



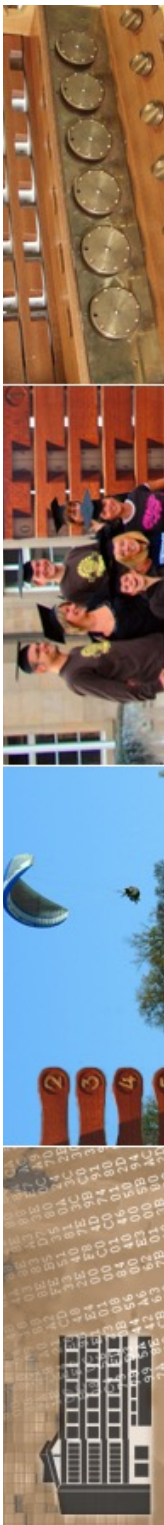
Einführung Internet-Technologien

INF3171

Clientseitige Web-Programmierung
mit JavaScript

Version 1.0

29.04.2026



Ransomware: Angreifer haben Oettinger-Brauerei erfolgreich attackiert

Die Großbrauerei Oettinger ist Opfer eines Cyberangriffs geworden. Die Täter haben offenbar Daten verschlüsselt und kopiert.



318

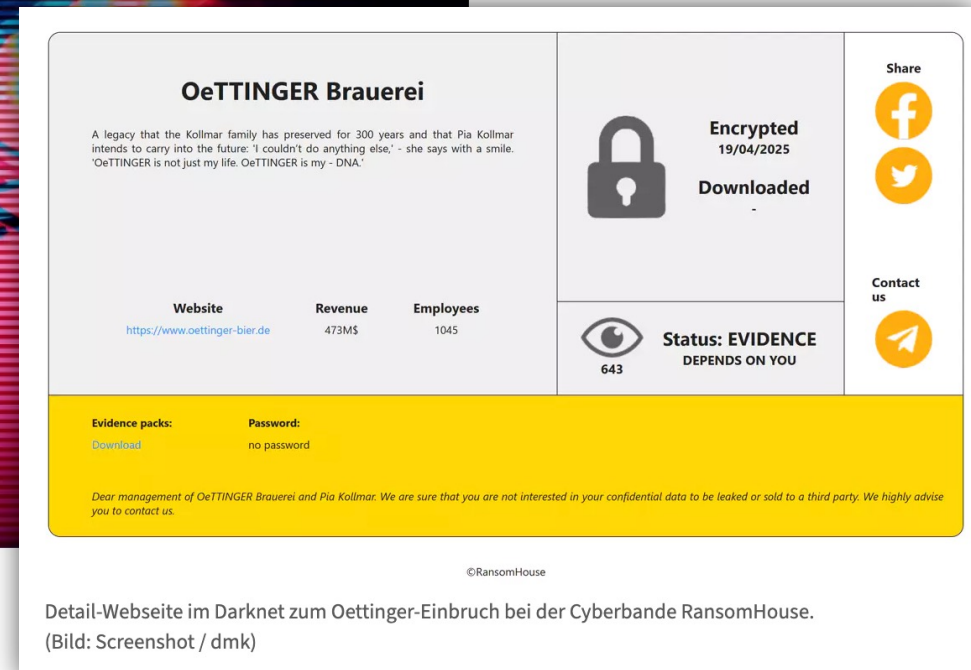


(Bild: janews/Shutterstock.com)

06.05.2025, 15:20 Uhr Lesezeit: 2 Min. | Security

Von Dirk Knop

Cyberkriminelle haben die Großbrauerei Oettinger angegriffen. Offenbar haben sie Daten verschlüsselt und kopiert, um das Unternehmen zu erpressen.



Detail-Webseite im Darknet zum Oettinger-Einbruch bei der Cyberbande RansomHouse.
(Bild: Screenshot / dmK)





Kritische Sicherheitslücke: Entwickler reparieren Nginx UI erneut

Wichtige Sicherheitsupdates schließen mehrere Schwachstellen im Web UI für nginx-Webserver Nginx UI. Angreifer können Instanzen übernehmen.



(Bild: VideoFlow/Shutterstock.com)

09:32 Uhr | Lesezeit: 2 Min. | Security

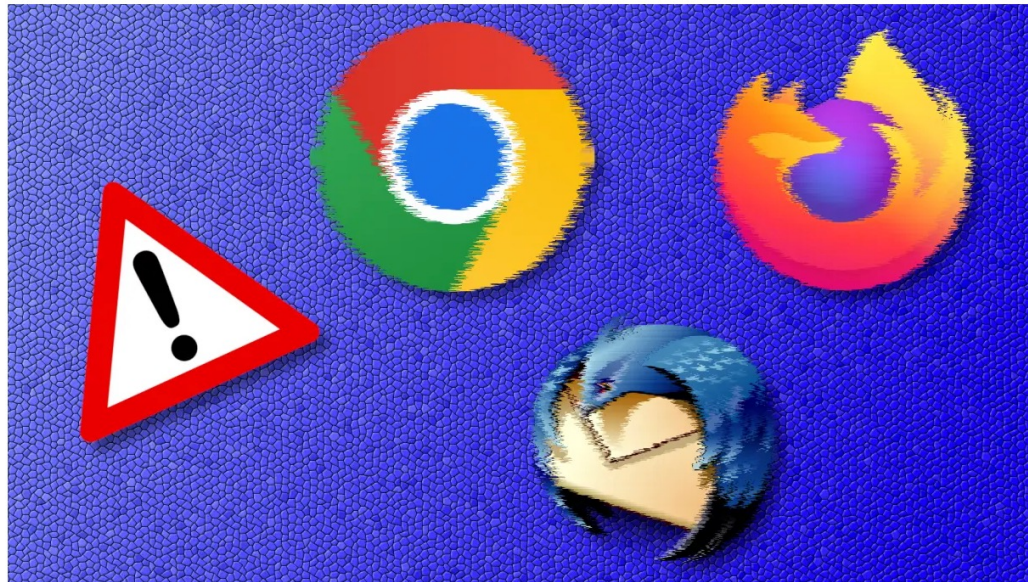
Von Dennis Schirmacher

Admins, die für ihre Webserver auf nginx-Basis die Weboberfläche Nginx UI nutzen, sollten die Software zeitnah auf den aktuellen Stand bringen. Geschieht das nicht, können Angreifer an mehreren Sicherheitslücken ansetzen und im schlimmsten Fall Systeme vollständig kompromittieren.



Kritische Lücken in Chrome und Firefox geschlossen

Die Webbrowser Chrome und Firefox sind in aktualisierten Versionen erschienen. Sie schließen kritische Sicherheitslücken.



(Bild: heise online / dmk)

08:45 Uhr Lesezeit: 3 Min. | Security

Von Dirk Knop

Wer weitverbreitete Webbrowser wie Googles Chrome oder Mozillas Firefox einsetzt, sollte zügig die bereitstehenden Aktualisierungen anwenden. Sie schließen als kritisches Risiko eingestufte Sicherheitslücken, die Angreifern unter anderem Codeschmuggel ermöglichen.



Klausurtermin

- 23.07. 8.00h st in N03

Einführung in
Internettechnologien
(Grundlagen der
Internettechnologien)



Walter

90

55 23.07.2026 08:00

wir suchen: HiWis

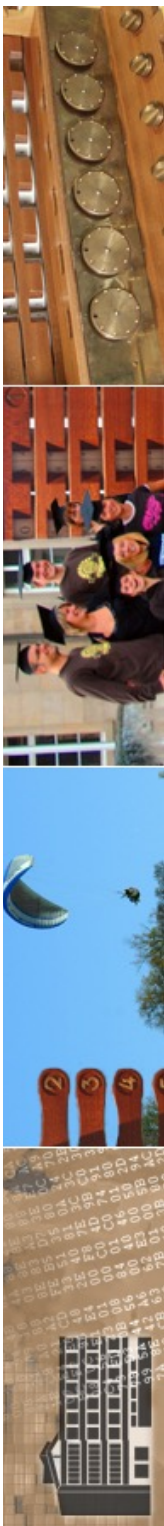
- am ZDV sind häufig HiWi-Stellen zu besetzen
- aktuell suchen wir ab SoSe 2026:

HiWis für die Genius-Bar

- ein sehr spannender, abwechslungsreicher und lehrreicher (!) Job

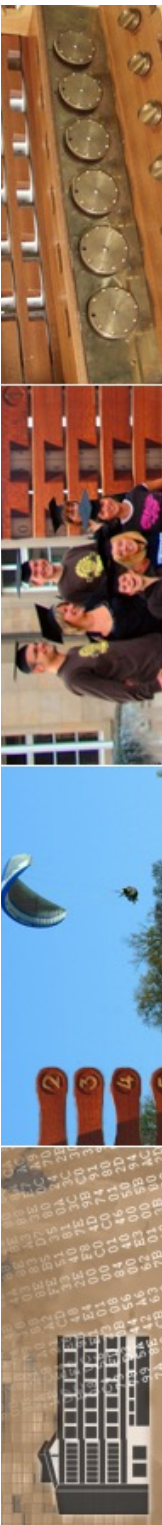
Access-Support

- bei Interesse Mail *mit dem Subject HiWi* an thomas.walter@uni-tuebingen.de



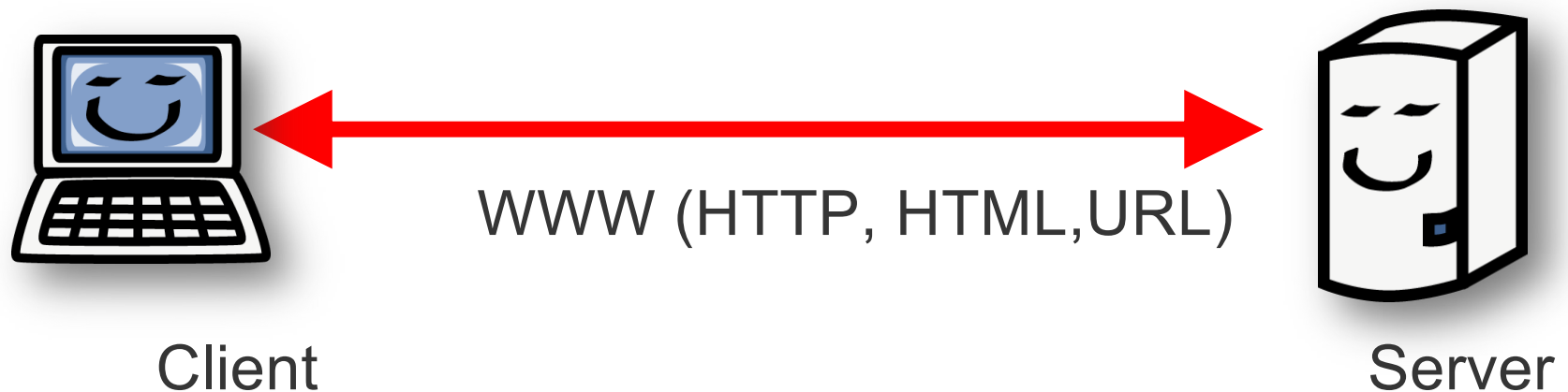
clientseitige Web-Programmierung

- es bestehen viele verschiedene Ansätze, u.a.
 - JavaScript
 - Java Applet
 - Adobe Flash
 - Microsoft Silverlight
 - Allgemeiner:
RIA: rich internet application

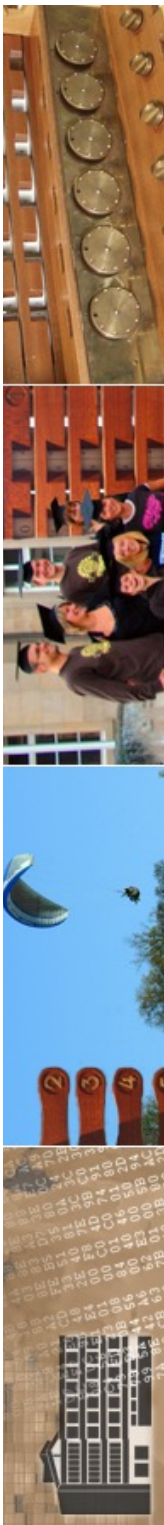


Architektur

- Situation:

