

Aktuelle Programmiersprachen

KAI JÄGER
JavaScript

1994

2

Microsoft[®]







Konkurrenz

Browser War



(fürs Erste)

1994



Wettrüsten

Features

User

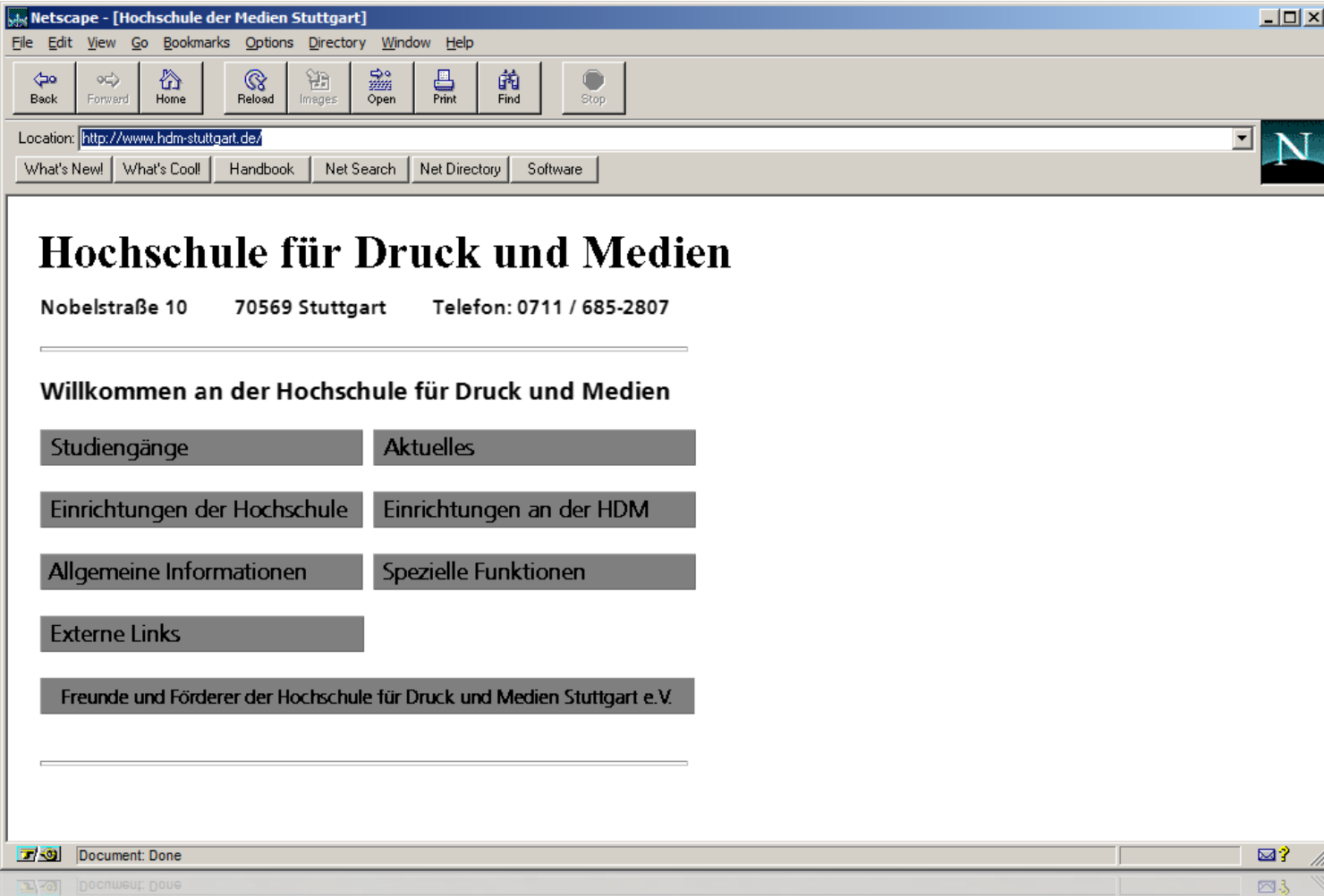
Erfolg



