



## Webdesign Javascript

## Waarden toekennen

### ➤ Gelijk aan teken (=) bij declaratie

```
var a = 5;
var a = "String";
var a = getFunction(param1, param2);
```

### ➤ Specialeke

a+= 5	=>	a = a+5	
a-= 5	=>	a = a-5	
a/=5	=>	a = a/5	
a*=5	=>	a = a*5	
a%=5	=>	a = a%5	(rest van een deling)
a++	=>	a = a+1	
a--	=>	a = a-1	



## Webdesign Javascript

## Waarden toekennen

### ➤ Plus teken (+)

Bij string: "aap" + " sta op!" => "aap sta op!"

Bij Number: 5 + 4 => 9

Bij alles: var a = +variabele => geeft de numerieke waarde van variabele

+true	=>	1
+null	=>	0
+"2.5"	=>	2.5
+"test"	=>	NaN



## Webdesign Javascript

### Waarden toekennen

➤ Uitroep teken (!)

Bij Boolean:

```
var a = !variabele
```

=> geeft tegengestelde waarde

```
!true    =>    false
```

```
!null    =>    true
```

```
!false   =>    true
```

```
!"test"  =>    false
```

```
!0       =>    true
```

```
!1       =>    false
```



## Webdesign Javascript

### Variabelen vergelijken

- Vergelijkingsoperatie (==)
- Verschiloperatie (!=)
- Strikte vergelijkingsoperatie (===)
- Strikte verschiloperatie (!==)
- Groter dan (>)
- Kleiner dan (<)
- Groter of gelijk aan (>=)
- Kleiner of gelijk aan (<=)



## Webdesign Javascript

## Variabelen vergelijken

### Vergelijkingsoperatie (==)

- Vergelijkt 2 waarden en geeft terug (boolean) of ze gelijk zijn
- Converteert variabelen als niet hetzelfde type

"5" == 5	=>	true
1 == 1	=>	true
0 == false	=>	true
null == undefined	=>	true
0 == undefined	=>	false
0 == null	=>	false

Front-end Webdesign Javascript 1-12 Operatoren en programma verloop

5



## Webdesign Javascript

## Variabelen vergelijken

### Verschiloperatie (!=)

- Vergelijkt 2 waarden en geeft terug (boolean) of ze verschillen
- Converteert variabelen als niet hetzelfde type

"5" != 5	=>	false
1 != 1	=>	false
0 != false	=>	false
null != undefined	=>	false
0 != undefined	=>	true
0 != null	=>	true

Front-end Webdesign Javascript 1-12 Operatoren en programma verloop

6

**Strikte vergelijkingsoperatie (===)**

- Vergelijkt 2 waarden en geeft terug (boolean) of ze gelijk zijn
- GEEN conversie (strict vergelijken)

"5" === 5	=>	false
1 === 1	=>	true
0 === false	=>	false
null === undefined	=>	false
0 === undefined	=>	false
0 === null	=>	false

**Strikte verschiloperatie (!==)**

- Vergelijkt 2 waarden en geeft terug (boolean) of ze verschillen
- GEEN conversie (strict vergelijken)

"5" !== 5	=>	true
1 !== 1	=>	false
0 !== false	=>	true
null !== undefined	=>	true
0 !== undefined	=>	true
0 !== null	=>	true



## Webdesign Javascript

## Variabelen vergelijken

- Wat is het verschil tussen
  - =
  - ==
  - ===
  
- En wat is het verschil tussen
  - !
  - !=
  - !==



## Webdesign Javascript

## Vergelijkingen samenvoegen

### Logische operatoren

- Logische AND (&&)
- Logische OF (||)
- Logische NIET (!)



## Webdesign Javascript

## Programma verloop

- if-else
- switch
  
- while
- do while
- for
  
- \* break
- \* continue



## Webdesign Javascript

## if-else

### if-else

```
if(boolean){  
    //code indien juist  
}  
else if(boolean2){  
    //code indien boolean fout en boolean2 juist  
}  
else{  
    //code indien boolean2 fout  
}
```

- else if en else zijn optioneel
- er kan maar 1 if en 1 else zijn per statement
- else if kan zo vaak als je wilt voorkomen



## Webdesign Javascript

## oefening 7

- Maak een script dat 2 getallen vraagt m.b.v. `prompt()`;
- Test of het weldegelijk 2 getallen zijn. Als dat niet het geval is wordt getoond dat geen 2 getal ingegeven werden

### Positie tonen

Je hebt geen 2 getallen ingegeven.