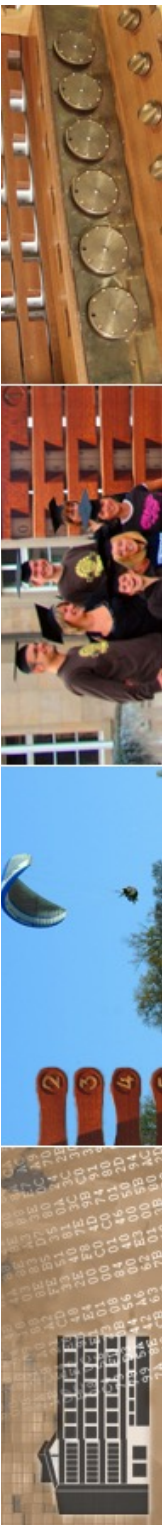


- "Webprogramme" können auf dem Client oder/und dem Server ausgeführt werden
 - wichtige Mischformen: **AJAX** und **WebSockets**



clientseitige Web-Programmierung

- Vorteil:
 - kein Netzwerk - schnelle Response
 - aktive Interaktion mit dem User
 - „Maus-Effekte“
 - Performance durch Nutzung der Client-CPU
 - jeder user hat eigene CPU im Gegensatz zur serverseitigen Web-Programmierung
- Nachteile:
 - Security auf dem Client
 - keine Kontrolle der Software auf dem Client
 - de facto keine Datenbank-Anbindung möglich



Programmiersprachen

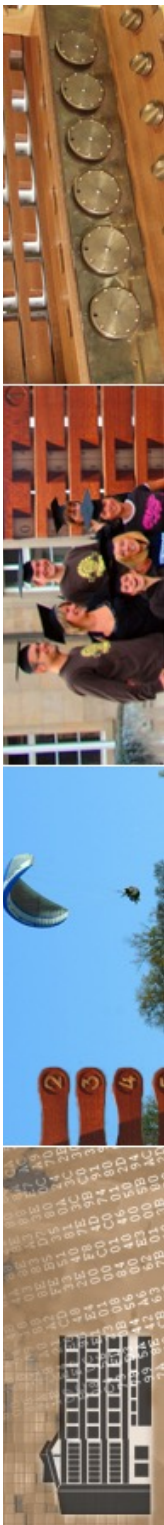
- Einschub: Was ist eine **Programmiersprache**?
 - "Notation für ein Computerprogramm"
 - maschinenauswertbare Notation eines **Algorithmus**
 - Unterscheidung
 - binäre Maschinensprache
 - Assembler
 - Hochsprachen
 - Algol, Fortran, Cobol, C, Pascal, Java, C#, ...
 - Basic, PHP, Perl, JavaScript, **Python**, Ruby, ...



Muhammed al-Chwarizmi
783 - 850

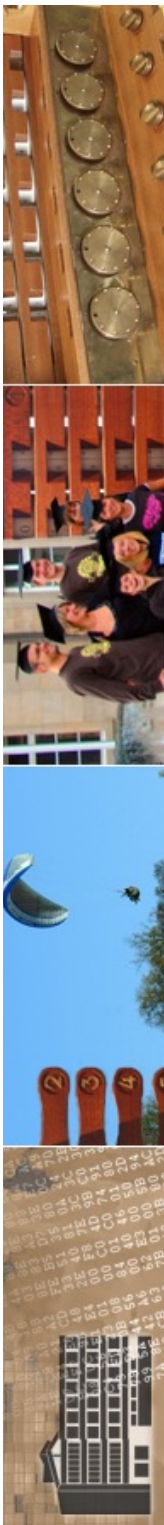
Programmiersprachen

- Hauptbestandteile von Programmiersprachen
 - **Variablen:** Name für Ort im Speicher
 - die „Subjekte und Objekte“ der Programmiersprache
 - ohne oder mit Typ für den Wert der Variablen
 - Typen: Zeichen (Char), ganze Zahl (Integer), Gleitkommazahl (float/double), Zeichenkette (String), ...
 - **Kontrollstrukturen:** Steuern den Programmfluss
 - Verzweigung (wenn - dann - sonst), Schleifen
 - **Anweisungen**
 - drucke, lese, beende, ...



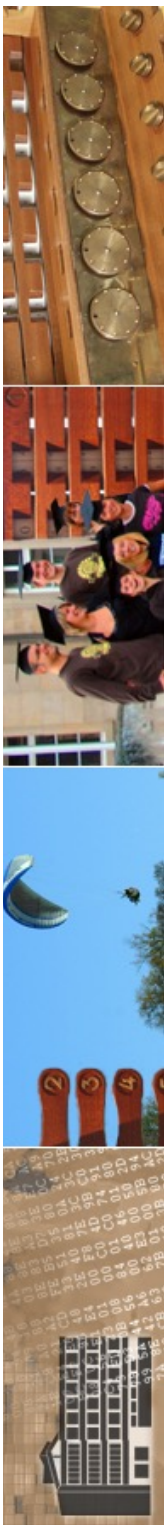
Programmiersprachen

- Unterscheidung: Compiler und Interpreter
 - aus dem Code der höheren Programmiersprache muss zum Ausführen Maschinencode werden
- **Compiler:** Übersetzt vor der Ausführung
 - keine Syntaxfehler bei der Ausführung
 - performant
- **Interpreter:** Übersetzen bei der Ausführung
 - flexibler



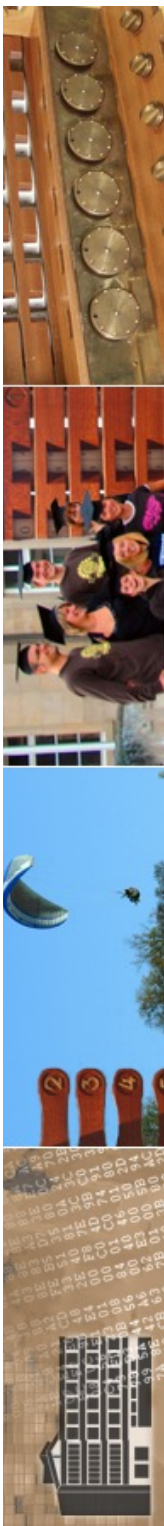
Scriptsprachen

- in dieser Veranstaltung benutzen wir drei Scriptsprachen:
 - JavaScript
 - clientseitige Web-Programmierung
 - PHP
 - serverseitige Script-Programmierung
 - Python
 - serverseitige CGI-Programmierung
 - Ruby
 - serverseitige CGI-Programmierung



JavaScript

- 1995 Brendan Eich: **LiveScript** (Netscape)
 - Netscape Navigator 2.0
- 1996 Umbenennung in **JavaScript**
- Klassiker: Version 1.5
- aktuell Version 1.8.5
 - Juli 2010
 - Sprachkern **ECMAScript**
- Literatur:
 - RRZN: JavaScript
 - Flanagan, David:



Den Kinderschuhen entwachsen – 25 Jahre JavaScript



Was 1995 als einfache Skriptsprache begann, ist heute die Grundlage praktisch jeder modernen Webanwendung. Zeit für einen Blick zurück, aber auch nach vorn.

Lesezeit: 6 Min.  In Pocket speichern

   181

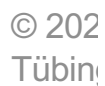


(Bild: ritchielee CC-BY-SA 2.0)

04.12.2020 18:35 Uhr | Developer

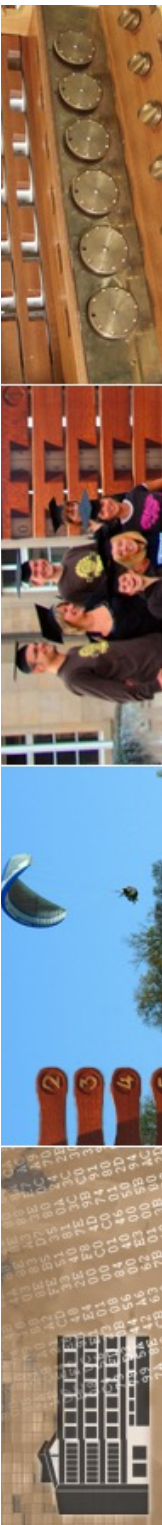
Von Golo Roden

Mit Jubiläen bei Programmiersprachen ist das so eine Sache. Was als Geburtsstunde gilt, kann variieren. Fakt ist, dass die Netscape Communications Corporation in Verbindung mit Sun Microsystems am 4. Dezember 1995, also vor genau 25 Jahren, eine neue Programmiersprache namens JavaScript ankündigte, die als offene und plattformunabhängige Sprache speziell für den Einsatz im Internet gedacht war. Ein wesentliches Ziel war von Anfang an, eine Sprache zu erschaffen, die zwar einfach, zugleich aber auch mächtig sein sollte.



Alternativen

- allgemeiner Ansatz: **ECMA-Script** (s.u.)
 - löst als Standard faktisch JavaScript ab
 - jährlich neue Version
- proprietäre Alternativen
(insbesondere von Microsoft)
 - JScript
 - TypeScript
 - ...und einige mehr...

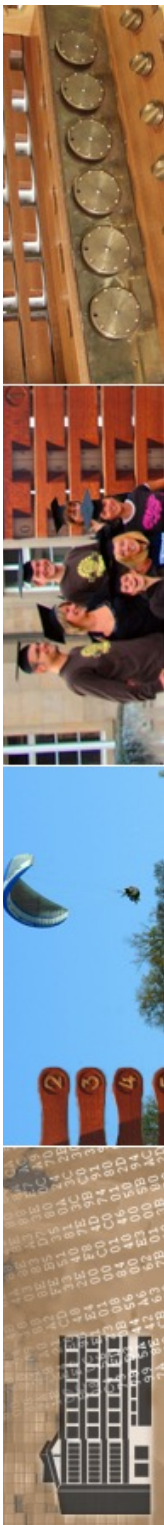


JavaScript

- zunächst ist JavaScript eine moderne, einfache Scriptsprache
 - nicht-typisiert („dynamisch typisiert“)
 - wird interpretiert
 - objektorientiert, aber *ohne Klassen* ☹️
- interessant wird JavaScript, weil diese Sprache *mit Webseiten "verbunden" werden kann*
 - Elementen in einer Web-Site werden JavaScript-Methoden zugeordnet
 - zentral: Verknüpfung von HTML und JavaScript: **DOM**

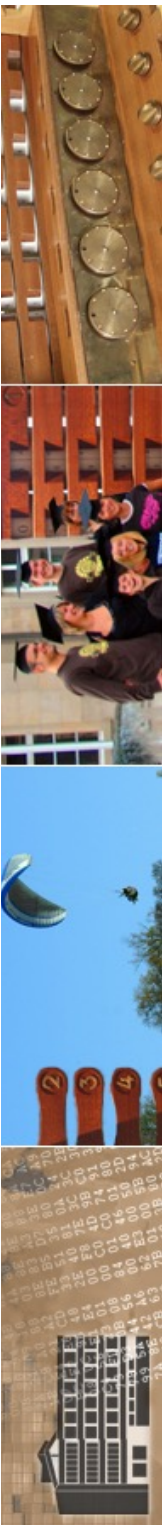
Grundprinzip

- mit JavaScript wird aktiver Code geschrieben, *der in einem HTML-Dokument integriert wird*
- Webseite „enthält das Programm“
- JavaScript wird vom jeweiligen Browser lokal ausgeführt



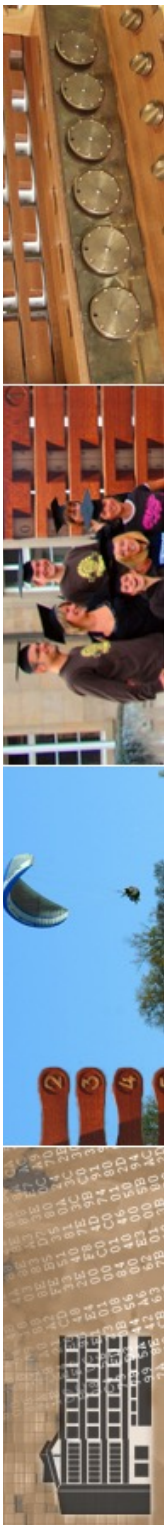
Editoren für JavaScript

- jeder Editor, der für HTML geeignet ist, ist auch für JavaScript geeignet
 - einfache Editoren wie notepad++ oder vi
 - HTML-Tools wie Dreamweaver
 - Programmierertools:
Eclipse mit Webtools

