

# JavaScript

**NOTE:** jullie kunnen programmeren dus ik focus op JavaScript specifieke zaken

Een if is een if

Een switch is een switch

Een for is een for

...

# JavaScript

is

waarschijnlijk de meest populaire  
programmeertaal

# JavaScript

is

waarschijnlijk de taal waarvan de  
meeste mensen hun haar uittrekken



uydurmahaber.com

## Small Quiz

Which of the following is the largest?

•A: A Peanut

•B: An Elephant

•C: The Moon

•D: A Kettle

```
<script>  
    alert(typeof (NaN));  
</script>
```

```
<script>
    if (Math.max() > Math.min()) {
        alert("max is bigger");
    }
    else {
        alert("min is bigger");
    }
</script>
```



A woman with short blonde hair, wearing a dark blue long-sleeved top and a light blue pleated skirt, is captured in a joyful spin in a vast field of yellow wildflowers. Her arms are outstretched, and her skirt is flaring out. In the background, majestic, rugged mountains with patches of snow rise against a clear blue sky with a few wispy clouds. The overall scene conveys a sense of freedom and happiness.

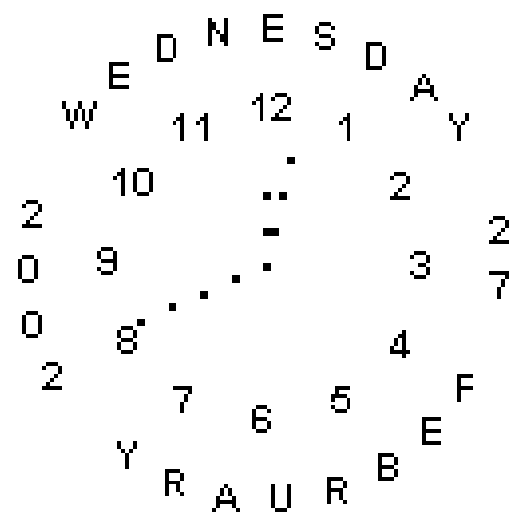
I <3 JavaScript



```
var v = "pretty normal";
```

```
var ୧_୦益୦_୧ = "crazy stuff";
```

[illegible]



# Het verhaal



THE BOLD AND THE BEAUTIFUL





**James Arthur Gosling**



<b>Born</b>	May 19, 1955 (age 58) Near Calgary, Alberta, Canada
<b>Residence</b>	San Francisco Bay Area, California, United States
<b>Nationality</b>	Canadian
<b>Institutions</b>	Sun Microsystems Oracle Corporation Google Liquid Robotics <sup>[1]</sup> Typesafe Inc.
<b>Alma mater</b>	Carnegie Mellon University University of Calgary
<b>Thesis</b>	<i>Algebraic Constraints</i> <a href="#">[1]</a> (1983)
<b>Doctoral advisor</b>	Bob Sproull <sup>[2]</sup>
<b>Known for</b>	Java (programming language)
<b>Notable awards</b>	Officer of the Order of Canada
	<b>Website</b> <a href="http://nighthacks.com">nighthacks.com</a> <a href="#">[1]</a>



Place:

done

# Welcome to HotJava!

After you have read the license, click on Accept or Reject to proceed.



1992 **1995** 1996 1998 1999 2005 2009 2011 2013





Location: [Welcome](#) [What's New!](#) [What's Cool!](#) [Questions](#) [Net Search](#) [Net Directory](#)

[Autos](#)  
[Careers & Jobs](#)  
[Computing](#)  
[Home & Real Estate](#)  
[Shopping](#)  
[Style](#)  
[Travel](#)

[Tools](#) • [Browser Central](#) • [White Pages](#) • [Yellow Pages](#) • [Get AOL Broadband](#)  
• [Download Netscape 7.0](#) • [Local Guide](#) • [My Netscape](#) • [Maps](#) • [Calendar](#) • [More Tools](#)

[CNN.com News](#) • [Get the latest news from CNN.com....](#)