1994

Vorsprung



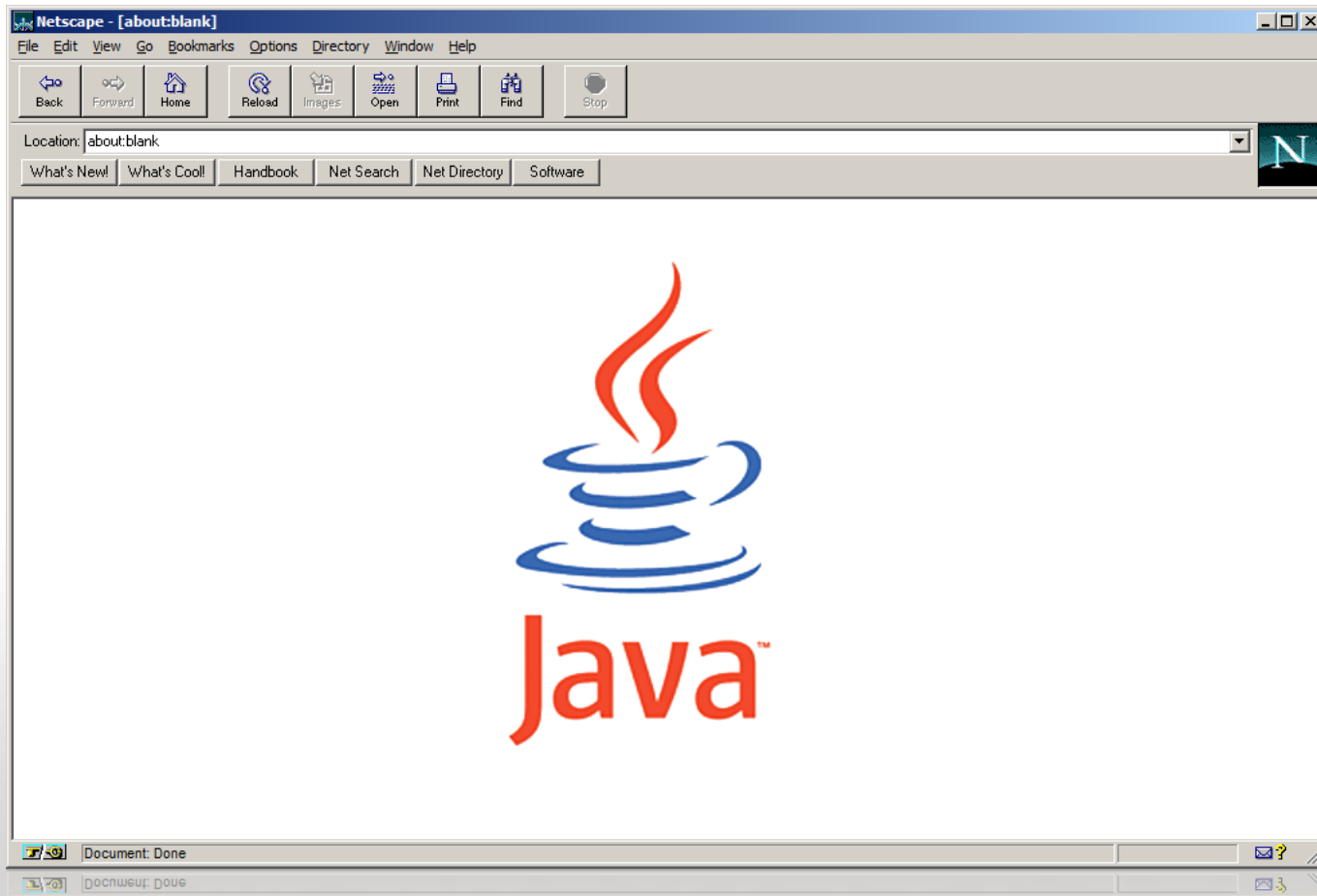




Sun®
microsystems









Vorsprung





Programmierer

~~Programmierer~~





Kai Jäger / Aktuelle Programmiersprachen / Wintersemester 2006/07.