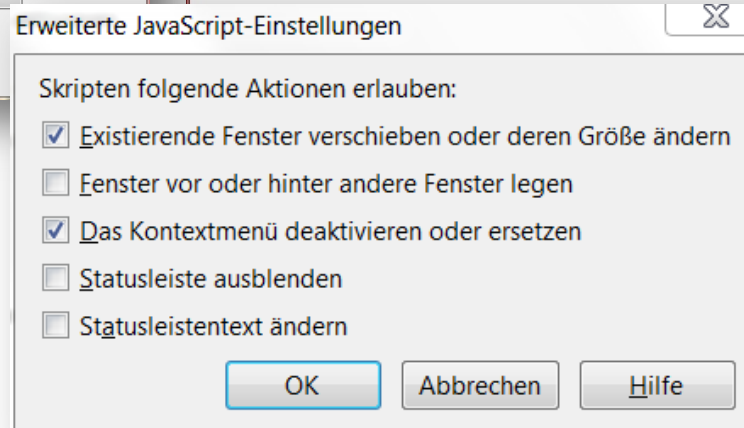
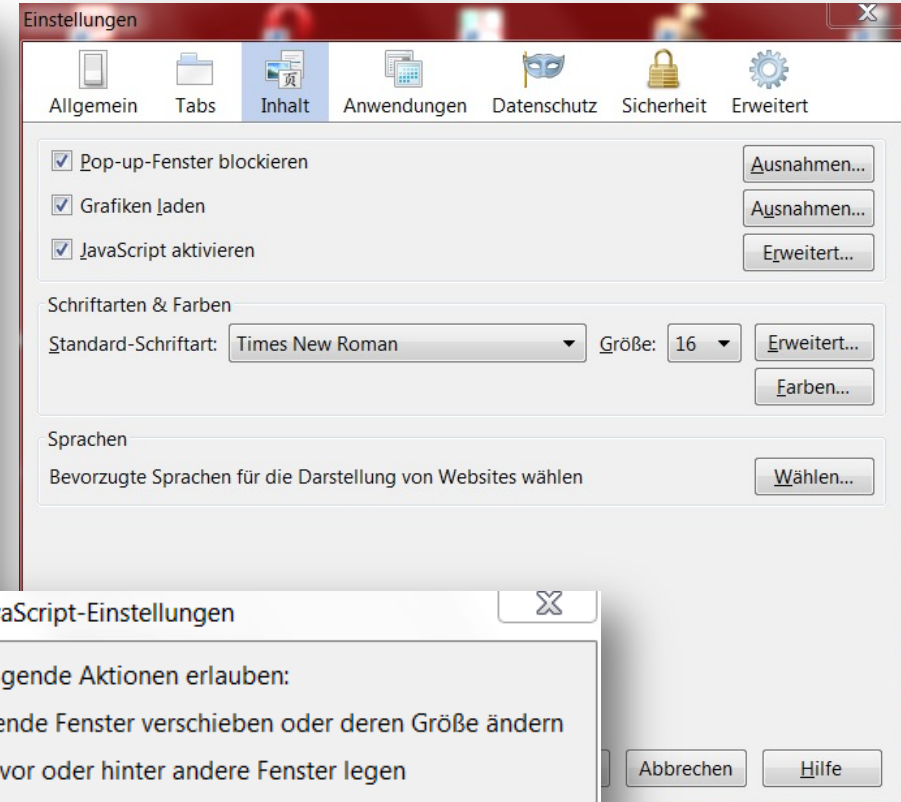
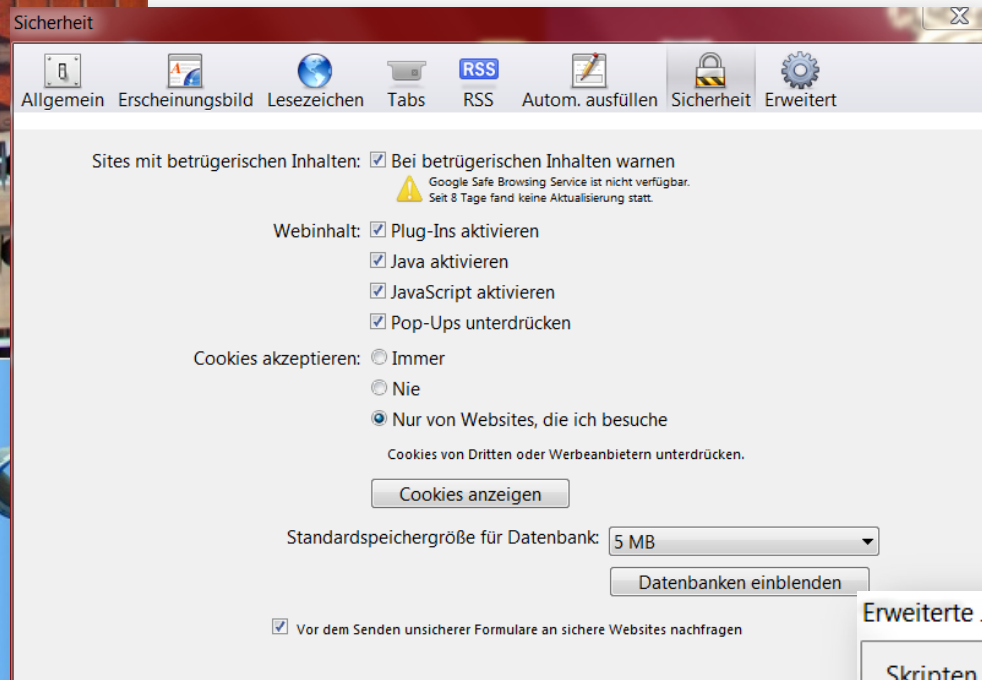


JavaScript und der Browser

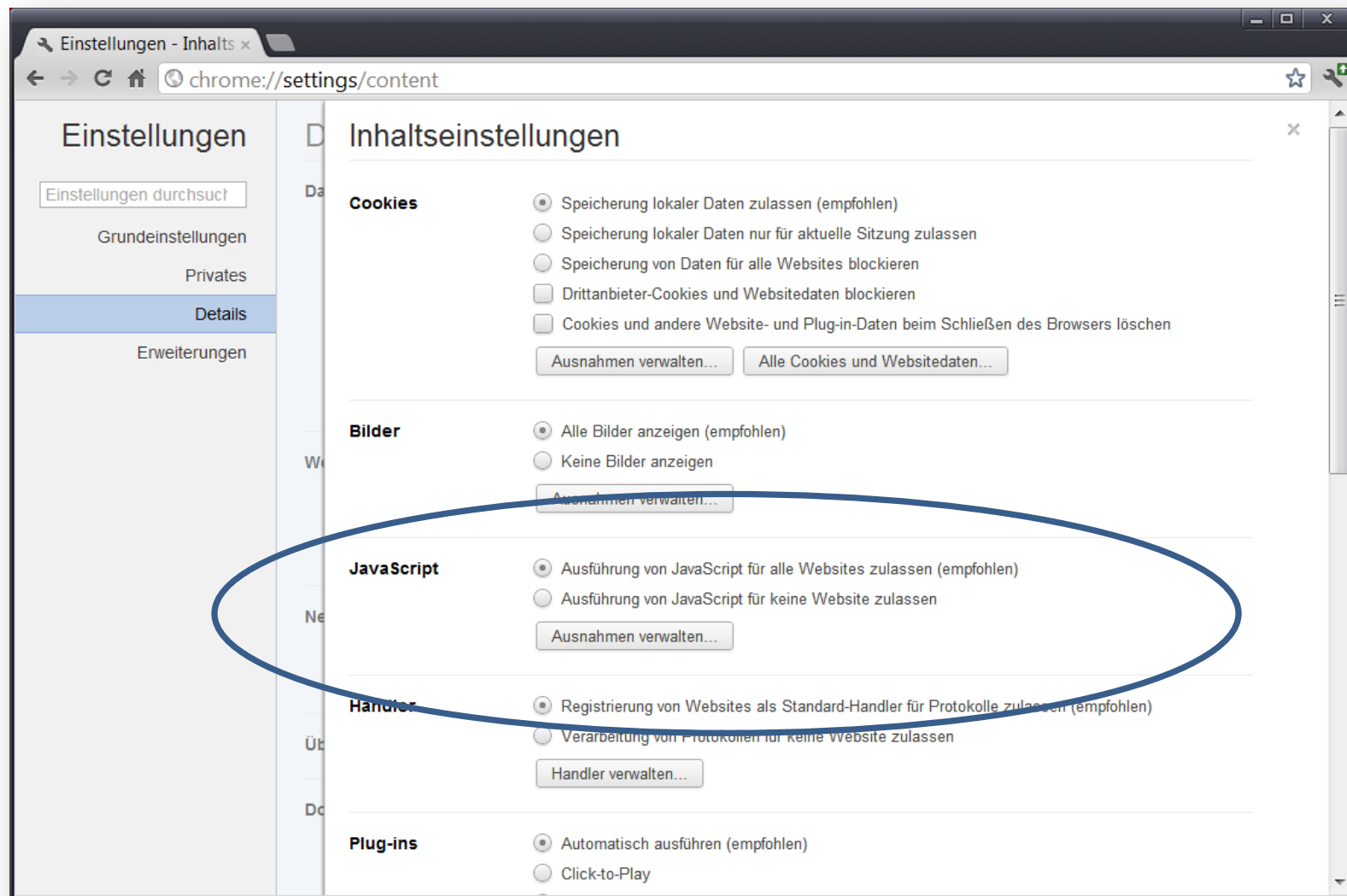
- JavaScript kann vom Benutzer erlaubt/verboten werden
- je nach Browser werden verschiedene Versionen von JavaScript unterstützt
– Problem für den Diensteanbieter
- engere Verknüpfung von HTML und JavaScript in HTML5



Browsereinstellungen



Browsereinstellungen





Inhaltseinstellungen

Cookies

☒ Speicherung lokaler Daten zulassen (empfohlen)
☐ Lokale Daten nach Schließen des Browsers löschen
☐ Speicherung von Daten für alle Websites blockieren
☐ Drittanbieter-Cookies und Websitedaten blockieren

Ausnahmen verwalten...

Alle Cookies und Websitedaten...

Bilder

☒ Alle Bilder anzeigen (empfohlen)
☐ Keine Bilder anzeigen

Ausnahmen verwalten...

JavaScript

☒ Ausführung von JavaScript für alle Websites zulassen (empfohlen)
☐ Ausführung von JavaScript für keine Website zulassen

Ausnahmen verwalten...

