

# Javascript -1

Tom Peeters

# Waarom javascript?

- Veel gebruikt, vele use cases, en steeds aan het groeien
  - Web pages, Ajax, Node.JS, Mobile hybrid frameworks (e.g. Ionic),
- Niets te maken met Java
- Ontwikkeld door Netscape in 1995 om applicaties te maken die in de browser “runnen”
- Voornamelijk client-side

# Wat kunnen we met JavaScript doen?

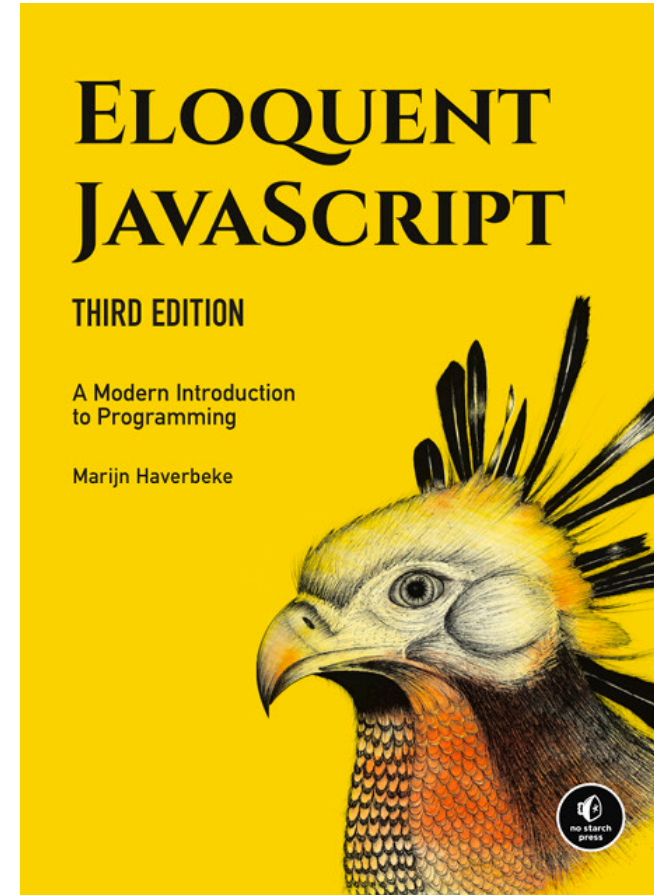
- Kleine uitbreiding binnen je webapplicatie
- Parallax (<http://matthew.wagerfield.com/parallax/> )
- Games in de browser
- “Internet of Things”: robots, wearable tech, ...
- Full Stack Web Development( AngularJS)

# Introductie

- Scripting taal nodig om je web pagina's interactief te maken
- Java <> JavaScript
- Scripting taal wil zeggen dat je code niet gecompileerd moet worden
  - Java compileert naar bytecode en wordt door de runtime interpreter uitgevoerd (source code krijgt men nooit te zien)
  - Javascript wordt niet gecompileerd maar direct geïnterpreteerd. De javascript broncode is opgenomen binnen de HTML pagina
- Principe van Loose typing: je variabele krijgen geen type mee

# Bijkomende literatuur

Heel goed boek. Gratis digitaal beschikbaar op <https://eloquentjavascript.net/>.



# Je eerste script

```
EersteScript.html x
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
3 "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
4
5 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
6   <head>
7     <title>EersteScript</title>
8
9   </head>
10
11   <body>
12
13     <script type="text/javascript">
14       document.write("Hallo");
15       console.log("Hallo");
16     </script>
17
18   </body>
19 </html>
20
```

EersteScript x  
127.0.0.1:8020/Javascript/EersteScript.html

Hallo

Elements Network Sources Timeline Profiles Resources Audits Console  
<top frame> Preserve log  
Hallo

# Je eerste script

```
EersteScript.html x
1  <?xml version="1.0" encoding="UTF-8"?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
3    "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
4
5  <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
6    <head>
7      <title>EersteScript</title>
8    </head>
9
10   <body>
11
12     <script type="text/javascript">
13       document.write("Hallo");
14       console.log("Hallo");
15     </script>
16
17   </body>
18 </html>
19
20
```

- De script-tag geeft aan dat er gebruik zal worden gemaakt van een scripttaal (het type attribuut, niet verplicht)

```
<script type="text/javascript">
  <!--
    hier komt de Javascript-code
  // -->
</script>
```

- Je kan vanuit een webpagina ook verwijzen naar een javascript bestand ( extensie .js)

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>RefJavascript</title>
    <script type="text/javascript" src="test.js"></script>
  </head>

