Dokument" ist im Standard weit gefaßt (d.h. vom "physikalischen" Vorhandensein eines XML-Dokuments unabhängig => XML als Art und Weise, Daten zu präsentieren).
- Alles was an "Struktur" (Text und Markup) in XML/HTML-Dokumenten gefunden werden kann, kann zugegriffen, verändert, gelöscht oder hinzugefügt werden (mit wenigen Ausnahmen).
- Anwendbarkeit in einer Reihe von Programmiersprachen
  - Definition in CORBA IDL
  - Bindung für Java und ECMA-Script (Java/JScript) im Standard enthalten
  - weitere außerhalb des Standards verfügbar (z.B. C++, Perl)





# DOM

- konkret: Objekte für die Interaktion  
JavaScript ↔ HTML
- drei Wurzeln (drei Teilbäume):
  - window: aktuelle Browserfenster
  - screen: Bildschirm
  - navigator: Browser
    - darunter jeweils weitere Hierarchie





# warum „navigator“?

Wikipedia | Die freie Enzyklopädie - Netscape

Datei Bearbeiten Ansicht Gehe Communicator Hilfe

Zurück Vor Neu laden Anfang Suchen Guide Drucken Sicherheit Shop Stop

Lesezeichen Adresse: <http://de.wikipedia.org/wiki/Wikipedia:Hauptseite> Verwandte Objekte

Instant Message Netscape.de Net2Phone Focus Online

Ansichten

- Projektseite

**Wikipedia:Hauptseite**

aus Wikipedia, der freien Enzyklopädie

Personen Wechseln zu: [Navigation](#), [Suche](#)

**Willkommen bei Wikipedia**

[Wikipedia](#) ist ein Projekt zum Aufbau einer Enzyklopädie, in der alle Menschen Wissen beitragen. Seit Mai 2001 sind so 801.583 Artikel [willkommen](#).

Suche

Artikel

Volltext

Nav

63% von 888

Start Wikipedia | Die freie ...



Version	Freigabedatum
Mosaic Netscape 0.9	13. Oktober 1994
Netscape Navigator 1.0	15. Dezember 1994
Netscape Navigator 1.1	März 1995
Netscape Navigator 1.22	August 1995
Netscape Navigator 2.0	März 1996 <sup>[8][9]</sup>
Netscape Navigator 3.0	19. August 1996
Netscape Navigator 3.04	4. Oktober 1996
Netscape Navigator 4.0	Juni 1997
Netscape Navigator 4.06	17. August 1998
Netscape Navigator 4.08	9. November 1998
Netscape Navigator 9.0.0.6	21. Februar 2008

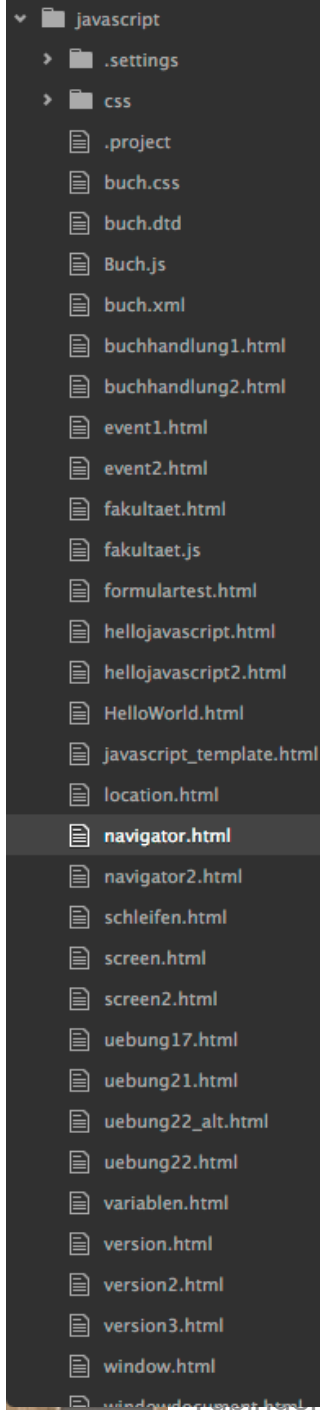




# das navigator-Objekt

- ...dient der Verwaltung des Browsers
- Attribute/Methoden:
  - `appName`
  - `appCodeName`
  - `appVersion`
  - `userAgent`
  - `platform`
  - `plugins`
  - `mimeTypes`
  - `language`
  - `javaEnabled()`
- Beispiel `navigator.html` und Browserweiche





```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <link rel="stylesheet" type="text/css" href="./css/webkompendium.css">
5     <link rel="shortcut icon" href="./css/favicon.ico">
6     <title>Grundlagen Internet-Technologien: JavaScript</title>
7   </head>
8   <body>
9     <hr /><center>
10    <h2>Grundlagen Internet-Technologien</h2>
11    <h3>JavaScript: das navigator-Objekt</h3><h4>
12
13    <script type="text/javascript">
14
15      document.write ("verwendeter Browser: ");
16      document.write (navigator.appName + " - " + navigator.appCodeName);
17
18      document.write ("<br>Version: ");
19      document.write (navigator.appVersion);
20
21      document.write ("<br><br>Meldet sich offiziell als:<br>");
22      document.write (navigator.userAgent);
23
24    </script>
25
26    </h4></center><hr />
27  </body>
28 </html>
```



Grundlagen Internet-Technologien

**JavaScript: das navigator-Objekt**

verwendeter Browser: Netscape - Mozilla  
Version: 5.0 (Windows; U; Windows NT 6.1; en-US)  
AppleWebKit/532.5 (KHTML, like Gecko)  
Chrome/4.1.249.1064 Safari/532.5

Meldet sich offiziell als:  
Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US)  
AppleWebKit/532.5 (KHTML, like Gecko)  
Chrome/4.1.249.1064 Safari/532.5

Grundlagen Internet-Technologien

**JavaScript: das navigator-Objekt**

verwendeter Browser: Opera - Mozilla  
Version: 9.80 (Windows NT 6.1; U; de)

Meldet sich offiziell als:  
Opera/9.80 (Windows NT 6.1; U; de) Presto/2.5.24  
Version/10.52





Grundlagen Internet-Technologien +

134.2.6.167/~zrvwa01/vorlesung/javascript/navigator.html

# Grundlagen Internet-Technologien

## JavaScript: das navigator-Objekt

**verwendeter Browser: Netscape - Mozilla**

**Version: 5.0 (Macintosh; Intel Mac OS X 11\_3\_1) AppleWebKit/537.36 (KHTML, like Gecko)  
Chrome/89.0.4389.128 Safari/537.36 OPR/75.0.3969.218**

**Meldet sich offiziell als:**

**Mozilla/5.0 (Macintosh; Intel Mac OS X 11\_3\_1) AppleWebKit/537.36 (KHTML, like Gecko)  
Chrome/89.0.4389.128 Safari/537.36 OPR/75.0.3969.218**



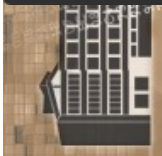


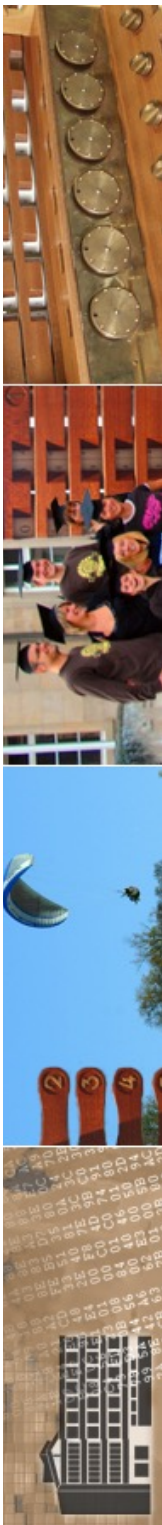


```

navigator2.html - /Users/thomas/data/www/webst1/javascript - Atom
version... x version... x version... x window... x window... x window... x navigat... x navigator2.html x variabl... x schleife... x helloja... x HelloW... x
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <link rel="stylesheet" type="text/css" href="./css/webkompendium.css">
5 <link rel="shortcut icon" href="./css/favicon.ico">
6 <title>Grundlagen Internet-Technologien: JavaScript</title>
7 </head>
8 <body>
9
10 <hr /><center>
11 <h2>Grundlagen Internet-Technologien</h2>
12 <h3>JavaScript: das navigator-Objekt, Teil 2</h3>
13 </center>
14
15 <script type="text/javascript">
16 for (eigenschaft in navigator)
17 document.write("<b>" + eigenschaft + ":</b> " + navigator[eigenschaft] + "<br />");
18
19 document.write("<b>javaEnabled():</b> " + navigator.javaEnabled() + "<br />");
20 </script>
21
22 <hr />
23
24 </body>
25 </html>
26
navigator2.html 1:1
CRLF ISO 8859-15 HTML 1 update

```





Grundlagen Internet- x

134.2.2.38/~zrvwa01/vorlesung/javascript/navigator2.html

Apps M240 D800 HIS U Tü Zabbix Support für SVP... Benutzerhandb... Sony : SVZ1311... Greg's Cable M...

# Grundlagen Internet-Technologien

## JavaScript: das navigator-Objekt, Teil 2

**serviceWorker:** [object ServiceWorkerContainer]  
**webkitPersistentStorage:** [object DeprecatedStorageQuota]  
**webkitTemporaryStorage:** [object DeprecatedStorageQuota]  
**geolocation:** [object Geolocation]  
**doNotTrack:** 1  
**onLine:** true  
**languages:** de-DE,de,en-US,en  
**language:** de  
**userAgent:** Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/42.0.2311.135 Safari/537.36  
**product:** Gecko  
**platform:** Win32  
**appVersion:** 5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/42.0.2311.135 Safari/537.36  
**appName:** Netscape  
**appCodeName:** Mozilla  
**hardwareConcurrency:** 4  
**maxTouchPoints:** 10  
**vendorSub:**  
**vendor:** Google Inc.  
**productSub:** 20030107  
**cookieEnabled:** true  
**mimeTypes:** [object MimeTypeArray]  
**plugins:** [object PluginArray]  
**javaEnabled:** function javaEnabled() { [native code] }  
**getStorageUpdates:** function getStorageUpdates() { [native code] }  
**getGamepads:** function getGamepads() { [native code] }  
**webkitGetUserMedia:** function webkitGetUserMedia() { [native code] }  
**vibrate:** function vibrate() { [native code] }  
**getBattery:** function getBattery() { [native code] }  
**sendBeacon:** function sendBeacon() { [native code] }  
**requestMediaKeySystemAccess:** function requestMediaKeySystemAccess() { [native code] }  
**registerProtocolHandler:** function registerProtocolHandler() { [native code] }  
**unregisterProtocolHandler:** function unregisterProtocolHandler() { [native code] }  
**javaEnabled():** true