Schlüsselgenerierung

☐ Schlüsselgenerierung in Formularen auf allen Websites zulassen

Fertig

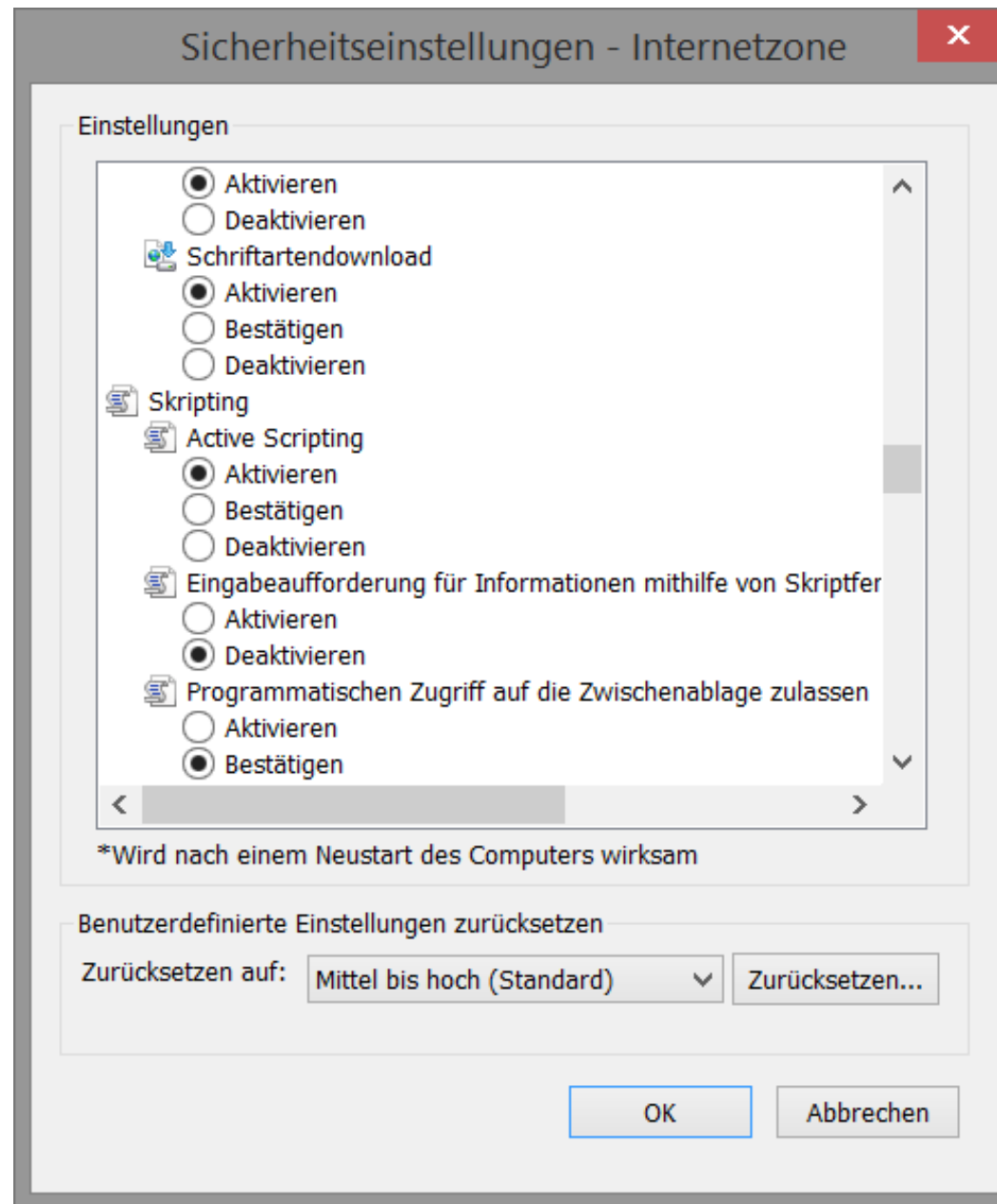
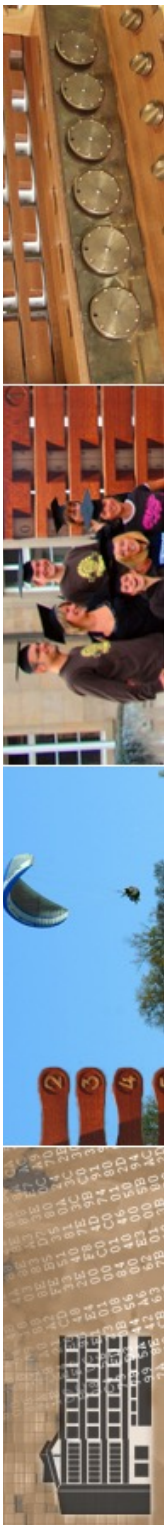


Firefox

about:config

Suchen: javascript

Einstellungsname	Status	Typ	Wert
browser.urlbar.filter.javascript	Standard	boolean	true
javascript.enabled	Standard	boolean	true
javascript.options.asmjs	Standard	boolean	true
javascript.options.baselinejit.chrome	Standard	boolean	true
javascript.options.baselinejit.content	Standard	boolean	true
javascript.options.gc_on_memory_pressure	Standard	boolean	true
javascript.options.ion.chrome	Standard	boolean	false
javascript.options.ion.content	Standard	boolean	true
javascript.options.ion.parallel_compilation	Standard	boolean	true
javascript.options.jit_hardening	Standard	boolean	true
javascript.options.mem.gc_allocation_threshold_mb	Standard	integer	30
javascript.options.mem.gc_decommit_threshold_mb	Standard	integer	32
javascript.options.mem.gc_dynamic_heap_growth	Standard	boolean	true
javascript.options.mem.gc_dynamic_mark_slice	Standard	boolean	true
javascript.options.mem.gc_high_frequency_heap_gro...	Standard	integer	300
javascript.options.mem.gc_high_frequency_heap_gro...	Standard	integer	150
javascript.options.mem.gc_high_frequency_high_limit_...	Standard	integer	500
javascript.options.mem.gc_high_frequency_low_limit_...	Standard	integer	100





🔍 In Einstellungen suchen

← JavaScript

🔍 Suchen

Zugelassen (empfohlen)



Blockieren

Hinzufügen

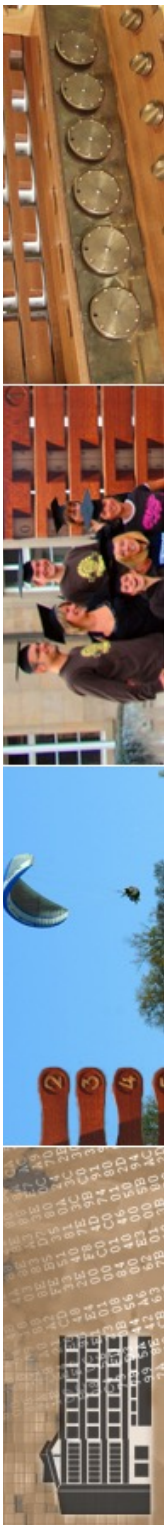
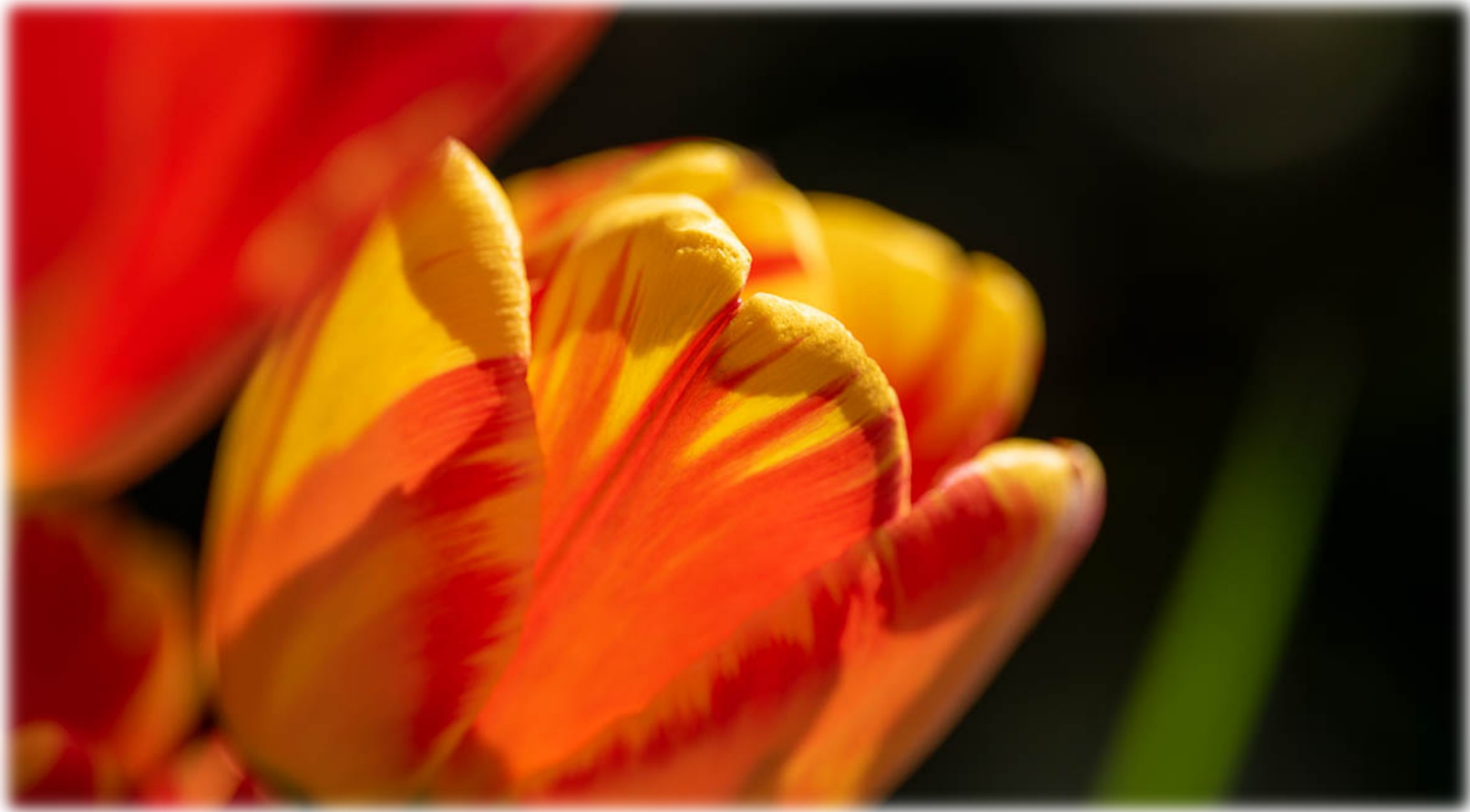
Keine Websites hinzugefügt

Zulassen

Hinzufügen

Keine Websites hinzugefügt







Kommentare in JavaScript

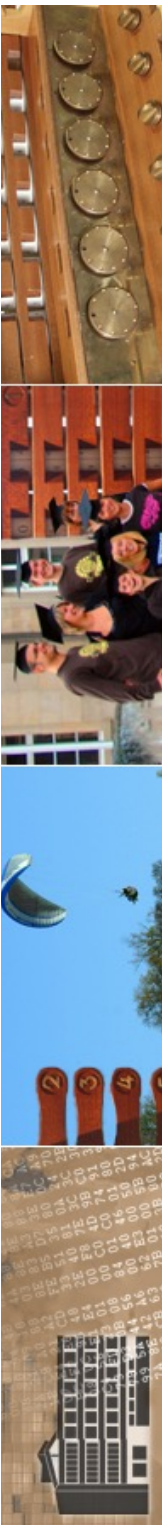
- Syntax der Kommentare in JavaScript:

`//` Zeilenkommentar

`/*`

Mehrzeilen-
kommentar

`*/`



HTML-Tags für JavaScript

- *verschiedene* Tags zur Kennzeichnung von JavaScript
 - `<SCRIPT>`
 `// JavaScript-Code`
 `</SCRIPT>`
 - `<SCRIPT language="JavaScript"> ...`
 - `<SCRIPT type="text/javascript"> ...`
 - auch mit Versionsangabe:
 - `<SCRIPT language="JavaScript1.5"> ...`
 - `<SCRIPT type="text/javascript1.5"> ...`

ein allererstes JavaScript



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
5 <title>Grundlagen Internet-Technologien</title>
6 </head>
7 <body>
8     <!-- ein erstes JavaScript -->
9     <script type="text/javascript">
10         // Kommentar in JavaScript
11         document.write("Grundlagen Internet-Technologien (INF3171)<BR />");
12         document.write("Hello <b>JavaScript</b>");
13     </script>
14 </body>
15 </html>
16
```

0 items

Description	Resource	Path	Location	Type
-------------	----------	------	----------	------

DOCTYPE:html Writable Smart Insert 1 : 1

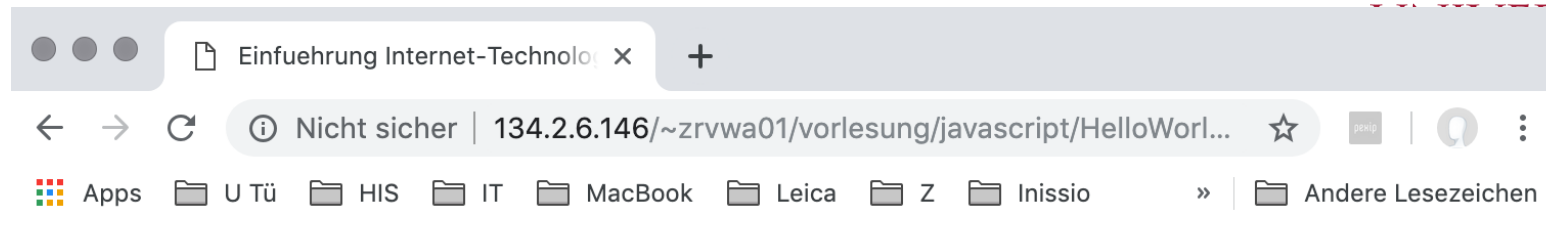
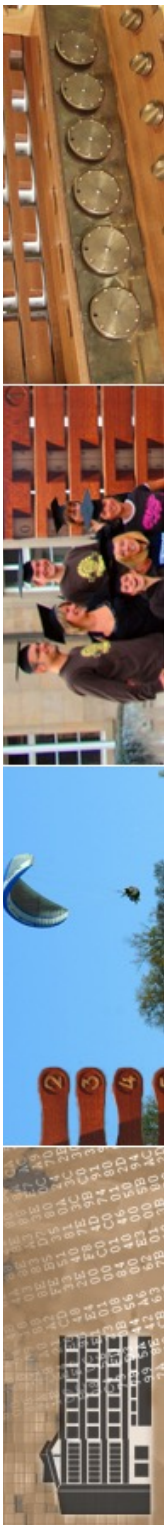


HelloWorld.html