  <body>

  </body>
</html>
```

# Syntax regels

- Javascript is hoofdlettergevoelig
- Voor kommagetallen gebruik je een punt ipv een komma
- Elke opdrachtlijn eindig je best met een puntkomma ( ; )
- Commentaar bij je javascript code kan als volgt worden ingevoerd:

```
<script>
var i=10; //Deze tekst is commentaar voor 1 regel

/*
 * Op deze manier kan ik meerdere regels in
 * commentaar plaatsen
 */
var i=11;

</script>
```



# Variabele

- Variabelen gebruik je om waarden in het geheugen van de computer vast te houden. Ze zijn fundamentele blokken van elke programmeertaal
- Variabelen kunnen getallen, tekenreeksen (strings genoemd), booleans, objecten, arrays bevatten, en kunnen op een willekeurige plaats in het programma worden geïnitieerd, uitgelezen of veranderd.
- In tegenstelling tot de meeste programmeertalen zijn variabelen in javascript untyped: het type (getal, karakter, boolean, object, ...) wordt automatisch toegewezen

# Variabele cont'd

- Variabelen gebruik je om waarden in het geheugen van de computer vast te houden. Ze zijn fundamentele blokken van elke programmeertaal
- Variabelen kunnen getallen, tekenreeksen (strings genoemd), booleans, objecten, arrays bevatten, en kunnen op een willekeurige plaats in het programma worden geïnitieerd, uitgelezen of veranderd.
- In tegenstelling tot de meeste programmeertalen zijn variabelen in javascript untyped: het type (getal, karakter, boolean, object, ...) wordt automatisch toegewezen

# Een variabele declareren en initialiseren

- Een variabele wordt in javascript met het woord var (of let) gedeclareerd (later uitleg wat het verschil is tussen var en geen var te gebruiken)

```
var i;
```

- Je kan ook verschillende variabelen tegelijk declareren:

```
var i, j;
```

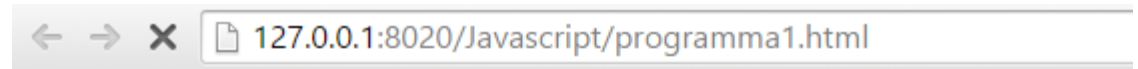
- Nadat je een variabele hebt gedeclareerd kan je er een waarde aan toewijzen. De toekenningoperator is in javascript het gelijkheidsteken =

```
var i = 7;
```

```
j = "Hallo";
```

# Voorbeeld

```
5< <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
6  <head>
7    <title>programma1</title>
8
9  <script>
10
11    cursus      = "cursus cmt4";
12    school      = "AP Hogeschool";
13    mededeling  = "Welkom";
14    dagen       = 1;
15    kleur       = "red";
16
17    alert(mededeling);
18  </script>
19  </head>
20
21  <body>
22    <h3>Welkom!</h3>
23    <script>
24      document.write("welkom in de " + cursus + "bij <b> " + school + "</b>");
25
26      document.write("<p style = 'color:" + kleur + "'> blabla </p>");
27    </script>
28  </body>
</html>
```



**Welkom!**

welkom in de cursus cmt4bij **AP Hogeschool**

blabla

De pagina op 127.0.0.1:8020 meldt het volgende: ✕

Welkom

OK

# Gegevenstypen

- Strings

Strings zijn stukjes tekst tussen enkele of dubbele aanhalingstekens

- Getallen

Voorbeelden zijn 12345 en 3.14 (een punt voor komma's)

- Boolean

2 mogelijkheden true of false

```
var stop = false;  
var start = true;
```

```
5 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">  
6   <head>  
7     <title>strings</title>  
8   </head>  
9   <body>  
10    <script>  
11      var oneLineStr = "this is one line";  
12      var multiStr = "This is the first line \  
13                      This is the second line \  
14                      This is more...";  
15  
16      document.write(oneLineStr);  
17      document.write(multiStr);  
18  
19    </script>  
20  </body>  
21 </html>
```

← → ↻ 127.0.0.1:8020/Javascript/strings.html

this is one lineThis is the first line This is the second line This is more...

# Variabele

- Het zou handig zijn als je “iets” had waaraan je een waarde kon toekennen die later eventueel zou kunnen veranderen. Dat “iets” noemen we een variabele  
De naam van een variabele moet beginnen met een letter, met het dollarteken of met een underscore
- ```
var groet = "Hallo WWW";  
document.write(groet);
```

Merk op dat op de tweede regel geen aanhalingstekens rond groet staat. Groet is namelijk een variabele en niet een stringconstante. Zouden we aanhalingstekens gebruiken, dan zou in plaats van de waarde van variabele groet (de woorden Hallo WWW) het woord “groet” op het scherm worden getoond
- Per statement een nieuwe regel! Enkel voor de leesbaarheid

# Expressies

- Als we constanten en variabelen van het type getal gebruiken in een expressie, zal die expressie zelf ook van het type getal zijn.
- Als we een getal variabele creëren (`var x = 1`), dan zal de expressie `x+1` het resultaat 2 opleveren.
- We kunnen dat resultaat toekennen aan een variabele `y` door middel van `y = x + 1`; (`y` is een getal variabele geworden die de waarde 2 bevat)
- Het `+` teken kan ook voor strings gebruikt worden. Woorden kunnen we niet optellen maar we samenvoegen (concatenate)

