

**Everything you  
know is wrong**

quelle: vvvv group

**JS 1**

**Javascript Basics**

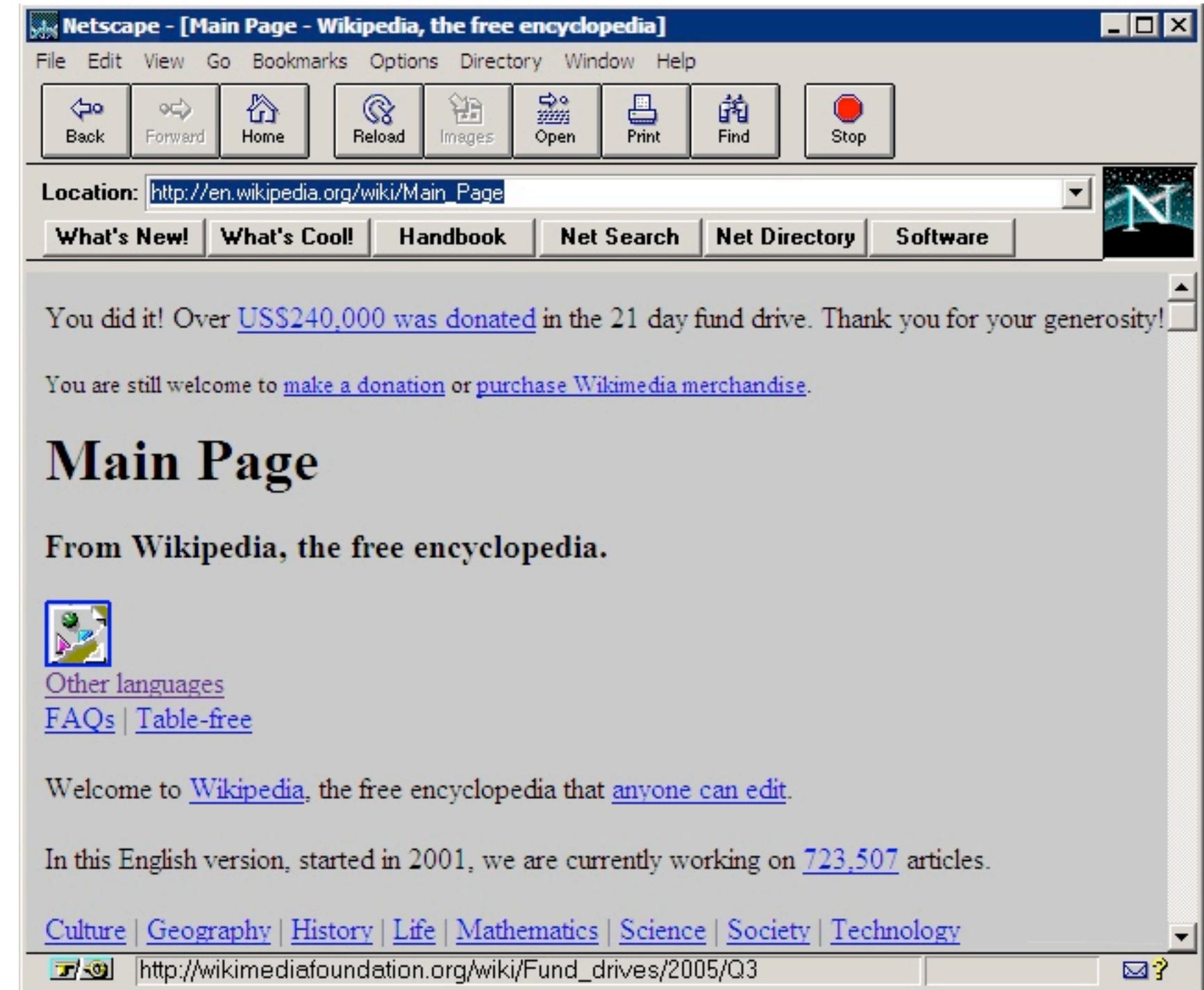
# JavaScript?

ECMAScript!

**Brendan Eich, Netscape 1995**



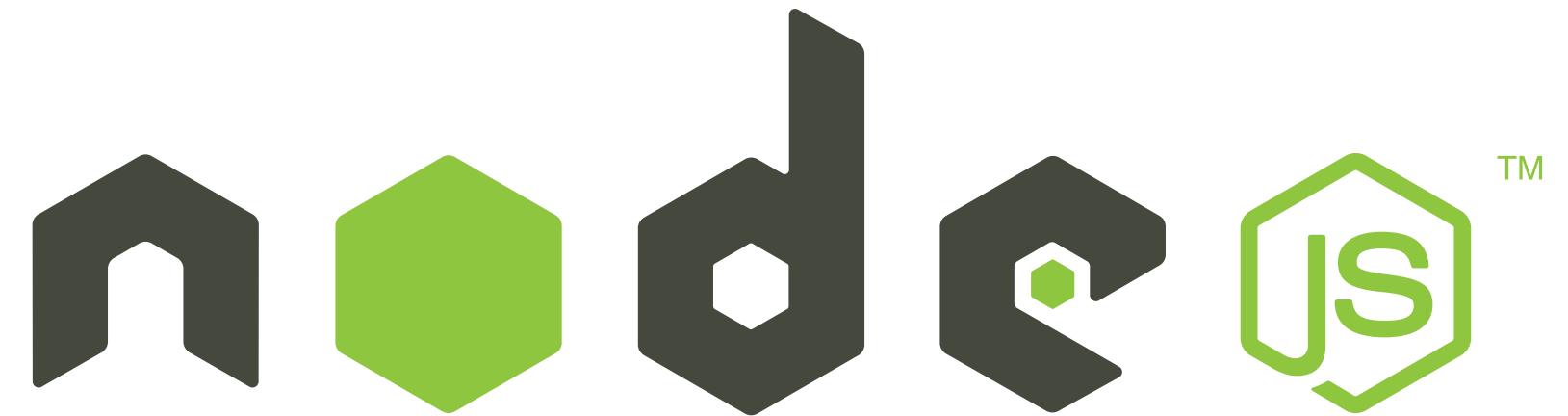
[http://en.wikipedia.org/wiki/Brendan\\_Eich](http://en.wikipedia.org/wiki/Brendan_Eich)