Opera

Grundlagen Internet-Techn x +

134.2.2.38/~zrvwa01/vorlesung/javascript/navigator2.html

## Grundlagen Internet-Technologien

### JavaScript: das navigator-Objekt, Teil 2

**serviceWorker:** [object ServiceWorkerContainer]  
**webkitPersistentStorage:** [object DeprecatedStorageQuota]  
**webkitTemporaryStorage:** [object DeprecatedStorageQuota]  
**geolocation:** [object Geolocation]  
**doNotTrack:** null  
**onLine:** true  
**languages:** de-DE,de,en-US,en  
**language:** de  
**userAgent:** Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/42.0.2311.90 Safari/537.36 OPR/29.0.1795.47 (Edition Campaign 21)  
**product:** Gecko  
**platform:** Win32  
**appVersion:** 5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/42.0.2311.90 Safari/537.36 OPR/29.0.1795.47 (Edition Campaign 21)  
**appName:** Netscape  
**appCodeName:** Mozilla  
**hardwareConcurrency:** 4  
**maxTouchPoints:** 10  
**vendorSub:**  
**vendor:** Google Inc.  
**productSub:** 20030107  
**cookieEnabled:** true  
**mimeType:** [object MimeTypeArray]  
**plugins:** [object PluginArray]  
**javaEnabled:** function javaEnabled() { [native code] }  
**getStorageUpdates:** function getStorageUpdates() { [native code] }  
**getGamepads:** function getGamepads() { [native code] }  
**webkitGetUserMedia:** function webkitGetUserMedia() { [native code] }  
**vibrate:** function vibrate() { [native code] }  
**getBattery:** function getBattery() { [native code] }  
**sendBeacon:** function sendBeacon() { [native code] }  
**requestMediaKeySystemAccess:** function requestMediaKeySystemAccess() { [native code] }  
**registerProtocolHandler:** function registerProtocolHandler() { [native code] }  
**unregisterProtocolHandler:** function unregisterProtocolHandler() { [native code] }  
**javaEnabled():** true



http://134.2.2.38/~zrvwa01 Grundlagen Internet-Techn... x



# Grundlagen Internet-Technologien

## JavaScript: das navigator-Objekt, Teil 2

```
appName: Mozilla
appMinorVersion: 0
browserLanguage: de-DE
cookieEnabled: true
cpuClass: x86
language: de-DE
mimeTypes: [object MimeTypeArray]
plugins: [object PluginArray]
systemLanguage: de-DE
userLanguage: de-DE
maxTouchPoints: 10
msManipulationViewsEnabled: true
msMaxTouchPoints: 10
msPointerEnabled: true
pointerEnabled: true
webdriver: false
geolocation: [object Geolocation]
appName: Netscape
appVersion: 5.0 (Windows NT 6.3; WOW64; Trident/7.0; Touch; .NET4.0E; .NET4.0C; .NET CLR 3.5.30729; .NET CLR 2.0.50727; .NET CLR 3.0.30729; Tablet PC 2.0; InfoPath.3; rv:11.0) like Gecko
platform: Win32
product: Gecko
userAgent: Mozilla/5.0 (Windows NT 6.3; WOW64; Trident/7.0; Touch; .NET4.0E; .NET4.0C; .NET CLR 3.5.30729; .NET CLR 2.0.50727; .NET CLR 3.0.30729; Tablet PC 2.0; InfoPath.3; rv:11.0) like Gecko
vendor:
onLine: true
msSaveBlob: function msSaveBlob() { [native code] }
msSaveOrOpenBlob: function msSaveOrOpenBlob() { [native code] }
confirmSiteSpecificTrackingException: function confirmSiteSpecificTrackingException() { [native code] }
confirmWebWideTrackingException: function confirmWebWideTrackingException() { [native code] }
removeSiteSpecificTrackingException: function removeSiteSpecificTrackingException() { [native code] }
removeWebWideTrackingException: function removeWebWideTrackingException() { [native code] }
storeSiteSpecificTrackingException: function storeSiteSpecificTrackingException() { [native code] }
storeWebWideTrackingException: function storeWebWideTrackingException() { [native code] }
javaEnabled: function javaEnabled() { [native code] }
taintEnabled: function taintEnabled() { [native code] }
msLaunchUri: function msLaunchUri() { [native code] }
javaEnabled(): true
```



# Grundlagen Internet-Technologien

## JavaScript: das navigator-Objekt, Teil 2

**vendorSub:**  
**productSub:** 20030107  
**vendor:** Google Inc.  
**maxTouchPoints:** 0  
**hardwareConcurrency:** 8  
**cookieEnabled:** true  
**appCodeName:** Mozilla  
**appName:** Netscape  
**appVersion:** 5.0 (Macintosh; Intel Mac OS X 10\_15\_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.129 Safari/537.36 OPR/68.0.3618.63  
**platform:** MacIntel  
**product:** Gecko  
**userAgent:** Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_15\_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.129 Safari/537.36 OPR/68.0.3618.63  
**language:** de-DE  
**languages:** de-DE,de,en-US,en  
**onLine:** true  
**doNotTrack:** null  
**geolocation:** [object Geolocation]  
**mediaCapabilities:** [object MediaCapabilities]  
**connection:** [object NetworkInformation]  
**plugins:** [object PluginArray]  
**mimeType:** [object MimeTypeArray]  
**webkitTemporaryStorage:** [object DeprecatedStorageQuota]  
**webkitPersistentStorage:** [object DeprecatedStorageQuota]  
**getBattery:** function getBattery() { [native code] }  
**sendBeacon:** function sendBeacon() { [native code] }  
**getGamepads:** function getGamepads() { [native code] }  
**javaEnabled:** function javaEnabled() { [native code] }  
**vibrate:** function vibrate() { [native code] }  
**userActivation:** [object UserActivation]  
**mediaSession:** [object MediaSession]  
**permissions:** [object Permissions]  
**javaEnabled():** false



## Grundlagen Internet-Technologien

### JavaScript: das navigator-Objekt, Teil 2

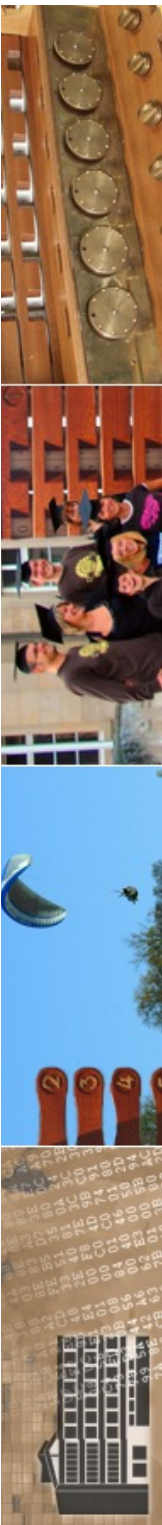
**vendorSub:**  
**productSub:** 20030107  
**vendor:** Google Inc.  
**maxTouchPoints:** 0  
**scheduling:** [object Scheduling]  
**userActivation:** [object UserActivation]  
**brave:** [object Brave]  
**globalPrivacyControl:** true  
**doNotTrack:** null  
**geolocation:** [object Geolocation]  
**plugins:** [object PluginArray]  
**mimeTypes:** [object MimeTypeArray]  
**pdfViewerEnabled:** true  
**webkitTemporaryStorage:** [object DeprecatedStorageQuota]  
**webkitPersistentStorage:** [object DeprecatedStorageQuota]  
**hardwareConcurrency:** 10  
**cookieEnabled:** true  
**appCodeName:** Mozilla  
**appName:** Netscape  
**appVersion:** 5.0 (Macintosh; Intel Mac OS X 10\_15\_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36  
**platform:** MacIntel  
**product:** Gecko  
**userAgent:** Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_15\_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36  
**language:** de-DE  
**languages:** de-DE,de,en-US,en  
**onLine:** true  
**webdriver:** false  
**getGamepads:** function getGamepads() { [native code] }  
**javaEnabled:** function javaEnabled() { [native code] }  
**sendBeacon:** function sendBeacon() { [native code] }  
**vibrate:** function vibrate() { [native code] }  
**ink:** [object Ink]  
**mediaCapabilities:** [object MediaCapabilities]  
**mediaSession:** [object MediaSession]  
**permissions:** [object Permissions]  
**windowControlsOverlay:** [object WindowControlsOverlay]  
**javaEnabled():** false

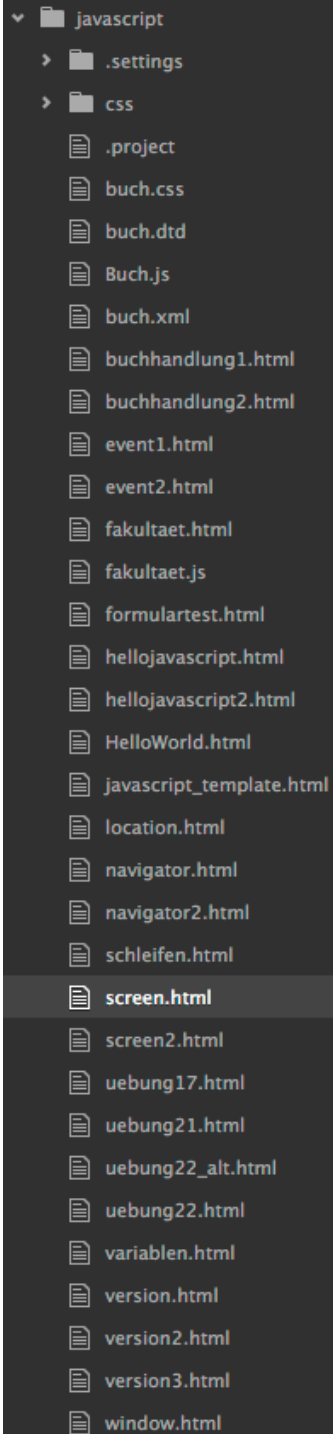




# das screen-Objekt

- Informationen zum **Bildschirm**
- Attribute:
  - `availableHeight`
  - `availableWidth`
  - `colorDepth`
- Beispiel `screen.html`
- Vorsicht bei *mehreren* Bildschirmen





```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
5     <link rel="stylesheet" type="text/css" href="./css/webkompendium.css">
6     <link rel="shortcut icon" href="./css/favicon.ico">
7     <title>Grundlagen Internet-Technologien: JavaScript</title>
8   </head>
9   <body>
10
11     <hr /><center>
12       <h2>Grundlagen Internet-Technologien</h2>
13       <h3>JavaScript: das screen-Objekt</h3><h4>
14
15         <script type="text/javascript">
16
17           document.write ("Bildschirmaufl&ouml;sung: ");
18           document.write (screen.availWidth + " x " + screen.availHeight);
19
20           document.write ("<br>Farbtiefe: ");
21           document.write (screen.colorDepth+" bit");
22
23         </script>
24
25       </h4></center><hr />
26
27     </body>
28 </html>
```