**Brendan Eich**



# ~~Mocha~~ LiveScript

**Born** 1961 (age 51–52)  
Pittsburgh, Pennsylvania

***Alma mater*** University of Illinois at Urbana–Champaign

**Occupation** CTO, Mozilla Corporation

**Known for** JavaScript

**Website**  
brendaneich.com

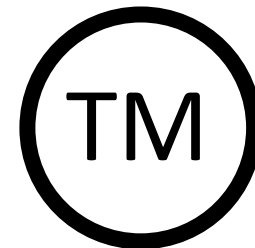


~~LiveScript~~

JavaScript

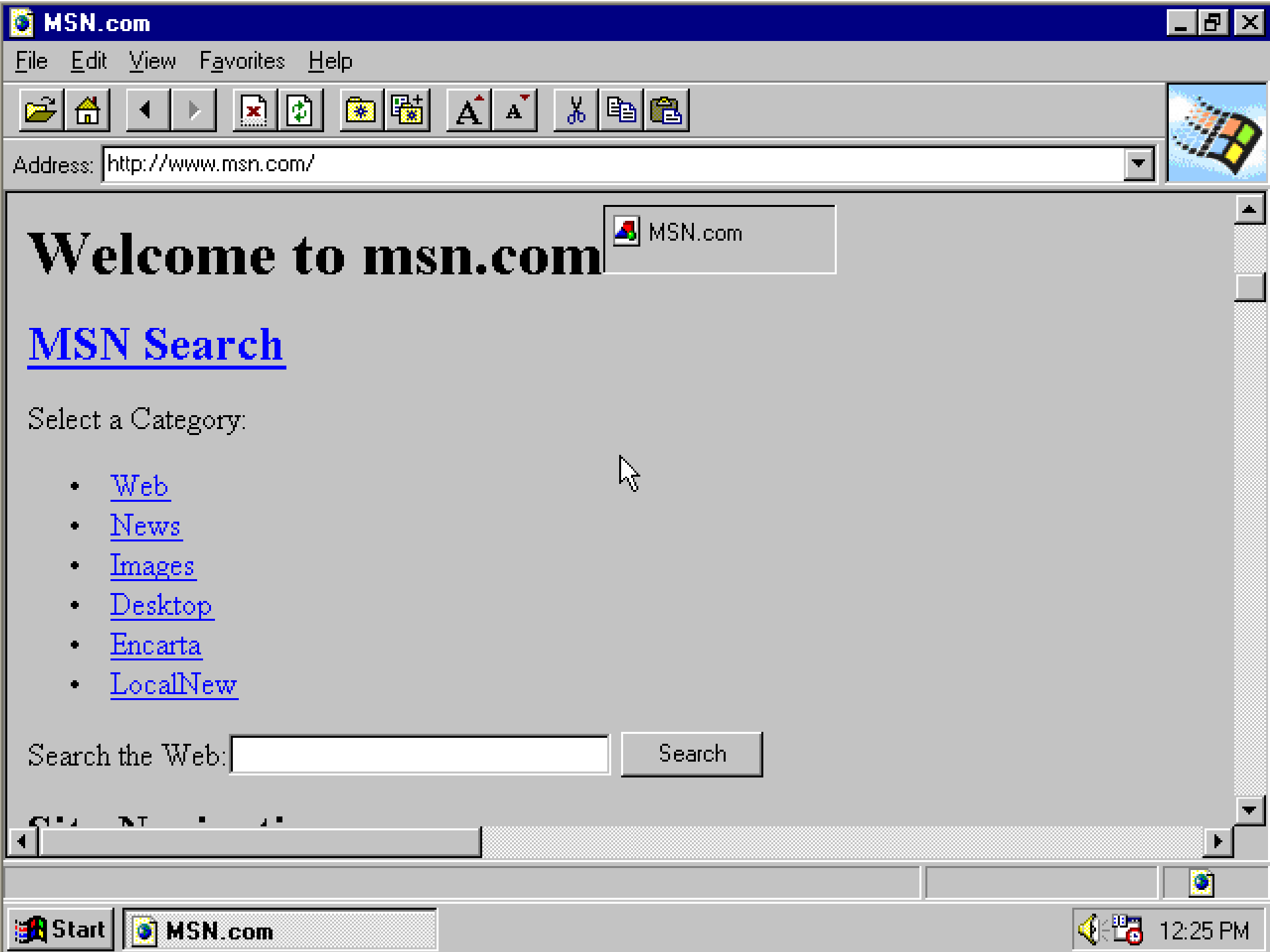


Java



1992 1995 **1996** 1998 1999 2005 2009 2011 2013



Address: 

# Welcome to msn.com



## [MSN Search](#)

Select a Category:

- [Web](#)
- [News](#)
- [Images](#)
- [Desktop](#)
- [Encarta](#)
- [LocalNew](#)

Search the Web:

Start

1992 1995 **1996** 1998 1999 2005 2009 2011 2013



ript



JavaScript



**ECMA**  
SCRIPT



# **JAVASCRIPT**

**≠**

# **ECMAScript**

# **JAVA**SCRIPT

is a dialect of

# **ECMA**SCRIPT

**J**AVASCRIPT **J**SCRIPT **A**CTIONS**C**RIP**T**

are dialects of

**E**CMA**A**SCRIPT

# ECMA SCRIPT

TC 39 controls what  
will be included in  
the spec

# **ECMA** SCRIPT 2

be inline with excisting  
international standards

# ECMA SCRIPT 3

~

features that are really  
essential to programming

regular expressions

more string methods

switch, do while

instanceof

exception handling

numeric formatting

# **ECMA** SCRIPT 4

~  
anticipated the future



# ~~ECMA SCRIPT 4~~

~~Incompatible with ES 3~~

# AJAX

~

Build complex web  
apps

# ECMA SCRIPT 5

Object.Create

defineProperty

Strict

Getter and setters

Semantic changes

json support

# **ECMA** SCRIPT 5.1

Revision of ES5

# ECMA SCRIPT 6

also called ES.next

# **ECMA** SCRIPT 6

~  
compatible with ES5

# **ECMA** SCRIPT 6

~  
be a better language



# **ECMA** SCRIPT 6

Final review of draft  
in november 2013

# ECMA SCRIPT 6

ECMA approval in  
december 2014

# ECMA HARMONY


---

Superset of ES.next  
future features in ES.next  
or ES.next.next

working\_draft\_ecma-262\_edition\_6\_05-14-13.doc [Compatibility Mode] - Word

FILE HOME INSERT DESIGN PAGE LAYOUT REFERENCES MAILINGS REVIEW VIEW

Kevin DeRudder

 **Draft**  
*Ecma/TC39/2013/0xx*

**Standard** ECMA-262  
6<sup>th</sup> Edition / Draft May 14, 2013

**Draft** **ECMAScript Language Specification**

Report Errors and Issues at: <https://bugs.ecmascript.org>

Product: Draft for 6th Edition  
Component: choose an appropriate one  
Version: Rev 15, May 14, 2013 Draft

**Standard**

Rue du Rhône 114 CH-1204 Geneva T: +41 22 849 6000 F: +41 22 849 6001

PAGE 1 OF 266 152672 WORDS ENGLISH (UNITED KINGDOM) 70%

# Linting



Visual Studio

Search Visual Studio with Bing

HOME SAMPLES LANGUAGES EXTENSIONS DOCUMENTATION COMMUNITY

Extensions > Tools > JSLint.VS2012

# JSLint.VS2012

Free

Adds JSLint support into Visual Studio

CREATED BY

Luke Page (Dennis Myren, Luke Page, Matthew Schaad)

LAST UPDATED

10/6/2012

VERSION

1.4.2

REVIEW

★★★★★ (4)






LICENSE

View

SUPPORTS

Visual Studio 2012

SHARE

DOWNLOADS

Download

(13,644)

FAVORITES

Add to favorites

TAGS

Javascript, jslint, jshint

DESCRIPTION

REVIEWS (4)

Q AND A (13)

No known issues at present, please report anything you find with 1.4.2.

- Choice of JSLint (current), JSLint (old) and JSHint.
- Simple .vsix double-click installation. Also available on the [Visual Studio Gallery](#) website, and downloadable directly from within Visual Studio using Tools->Extension Manager.
- Configure from the Tools menu. [Screenshot](#)
- Import/Export Options

get started for free

Report abuse to Microsoft

More From Luke Page

JSLint.VS2012

★★★★★

<script>

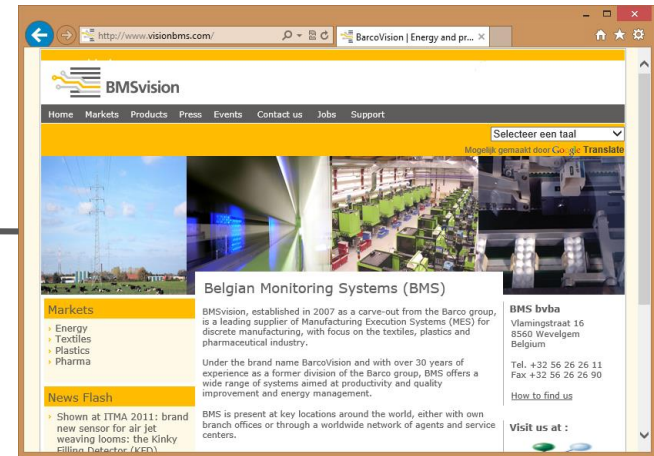


# Script

- JavaScript in externe files plaatsen
  - Vermijd inline script
  - Herbruikbaarheid, script caching!!
  - minifying
- Script files **onderaan** de HTML definiëren
  - Het uitvoeren van JavaScript blokkeert alles
  - Tenzij het niet anders kan: JavaScript die direct nodig is

```
<script src="demo.js"></script>
```

# Content Delivery Network



- CDN
- JS files gehost op een externe server
- JS zijn gechached waardoor een betere performantie mogelijk is

# Voorbeelden CDN

<http://www.asp.net/ajaxlibrary/cdn.ashx>

<http://www.bootstrapcdn.com/>

<https://developers.google.com/speed/libraries/devguide>

<http://aws.amazon.com/s3/>

<http://www.maxcdn.com/>

```
<script  
    src="http://ajax.aspnetcdn.com/ajax  
        /jquery/jquery-1.9.0.js">  
</script>
```

# Syntax

;