Kompromiss



einfach



Mocha

1995





Brendan Eich

Chief Technology Officer





Brendan Eich

LiveScript



Sun®
microsystems



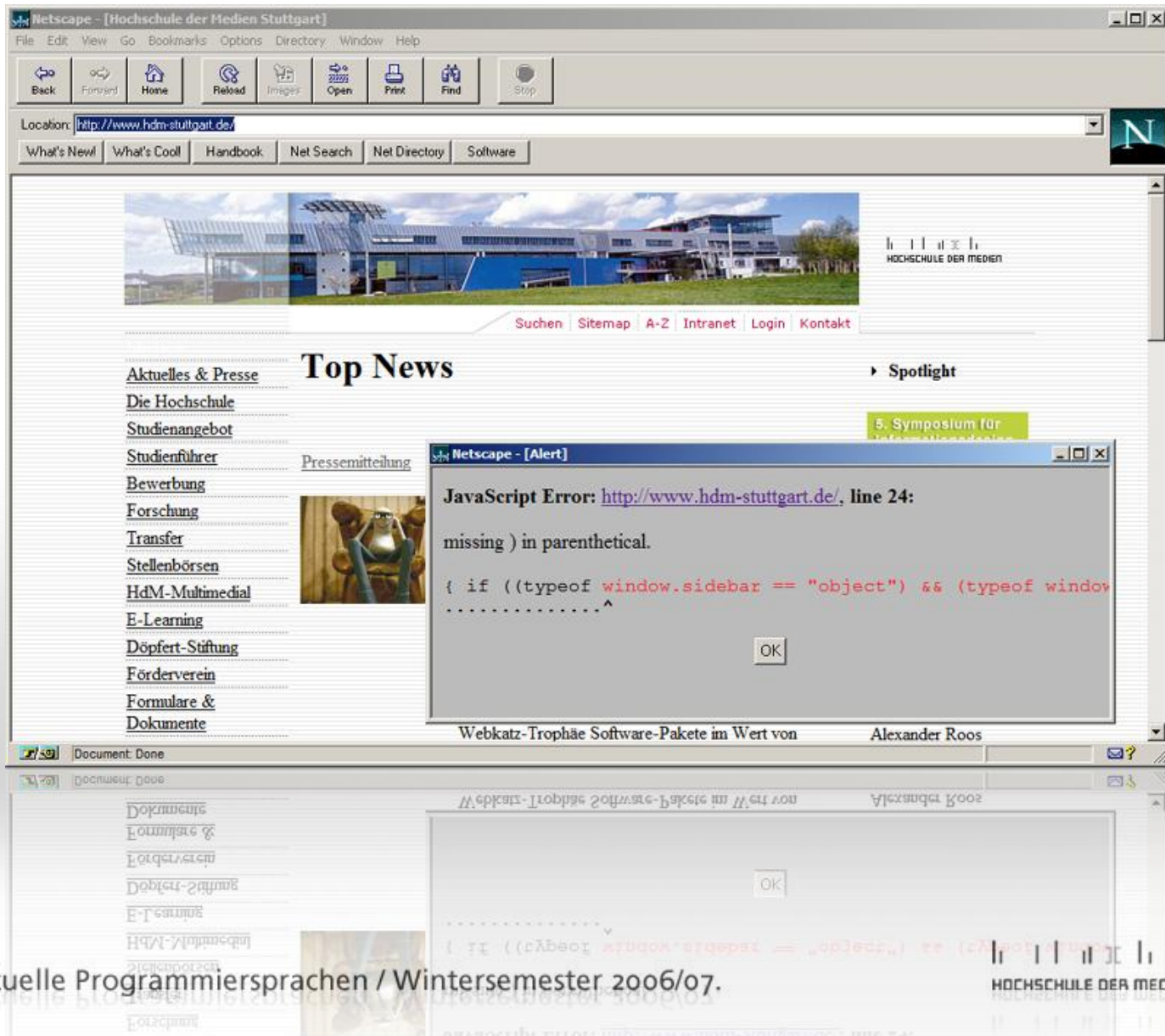






populär

1995



</geschichte>

heute

ECMA-262

JavaScript 1.7







Search Mail Search the Web

[Show search options](#) [Create a filter](#)