Grundlagen Internet-Techn... x Thomas

134.2.2.38/~zrvwa01/vorlesung/javascript/screen.html

Apps U Tü PetitMouton HIS IT » Andere Lesezeichen

# Grundlagen Internet-Technologien

## JavaScript: das screen-Objekt

Bildschirmauflösung: 1920 x 1128  
Farbtiefe: 24 bit

Informationsdienste - AI... x Grundlagen Internet-Tec... x

127.0.0.1/webst1/javascript/scree

Google

# Grundlagen Internet-Technologien

## JavaScript: das screen-Objekt

Bildschirmauflösung: 1536 x 922  
Farbtiefe: 24 bit

Grundlagen Internet... x Google

http://134.2.2.38/~zrvwa01,

# Grundlagen Internet-Technologien

## JavaScript: das screen-Objekt

Bildschirmauflösung: 1920 x 1152  
Farbtiefe: 32 bit



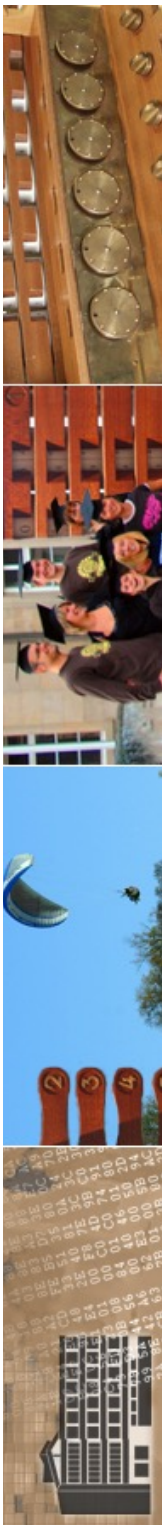


◀ ▶ ↻ 🔖 ⚠ Nicht sicher | 134.2.6.167/~... 🔍 📄 🏠 📁 VPN Aktualisieren ☰

# Grundlagen Internet-Technologien

## JavaScript: das screen-Objekt

**Bildschirmauflösung: 3200 x 1723**  
**Farbtiefe: 24 bit**





```

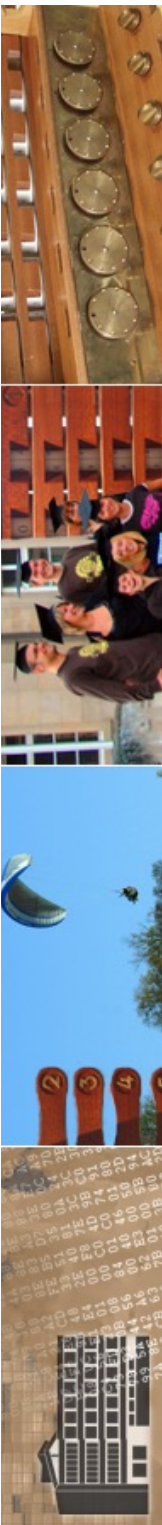
screen2.html - /Users/thomas/data/www/webst1/javascript - Atom
screen2.html x
versi... x
versi... x
versi... x
wind... x
wind... x
wind... x
navi... x
navi... x
scre... x
screen2.html x
varia... x
schl... x
hell... x
Hell... x

1  <!DOCTYPE html>
2  <html>
3    <head>
4      <link rel="stylesheet" type="text/css" href="./css/webkompendium.css">
5      <link rel="shortcut icon" href="./css/favicon.ico">
6      <title>Grundlagen Internet-Technologien: JavaScript</title>
7    </head>
8    <body>
9
10     <hr /><center>
11       <h2>Grundlagen Internet-Technologien</h2>
12       <h3>JavaScript: das screen-Objekt, Teil 2</h3>
13     </center>
14
15     <script type="text/javascript">
16       for (eigenschaft in screen)
17         document.write("<b>" + eigenschaft + " :</b> " + screen[eigenschaft] + "<br />");
18     </script>
19
20     <hr />
21
22   </body>
23 </html>
24

screen2.html 2:1 (2, 17)
CRLF ISO 8859-15 HTML 1 update

```







Informationsdienste - Allg... x Grundlagen Internet-Techn... x +

127.0.0.1/webst1/javascript/screen2.html Google

# Grundlagen Internet-Technologien

## JavaScript: das screen-Objekt, Teil 2

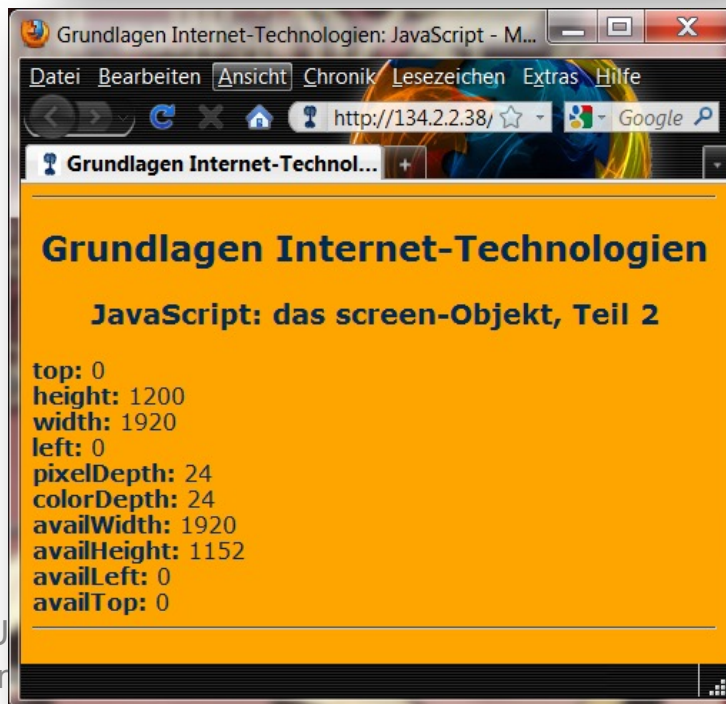
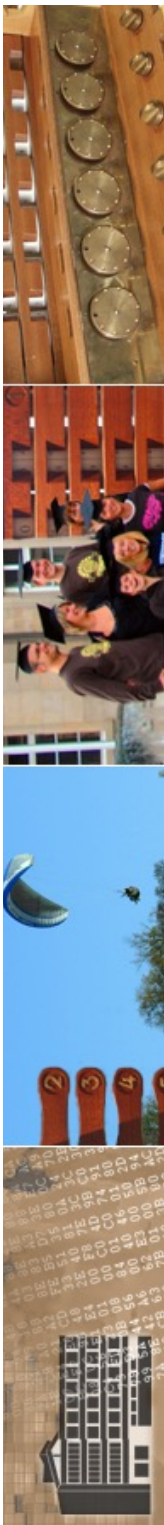
```

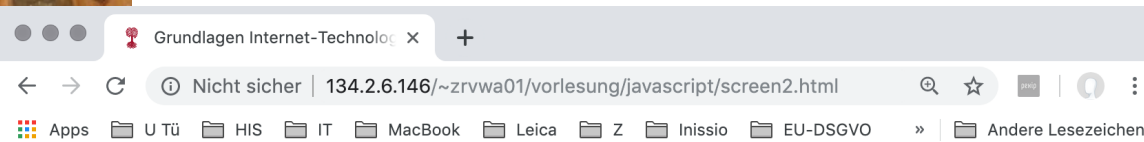
mozLockOrientation: function mozLockOrientation() { [native code] }
mozUnlockOrientation: function mozUnlockOrientation() { [native code] }
availWidth: 1536
availHeight: 922
width: 1536
height: 960
colorDepth: 24
pixelDepth: 24
top: 0
left: 0
availTop: 0
availLeft: 0
mozOrientation: landscape-primary
onmozorientationchange: null
addEventListener: function addEventListener() { [native code] }
removeEventListener: function removeEventListener() { [native code] }
dispatchEvent: function dispatchEvent() { [native code] }
    
```





# so schön ist JavaScript...





# Grundlagen Internet-Technologien

## JavaScript: das screen-Objekt, Teil 2

**availWidth:** 3008  
**availHeight:** 1623  
**width:** 3008  
**height:** 1692  
**colorDepth:** 24  
**pixelDepth:** 24  
**availLeft:** 0  
**availTop:** 23  
**orientation:** [object ScreenOrientation]



# Grundlagen Internet-Technologien

## JavaScript: das screen-Objekt, Teil 2

**availWidth:** 3008  
**availHeight:** 1623  
**width:** 3008  
**height:** 1692  
**colorDepth:** 24  
**pixelDepth:** 24  
**availLeft:** 0  
**availTop:** 23  
**orientation:** [object ScreenOrientation]





# Grundlagen Internet-Technologien

## JavaScript: das screen-Objekt, Teil 2

**availWidth:** 3200  
**availHeight:** 1723  
**width:** 3200  
**height:** 1800  
**colorDepth:** 24  
**pixelDepth:** 24  
**availLeft:** 0  
**availTop:** 25  
**orientation:** [object ScreenOrientation]  
**addEventListener:** function addEventListener() { [native code] }  
**dispatchEvent:** function dispatchEvent() { [native code] }  
**removeEventListener:** function removeEventListener() { [native code] }







# Baumstruktur



screen

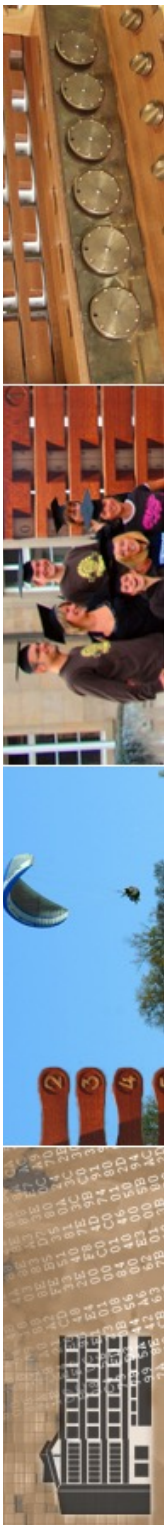
- availHeight
- availLeft
- availTop
- availWid
- colorDej
- height
- pixelDepth
- width



navigator

- appCodeName
- appName
- appVersion
- userAgent
- platform
- javaEnabled
- plugins
- mimeTypes