# Optional ;

- ; wordt gebruikt om statements van elkaar te scheiden
- Wanneer 2 statements op een verschillende lijn staan dan hoef je de ; niet te schrijven
  - GEEN BEST PRACTICE
  - MINIFYING
- Exceptions
  - Return
  - Break
  - continue



# Quiz vraagje 😊

```
var x = 12, y= 42;
```

x

++

y

```
alert(x);
```

```
alert(y);
```

```
if(caseSensitive != CaSeSenSiTive){  
    alert('JavaScript is case sensitive');  
}
```

```
var While = "demo case sensitive"; // OK  
var while = "demo case sensitive"; // NOK
```

“ “ & ’ ’

# Syntax

- Unicode character set
  - Elke taal kan zijn ding doen
- JavaScript negeert whitespaces
  - Ideaal voor code formatting

- Commentaar

// single line commentaar

/\* multiple lines

commentaar \*/

# Literals en Identifiers

- Literal: Data die direct in het programma voorkomt

```
42;  
true;  
null;  
"Hello World";
```

- Identifier: naam die je toekent aan literal
  - Begint met een letter, \_ of \$
  - Erna mogen letters, cijfers, \_ en dollar tekens

# Reserved words

|          |            |
|----------|------------|
| break    | instanceof |
| case     | new        |
| catch    | null       |
| continue | return     |
| debugger | switch     |
| default  | this       |
| delete   | throw      |
| do       | true       |
| else     | try        |
| false    | typeof     |
| finally  | var        |
| for      | void       |
| function | while      |
| if       | with       |
| in       |            |

# Gereserveerde Reserved words

class

const

enum

export

extends

import

super

Gedefinieerd sinds ES5  
voor volgende versies



# Reserved words in strict mode

Implements

Interface

Let

Package

Private

Protected

Public

Static

Yield

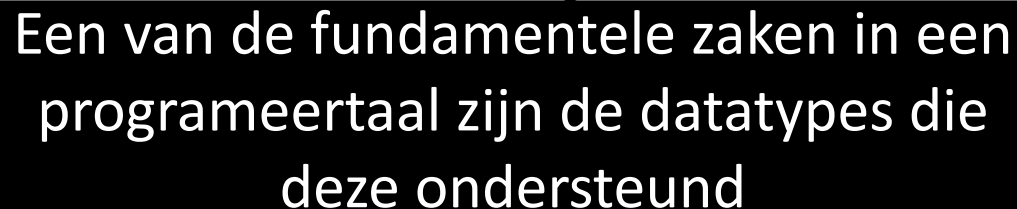
# Naamgevingen

- camelCase
  - Variabelen
  - Functies
- PascalCase
  - Objecten
  - Constructor functies

# Datatypes

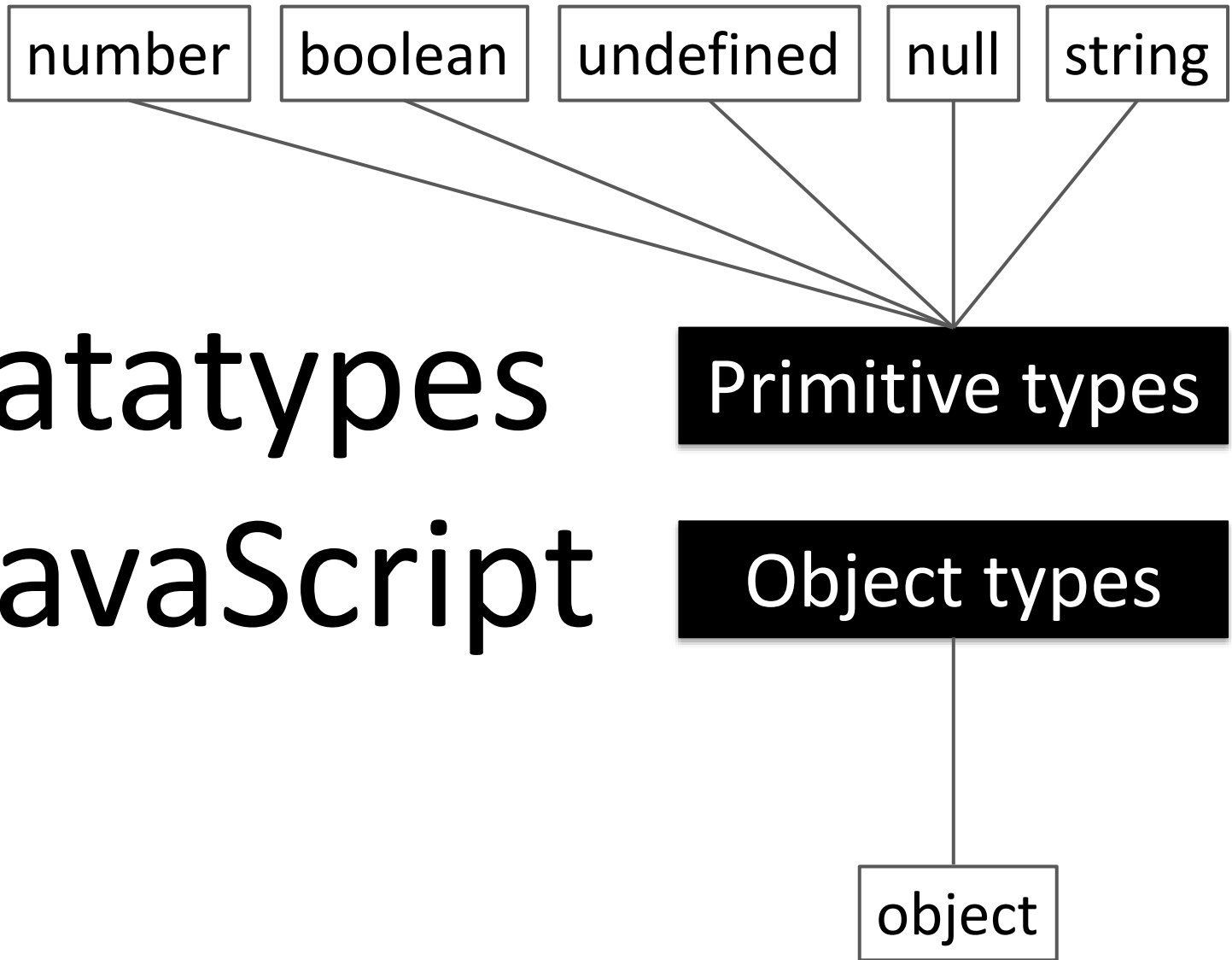
In IT bestaat er alleen data. Wat geen data is  
bestaat gewoon niet!!