```
<script>
  var tekst1 = "hallo";
  var tekst2 = "Tom";

  var tekst3 = tekst1 + tekst2;

  document.write(tekst3);
</script>
```

# Increment & Decrement

- Een aantal wiskundige operaties worden zoveel gebruikt dat ze een eigen operator hebben gekregen
- Increment : ++
- Decrement: --

```
<script>
```

```
var getal = 5;  
getal = getal + 1;  
document.write(getal);
```

```
var getal2 = 10;  
getal2++;  
document.write(getal2);
```

```
</script>
```



# Oefeningen

- Gebruik de functie `prompt()` om aan de gebruiker een getal te vragen (bewaar dit getal in een variabele). Dit script vraagt 2 getallen aan de gebruiker en telt deze getallen op. Toon het resultaat op het scherm.
  - Wat krijg je te zien indien je eerste getal 4 is en je tweede getal 5 ?
- Schrijf een script dat graden converteert naar graden Fahrenheit met volgende formule:  
$$\text{gradenFahren} = 9/5 * \text{gradenCelsius} + 32$$

Vraag de gebruiker een getal in te geven en schrijf het resultaat op het scherm

# Oplossing 1

```
5 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">  
6   <head>  
7     <title>Oefening1</title>  
8  
9   </head>  
10  
11   <body>  
12  
13     <script>  
14  
15       var getal1 = prompt("geef getal 1 in");  
16       var getal2 = prompt("geef getal 2 in");  
17  
18       var resultaat = parseInt(getal1) + parseInt(getal2);  
19       document.write(resultaat);  
20     </script>  
21  
22   </body>  
23 </html>
```

# Oplossing 2

```
5⊖ <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
6⊖ <head>
7    <title>Oefening2</title>
8
9 </head>
10
11⊖ <body>
12
13⊖ <script>
14
15    var tempCelsius = prompt("geef een temperatuur in celsius");
16
17    var tempFahrenheit = 9/5 * tempCelsius + 32;
18
19    document.write(tempFahrenheit);
20 </script>
21
22 </body>
23 </html>
24
```

Meer oefeningen: zie [Learning.ap.be](http://Learning.ap.be)

# Toekenningoperator =

- Dit wordt gebruikt om het resultaat van een expressie toe te kennen aan een variabele

```
<script>
```

```
var getal = 123 + 345;
```

```
</script>
```

- += : De opdracht `x += y` betekent hetzelfde als `x = x + y`
- -= : De opdracht `x -= y` betekent hetzelfde als `x = x - y` (idem voor `*=` )

```
<script>
```

```
var x = 5;
```

```
var y = 3;
```

```
x += y;
```

```
</script>
```

# Logische operators

| Betekenis        | Operator | Voorbeeld              |
|------------------|----------|------------------------|
| Optellen         | +        | som = a + b;           |
| Aftrekken        | -        | verschil = a - b;      |
| Tegengestelde    | -        | tegengestelde = -b;    |
| Vermenigvuldigen | *        | product = a * b;       |
| Delen            | /        | quotiënt = a / b;      |
| Modulo           | %        | rest = a % b;          |
| Verhogen met 1   | ++       | a++ of ++a (a = a + 1) |
| Verlagen met 1   | --       | a-- of --a (a = a - 1) |

# Logische test: if else

- We kijken of aan een bepaalde voorwaarde voldaan wordt: daartoe gebruiken we de if opdracht en de verwante if ... else opdracht ( de voorwaarde staat tussen haakjes / de actie tussen accolades { } )

Er wordt slechts 1 blok

uitgevoerd, afhankelijk van c  
variabele getal

- Je mag gerust enkel een if-statement plaatsen (zonde if/else, of else blok )