[Compose Mail](#)

« [Back to Inbox](#) [Archive](#) [More actions...](#)

1 of 1

Inbox

- [Starred](#) ★
- [Sent Mail](#)
- [All Mail](#)
- [Spam](#)
- [Trash](#)

▼ **Labels**

[Edit labels](#)

Gmail is different. Here's what you need to know. [Inbox](#) [Apply label...](#)

★ **Gmail Team** <gmail-noreply@google.com> to me [More options](#) 9:14pm (2 minutes ago)

First off, welcome. And thanks for agreeing to help us test Gmail. By now you probably know the key ways in which Gmail differs from traditional webmail services. Searching instead of filing. A free gigabyte of storage. Messages displayed in context as conversations.

So what else is new?

Gmail has many other special features that will become apparent as you use your account. You'll find answers to most of your questions in our searchable [help](#) section, which includes a [Getting Started guide](#). You'll find information there on such topics as:

- How to use address auto-complete
- Setting up filters for incoming mail
- Using advanced search options

You may also have noticed some text ads or related links to the right of this message. They're placed there in the same way that ads are placed alongside Google search results and, through our AdSense program, on content pages across the web. The matching of ads to content in your Gmail messages is performed entirely by computers; never by people. Because the ads and links are matched to information that is of interest to you, we hope you'll find them relevant and useful.

You're one of the very first people to use Gmail. Your input will help determine how it evolves, so we encourage you to send your feedback, suggestions and questions to us. But mostly, we hope you'll enjoy experimenting with Google's approach to email.

Speedy Delivery,

The Gmail Team

p.s. You can sign in to your account any time by visiting <http://gmail.google.com>

[Reply](#) [Forward](#)

- [Open in new window](#)
- [Print conversation](#)
- [Expand all](#)

Sponsored Links

[Free HighRankings Secrets](#)

Learn meta tags, page rank, SEO, linking, writing for search engines
highrankings.com/search-engine-help

Related Pages

[PHP Warped/Web Net - PHP Scripts: PHPEmailArchiver](#)

Free PHP Scripts to download including a search engine, mailing ...
php.warpedweb.net

[Searching With Opera](#)

Opera has a built in search function that lets you search the Web ...
www.opera.com

[About these links](#)

Reply Forward

p.s. You can sign in to your account any time by visiting <http://gmail.google.com>

The Gmail Team

Speedy Delivery,





z. B. "10 market st, san francisco" oder "Hotels in der Nähe von LAX"

Nobelstraße 10, stuttgart, germany

Maps-Suche

Karte durchsuchen

Firmen finden

Route berechnen

Karten

[Drucken](#) | [E-Mail](#) | [URL zu dieser Seite](#)

Karte | Satellit | Hybrid



©2006 Google - Grafiken ©2006 DigitalGlobe, GeoContent - Nutzungsbedingungen

©2006 Google - Grafiken ©2006 DigitalGlobe, GeoContent - Nutzungsbedingungen




logout Send feedback
 Terms of Service Privacy Policy

fluxiom


Assets

1 of 14 assets — 5 GB available


Reset	My assets	Today	Last week	Favourites	Shared	Documents	Images	Untagged
9rules	cars							




JPG 2005-Lotus-Exige-Esp
2 hours ago




JPG F430 wheel
2 hours ago




JPG deviance-final
2 hours ago




JPG 2005-Nissan-GT-R-Pr
2 hours ago




JPG 2006-Porsche-Cayman
2 hours ago




JPEG mist10k
3 hours ago




ZIP k2 backup 5-10-06
3 hours ago




PHP def-subscribe
3 hours ago




JPG 9rad
3 hours ago




PSD xifooter_square
3 hours ago




PSD header_square
3 hours ago



DOC podcasting paper
2 weeks ago



ZIP defiance redesign
2 weeks ago



ZIP April 24 06 k2 backup
2 weeks ago

Add
Edit
Preview
Tags
Details
Download
Share
RSS Feed
Delete



alles gut?

The world's most misunderstood programming language.

—Douglas Crockford, Architect | Yahoo!