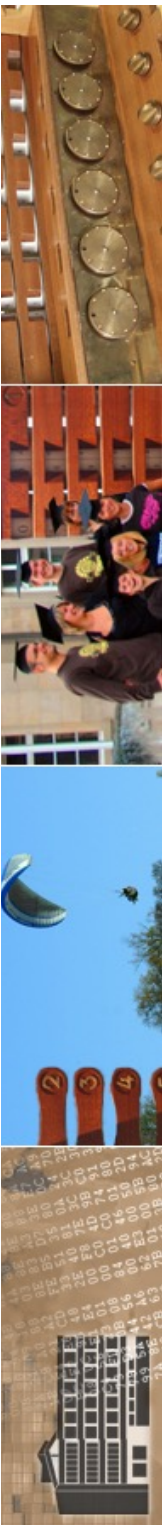






# das window-Objekt

- Spitze der **Dokumentenhierarchie**
  - komplex
  - wichtige Untergliederungen: document, frames, event, history, location
  - window:
    - alert(...), blur(), close(), confirm(...), defaultStatus(...), fokus(), open(url,name, position), status
  - Beispiel window.html
  - vorsichtiger Einsatz
  - **window** ist Standard, kann meistens weggelassen werden





```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
    <link rel="stylesheet" type="text/css" href="./css/webkompendium.css">
    <link rel="shortcut icon" href="./css/favicon.ico">
    <title>Grundlagen Internet-Technologien: JavaScript</title>

    <script type="text/javascript">
      <!--
        window.defaultStatus = "Grundlagen Internet-Technologien mit JavaScript";
      // -->
    </script>

  </head>
  <body>

    <hr><center>
      <h2>Grundlagen Internet-Technologien</h2>
      <h3>JavaScript: das window-Objekt</h3><h4>
    </center><hr>

    <script type="text/javascript">

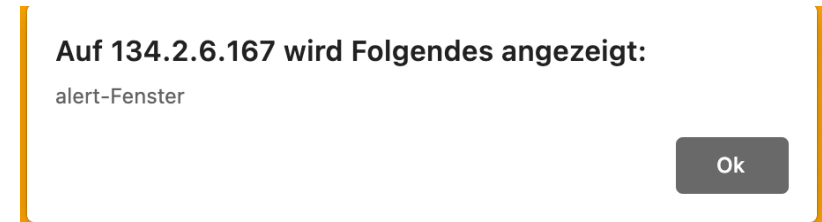
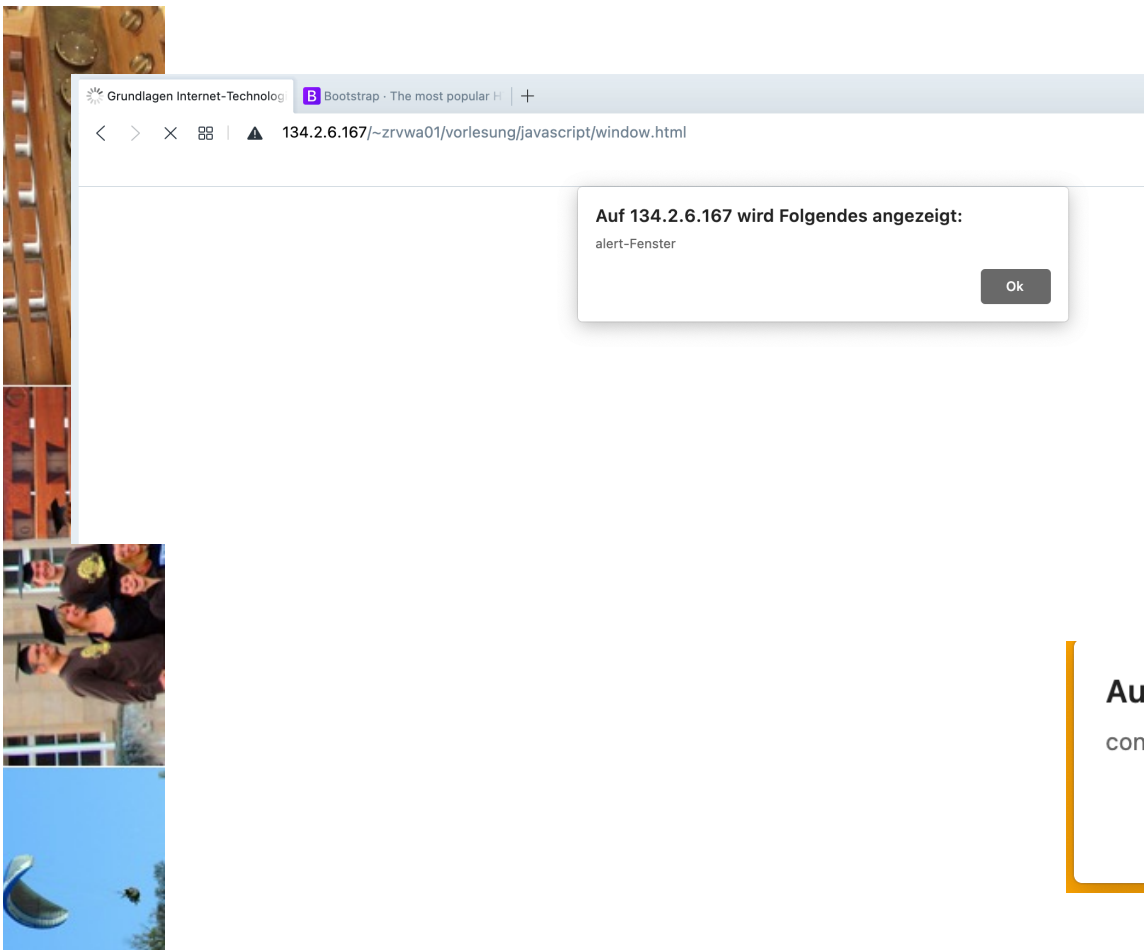
      window.alert("alert-Fenster");
      window.confirm("confirm-Fenster");

      window.open("http://www.uni-tuebingen.de","Uni Tuebingen");

    </script>

    <a href="javascript:window.close()">Fenster schließen</a>

  </body>
</html>
```



**Grundlagen Internet-Technologien**

**JavaScript: das window-Objekt**

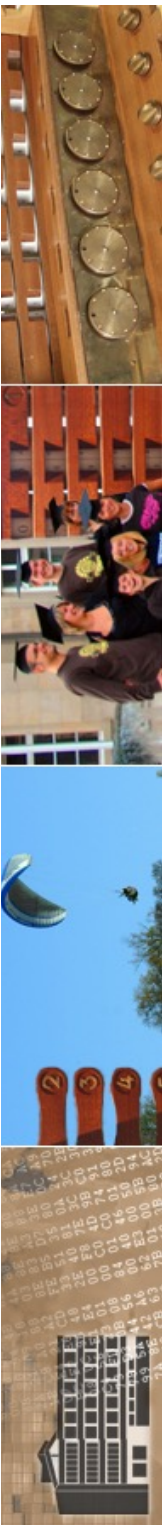
[Fenster schließen](#)





# window.location

- verwaltet das aktuell angezeigte Objekt
  - protocol
  - hostname
  - port
  - pathname
  - href
  - host
  - reload()





```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" type="text/css" href="./css/webkompendium.css">
    <link rel="shortcut icon" href="./css/favicon.ico">
    <title>Grundlagen Internet-Technologien: JavaScript</title>
  </head>
  <body>

    <hr><center>
      <h2>Grundlagen Internet-Technologien</h2>
      <h3>JavaScript: das window.location</h3>
    </center>

    <script type="text/javascript">
      for (eigenschaft in window.location)
        document.write("<b>" + eigenschaft + " :</b> " + window.location[eigenschaft] + "<br />");
    </script>

    <hr>

  </body>
</html>
```





# Grundlagen Internet-Technologien

## JavaScript: das window.location

**ancestorOrigins:** [object DOMStringList]  
**href:** http://134.2.6.167/~zrvwa01/vorlesung/javascript/windowlocation.html  
**origin:** http://134.2.6.167  
**protocol:** http:  
**host:** 134.2.6.167  
**hostname:** 134.2.6.167  
**port:**  
**pathname:** /~zrvwa01/vorlesung/javascript/windowlocation.html  
**search:**  
**hash:**  
**assign:** function assign() { [native code] }  
**reload:** function reload() { [native code] }  
**replace:** function replace() { [native code] }  
**toString:** function toString() { [native code] }







# window.history und window.frames

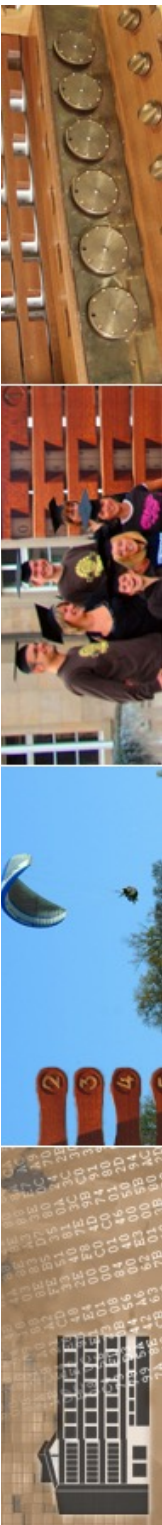
- history verwaltet Browser-History
  - `back()`
  - `forward()`
  - `go(n)`
  
- frame verwaltet Frameset





# window.document

- das angezeigte HTML-Dokument
  - `write (...)`
  - `title`
  - `lastModified`
  - `bgColor` und `fgColor`
  - `<script>`  
`window.document.write (document.lastModified) ;`  
`</script>`





```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <link rel="stylesheet" type="text/css" href="./css/webkompendium.css">
5     <link rel="shortcut icon" href="./css/favicon.ico">
6     <title>Grundlagen Internet-Technologien: JavaScript</title>
7   </head>
8   <body>
9
10    <hr><center>
11      <h2>Grundlagen Internet-Technologien</h2>
12      <h3>JavaScript: das window.document</h3>
13    </center>
14
15    <script type="text/javascript">
16      for (eigenschaft in window.document)
17        document.write("<b>" + eigenschaft + " :</b> " + window.document[eigenschaft] + "<br />");
18    </script>
19
20    <hr>
21
22  </body>
23 </html>

```





# Grundlagen Internet-Technologien

## JavaScript: das window.document

**vlinkColor:**  
**linkColor:**  
**alinkColor:**  
**fgColor:**  
**bgColor:**  
**compatMode:** CSS1Compat  
**all:** [object HTMLAllCollection]  
**onautocompleteerror:** null  
**onautocomplete:** null  
**ontouchstart:** null  
**ontouchmove:** null  
**ontouchend:** null  
**ontouchcancel:** null  
**rootElement:** null  
**childElementCount:** 1  
**lastElementChild:** [object HTMLHtmlElement]  
**firstElementChild:** [object HTMLHtmlElement]  
**children:** [object HTMLCollection]  
**onwaiting:** null  
**onvolumechange:** null  
**ontoggle:** null  
**ontimeupdate:** null  
**onsuspend:** null  
**onsubmit:** null  
**onstalled:** null



Fügen Sie mit dem Herzsymbol in der Adressleiste Ihr erstes Lesezeichen hinzu oder verwenden Sie [Lesezeichen jetzt importieren](#)