Elk stukje data heeft een bepaalde rol. Op basis van de rol bepaal je het datatype



Een van de fundamentele zaken in een programmeertaal zijn de datatypes die deze ondersteund

# 2 datatypes in JavaScript



*(array, function zijn speciale soorten objecten)*

var

# Variabelen declareren

- Voor je een variabele kan gebruiken moet je die declareren
  - Via het **var** keyword

```
var i;  
var age;  
var name;  
  
// OF  
  
var i, age, name;
```

- Wanneer je geen value meegeeft is de waarde **'undefined'**



# Loosely typed

- JavaScript is heel flexibel als het gaat om het aanmaken van variabelen

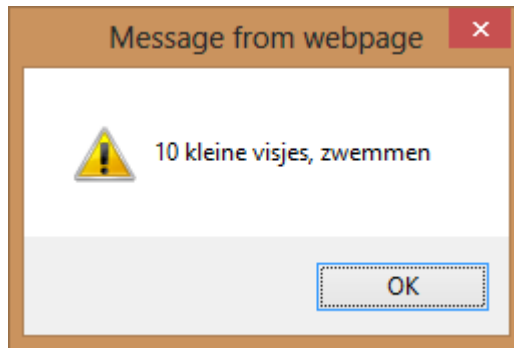
Is ook een van de redenen waarom intellisense zo moeilijk is

```
var demo = "een string";  
demo = 12;  
demo = true;  
demo = { type: "object" };
```

# Maar let op!!

Wat zit er in demo?

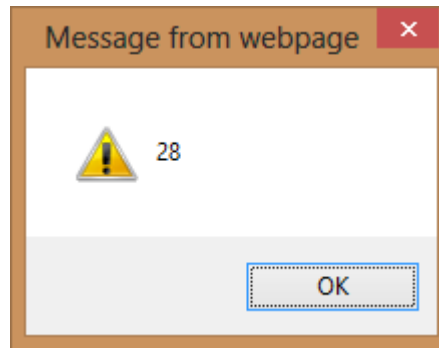
```
var demo = 10 + " kleine visjes, zwemmen";
```



# Maar let op!!

Wat zit er in demo?

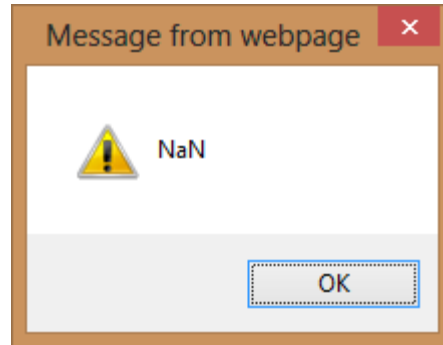
```
var demo = "7" * "4";
```



# Maar let op!!

Wat zit er in demo?