- Vergelijk de 2 getallen. En toon of het eerste getal gelijk, groter of kleiner is dan het 2<sup>e</sup> getal.

### Positie tonen

Het eerste getal (33) is gelijk aan het tweede getal (33).

### Positie tonen

Het eerste getal (12) is groter aan het tweede getal (3).

### Positie tonen

Het eerste getal (5) is kleiner aan het tweede getal (35).



## Webdesign Javascript

## switch

### switch

```
switch(expression) {
  case a:
    //code
    break;
  case b:
    //code
    break;
  default:
    //code
    break;
}
```



## Webdesign Javascript

### switch

#### switch

```
var dag = new Date().getDay(); //geeft dag in nummer van week mee
switch(dag){
  case 0:
    alert("Het is zondag!");
    break;
  case 6:
    alert("Het is zaterdag!");
    break;
  default:
    alert("Als het geen weekend is, interesseert het me niet");
    break;
}
```



## Webdesign Javascript

### while

#### while

```
var a = 0;
while(a<=5){
  alert(a);
  a++;
} //Deze code zal 6 keer worden uitgevoerd
```

- While = "zolang dat"  
=> voert code uit zolang het statement klopt
- While test EERST voorwaarde en voert dan uit





## Webdesign Javascript

### do while

#### do while

```
var a = 12;  
do{  
    alert(a);  
    a++;  
}  
while(a<=5)  
//Deze code zal toch 1 keer worden uitgevoerd !!!
```

- Do While = "doe zolang dat"  
=> voert code uit zolang het statement klopt
- Do while voert eerst uit en test DAARNA voorwaarde



## Webdesign Javascript

### oefening 8

- Pas oefening 7 aan:
- Blijf getal1 en getal2 opnieuw vragen zolang geen 2 getallen ingegeven werden.



## Webdesign Javascript

for

### for

```
for(var a=0; a<=5; a++){  
    // this code will run 6 times  
}
```

- 3 gedeelten:
  - Initialisatie voorwaarde
  - Voorwaarde
  - Aanpassen voorwaarde



## Webdesign Javascript

for/in en for/of

### Ook 2 speciale vormen

- for(... in ...)

Itereert over alle property **namen** in een object/array

Bv: 

```
var persoon={naam:"Willems", Voornaam:"Mimi", nummer:46}  
for(var a in persoon){  
    alert(a+" is "+persoon[a]); }  
geeft: "naam is Willems", "Voornaam is Mimi" en "nummer is 46"
```

- for(... of ...)

Itereert over alle property **waarden** in een object/array

Bv: 

```
for(var a of persoon){  
    alert(a); }  
geeft: "Willems", "Mimi" en "46"
```



## Webdesign Javascript

## oefening 9

- Maak een script dat 1 getal vraagt m.b.v. `prompt()`;
- Controleer of dit een getal is tussen 1 en 20. Als dat niet het geval is, vraag het getal opnieuw.
- Als het een getal is van 1 tot 20 toon dan de tafels van 1 tot 20 van dit getal.
- Het resultaat zou er als volgt kunnen uitzien:

### Tafels van 15

```
1 * 15 = 15
2 * 15 = 30
3 * 15 = 45
4 * 15 = 60
5 * 15 = 75
6 * 15 = 90
7 * 15 = 105
8 * 15 = 120
9 * 15 = 135
10 * 15 = 150
11 * 15 = 165
12 * 15 = 180
13 * 15 = 195
14 * 15 = 210
15 * 15 = 225
16 * 15 = 240
17 * 15 = 255
18 * 15 = 270
19 * 15 = 285
20 * 15 = 300
```



## Webdesign Javascript

## break

### break

- Bij switch al gezien
- Stopt de volledige loop (code na loop wordt meteen uitgevoerd)
- Vooral handig voor optimalisatie van loop

```
for(var a = 0; a<=5; a++){
    if(a==3){break;}
    alert(a);
} //zal enkel 0,1 & 2 tonen
```



## Webdesign Javascript

### continue

#### **continue**

- slaat enkel huidige iteratie over in loop
- Handig voor wanneer specifiek geval 'anders' behandeld moet worden

```
for(var a = 0; a<=5; a++){  
    if(a==3){continue;}  
    alert(a);  
} //zal enkel 0,1,2,4 & 5 tonen, NIET 3
```



## Webdesign Javascript

### return

#### **return**

- Wordt enkel gebruikt binnen een functie.
- Stopt niet enkel de loop, maar ook de volledige functie.
- Keert terug naar het aanroepende statement.
- return kan eventueel een waarde teruggeven, maar dat is niet verplicht.