# Grundlagen Internet-Technologien

## JavaScript: das window.document

**location:**

`http://134.2.6.146/~zrvwa01/vorlesung/javascript/windowdocument.html`

**implementation:** [object DOMImplementation]

**URL:** `http://134.2.6.146/~zrvwa01/vorlesung/javascript/windowdocument.html`

**documentURI:**

`http://134.2.6.146/~zrvwa01/vorlesung/javascript/windowdocument.html`

**compatMode:** CSS1Compat

**characterSet:** windows-1252

**charset:** windows-1252

**inputEncoding:** windows-1252

**contentType:** text/html

**doctype:** [object DocumentType]

**documentElement:** [object HTMLHtmlElement]

**xmlEncoding:** null

**xmlVersion:** null

**xmlStandalone:** false

**domain:** 134.2.6.146

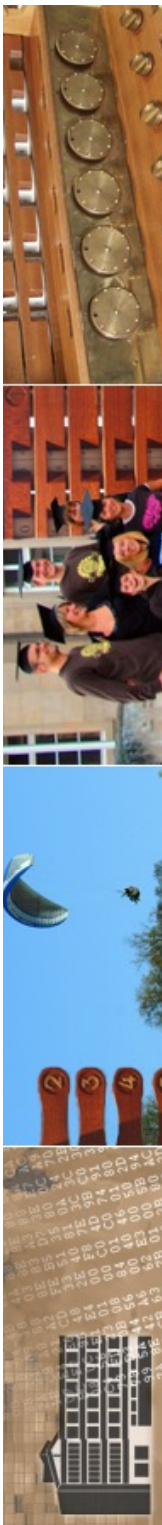
**referrer:**



# Grundlagen Internet-Technologien

## JavaScript: das window.document

**location:** <http://134.2.6.146/~zrvwa01/vorlesung/javascript/windowdocument.html>  
**implementation:** [object DOMImplementation]  
**URL:** <http://134.2.6.146/~zrvwa01/vorlesung/javascript/windowdocument.html>  
**documentURI:** <http://134.2.6.146/~zrvwa01/vorlesung/javascript/windowdocument.html>  
**compatMode:** CSS1Compat  
**characterSet:** windows-1252  
**charset:** windows-1252  
**inputEncoding:** windows-1252  
**contentType:** text/html  
**doctype:** [object DocumentType]  
**documentElement:** [object HTMLHtmlElement]  
**xmlEncoding:** null  
**xmlVersion:** null  
**xmlStandalone:** false  
**domain:** 134.2.6.146  
**referrer:** <http://134.2.6.146/~zrvwa01/vorlesung/javascript/>  
**cookie:** PHPSESSID=9uvp406gl4hjco5tvshr6uo2s1  
**lastModified:** 05/04/2014 11:03:01  
**readyState:** loading  
**title:** Grundlagen Internet-Technologien: JavaScript  
**dir:**  
**body:** [object HTMLBodyElement]  
**head:** [object HTMLHeadElement]  
**images:** [object HTMLCollection]  
**embeds:** [object HTMLCollection]  
**plugins:** [object HTMLCollection]  
**links:** [object HTMLCollection]  
**forms:** [object HTMLCollection]



# Grundlagen Internet-Technologien

## JavaScript: das window.document

**location:**  
`http://134.2.6.167/~zrvwa01/vorlesung/javascript/windowdocument.html`

**implementation:** [object DOMImplementation]

**URL:** `http://134.2.6.167/~zrvwa01/vorlesung/javascript/windowdocument.html`

**documentURI:**  
`http://134.2.6.167/~zrvwa01/vorlesung/javascript/windowdocument.html`

**compatMode:** CSS1Compat

**characterSet:** windows-1252

**charset:** windows-1252

**inputEncoding:** windows-1252

**contentType:** text/html

**doctype:** [object DocumentType]

**documentElement:** [object HTMLHtmlElement]

**xmlEncoding:** null

**xmlVersion:** null

**xmlStandalone:** false

**domain:** 134.2.6.167

**referrer:** `http://134.2.6.167/~zrvwa01/vorlesung/javascript/`

**cookie:** PHPSESSID=1te0125u54kql4015iephss4pu

**lastModified:** 05/04/2014 11:03:01

**readyState:** loading

**title:** Grundlagen Internet-Technologien: JavaScript

**dir:**

**body:** [object HTMLBodyElement]

**head:** [object HTMLHeadElement]

**images:** [object HTMLCollection]

**embeds:** [object HTMLCollection]

**plugins:** [object HTMLCollection]

**links:** [object HTMLCollection]

**forms:** [object HTMLCollection]

**scripts:** [object HTMLCollection]

**currentScript:** [object HTMLScriptElement]

**defaultView:** [object Window]

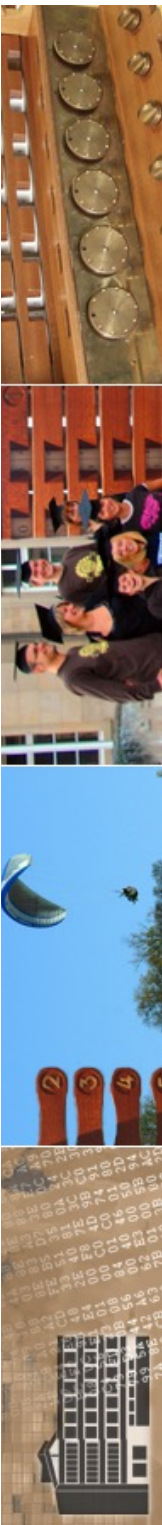
**designMode:** off

**onreadystatechange:** null



# window.document.forms

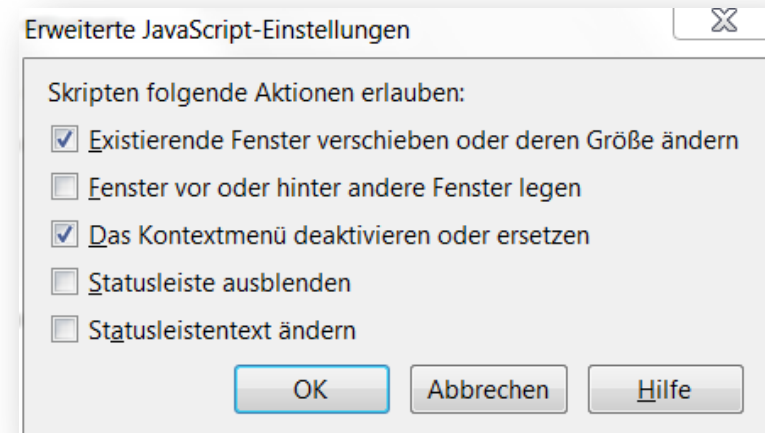
- Verwaltet Formular
- → behandeln wir in den Übungen





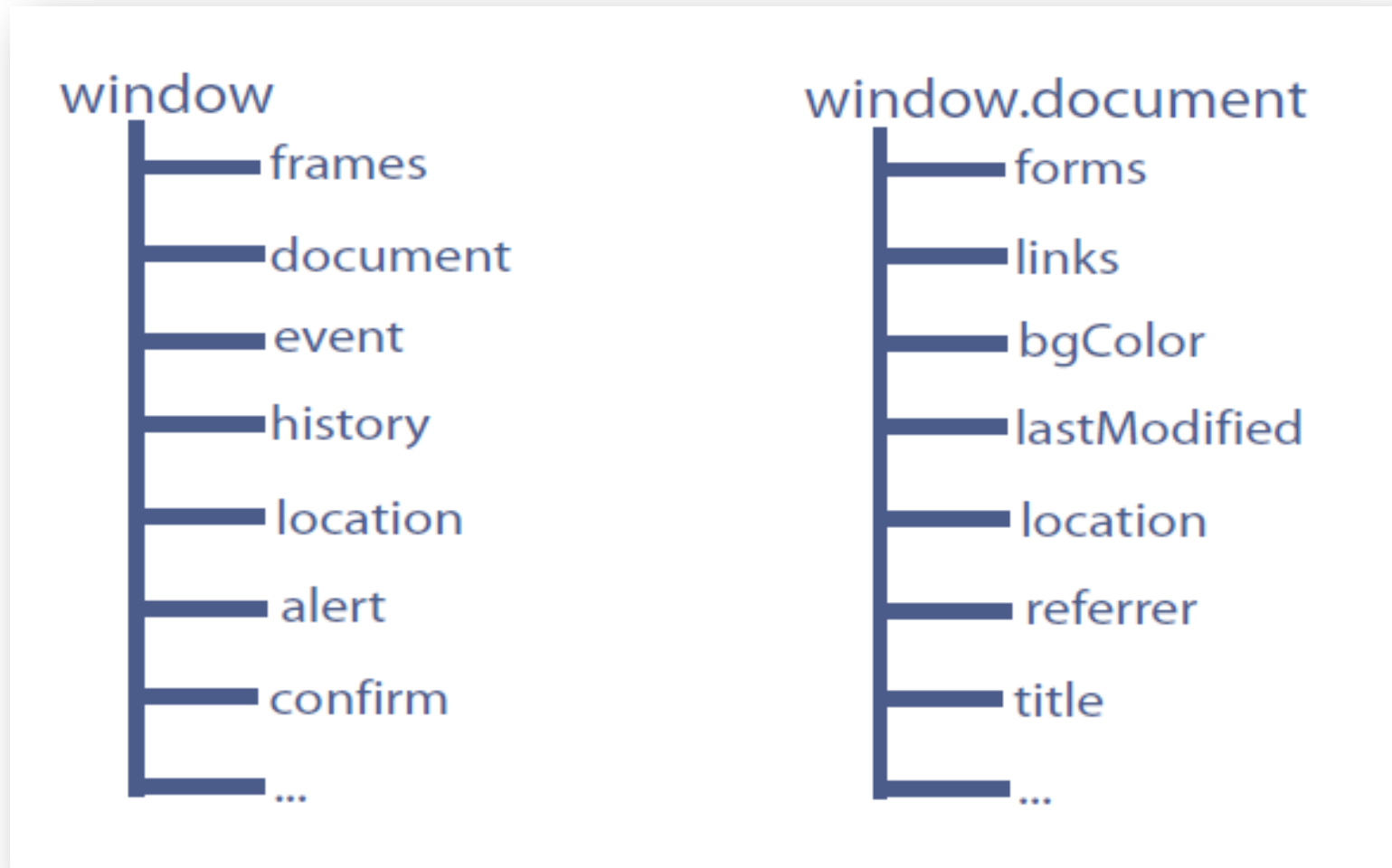
# Einfluss auf Browser

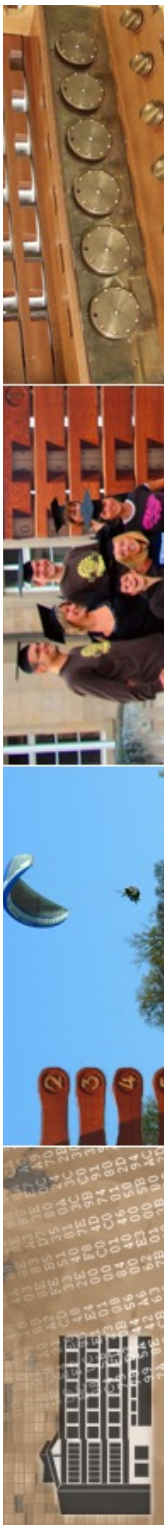
- mit JavaScript können auch spezielle Eigenschaften des Browsers über das Window-Objekt gesetzt werden
  - vielfältige Möglichkeiten
  - Gefahr für Benutzer
  - kein guter Stil





# Baumstruktur







# Event-Behandlung mit JavaScript

- Event: Ereignis
  - typische Events:
    - Mausklick
    - "Mouseover"
    - Fenster schließen
    - Seite neu laden
  - nun: Verbinden von Events mit JavaScript-Methoden





# Event-Behandlung mit JavaScript

- zwei Möglichkeiten
  - direkter JavaScript-Link
    - `<a href="javascript:window.close()">...</a>`
  - Event-Handle: Beispiel onFocus als HTML-Attribut:
 

```
<INPUT onFocus="jsfunction()">
```
- Beispiel even1.html, event2.html