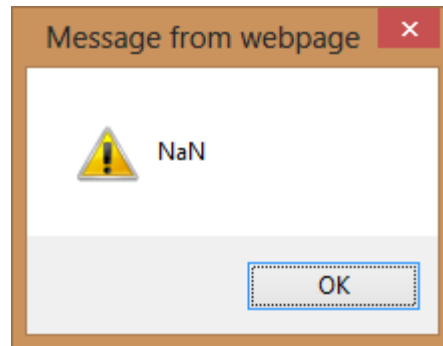
```
var demo = 1 - "x";
```



# Maar let op!!

Welke type is dit

```
var demo = 1 - "x";
```



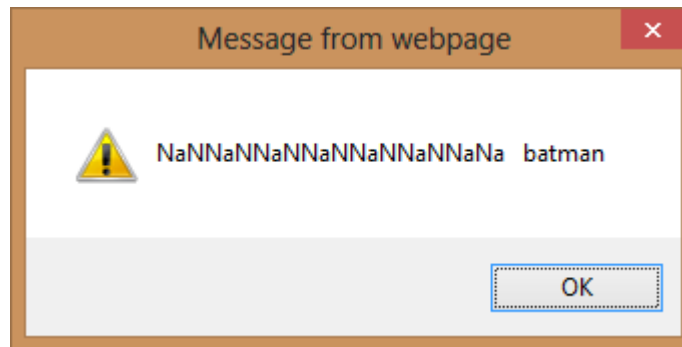
Number 😊



# Maar let op!!

Nu wordt het helemaal tof?? Wat is de output?

```
var s="";  
for (var i = 0; i < 7 ; i++) {  
    s += demo + " ";  
}  
  
s += "a   batman";
```



Doordat het converteren in JavaScript flexibel is, is de 'gelijk aan' operator dat ook

```
var numberTwelve = 12;  
var stringTwelve = "12";  
  
numberTwelve == stringTwelve; // true
```



Daarom is het beter op de 'strict equality'  
operator te gebruiken

```
var numberTwelve = 12;  
var stringTwelve = "12";  
  
numberTwelve === stringTwelve; // false
```

# Expliciete converties

```
var aNumber = Number("42");  
var aString = String("Kevin");  
var aString2 = "kevin".toString();  
var aBoolean = Boolean("42");  
var anObject = Object(42);
```

# Numbers

- Geen verschil tussen integer of floating point getallen
  - Elke nummer is een floating point
  - 64-bit
- Hexadecimal values begin with “0x” of “0X”

```
var demo = 0xFF; // 255
```

- Octal
  - Not supported by ECMA
  - Some browsers do support them

```
var demo = 0377; // 255
```

# Math

```
Math.pow();  
Math.round();  
Math.ceil();  
Math.floor();  
Math.abs();  
Math.max();  
Math.min();  
Math.random();  
Math.PI;  
Math.sqrt();  
Math.sin();  
Math.log();  
Math.exp();
```

# Date

- JavaScript heeft een Date constructor
  - Heeft methodes die u toelaten om met datums te werken

```
var aDate = new Date(2013, 08, 23);  
var now = new Date();
```

```
aDate.getDay();  
aDate.getHours();
```

# Strings en tekst

- String is een immutable sequentie van unicode karakters
- Single quotes of dubbele quotes kunnen gebruikt worden om een string aan te duiden
  - Probeer een afspraak te maken, een conventie
- `\` kan gebruikt worden als escape character



# Escape characters

| Escape character | Beschrijving    |
|------------------|-----------------|
| <code>\0</code>  | NUL character   |
| <code>\b</code>  | Backspace       |
| <code>\t</code>  | Horizontale tab |
| <code>\n</code>  | Nieuwe lijn     |
| <code>\v</code>  | Form feed       |
| <code>\r</code>  | Carriage return |
| <code>\"</code>  | Dubbele quote   |
| <code>'</code>   | Single quote    |
| <code>\\</code>  | Backslash       |

# String bewerkingen

- + operator om strings aan elkaar te plakken
- Herinner u dat een string immutable is
  - Er wordt telkens een nieuwe string aangemaakt

```
var s = "hello, world" // Start with some text.
s.charAt(0) // => "h"
s.charAt(s.length - 1) // => "d": laatste karakter
s.substring(1, 4) // => "ell": 2e, 3e, 4e karakter
s.slice(1, 4) // => "ell": 2e, 3e, 4e karakter
s.slice(-3) // => "rld": laatste 3 karakters
s.indexOf("l") // => 2: positie van de eerste opgegeven karakter
s.lastIndexOf("l") // => 10: positie van laatste opgegeven karakter
s.indexOf("l", 3) // => 3: pos van eerste "l" op of achter plaats 3
s.split(", ") // => ["hello", "world"]
s.replace("h", "H") // => "Hello, world"
s.toUpperCase() // => "HELLO, WORLD"
```