```

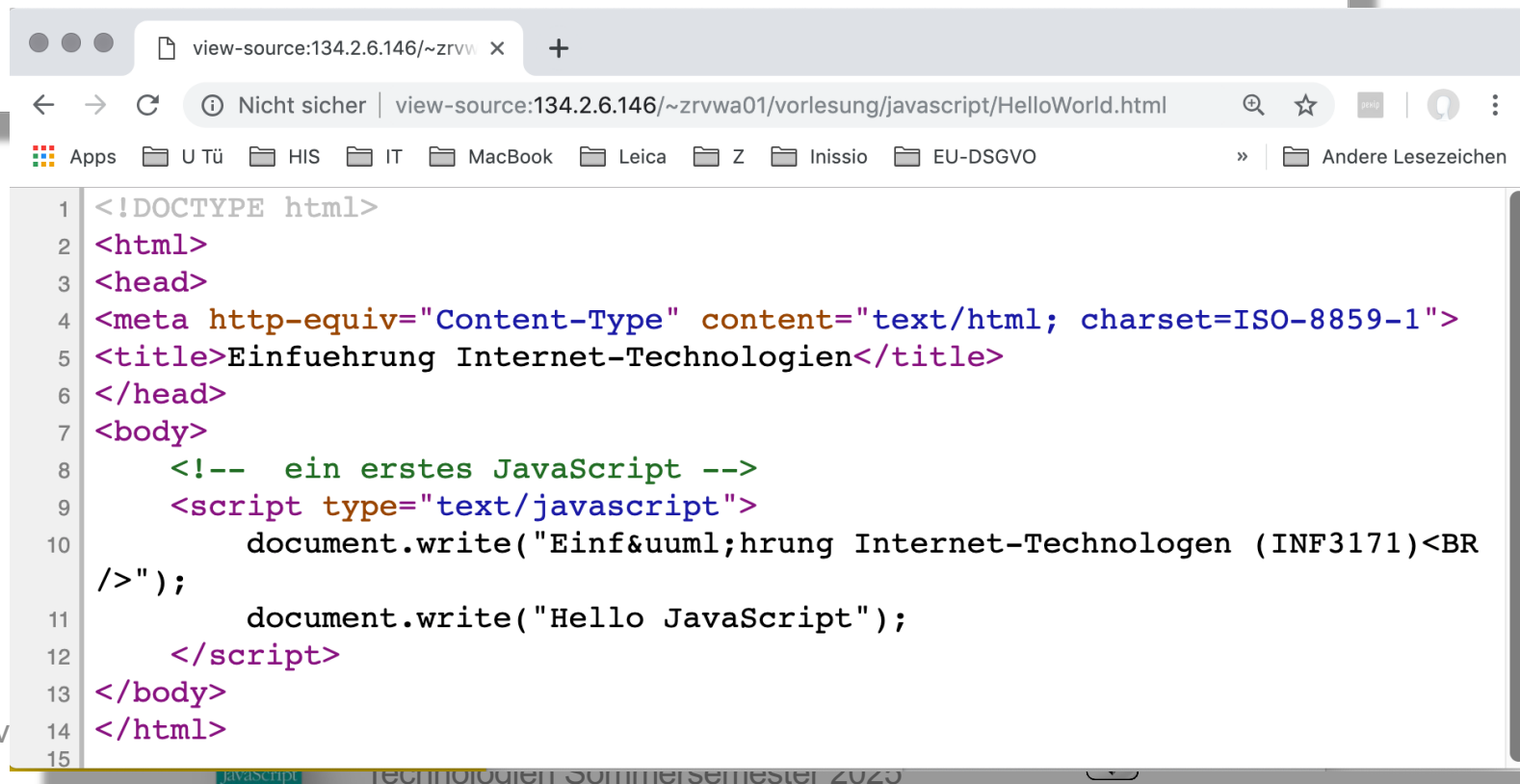
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
5  <title>Grundlagen Internet-Technologien</title>
6  </head>
7  <body>
8      <!-- ein erstes JavaScript -->
9      <script type="text/javascript">
10         // Kommentar in JavaScript
11         document.write("Grundlagen Internet-Technologien (INF3171)<BR />");
12         document.write("Hello <b>JavaScript</b>");
13     </script>
14 </body>
15 </html>

```



Einführung Internet-Technologien (INF3171)

Hello JavaScript





JavaScript - GIT JavaScript/HelloWorld.html - Eclipse - /Users/thomas/Documents/eclipse32/workspace

Quick Access

Java Perl JavaScript PHP Ruby Resource PyDev Web

Projec Naviga

Git JavaScript

- JavaScript Resources
- css
- buch.css
- buch.dtd
- Buch.js
- buch.xml
- buchhandlung1.html
- buchhandlung2.html
- event1.html
- event2.html
- fakultaet.html
- fakultaet.js
- formulartest.html
- hellojavascript.html
- hellojavascript2.html
- HelloWorld.html**
- javascript_template.html
- location.html
- navigator.html
- navigator2.html
- schleifen.html

HelloWorld.html

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
5 <title>Grundlagen Internet-Technologien</title>
6 </head>
7 <body>
8 <!-- ein erstes JavaScript -->
9 <script type="text/javascript">
10 // Kommentar in JavaScript
11 document.write("Grundlagen Internet-Technologien (INF3171)<BR />");
12 document.write("Hello <b>JavaScript</b>");
13 </script>
14 </body>
15 </html>
16
```

Outline

- DOCTYPE:html
- html
 - head
 - meta http-equiv=Content-Type
 - title
 - body
 - #comment
 - script type=text/javascript

Problems @ Documentation Declaration

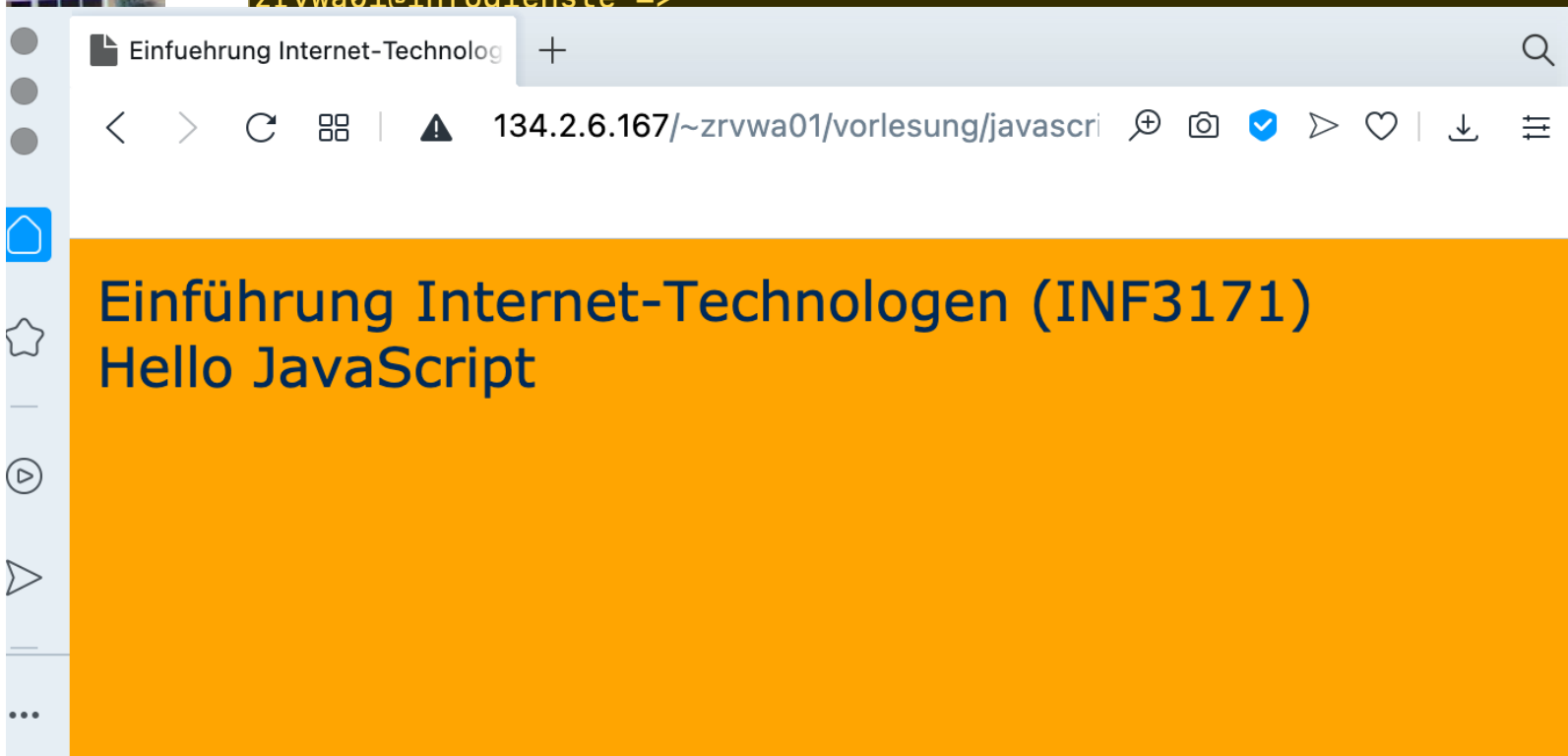
0 items

Description	Resource	Path	Location	Type
-------------	----------	------	----------	------

<> html/body/script



```
zrvwa01@infodienste =>
zrvwa01@infodienste => more HelloWorld.html
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
  <title>Einfuehrung Internet-Technologien</title>
  <link rel="stylesheet" type="text/css" href="./css/webkompendium.css">
</head>
<body>
  <!-- ein erstes JavaScript -->
  <script type="text/javascript">
    document.write("Einf&uuml;hrung Internet-Technologen (INF3171)<BR />");
    document.write("Hello JavaScript");
  </script>
</body>
</html>
zrvwa01@infodienste =>
```





< > ↺ ☰ | ⚠ view-source:134.2.6.146/~zrvwa01/vorlesung/javascript/HelloWorld.html

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
5     <title>Einfuehrung Internet-Technologien</title>
6     <link rel="stylesheet" type="text/css" href="._/css/webkompendium.css">
7 </head>
8 <body>
9     <!-- ein erstes JavaScript -->
10    <script type="text/javascript">
11        document.write("Einf&uuml;hrung Internet-Technologen (INF3171)<BR />");
12        document.write("Hello JavaScript");
13    </script>
14 </body>
15 </html>
16
```



Testscript version. html

- testen der
JavaScript-
Version,
die der
Browser
unterstützt



```
zrvwa01@infodienste => more version.html
<!DOCTYPE html>
<html>

  <head>
    <link rel="stylesheet" type="text/css" href="./css/webkompendium.css">
    <link rel="shortcut icon" href="./css/favicon.ico">
    <title>Internet-Technologien: erstes JavaScript</title>
  </head>
  <body>
    <hr><center>
      <H2>Grundlagen Internet-Technologien</H2>
      <H3>Unterstützte JavaScript-Versionen</H3>

      <script language="JavaScript1.1">
        document.write("JavaScript 1.1 unterstützt<br>");
      </script>

      <script language="JavaScript1.2">
        document.write("JavaScript 1.2 unterstützt<br>");
      </script>

      <script language="JavaScript1.3">
        document.write("JavaScript 1.3 unterstützt<br>");
      </script>

      <script language="JavaScript1.4">
        document.write("JavaScript 1.4 unterstützt<br>");
      </script>

      <script language="JavaScript1.5">
        document.write("JavaScript 1.5 unterstützt<br>");
      </script>

      <script language="JavaScript1.6">
        document.write("JavaScript 1.6 unterstützt<br>");
      </script>

      <script language="JavaScript1.7">
        document.write("JavaScript 1.7 unterstützt<br>");
      </script>

      <script language="JavaScript1.8">
        document.write("JavaScript 1.8 unterstützt<br>");
      </script>