[http://upload.wikimedia.org/wikipedia/en/2/2d/Netscape\\_2.02.png](http://upload.wikimedia.org/wikipedia/en/2/2d/Netscape_2.02.png)

# Javascript heute?

**Immer noch im Web!**



**... und in Adobe Suite, Spielen,  
Servern, u.v.m.**

Und?

**JavaScript** (sometimes abbreviated **JS**) is a prototype-based scripting language that is dynamic, weakly typed, general purpose programming language and has first-class functions. It is a multi-paradigm language, supporting object-oriented,<sup>[5]</sup> imperative, and functional<sup>[1][6]</sup> programming styles.

wtf(„javascript“);

- + einfache Syntax
- + läuft im Webbrowser
- + kann mit jedem Texteditor geschrieben werden
- + es gibt viele Frameworks/Bibliotheken
- + unterstützt viele Programmier-Weisen

- unterstützt viele Programmier-Weisen
- vorsicht mit „var“
- Zugriff auf Dateien nicht einfach
- Welches Framework passt für mein Vorhaben?

# **Beispiel**

```
var name = „Jens“;  
console.log(„Mein name ist“ + name);
```

# PROGRAMMIEREN

<http://de.wikipedia.org/wiki/Programmierung>



# Computerprogramme

<http://de.wikipedia.org/wiki/Computerprogramm>

Ein **Computerprogramm** oder kurz **Programm** ist eine Folge von den Regeln der jeweiligen Programmiersprache genügenden Anweisungen, die auf einem Computer ausgeführt werden können, um damit eine bestimmte Funktionalität zur Verfügung zu stellen.

**Was braucht man zum  
„Programmieren“  
?**

# Speicher

```
var a = 1;
```

```
a = {eins: "zwei"};
```

```
a = [1, 2, "drei"];
```

# Routinen

```
function sayHi() {  
    console.log("Hi");  
}  
  
function addiere(a,b) {  
    return a + b;  
}
```

# **Bedingungen**

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

```
switch(){  
    case 1: break;  
}
```

# Wiederholungen

for (...) {}

while (...) {}

[1,2,3].map(function(el){...});

# Kommentare

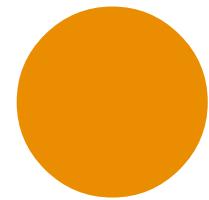
```
/*
```

Kommentare sind wichtig  
und können mehrere  
Zeilen lang sein.

```
*/
```

// oder auch nur ein Zeile

**ACHTUNG! besondere Zeichen:**



‘

’

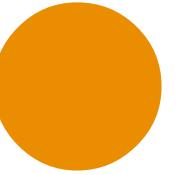
’

’

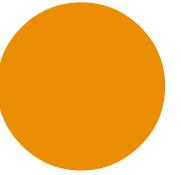
”

”

?



!



**wat(...);**



Nutze die Konsole!

Developer Tools – chrome://newtab/

Elements Resources Network Sources Timeline Profiles Audits Console

```
> console.log("Hello World!");
Hello World!
< undefined
> console._defineGetter_
  __defineGetter__
  __defineSetter__
  __lookupGetter__
  __lookupSetter__
  assert
  constructor
  count
  debug
  dir
  dirxml
  error
  group
  groupCollapsed
  groupEnd
  hasOwnProperty
  info
  isPrototypeOf
  log
```

Errors Warnings Logs

<http://jsconsole.com>

```
console.log(“Hello World!”);
```

```
var name = "Jens";
console.log("Mein name ist" + name);
```

„dynamic & weakly typed“??

```
var eineVariable = 1;
```



Eine Zahl!

```
eineVariable = „Hallo“;
```



Jetzt ein Text!

```
eineVariable = 42;
```

Jetzt wieder eine Zahl

# **Speicher (Variablen)**

```
var name = "Jens";
var status;
status = "spricht über JS";
console.log(name, status);
```

# Operationen

**... auf & mit Variablen**

# **Rechner rechnen:**

```
var level = 1;  
level = level + 1;  
level = level - 1;  
level = level * 1;  
level = level / 2; // ?
```

... mit *kombinierten* Operatoren

```
var level = 1;  
level += 1;  
level -= 1;  
level *= 3;  
level /= 2;
```

**„rechnen“ mit Text**

```
var name = "Klaus";
name = name + " Dieter";
name += " Müller";
var level = 1;
status += name + " ist im ";
status += level + " Level!";
```

NaN

// NaN heißt „Not a Number“  
2 + ”Jens”; // geht

^ JS konvertiert die 2 automatisch in Text, daher OK

2.33 \* ”Jens”; // NaN!

;

# Github?

# Aufgabe:

Finde 5 kreative NaNs