JavaScript



Script

Java
light
TRADEMARK
REGD

get a **taste** of it™

JavaScript

Self

Lisp

Die Sprache

JavaScript

Syntax



```
if (...) {  
} else if (...) {  
} else {  
}
```

```
switch (...) {  
  case ....:  
    break;  
}
```

```
for (...) {  
}
```

```
while (...) {  
}
```

```
do {  
}  
while (...)
```

```
try {  
} catch (...) {  
} finally {  
}
```

Typisierung

Dynamisch typisiert

```
var x;
```

```
var x;  
x = 17;
```

```
var x;  
x = 17;  
x = "Hello World";
```

```
var x;  
x = 17;  
x = "Hello World";  
x = false;
```

Komplexe Typen

Object

Arrays

Funktionen

```
function greet(name) {  
    document.write("Hello " + name);  
}
```

```
var greet = function (name) {  
    document.write("Hello " + name);  
}
```

Anonyme Funktion

```
function doGreet(f) {  
    f("Fritz");  
}  
  
doGreet(  
    function (name) {  
        document.write("Hello " + name);  
    }  
);
```

```
function makeGreet() {  
    return function (name) {  
        document.write("Hello " + name);  
    }  
}
```

Higher order functions

Funktionalen Sprachen

Lisp

```
function outer() {  
    function inner(name) {  
        document.write("Hello " + name);  
    }  
    inner("Fritz");  
}
```

Funktionen in Funktionen

```
function makeGreet(name) {  
    var salutation = "Hello ";  
    return function() {  
        alert(salutation + name);  
    }  
}  
  
var greetFunc = makeGreet("Fritz");  
greetFunc();
```

```
function makeGreet(name) {  
    var salutation = "Hello ";  
    return function() {  
        alert(salutation + name);  
    }  
}
```

```
var greetFunc = makeGreet("Fritz");  
greetFunc();
```

```
function makeGreet(name) {  
    var salutation = "Hello ";  
    return function() {  
        alert(salutation + name);  
    }  
}
```

```
var greetFunc = makeGreet("Fritz");  
greetFunc();
```

```
function makeGreet(name) {  
    var salutation = "Hello ";  
    return function() {  
        alert(salutation + name);  
    }  
}
```

```
var greetFunc = makeGreet("Fritz");  
greetFunc();
```

```
function makeGreet(name) {  
    var salutation = "Hello ";  
    return function() {  
        alert(salutation + name);  
    }  
}
```

```
var greetFunc = makeGreet("Fritz");  
greetFunc();
```

```
function makeGreet(name) {  
    var salutation = "Hello ";  
    return function() {  
        alert(salutation + name);  
    }  
}
```

```
var greetFunc = makeGreet("Fritz");  
greetFunc();
```

```
function makeGreet(name) {  
    var salutation = "Hello ";  
    return function() {  
        alert(salutation + name);  
    }  
}
```

```
var greetFunc = makeGreet("Fritz");  
greetFunc();
```