    </center><hr>

  </body>
</html>
zrvwa01@infodienste =>
```





Grundlagen Internet-Technologien

Unterstützte JavaScript-Versionen

JavaScript 1.1 möglich
JavaScript 1.2 möglich
JavaScript 1.3 möglich

Grundlagen Internet-Technologien

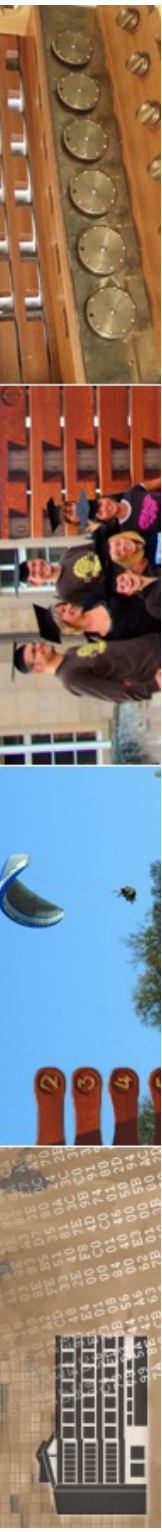
Unterstützte JavaScript-Versionen

JavaScript 1.1 möglich
JavaScript 1.2 möglich
JavaScript 1.3 möglich
JavaScript 1.4 möglich
JavaScript 1.5 möglich
JavaScript 1.6 möglich
JavaScript 1.7 möglich

Grundlagen Internet-Technologien

Unterstützte JavaScript-Versionen

JavaScript 1.1 möglich
JavaScript 1.2 möglich
JavaScript 1.3 möglich
JavaScript 1.4 möglich
JavaScript 1.5 möglich



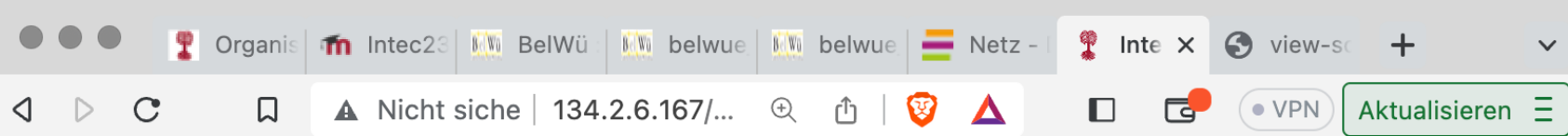
Internet-Technologien: erstes J x +

Nicht sicher | 134.2.6.167/~zrvwa01/vorlesung/javascript/version.html

Grundlagen Internet-Technologien

Unterstützte JavaScript-Versionen

- JavaScript 1.1 möglich
- JavaScript 1.2 möglich
- JavaScript 1.3 möglich
- JavaScript 1.4 möglich
- JavaScript 1.5 möglich
- JavaScript 1.6 möglich
- JavaScript 1.7 möglich



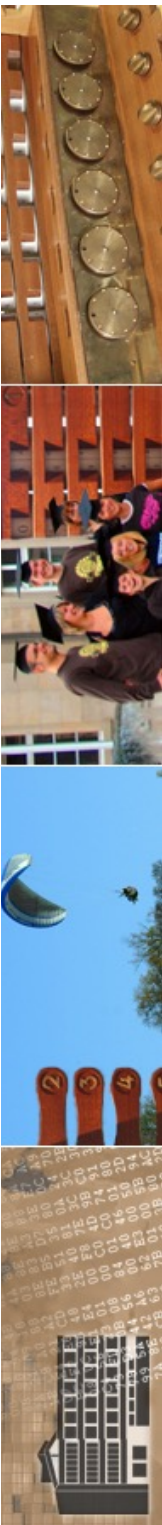
Grundlagen Internet-Technologien

Unterstützte JavaScript-Versionen

JavaScript 1.1 möglich
JavaScript 1.2 möglich
JavaScript 1.3 möglich
JavaScript 1.4 möglich
JavaScript 1.5 möglich

des „Anti-Tag“

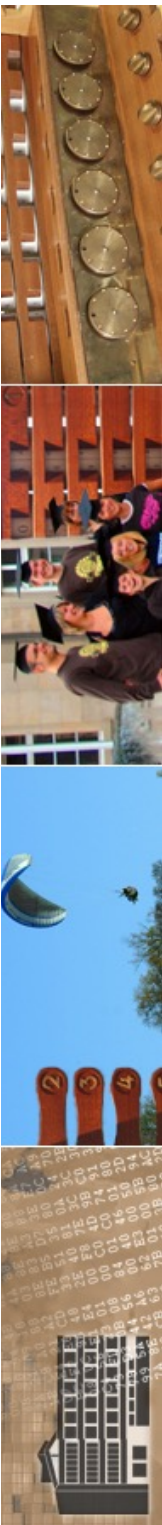
- es gibt auch die umgekehrte Intention:
 - `<noscript>`
 - ...
 - `</noscript>`
 - ist ein HTML-Tag
 - wird *nur* angezeigt, wenn kein JavaScript ausgeführt werden kann
 - sinnvoll für Warnungen, dass JavaScript fehlt





Debugging von JavaScript

- über Browser-Funktionen
 - Fehler-Konsole von Firefox
 - JavaScript-Konsole von Chrome
 - Firebug-Plugin
 - Entwickler-Tools von Chrome

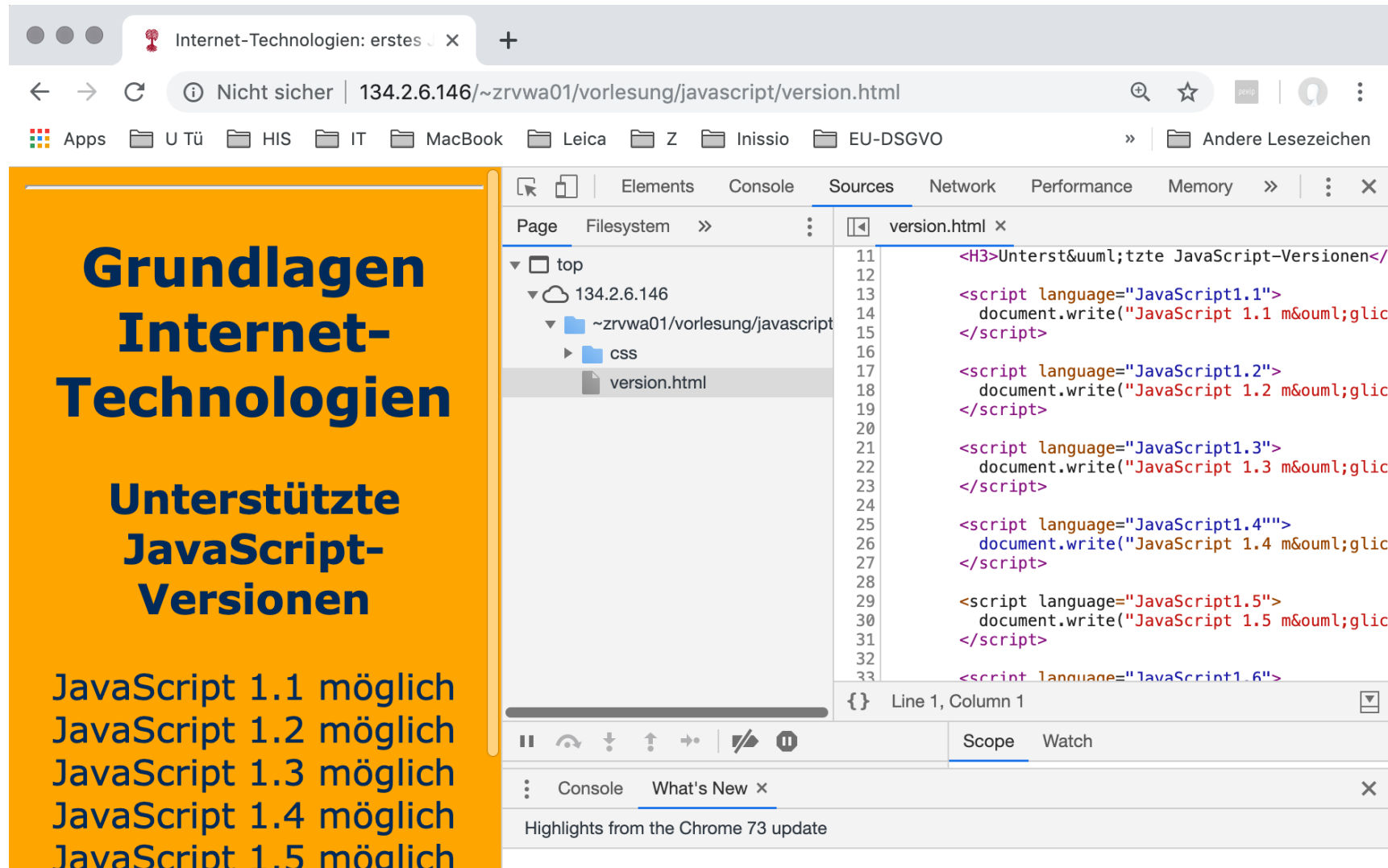


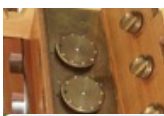


Grundlagen Internet- Technologien

Unterstützte JavaScript- Versionen

JavaScript 1.1 möglich
JavaScript 1.2 möglich
JavaScript 1.3 möglich
JavaScript 1.4 möglich
JavaScript 1.5 möglich





134.2.6.146/~zrvwa01/vorlesung/javascript/version.html

Grundlagen Internet-Technologien

Unterstützte JavaScript-Versionen

JavaScript 1.1 möglich
JavaScript 1.2 möglich
JavaScript 1.3 möglich
JavaScript 1.4 möglich
JavaScript 1.5 möglich
JavaScript 1.6 möglich
JavaScript 1.7 möglich

Elements
Console
Sources
Network
Performance

```

<!doctype html>
<html>
  <head>...</head>
  <body>
    <hr>
    <center>
      <h2>Grundlagen Internet-Technologien</h2>
      <h3>Unterstützte JavaScript-Versionen</h3>
      ...
      <script language="JavaScript1.1">
        document.write("JavaScript 1.1 möglich<br>");
      </script>
      "JavaScript 1.1 möglich"
      <br>
      <script language="JavaScript1.2">
        document.write("JavaScript 1.2 möglich<br>");
      </script>
      "JavaScript 1.2 möglich"
      <br>
      <script language="JavaScript1.3">
        document.write("JavaScript 1.3 möglich<br>");
      </script>
      "JavaScript 1.3 möglich"
      <br>
      <script language="JavaScript1.4">
        document.write("JavaScript 1.4 möglich<br>");
      </script>
    </center>
  </body>
</html>

```

html
body
center
script

Styles
Event Listeners
DOM Breakpoints
Properties
Accessibility

Filter
:hov .cls +

```

element.style {
}

script {
  display: none;
}

Inherited from center
center {
  text-align: -webkit-center;
}

Inherited from body
body {
  background-color: #FFA500;
  font-family: Verdana, Arial, Helvetica, sans-serif;
  color: #002A5A;
}

```

margin
border
padding
auto x auto

Filter
☐ Show all

```

> color
  rgb(0, 42, 90)
> display
  none
> font-family

```

© 2026 Universität
Tübingen

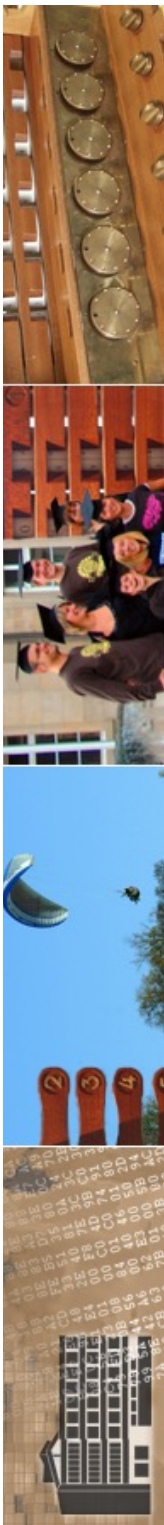
INF3171 Grundlagen Internet-Technologien Sommersemester 2025

45



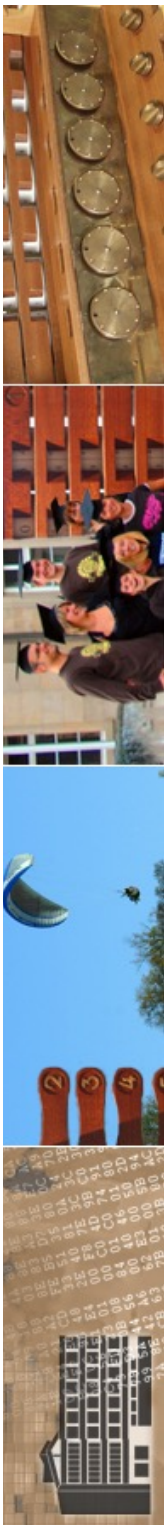
Variablen in JavaScript

- JavaScript ist eine **Scriptsprache**:
Variablen sind **nicht explizit typisiert**
- optionales Schlüsselwort **var** :
var sinn_des_lebens = 42;
- Gültigkeit:
 - Variablen in Methoden sind **lokal**
 - Variablen im Hauptblock sind **global**



dynamische Datentypen

- intern verwendet JavaScript folgende Datentypen:
 - ganze Zahlen
 - Gleitkommazahlen
 - Zeichenketten
 - Boolesche Werte
 - Objekte
- Typabfrage: **`typeof`** **variablenbezeichner**





Zeichenketten

- Zeichenketten werden durch ' oder " begrenzt
- Zeichenkettenkomposition durch + Operator