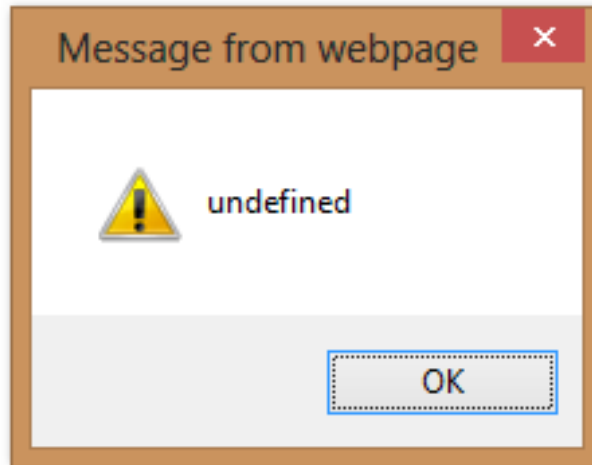
# Boolean

- True of false, ja of neen, 1 of 0
- Vooral gebruikt bij controle structuren
  - If – else
  - While
  - Do while

# NULL vs UNDEFINED

- Variabele of property die gedefinieerd is is “undefined”

```
alert(product.UndefinedProperty);
```



Lijkt simpel toch?

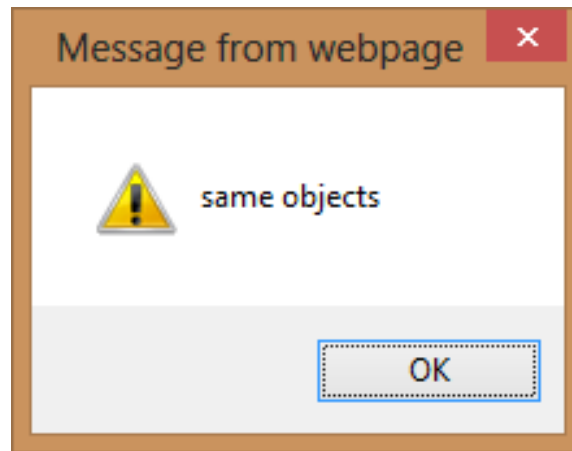
Waarom dan zo veel misverstanden

Als we kijken naar wat `null` betekent: `null` duidt aan dat er geen waarde gedefinieerd is

is `null` dan hetzelfde als `undefined`

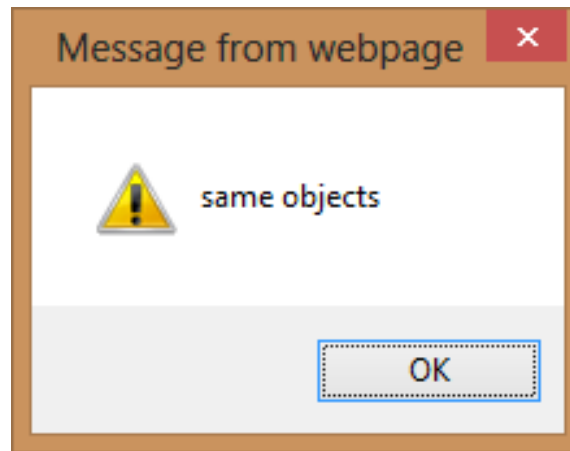


```
if (null == undefined) {  
    alert("same objects");  
}  
else {  
    alert("not the same object");  
}
```





```
if (null == undefined) {  
    alert("same objects");  
}  
else {  
    alert("not the same object");  
}
```



==

kijkt enkel naar de waarde, niet naar het type

===

kijkt naar de waarde en het type

```
' ' == '0'    // false
0  == ' '    // false
0  == '0'    // true
```

```
false == 'false' // false
false == '0'      // true
```

```
false == undefined // false
false == null       // false
```

Vermijd == of !=

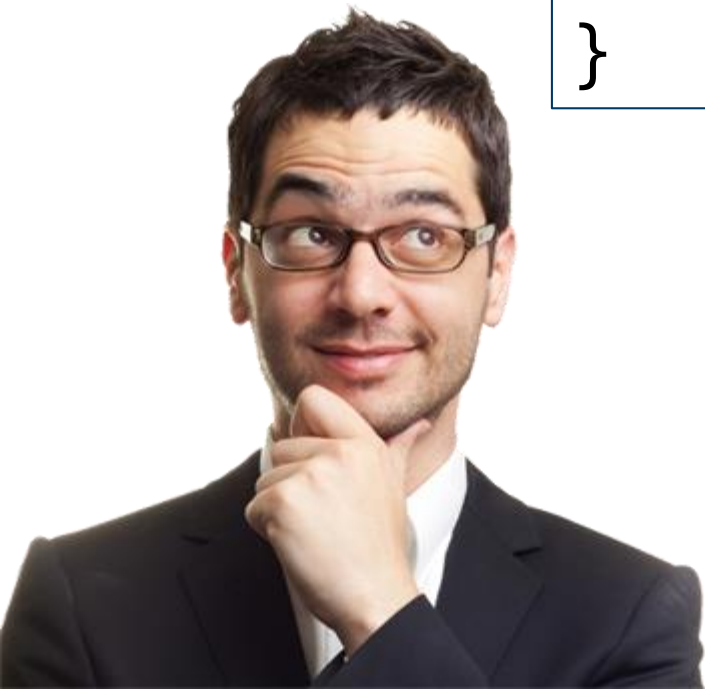
`Null` wordt gezet door de developer in code