```
--  
11  <body>  
12  
13  <script>  
14  
15      var getal = prompt("geef een getal op");  
16  
17      if(getal < 10)  
18      {  
19          document.write("het getal is kleiner dan 10");  
20      }  
21      else if(getal >= 10 && getal < 20)  
22      {  
23          document.write("het getal is groter of gelijk aan 10 maar kleiner dan 20");  
24      }  
25      else  
26      {  
27          document.write("het getal is groter dan 20");  
28      }  
29  
30  </script>  
31
```

# Enkel if statement

```
11  <body>
12
13  <script>
14    var prijs = prompt("Geef een prijs op");
15
16    if (prijs < 50)
17      prijs = 50;
18
19    alert(prijs);
20
21
22
23  </script>
24  </body>
```

Je kan ook if-statements nesten: een if- statement binnen een ander if- statement



# Logische operators

- De logische ontkenning, weergegeven door een uitroepteken ( ! ), en uitgesproken als NOT
- Logische vergelijkingen:
  - $<$  : kleiner dan
  - $>$  : groter dan
  - $\leq$  : kleiner dan of gelijk aan
  - $\geq$  : groter dan of gelijk aan

# Logische operators

- Met de operator `!=` kunnen we testen op ONGELIJKHEID
  - De expressie `x != y` heeft de waarde `true` als variabele `x` een andere waarde heeft dan variabele `y`, of `false` als de twee variabelen dezelfde waarden hebben
- De logische operator `==` test op gelijkheid ( let op de 2 gelijkheidstekens!)
  - `X == Y` : is `true` als de twee variabelen dezelfde waarde hebben , maar `false` als ze verschillende waarden hebben

# Logische operators

- De operators `&&` en `||` maken het mogelijk om 2 expressies te combineren met respectievelijk een logische AND of een logische OR
  - `X && Y`: is alleen true als X en Y de waarde true heeft
  - `X || Y` : is true als X true is, als Y true is of beiden zijn true

```
<script>

var a = prompt("geef een getal in: ");
var b = prompt("geef een getal in: ");

if (a == b )
{
    alert("de getallen zijn gelijk");
}
else
    alert ("de getallen zijn niet gelijk");
</script>
```

# Math object

- Een Math object geeft de mogelijkheid om bepaalde wiskundige taken te berekenen
  - `var x = Math.PI;`
  - `var y = Math.sqrt(16);`
- VRAAG: ga na wat het resultaat van `Math.random()` is.

# Oefening

Schrijf de huidige dag (in het Nederlands) op je webpagina

# switch

```
switch(expression) {  
    case n:  
        code block  
        break;  
    case n:  
        code block  
        break;  
    default:  
        default code block  
}
```

