



WebAdvanced

# Hoofdstuk 1

Values, types & operators

**DE HOGESCHOOL  
MET HET NETWERK**

Hogeschool PXL – Elfde-Liniestraat 24 – B-3500 Hasselt  
[www.pxl.be](http://www.pxl.be) - [www.pxl.be/facebook](http://www.pxl.be/facebook)



# Variables / constants

let	<p>variabele</p> <p>declaratie &amp; initialisatie in 1 keer</p> <pre>let aantal = 1;</pre> <p>declaratie &amp; initialisatie verspreid</p> <pre>let waarde; waarde = 1;</pre> <p>meerdere declaraties / initialisaties op 1 regel</p> <pre>let a, b = 1, c;</pre>
var	<p>variabele</p> <p><u>nooit</u> gebruiken</p>
const	<p>constante (niet wijzigbaar na toekenning)</p> <p>declaratie &amp; initialisatie in 1 keer</p> <pre>const een = 1;</pre> <p>meerdere declaraties /initialisaties</p> <pre>const twee = 2, drie = 2;</pre>



# Types

## **Primitive datatypes**

- number
- string
- boolean
- null
- undefined

## **Reference datatypes (later)**

- array
- function
- object



# Types: number

Geen onderscheid tussen komma- en gehele getallen

```
let number = 13;  
let decimalNumber = 3.14;  
let avagadroNumber = 6.0221e23;  
  
let teGrootGetal = 1e1000; //Infinity  
  
if (teGrootGetal === Infinity) {  
    console.log("Infinity!");  
}  
  
console.log(1 / teGrootGetal); // 0  
  
console.log(1 / 0); // Infinity
```



# Types: number

Bewerkingen: +, -, \*, /, %

```
let number = 13;  
console.log( number / 2 );    // 6.5  
console.log( number % 2 );    // 1
```



# Types: string

## Single & double quotes

```
let welcome = "Hello\nWorld";
```

## Backticks

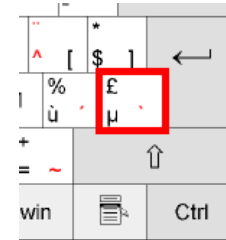
```
let getal=12;  
console.log(`${getal} + 1 = ${getal + 1} !`);  
// 12 + 1 = 13 !  
console.log(`half of 100 is ${100 / 2}`);  
// half of 100 is 50
```

## Geen char

```
let eersteSymbool = welcome[0];  
console.log(eersteSymbool); //H  
console.log(typeof eersteSymbool); // string
```

## Concatenatie: +

```
console.log("a"+"bc"); //abc
```



# Types: boolean

true, false

```
let goedGekeurd = true;
```

negatie: !

```
if ( !goedGekeurd ) {
```

&&, ||

```
if (name == "tim" && age > 2) {
```

# Types: undefined

waarde nog niet toegekend

```
let getal;
```

```
console.log(typeof getal); // undefined
```

# Types: null

```
let a = null;
```



# Wrappers

**Primitives: string, number**

**Reference types: String, Number**

```
let name = "Sofie";  
console.log(typeof name);           // string
```

```
let street = new String("Wetstraat");  
console.log(typeof street);         // object
```

```
console.log(name.toUpperCase());  
    // name wordt omgezet naar String-object  
    // toUppercase van String-object wordt  
    // uitgevoerd
```





# Comparison

```
1 < 2 // true
```

```
1 == 1 // true
```

```
"aardvark" < "zebra" // true
```

```
"a" == "jan" // false
```

## Ternary if / Inline if

```
let age = 19;  
let usertype = age < 18 ? 'minor' : 'adult';
```



# Type coercion

Bij een operatie tussen één of meerdere datatypes  
automatische conversie van datatype

```
console.log(8 * null); // 0 (number)
```

```
console.log("5" - 1); // 4 (number)
```

```
console.log("5" + 1); // 51 (string)
```

```
console.log("five" * 2); // NaN (number)
```

```
console.log(false == 0); // true (boolean)
```

```
console.log(false === 0); // false (boolean)
```



# Type coercion

## Truthy / falsy

```
if ( "a" ){  
    console.log("de string a is truthy");  
}
```

```
if ( !null ){  
    console.log("null is not truthy");  
}
```



# Type coercion bij && ||

```
let result = a || b;  
// als a truthy dan result = a  
// anders result = b
```

```
function assignName( name ){  
    return name || "unknown";  
}
```

```
let person1 = assignName( null );  
let person2 = assignName( "sofie" );  
console.log( person1 ); // unknown  
console.log( person2 ); // sofie
```



# Besluit

**primitive:** numbers, strings, booleans, null & undefined

## binary operators

- arithmetic ( + , - , \* , / , % )
- string concatenation ( + ),
- comparison ( == , != , === , !== , < , > , <= , >= )
- logic ( && , || )

## unary operators

- negatie (`getal = -getal;`)
- not (`if (!geslaagd) { ... }`)
- typeof

## ternary operator ? :

- `let usertype = age < 18 ? 'minor' : 'adult';`