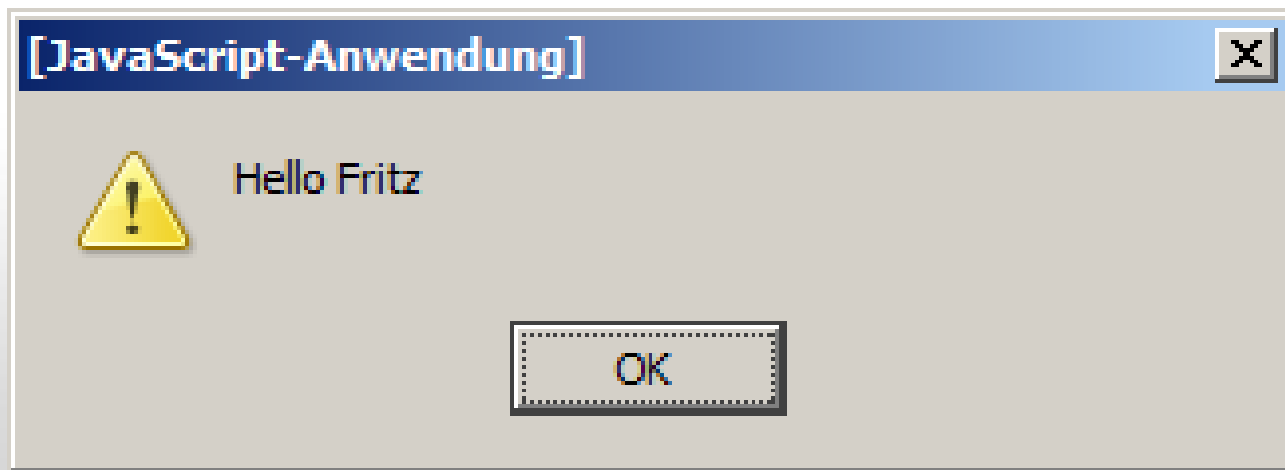
```
function makeGreet(name) {  
    var salutation = "Hello ";  
    return function() {  
        alert(salutation + name);  
    }  
}
```

```
var greetFunc = makeGreet("Fritz");  
greetFunc();
```

```
function makeGreet(name) {  
    var salutation = "Hello ";  
    return function() {  
        alert(salutation + name);  
    }  
}  
  
var greetFunc = makeGreet("Fritz");  
greetFunc();
```

```
function makeGreet(name) {  
    var salutation = "Hello ";  
    return function() {  
        alert(salutation + name);  
    }  
}  
  
var greetFunc = makeGreet("Fritz");  
greetFunc();
```

```
function makeGreet(name) {  
    var salutation = "Hello ";  
    return function() {  
        alert(salutation + name);  
    }  
}  
  
var greetFunc = makeGreet("Fritz");  
greetFunc();
```



```
function makeGreet(name) {  
    var salutation = "Hello ";  
    return function() {  
        alert(salutation + name);  
    }  
}  
  
var greetFunc = makeGreet("Fritz");  
greetFunc();
```

Closure

Funktionsabschluss

Objektorientierung

```
var auto = new Object();
```

```
var auto = new Object();  
auto.farbe = "silber";
```

```
var auto = new Object();  
auto.farbe = "silber";  
auto.hupe = function() {  
    alert("Hup!");  
}
```

```
var auto = new Object();  
auto.farbe = "silber";  
auto.hupe = function() {  
    alert("Hup!");  
}  
  
auto.hupe();
```

JavaScript

keine Klassen

Konstrukturen

```
function Auto() {  
}
```

```
var x = new Auto();
```

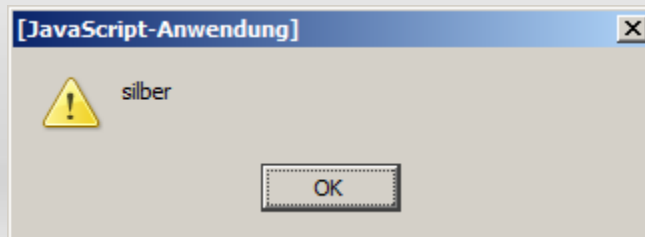
```
function Auto() {  
    this.farbe = "silber";  
}  
  
var x = new Auto();
```

```
function Auto() {  
    this.farbe = "silber";  
}
```

```
var x = new Auto();  
alert(x.farbe);
```

```
function Auto() {  
    this.farbe = "silber";  
}
```

```
var x = new Auto();  
alert(x.farbe);
```



```
function Auto() {  
    this.farbe = "silber";  
}
```

```
var x = new Auto();  
var y = new Auto();  
var z = new Auto();
```

Was ist da passiert?

```
function Auto() {  
    this.farbe = "silber";  
}
```

```
var x = new Object();
```

```
function Auto() {  
    this.farbe = "silber";  
}
```

```
var x = new Object();  
Auto.call(x);
```

```
function Auto() {  
    x.farbe = "silber";  
}
```

```
var x = new Object();  
Auto.call(x);
```

```
function Auto() {  
    this.farbe = "silber";  
}
```

```
var x = new Object();  
x.farbe = "silber";
```