`Undefined` wordt door JavaScript bepaald

# Truthy vs Falsy

# Waarom schrijven we dit

```
function demo() {  
    if (variabele) {  
  
    }  
}
```





# Truthy vs Falsy

- › Boolean is een van de beschikbare datatypes
- › Daarenboven heft JavaScript ook iets als een 'inherent Boolean' value
  - › Gekend als truthy of falsy

# Volgende zaken zijn falsy

› false

› 0

› ""

› null

› undefined

› NaN

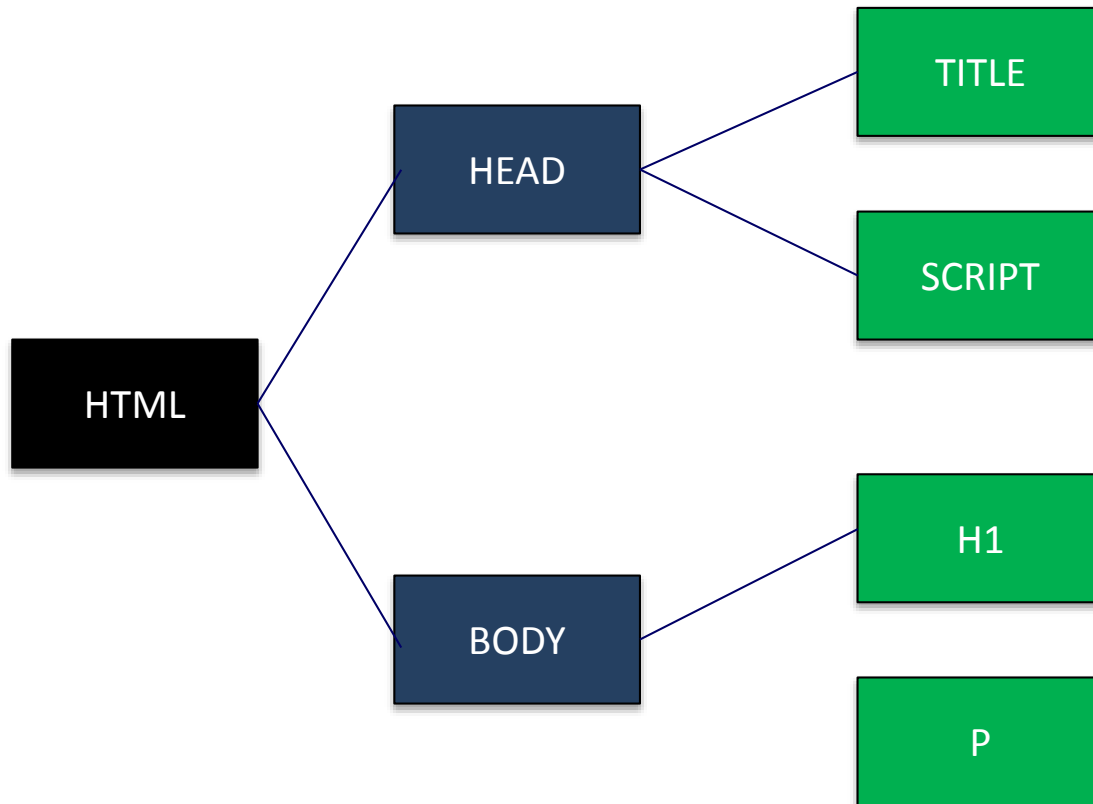
```
var a = (false == 0); // true
var b = (false == ""); // true
var c = (0 == ""); // true
```

# JavaScript en DOM

JavaScript heeft weinig tot niets te maken  
met het **D**ocument **O**bject **M**odel

# Document Object Model

- Cross-platform en programmeertaal onafhankelijke API
  - Dient om met XML en HTML documenten te communiceren



Performantie!! 😞







Onthoud dat JavaScript een tol moet betalen  
wanneer die met het DOM communiceert

# Navigeren door het DOM

```
document.body  
document.body.parentNode  
document.body.childNodes.length  
document.documentElement  
document.documentElement.lastChild  
document.documentElement.firstChild  
document.documentElement.lastChild.firstChild.nextSibling
```

# Element selecteren

`document.get`


-  `getElementById`
-  `getElementsByClassName`
-  `getElementsByName`
-  `getElementsByTagName`
-  `getElementsByTagNameNS`
-  `getSelection`





# Elementen selecteren

- Op basis van een CSS selector
  - Zoals jQuery al jaren doet

`document.query`



-  `querySelector`
-  `querySelectorAll`