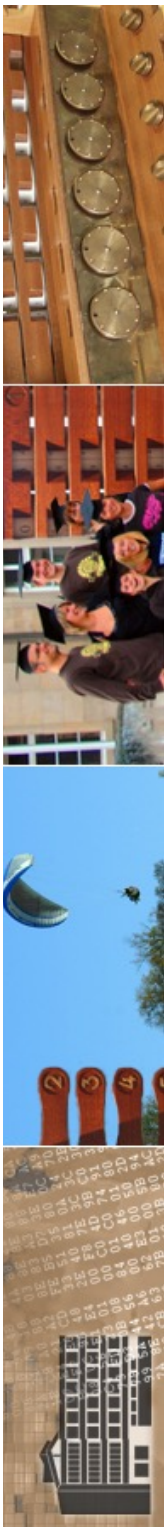


```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <link rel="stylesheet" type="text/css" href="./css/webkompendium.css">
5     <link rel="shortcut icon" href="./css/favicon.ico">
6     <title>Grundlagen Internet-Technologien: JavaScript</title>
7   </head>
8   <body>
9
10    <hr><center>
11      <h2>Grundlagen Internet-Technologien</h2>
12      <h3>Events in HTML & JavaScript</h3>
13
14      <form method="get" action="...">
15        <table>
16          <tr><td><input name="a" onFocus="this.value='1. Eingabefeld'"></td></tr>
17          <tr><td><input name="b" onFocus="this.value='2. Eingabefeld'"></td></tr>
18          <tr><td><input name="c" onFocus="this.value='3. Eingabefeld'"></td></tr>
19        </table>
20      </form>
21
22    </center><hr>
23
24  </body>
25 </html>
26

```





# Grundlagen Internet-Technologien

## Events in HTML & JavaScript

1. Eingabefeld

# Grundlagen Internet-Technologien

## Events in HTML & JavaScript

1. Eingabefeld

2. Eingabefeld





```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <link rel="stylesheet" type="text/css" href="./css/webkompendium.css">
5     <link rel="shortcut icon" href="./css/favicon.ico">
6     <title>Grundlagen Internet-Technologien: JavaScript</title>
7   </head>
8   <body>
9
10    <hr><center>
11      <h2>Grundlagen Internet-Technologien</h2>
12      <h3>Events in HTML & JavaScript</h3>
13
14      <form>
15        <table>
16          <tr><td><input size="30" name="ausgabe" readonly></td></tr>
17          <tr><td><input type="button" value="Stand"
18            onClick="this.form.elements[0].value=document.lastModified"></td></tr>
19        </table>
20      </form>
21
22    </center><hr>
23
24  </body>
25 </html>
```







The screenshot shows a web browser window with the following details:

- Address bar: `134.2.6.167/~zrvwa01/vorlesung/javascript/event2.html`
- Page Title: **Grundlagen Internet-Technologien**
- Section Title: **Events in HTML & JavaScript**
- Text: `05/14/2023 15:31:40`
- Button: `Stand der Datei`

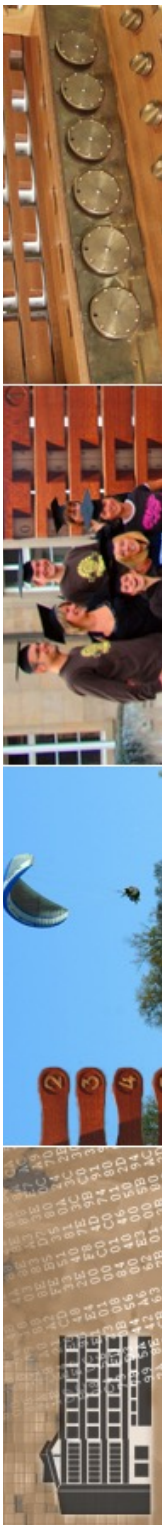




# Übersicht Event-Handles

- viele, aber wieder *stark abhängig von der JavaScript-Version*
- `onAbort` `onBlur` `onChange` `onClick`  
`onDbClick` `onError` `onFocus`  
`onKeyDown` `onKeyPress` `onKeyUp`  
`onLoad` `onMouseDown` `onMouseMove`  
`onMouseout` `onMouseOver` `onMouseUp`  
`onReset` `onSelect` `onSubmit`  
`onUnload` `javascript:`





## Was für Event-Handler gibt es? [\[ Bearbeiten \]](#)

Für jedes Event ist in JavaScript ein spezieller Event-Handler vorgesehen. Hier eine Übersicht:<sup>[1]</sup>



### Event-Handler für Maus-Aktionen [\[ Bearbeiten \]](#)

- onclick - Mausklick (auf ein Element)
- ondblclick - doppelter Mausklick
- onmousedown - Mausschalter wird gedrückt
- onmousemove - Maus wird bewegt
- onmouseout - Mauszeiger wird aus einem Element herausbewegt
- onmouseover - Mauszeiger berührt ein Element
- onmouseup - (der zunächst gedrückte) Mausschalter wird losgelassen

### Event-Handler für Tastatur-Aktionen [\[ Bearbeiten \]](#)

- onkeydown - eine Taste wird gedrückt
- onkeypress - eine Taste wird gedrückt und festgehalten
- onkeyup - eine (zunächst gedrückte) Taste wird losgelassen

### Event-Handler für Formulare und Formular-Elemente [\[ Bearbeiten \]](#)

- onblur - ein Element verliert den Fokus
- onchange - der Inhalt eines Formularelements wird verändert
- onfocus - ein Element gelangt in den Fokus
- onreset - Formular wird mit *reset* in den Ausgangszustand zurückgesetzt
- onselect - ein Element wird ausgewählt/markiert
- onsubmit - Formular wird mit *submit* abgeschickt

Siehe auch: [Dynamisierung von Formularen](#)

### Event-Handler für Dokument- und Bild-Aktionen [\[ Bearbeiten \]](#)

- onabort - Laden eines Bildes wird abgebrochen
- onerror - Fehler beim Laden eines Dokuments oder eines Bildes
- onload - Dokument wird geladen
- onunload - Dokument wird geschlossen

### Event-Handler für Fenster- und Frame-Aktionen [\[ Bearbeiten \]](#)

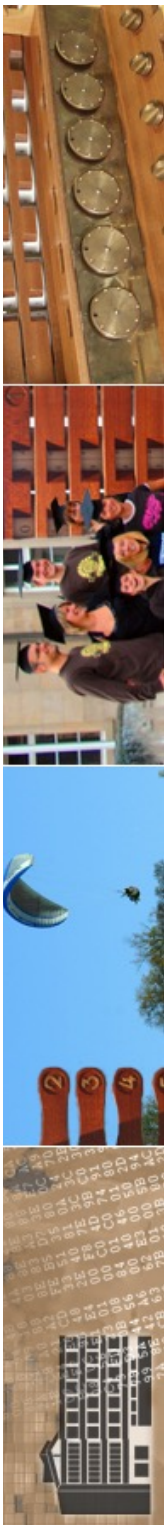
- onresize - Fenster oder Frame wird in der Größe verändert



# weitere Punkte zu JavaScript

- gemischtes Server-Client-Script
  - `<?php echo("<SCRIPT>document.write('Hello');</SCRIPT>"); ?>`
  - interessant: **AJAX**  
Asynchronous JavaScript And XML
  - HTML5: WebSockets
- Sicherheit des Clients
- Eingabefokus steuern (google)
  - `<BODY onLoad="self.focus(); document.formular.feld.focus()">`
- destruktive Skripte
- nicht auf JavaScript auf dem Client verlassen

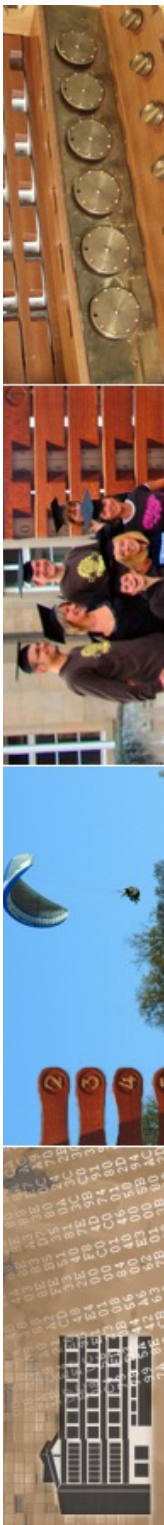
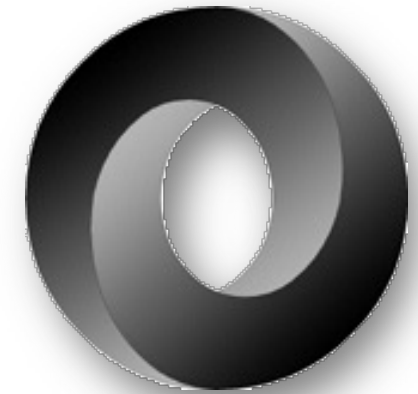






# JSON

- JSON: JavaScript Object Notation
- kompaktes **Datenaustauschformat**
  - JavaScript-Notation für Objektliterale
  - <http://www.json.org>
- Textformat
  - ungeordneter Container von key-value-Paaren
- wichtige Alternative zu XML als Datenaustauschformat



Fügen Sie mit dem Herzsymbol in der Adressleiste Ihr erstes Lesezeichen hinzu oder verwenden Sie [Lesezeichen jetzt importieren](#)



# Introducing JSON

*How JavaScript Works* by Douglas Crockford

العربية Български 中文 Český Dansk Nederlands English Esperanto Français Deutsch Ελληνικά Ελληνικά Magyar Indonesia Italiano 日本 한국어 فارسی Polski Português Română Русский Српско-хрватски Slovenščina Español Svenska Türkçe Tiếng Việt

ECMA-404 The JSON Data Interchange Standard.

**JSON** (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. It is based on a subset of the [JavaScript Programming Language, Standard ECMA-262 3rd Edition - December 1999](#). JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others. These properties make JSON an ideal data-interchange language.

JSON is built on two structures:

- A collection of name/value pairs. In various languages, this is realized as an *object*, record, struct, dictionary, hash table, keyed list, or associative array.
- An ordered list of values. In most languages, this is realized as an *array*, vector, list, or sequence.

These are universal data structures. Virtually all modern programming languages support them in one form or another. It makes sense that a data format that is interchangeable with programming languages also be based on these structures.