```
- var s = "Hello "; var t = "World";  
  s = s + t;
```

```
var n = s.length;    // n bekommt Wert 11  
var u = s.substring(1,4); // ell
```

```
// fuer Informatiker:  
var v = new String("FBI");
```



```
zrvwa01@infodienste => more variablen.html
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" type="text/css" href="./css/webkompendium.css">
    <link rel="shortcut icon" href="./css/favicon.ico">
    <title>Internet-Technologien: JavaScript</title>
  </head>
  <body>

    <hr><center>
      <h2>Einführung Internet-Technologien</h2>
      <h3>Variablen in JavaScript</h3>

      <script type="text/javascript">
        <!--
          var sinn_des_lebens = 042;
          var wk = "Grundlagen " + "Internet-Technologien";
          var s = wk.substring(11,19)

          document.write("Substring: "+s);
          document.write("<BR /> Wert der Variablen: "+sinn_des_lebens);

          // -->
        </script>

      </center><hr>

    </body>
  </html>
zrvwa01@infodienste =>
```



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

    <hr><center>
      <h2>Einführung Internet-Technologien</h2>
      <h3>Variablen in JavaScript</h3>

      <script type="text/javascript">
        <!--
        var sinn_des_lebens = 042;
        var wk = "Grundlagen " + "Internet-Technologien";
        var s = wk.substring(11,19)

        document.write("Substring: "+s);
        document.write("<BR /> Wert der Variablen: "+sinn_des_lebens);

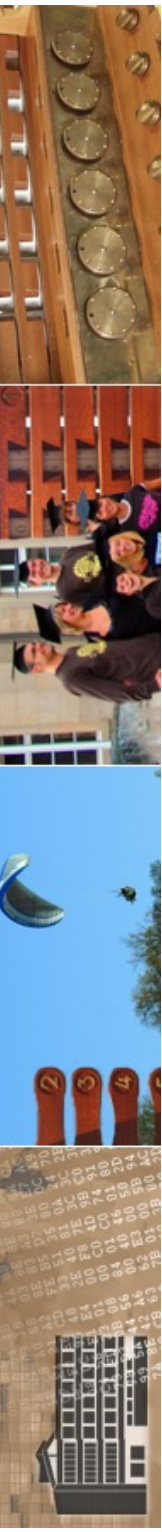
        // -->
      </script>

    </center><hr>

  </body>
</html>
```

~



Internet-Technologien: JavaScript x +

Nicht sicher | 134.2.6.167/~zrvwa01/vorlesung/javascript/variablen.html

Einführung Internet-Technologien

Variablen in JavaScript

Substring: Internet
Wert der Variablen: 34



Arrays in JavaScript

- Arrays sind Listen
- numerischer Index genauso wie assoziatives Array
 - `var liste = new array();`
`liste[0] = 42;`
`liste["tuebingen"] = "FBI";`
- Listen-Methoden wie
 - `push` `pop` `shift` `sort` ...

Logik, Kontrollstrukturen

- ...das übliche aus Java:
 - `!    ||    &&`
- Kontrollstrukturen: ebenfalls (fast) wie in Java
 - `{ ... }`
 - `if (boole)    {...}    else    {...}`
 - `switch (n) {    case ...    default ... }`
 - `while (boole) {...}`
 - `do {...}    while (boole);`
 - `for (init;boole;expr)`
 - `break    continue`
 - `for (laufvariable in struktur)`



```
zrvwa01@infodienste => more schleifen.html
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<link rel="stylesheet" type="text/css" href="./css/webkompendium.css">
```

```
<link rel="shortcut icon" href="./css/webkompendium.ico">
```

```
<title>Internet-Technologien: JavaScript</title>
```

```
</head>
```

```
<body>
```

```
<hr><center>
```

```
<h2>Grundlagen Internet-Technologien</h2>
```

```
<h3>Schleifen in JavaScript</h3>
```

```
<script type="text/javascript">
```

```
var max = 100;
```

```
// while-Schleife
```

```
var summe = 0;
```

```
var i = 1;
```

```
while (i <= max)
```

```
    summe += i;
```

```
document.write("<BR>Summierung mittels while: " + summe);
```

```
// do-while
```

```
summe = 0; i = 1;
```

```
do {
```

```
    summe += i++
```

```
} while (i <= max);
```

```
document.write("<BR>Summierung mittels do-while: " + summe);
```

```
// for-schleife
```

```
summe = 0;
```

```
for (i = 1; i <= max; i++)
```

```
    summe += i;
```

```
document.write("<BR>Summierung mittels for: " + summe);
```

```
</script>
```

```
</center><hr>
```

```
</body>
```

```
</html>
```

```
zrvwa01@infodienste =>
```



Grundlagen Internet-Technologien

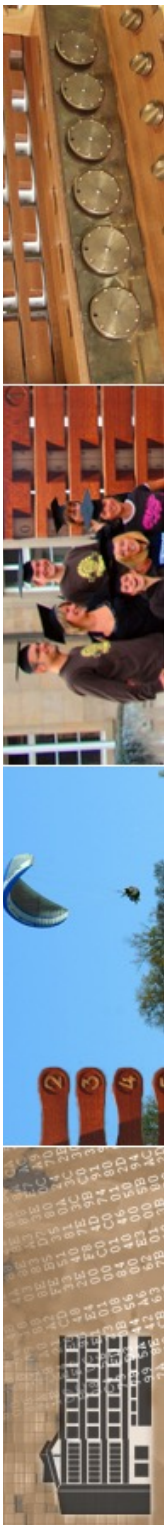
Schleifen in JavaScript

Summierung mittels while: 5050
Summierung mittels do-while: 5050
Summierung mittels for: 5050

$$\sum_{i=1}^{100} i = \frac{n(n+1)}{2}$$

$$n = 100$$







Methoden in JavaScript

- Schlüsselwort `function` :
 - `function fakultaet(n) {`
 - `if (n < 1) return 1;`
 - `return n * fakultaet(n-1);`
 - `}`
- einige besondere Methoden:
 - `alert(arg)`
 - `parseInt(arg)`
 - `parseFloat(arg)`
 - `isNaN(arg)`



Importieren von JavaScript

- JavaScript-Code kann nachgeladen werden
 - `<SCRIPT language="JavaScript" src="file.js"/>`
- zentrale Funktionalität für verteilten Code
- src von gleichem Server wie Ausgangsdokument oder mittels http von anderem Server

```

zrvwa01@infodienste =>
zrvwa01@infodienste => more fakultaet.html
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" type="text/css" href="./css/webkompendium.css">
    <link rel="shortcut icon" href="./css/favicon.ico">
    <title>Internet-Technologien: JavaScript</title>

    <script src="http://134.2.6.146/~zrvwa01/vorlesung/javascript/fakultaet.js"></script>

  </head>
  <body>

    <hr><center>
      <h2>Grundlagen Internet-Technologien</h2>
      <h3>Externes JavaScript</h3>

      <h4> Fakult&auml;
zrvwa01@infodienste => more fakultaet.js
      <script type="text/javascript">
        /*
        * Grundlagen Internet-Technologien
        * rekursive Fakult&auml;t in JavaScript
        */
        function fakultaet(n) {
          if (n < 1) return 1;
          return n * fakultaet(n-1);
        }
zrvwa01@infodienste =>

```





Internet-Technologien: JavaScript

Nicht sicher | 134.2.6.167/~zrvwa01/vorlesung/javascript/fakultaet.html

Grundlagen Internet-Technologien

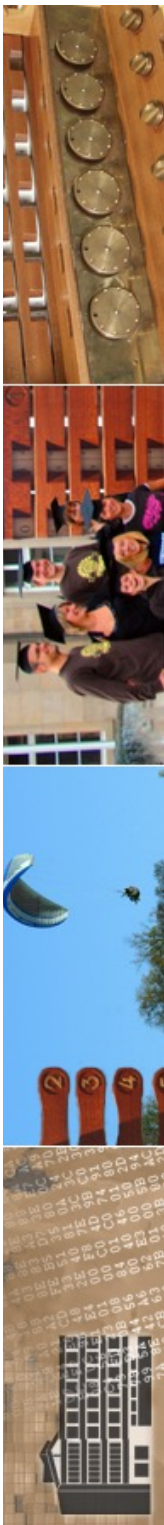
Externes JavaScript

Fakultät(5) = 120



Objektorientierung in JavaScript

- JavaScript ist OO
 - ...zumindest in der Werbung...
- **kein** Schlüsselwort class
- *Klassen werden über eine function definiert, die den Konstruktor bildet*
 - keine Klassen, Instanzen, ... - rein funktionaler Zugang
 - keine Vererbung im eigentlichen Sinne da keine Klasse
 - keine Datenkapselung



```
[zrvwa01@infodienste => more Buch.js
/*
 * Grundlagen Internet-Technologien
 * Objektorientierung in JavaScript
 */

// Konstruktor
function Buch(titel, autor, verlag, jahr) {

    this.titel = titel;
    this.autor = autor;
    this.verlag = verlag;
    this.jahr = jahr;

    this.toString = toString;
    this.htmlRow = htmlRow;

} // Konstruktor

// reine Textausgabe
function toString() {
    return this.autor + ": "+this.titel+" ("+this.verlag+", "+this.jahr+");
} // toString

// HTML-Tabellenzeile
function htmlRow() {
    return "<tr><td>"+this.titel+"</td><td>"+this.autor+
        "</td><td>"+this.verlag+"</td><td>"+this.jahr+"</td></tr>";
} // htmlRow
[zrvwa01@infodienste =>
```



```
zrvwa01@infodienste => more buchhandlung2.html
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <link rel="stylesheet" type="text/css" href="./css/webkompendium.css">
    <link rel="shortcut icon" href="./css/springer_icon.ico">
    <title>Internet-Technologien: JavaScript</title>

    <script src="Buch.js" type="text/javascript"></script>

    <script type="text/javascript">
      var wk = new Buch("Webkompendium", "Thomas Walter","Springer",2008);
      var mf = new Buch("MediaFotografie", "Thomas Walter","Springer",2005);
    </script>

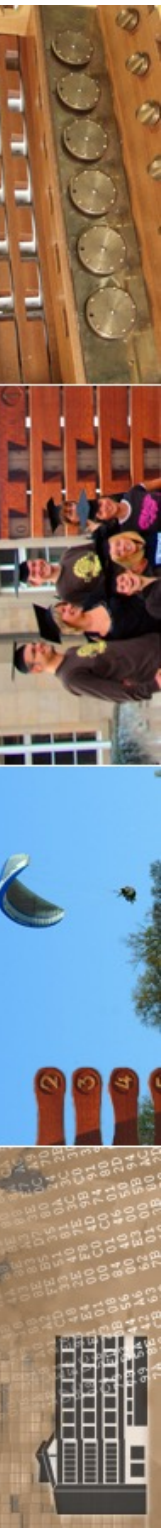
  </head>
  <body>

    <hr><center>
      <h2>Grundlagen Internet-Technologien</h2>
      <h3>Objektorientierung in JavaScript</h3>

      <table>
        <script type="text/javascript">
          document.write(wk.htmlRow());
          document.write(mf.htmlRow());
        </script>

      </table>
    </center><hr>

  </body>
</html>
zrvwa01@infodienste =>
```



Internet-Technologien: JavaScript x +

Nicht sicher | 134.2.6.167/~zrvwa01/vorlesung/javascript/buchhandlun...