Information hiding

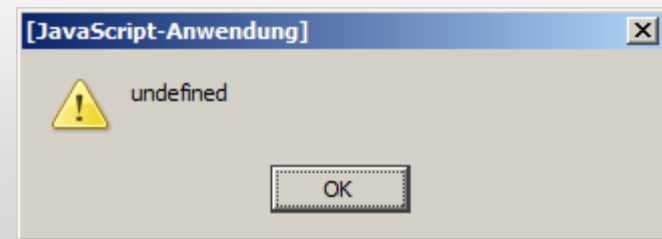
```
function Auto() {  
    var sn = 523523;  
  
    this.farbe = "silber";  
  
    this.getSN = function() {  
        return sn;  
    }  
}  
  
var x = new Auto();
```

```
function Auto() {  
    var sn = 523523;  
  
    this.farbe = "silber";  
  
    this.getSN = function() {  
        return sn;  
    }  
}
```

```
var x = new Auto();  
alert(x.sn);
```

```
function Auto() {  
    var sn = 523523;  
  
    this.farbe = "silber";  
  
    this.getSN = function() {  
        return sn;  
    }  
}
```

```
var x = new Auto();  
alert(x.sn);
```

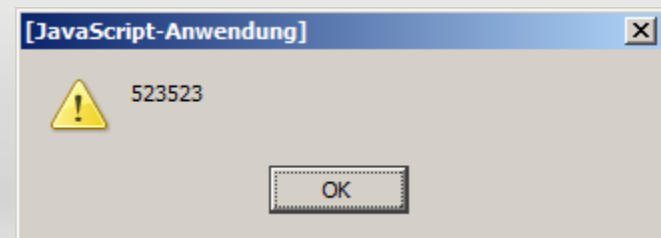


```
function Auto() {  
    var sn = 523523;  
  
    this.farbe = "silber";  
  
    this.getSN = function() {  
        return sn;  
    }  
}
```

```
var x = new Auto();  
alert(x.sn);  
alert(x.getSN());
```

```
function Auto() {  
    var sn = 523523;  
  
    this.farbe = "silber";  
  
    this.getSN = function() {  
        return sn;  
    }  
}
```

```
var x = new Auto();  
alert(x.sn);  
alert(x.getSN());
```



Vererbung

Prototypen

Self

```
function Auto() {  
    this.farbe = "silber";  
}
```

```
function Fahrzeug() {  
    this.raeder = 4;  
}
```

```
function Auto() {  
    this.farbe = "silber";  
}
```

```
function Fahrzeug() {  
    this.raeder = 4;  
}
```

```
function Auto() {  
    this.farbe = "silber";  
}
```

```
Auto.prototype = new Fahrzeug();
```

```
function Fahrzeug() {  
    this.raeder = 4;  
}
```

```
function Auto() {  
    this.farbe = "silber";  
}
```

```
Auto.prototype = new Fahrzeug();
```

```
var x = new Auto();
```

```
function Fahrzeug() {  
    this.raeder = 4;  
}
```

```
function Auto() {  
    this.farbe = "silber";  
}
```

```
Auto.prototype = new Fahrzeug();
```

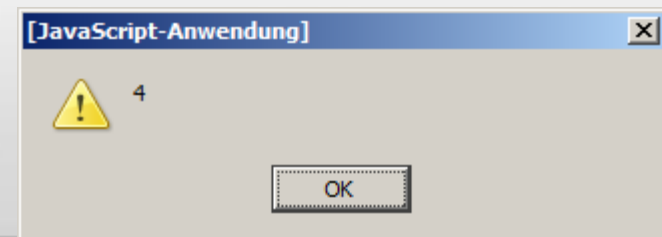
```
var x = new Auto();  
alert(x.raeder);
```

```
function Fahrzeug() {  
    this.raeder = 4;  
}
```

```
function Auto() {  
    this.farbe = "silber";  
}
```