In JSON, they take on these forms:

An *object* is an unordered set of name/value pairs. An object begins with { (left brace) and ends with } (right brace). Each name is followed by : (colon) and the name/value pairs are separated by , (comma).

```
json
element

value
object
array
string
number
"true"
"false"
>null

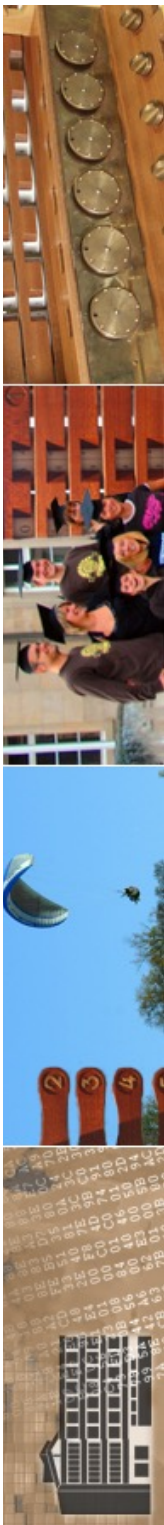
object
{' ws '}
{' members '}

members
member
member ', members
```



# JSON

- "JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. It is based on a subset of the JavaScript Programming Language, Standard ECMA-262 3rd Edition - December 1999. JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others. These properties make JSON an ideal data-interchange language."

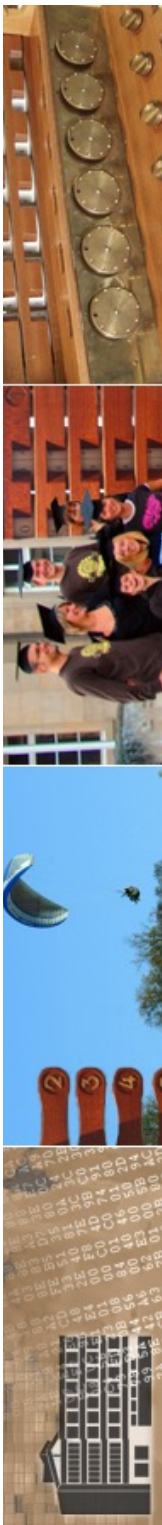


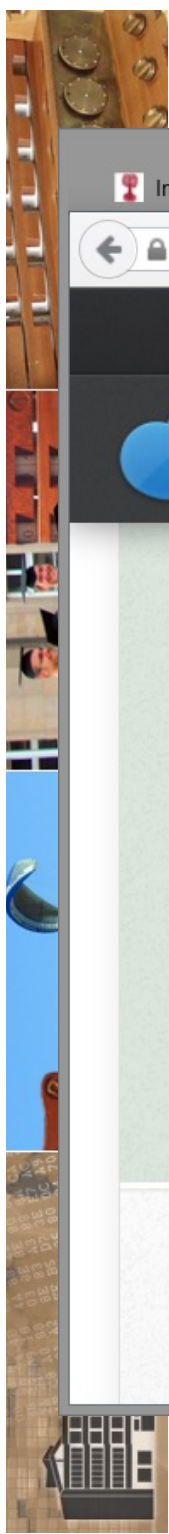




# AJAX

- Asynchronous JavaScript And XML
- **asynchrone** Kommunikation zwischen Client und Server
- gesteuert über JavaScript-Objekt **XMLHttpRequest**
  - damit Nachladen von Server-Inhalten/kontinuierliche Kommunikation zwischen Webclient und Webserver
  - Austauschformat XML, JSON, ...





Give us a shout out:

Username or e-mail Password [Sign In](#) [Sign up](#)

**9 Cloud9 IDE**  
*Your code anywhere, anytime*

PRICING FEATURES BLOG COMPANY SUPPORT

# Your code anywhere, anytime...

Write, run, and debug your code with our powerful and flexible cloud IDE. Collaborate on your workspaces publicly, or keep them private. The choice is yours!

[Sign me up!](#)

Or login instantly with  
GitHub or BitBucket:

*And yes, there is a free plan!*

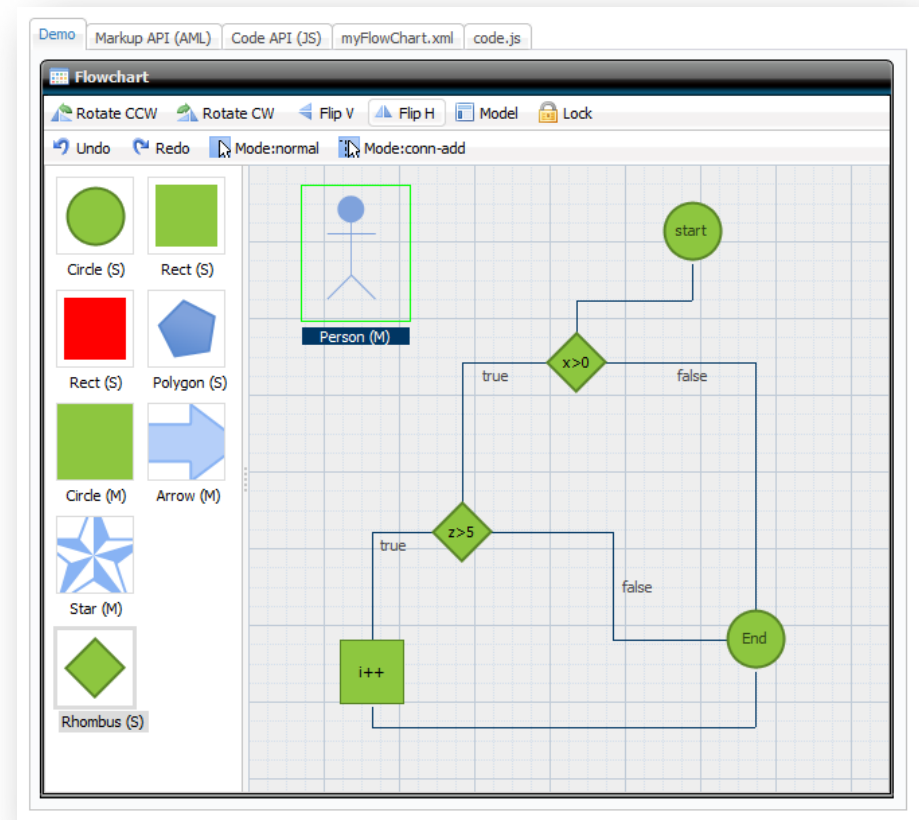
## Code smarter. Code together.

No bug is too great when you've got someone to help you out. Cloud9 IDE now



# AJAX

- AJAX wird erst interessant, wenn auch serverseitige Techniken eingesetzt werden
  - deshalb behandeln wir es etwas später genauer





# HTML5: WebSockets

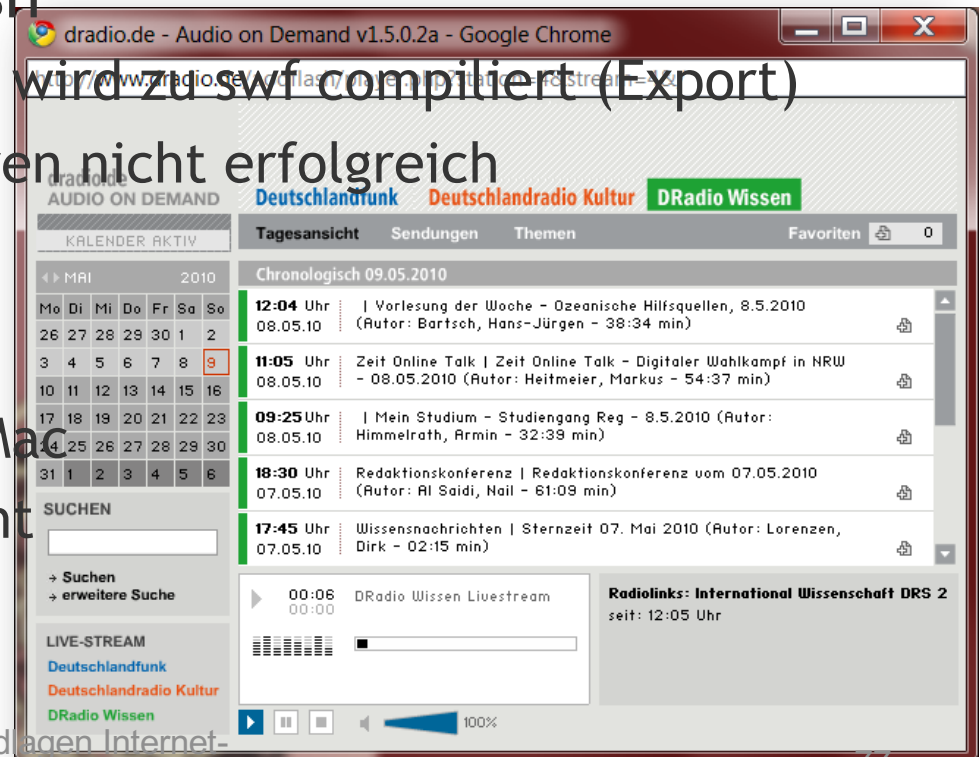
- Alternative zu AJAX: WebSockets von HTML5
  - Client öffnet Verbindung zum Server, welche asynchron und bidirektional genutzt werden kann

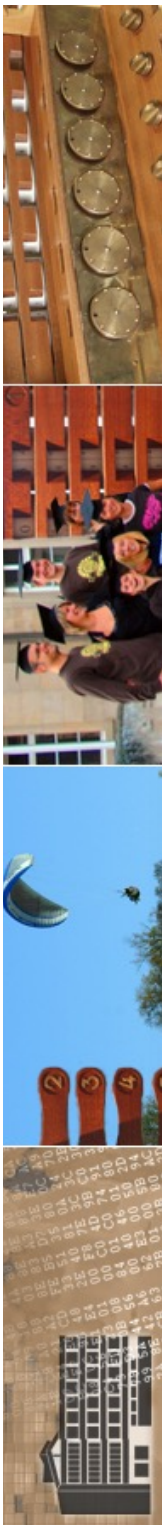
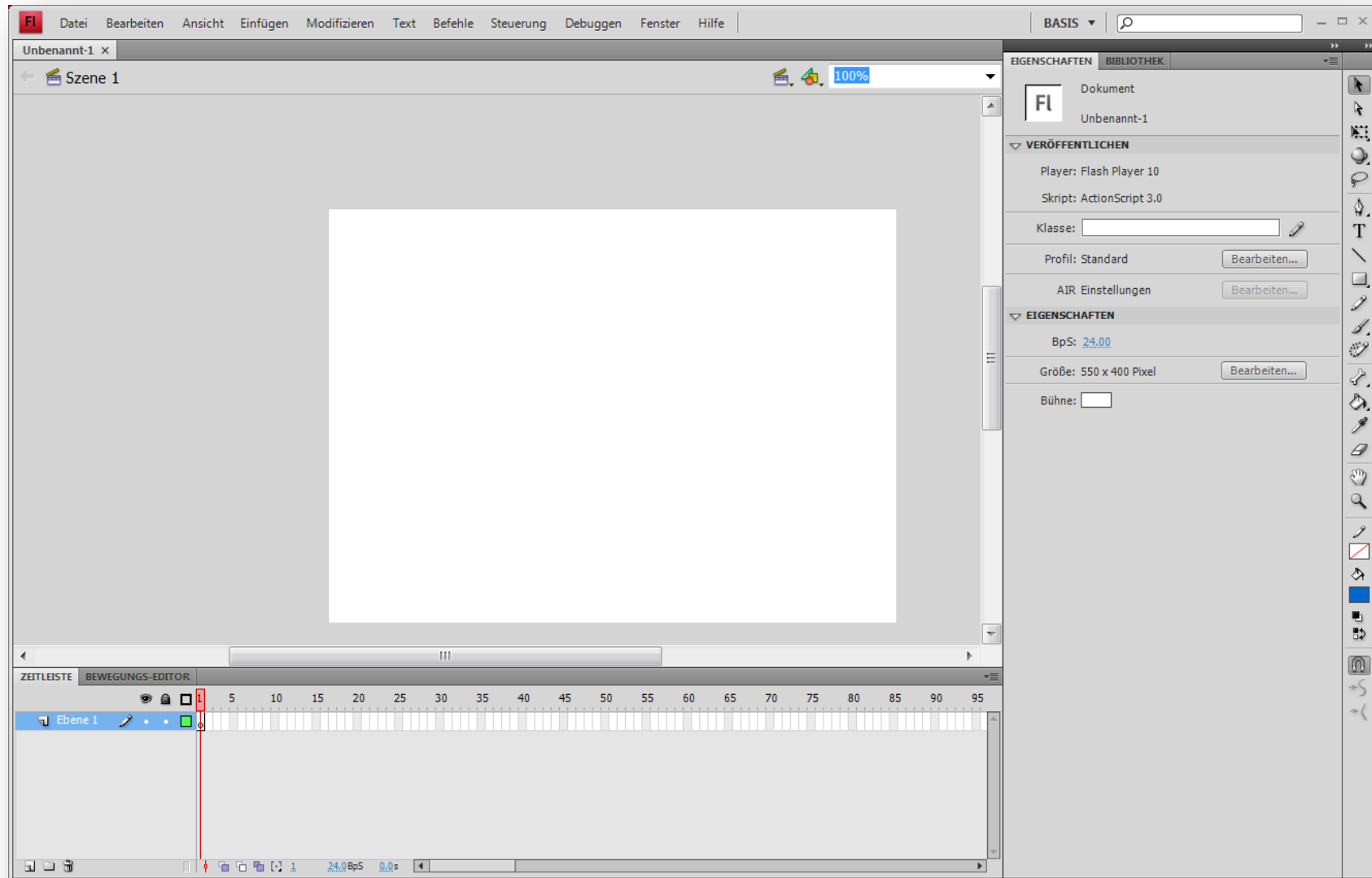