```
switch (new Date().getDay()) {  
    case 0:  
        day = "Sunday";  
        break;  
    case 1:  
        day = "Monday";  
        break;  
    case 2:  
        day = "Tuesday";  
        break;  
    case 3:  
        day = "Wednesday";  
        break;  
    case 4:  
        day = "Thursday";  
        break;  
    case 5:  
        day = "Friday";  
        break;  
    case 6:  
        day = "Saturday";  
        break;  
}
```

# Oefening

- Ontwikkel een tekst adventure game
- We maken een scenario waar we tegen een zombie vechten. Vooraleer je vecht kies je een wapen (bijvoorbeeld een pijl & boog, bijl of een kip). Daarna start je het gevecht.
- Deel 1: Ontwikkel je spel zodat je 50% kans hebt om te winnen ( denk aan `Math.random()`).
- Deel 2: tracht je webpagina attractief te maken en afhankelijk van het wapen stijgen je kansen tot winst.

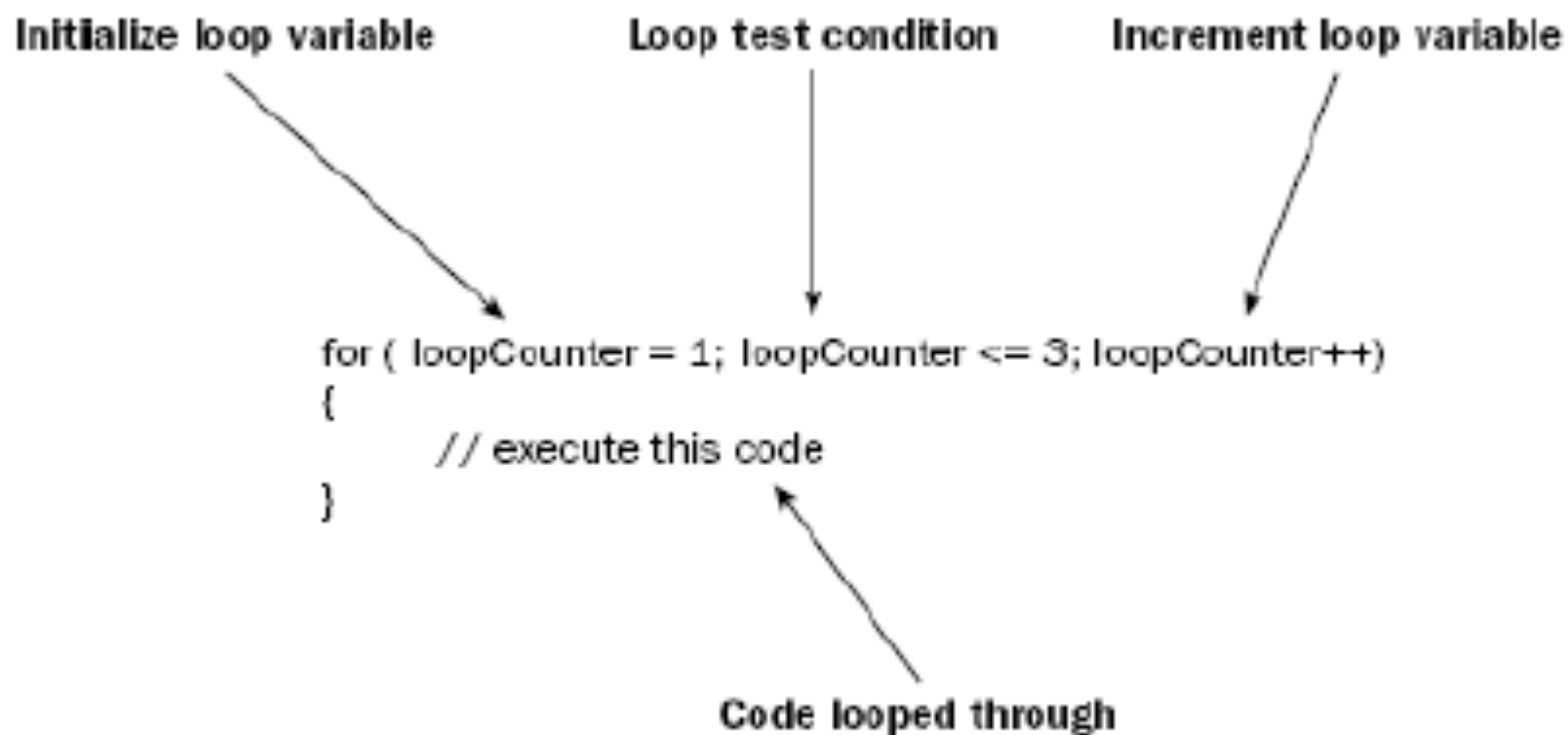
# Herhalingslussen : for

- Stel je eens voor dat we ons niet hebben gehouden aan de JavaScript etiquette en dat we als straf duizend keer de regel “ ik zal verwijzen naar de oorspronkelijke auteur als ik een script kopieer” op het scherm moeten tonen

```
<script>
    for(i=0; i<100;i++)
    {
        document.write("ik zal verwijzen naar de oorspronkelijke auteur als ik een script kopieer </br>");
    }
</script>
```



# For lus



```
<script>
    for(i=0; i<100;i++)
    {
        document.write("ik zal verwijzen naar de oorspronkelijke auteur als ik een script kopieer </br>");
    }
</script>
```

- De for lus bevat 3 elementen tussen de haakjes
- Voordat de lus wordt uitgevoerd wordt de i variabele geïnitieerd : hier op 0.
- Vervolgens wordt gekeken of aan de voorwaarde voldaan wordt: is i kleiner dan 100? Is dit het geval, dan wordt de opdracht (blok code tussen accolades) 1 keer uitgevoerd
- Tenslotte wordt de i variabele met 1 verhoogd en wordt daarna opnieuw gekeken of aan de voorwaarde voldaan is.

# Voorbeeld van een eindeloze lus

```
For (i=1 ; i> 0 ; i++)
```

# Oefening

- Breng de tafel van vermenigvuldiging voor het getal 7 op het scherm. ( in tabel vorm)

# WHILE

- Een while statement voert een opdracht telkens opnieuw uit tot de voorwaarde voor de while lus onwaar wordt

Condition - keep looping while this condition is still true

```
while ( degCent != 100)
{
    // some code
}
```

Code looped through

```
<script>
```

```
var teller = 1;
while (teller <= 10)
{
    document.write("de teller is :" + teller + "<br>");
    teller++;
}
```

```
</script>
```

# Oefeningen

Schrijf een script dat een random getal tussen 50 en 100 toont. Doe dit zolang de gebruiker nee ( of n ) in je prompt venster invult