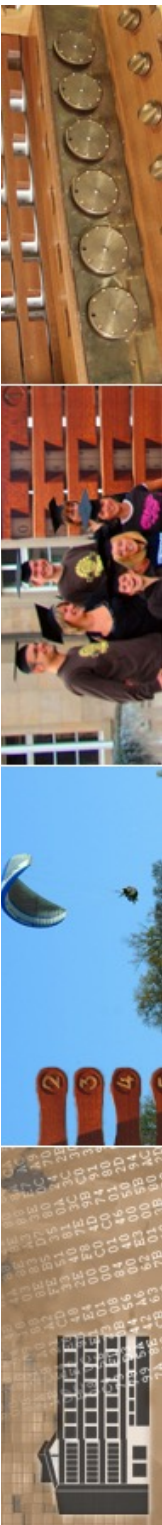
Grundlagen Internet-Technologien

Objektorientierung in JavaScript

Webkompendium Thomas Walter Springer 2008
MediaFotografie Thomas Walter Springer 2005

JavaScript und Sicherheit

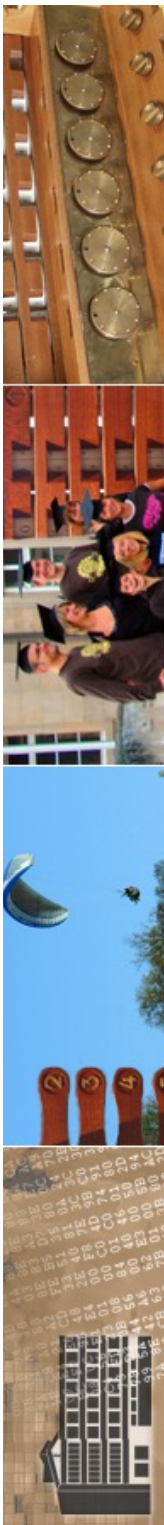
- JavaScript ist aktiver Code, der im Browser ausgeführt wird:
potenzielles Sicherheitsrisiko
- Grundprinzip der Sicherheit von JavaScript:
Sandbox
 - JavaScript-Programm „eingesperrt“ in Sandkasten, etwa ohne Zugriff auf das Dateisystem





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"`



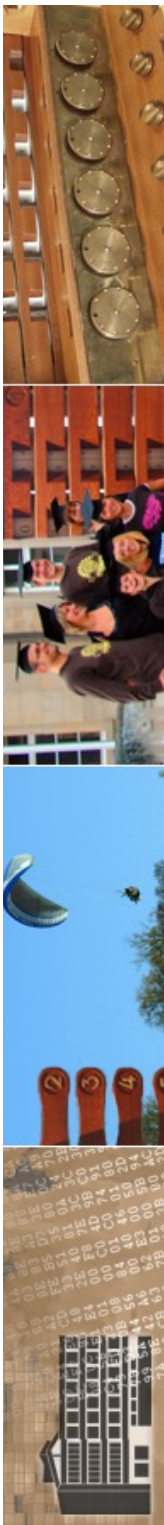
JavaScript und Browser-Erscheinung

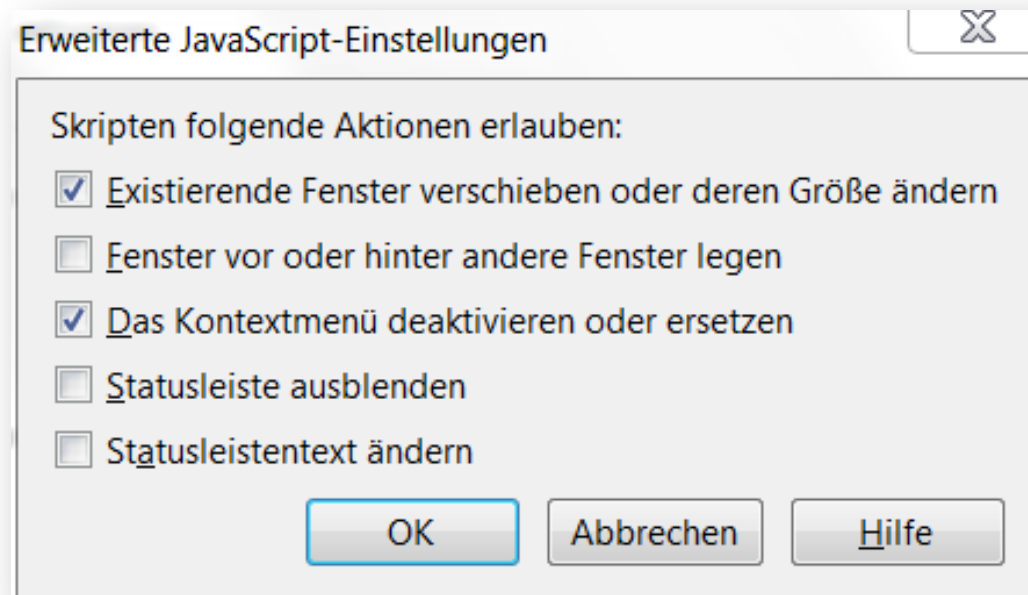
- von Anfang an kritisch:

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





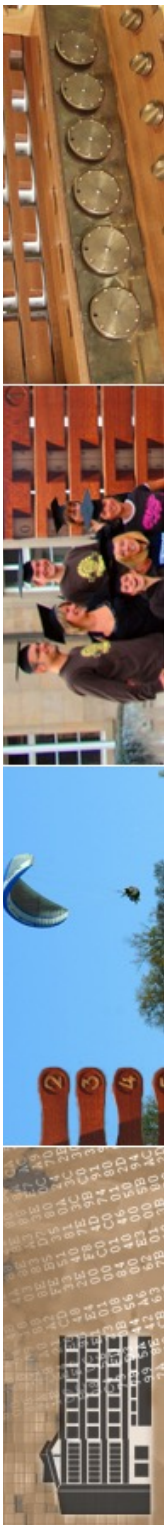
destruktives JavaScript

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

einfaches

`window.open ( . . . )`

wird deshalb von Popup-Blocker blockiert



XSS

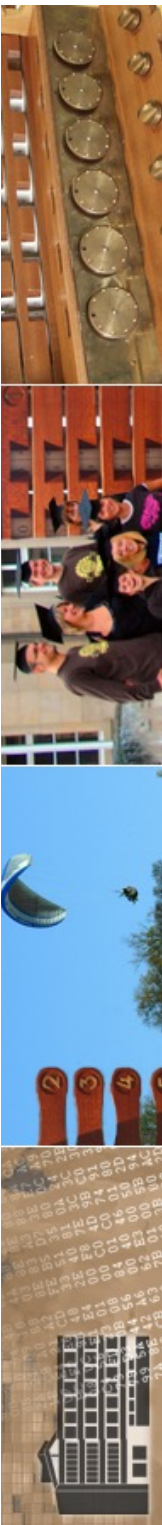
- Cross-Site-Scripting: Einbinden von fremden (schädlichen) JavaScript in eine andere, an sich harmlose und vertrauenswürdige Seite

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.

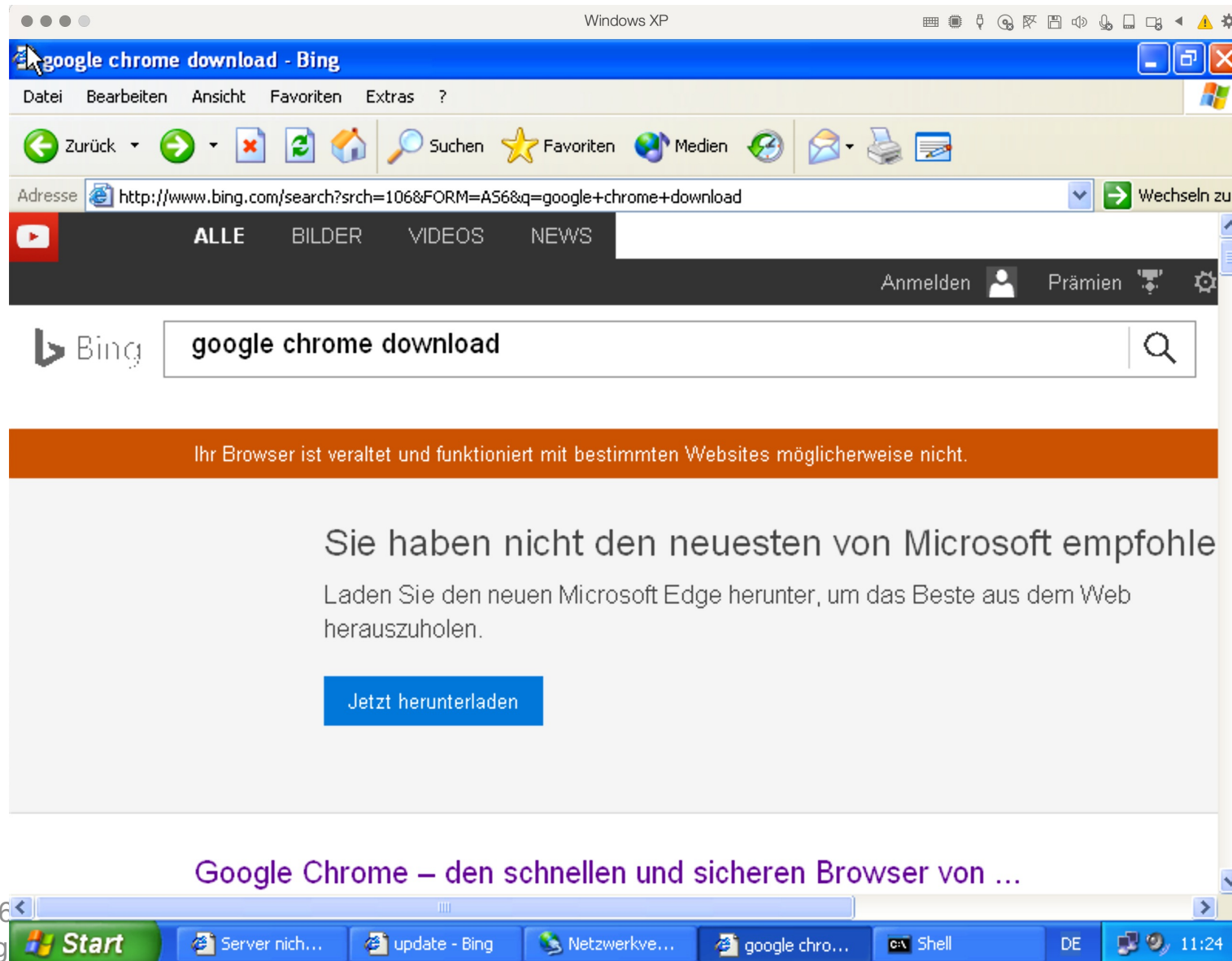
- https://wiki.selfhtml.org/wiki/JavaScript/Tutorials/Cross_Site_Scripting
- Cookie-Diebstahl
- Keylogging
- Phishing

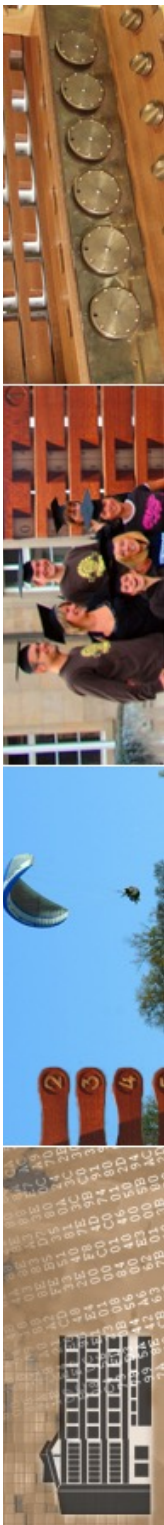
XSS

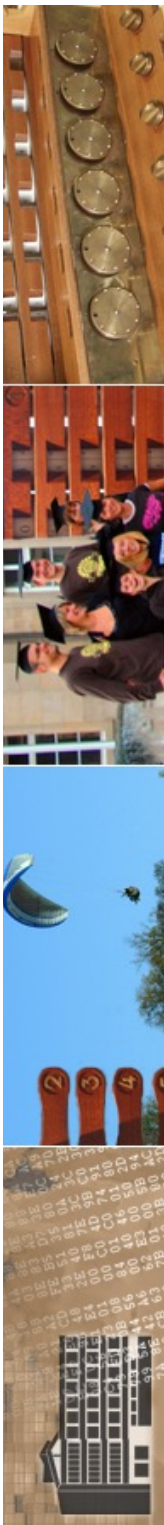
- Um XSS zu verhindern, muss verhindert werden, dass eingegebene Daten Skripte enthalten können
 - das ist teilweise schwieriger als es zunächst erscheint



ganz alte Browser







...und nun...

- haben wir das Prinzip der clientseitigen Scriptsprache JavaScript kennen gelernt
- als nächstes:

JavaScript und DOM:

die **Verbindung von JavaScript zu HTML**