# weitere Client-Techniken

- Java Applet
  - echtes Java auf dem Client
- Adobe (Macromedia) Flash
  - Bühne in Flash, .fla-Datei, wird zu swf kompiliert (Export)
  - proprietär, freie Alternativen nicht erfolgreich
- MS Silverlight
  - beschränkt auf Windows/Mac
  - Alternative Mono/Moonlight

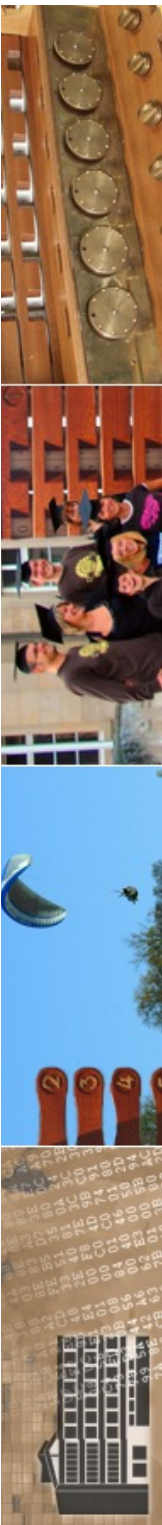






# aktuelle Ansätze

- neue Ansätze:
  - JavaScript-Frameworks
    - Google Web Toolkit GWT
    - qooxdoo
    - dojo
- Hauptziel: RIA - rich internet application  
Web-Anwendung mit GUI wie lokale  
Applikation





# GWT

- Idee: JavaScript-Client wird in Java geschrieben und mittels Compiler nach Javascript überführt
- <http://code.google.com/intl/de-DE/webtoolkit/>

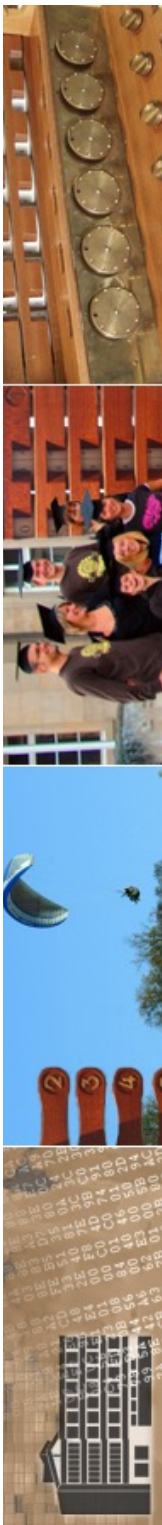






# GWT

- „With Google Web Toolkit (GWT), you write your **AJAX front-end in the Java programming language** which GWT then cross-compile into optimized JavaScript that automatically works across all major browsers. During development, you can iterate quickly in the same "edit - refresh - view" cycle you're accustomed to with JavaScript, with the added benefit of being able to debug and step through your Java code line by line. When you're ready to deploy, GWT compiles your Java source code into optimized, standalone JavaScript files. Easily build one widget for an existing web page or an entire application using Google Web Toolkit.”





# qooxdoo

- komplexes JavaScript/AJAX Framework:  
**qooxdoo Web Toolkit (QWT)**  
– auch Support für Mobile Devices
- <http://qooxdoo.org/>





Donate

v7.6.0

IN ACTION

DOCUMENTATION

COMMUNITY

ABOUT

Single page applications Wide range of UI components Extensive documentation Mature, stable & open source Sophisticated object model Comprehensive test suite 100% JavaScript toolchain Battle tested

# A universal JavaScript framework

TRUSTED BY








# dōjō      dojo

- dojo-Toolkit: [www.dojotoolkit.org](http://www.dojotoolkit.org)
  - sehr leistungsfähiges Toolkit, ab 2004 von Alex Russell, Dylan Schiemann und David Schontzer entwickelt
  - verfügt über Widgets, asynchroner Kommunikation, Möglichkeiten der persistenten Datenspeicherung und mehr

dojo  
toolkit





Visit [dojo.io](https://dojo.io) for documentation, examples, and more on Dojo, the Progressive Framework for Modern Web Apps and successor to the Dojo Toolkit

# Dojo Toolkit 1.17

A JavaScript toolkit that saves you time and scales with your development process. Provides everything you need to build a Web app. Language utilities, UI components, and more, all in one place, designed to work together perfectly.

GET DOJO

## TRUSTED BY



## USE DOJO



### REFERENCE GUIDE

Documentation and examples for every part of The Dojo Toolkit.



### TUTORIALS

Step-by-step guides focused on using Dojo to develop web apps.



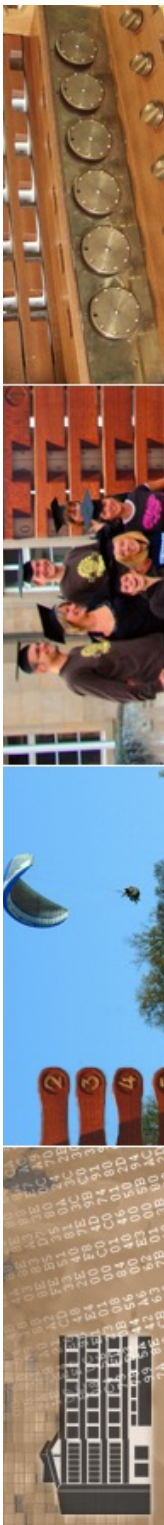
### API

The hardcore, no-fluff API documentation for the Dojo Toolkit.



# Gemeinsamkeiten

- Web-Applikationen mit dem „Look&Feel“ von klassischen Clientapplikationen
- vergleichsweise einfach zu entwickeln
- es gibt gute Einbindung in Entwicklertools
- schnelle Weiterentwicklung der Frameworks



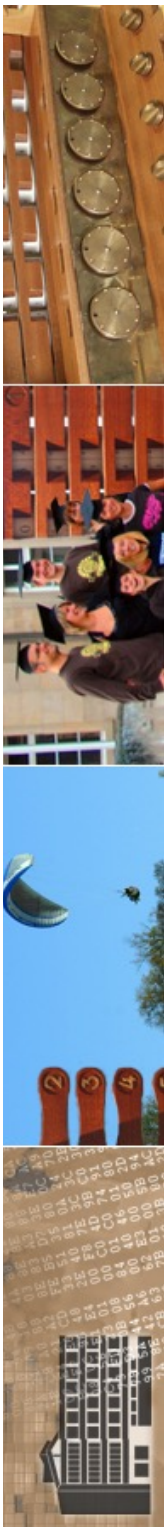


# ECMA Script

- ECMA Script 262 ist Versuch für einen Standard für Sprachen wie JavaScript (und JSON), 1997
  - ECMA: European Computer Manufacturers Association
  - JavaScript erfüllt ECMA weitgehend
  - siehe <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-262.pdf>
- aktuell ECMAScript 2022







[Back to the list](#)

# ECMA-262

ECMAScript® 2022 language specification

13th edition, June 2022

This Standard defines the ECMAScript 2022 general-purpose programming language.

Kindly note that the normative copy is the HTML version; the PDF version has been produced to generate a printable document.

This 13<sup>th</sup> edition has been prepared under the Ecma RF patent policy.

Please note that for ECMAScript Edition 4 the Ecma standard number “ECMA-262 Edition 4” was reserved but not used in the Ecma publication process. Therefore “ECMA-262 Edition 4” as an Ecma International publication does not exist.

The latest drafts are available at: <https://tc39.github.io/ecma262/>. Reporters should generally only file bugs if the bug is still present in the latest drafts.

Please find hereafter the place to file bugs: <https://github.com/tc39/ecma262#ecmascript>.

Download this standard

View in HTML



# JavaScript-Standard ECMAScript 2022 führt private Properties ein

Neben erweiterten Feldern bringt ES2022 Top-Level Await und einen erweiterten Zugriff auf einzelne Elemente in Arrays und Strings.

Lesezeit: 4 Min.  In Pocket speichern

   24



(Bild: Trismegist san/Shutterstock.com)

28.06.2022 11:23 Uhr | Developer

Von Rainald Menge-Sonntag

Ecma International hat mit ECMAScript 2022 den aktuellen Sprachstandard für JavaScript freigegeben. Die dreizehnte Auflage der seit 1997 veröffentlichten Spezifikation erweitert die Properties von Objekten unter anderem um private Felder. Die `at()`-Methode erlaubt den Zugriff auf indizierbare Typen mit negativen Indizes, und für asynchrone Programmausführung existiert nun `Await` auf Modulebene.

# ...und nun...

- haben wir die Möglichkeiten von JavaScript genauer kennen gelernt
- wir kennen Begriffe wie DOM, Events in JavaScript, Frameworks in JavaScript und ECMA Script
- Sicherheit und JavaScript beachtet
- Einstieg in die **Serverseitige Web-Programmierung**