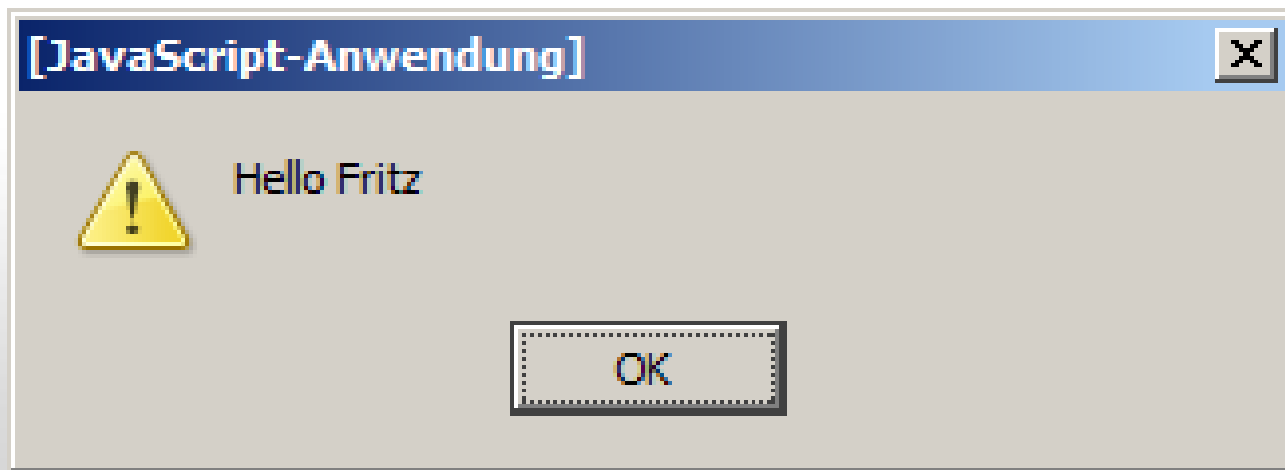
```
Auto.prototype = new Fahrzeug();
```

```
var x = new Auto();  
alert(x.raeder);
```



```
String.prototype.greet = function() {  
    alert("Hello " + this);  
}
```

```
var name = "Fritz";  
name.greet();
```



Fazit

JavaScript

funktional

objektorientiert

Ausdrucksstärke



Performance

~~Performance~~

Syntactic sugar

JavaScript 2.0

Frameworks

**Vielen Dank für die
Aufmerksamkeit.**

Fragen?

Interessante Links

Douglas Crockford's Seiten über JavaScript. Sehr gute Sammlung von Artikeln über OOP in JavaScript und über JSON: <http://javascript.crockford.com/>

JavaScript Closures. Ausführlicher Artikel und FAQ über Closures in JavaScript.
http://jibbering.com/faq/faq_notes/closures.html

Javafied JavaScript. Artikel darüber, wie man in JavaScript Java-ähnlich objektorientiert programmieren kann: <http://www.u.arizona.edu/~jscully/javafication.html>

ECMA-262. Die offizielle JavaScript Spezifikation:
<http://www.ecma-international.org/publications/standards/Ecma-262.htm>

Quirksmode. Riesige Sammlung von Artikeln zu allen möglichen Browser-spezifischen JavaScript Themen. <http://www.quirksmode.org/resources.html>

Using the XMLHttpRequest object. Kurze Einführung in AJAX.
<http://www.jibbering.com/2002/4/httprequest.html>

Ajaxian. Weblog mit News über AJAX.
<http://www.ajaxian.com/>

Quellen

Dieser Vortrag ist stark beeinflusst vom einzigartigen Präsentationsstil von *Prof. Lawrence Lessig* (<http://www.lessig.org/>).

Die Aufmachung der Folien richtet sich teilweise nach Empfehlung von *Garr Reynolds* Website (<http://www.garrreynolds.com/>).

Informationen zur Geschichte von JavaScript entnommen aus *Wikipedia* (<http://en.wikipedia.org/wiki/Javascript>) und *Innovators of the Net* (http://wp.netscape.com/comprod/columns/techvision/innovators_be.html)

Zitat *Douglas Crockford* entnommen von seiner Website (<http://javascript.crockford.com/>).

Einzelheiten zum Sprachstandard von JavaScript stammen aus der ECMA-Spezifikation (<http://www.ecma-international.org/publications/standards/Ecma-262.htm>).

Bilder stammen zum größten Teil aus freien Quellen (<http://www.sxc.hu>). Logos stammen in der Regel von den Firmenwebseiten.