



Webscripting

Hoofdstuk 3

Functions

DE HOGESCHOOL MET HET NETWERK

Hogeschool PXL – Elfde-Liniestraat 24 – B-3500 Hasselt
www.pxl.be - www.pxl.be/facebook



Definitie via binding

Binding tussen naam en stuk code

De functie is enkel beschikbaar na de definitie

const: niet wijzigbaar door toekenning

let: wel wijzigbaar

scope
square
↓

```
const square = function( x ) {  
    return x * x;  
};  
console.log( square( 2 ) );
```



Definitie via binding

Binding tussen naam en stuk code

Functie enkel beschikbaar na de definitie

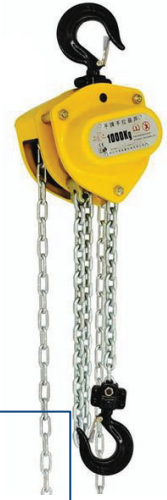
const: niet wijzigbaar door toekenning

let: wel wijzigbaar

```
let safeMode = true;
let launchMissiles = function() {
    console.log("launching missiles");
};
if ( safeMode ) {
    launchMissiles = function() {
        console.log( "nothing happens" );
    };
}
launchMissiles();
```



Definitie via declaration



Function declaration is 'hoisted' (omhooggebracht)
Je mag de functie aanroepen voor de declaratie:

```
future();  
function future() {  
    console.log("You'll never have flying cars");  
}
```

```
You'll never have flying cars
```

```
future();  
const future = function() {  
    console.log("You'll never have flying cars");  
}
```

```
ReferenceError: future is not defined
```



Definitie via arrow-notation

```
const power = (base, exponent) => {  
  let result = 1;  
  for (let i = 0; i < exponent ; i++) {  
    result *= base;  
  }  
  return result;  
}  
console.log(power(2,3)); // 8
```

```
const square = (x) => {return x*x;}  
console.log(square(2)); // 4
```

```
// 1 argument: haakjes mogen weg  
const square2 = x => {return x*x;}
```

```
// zonder accolades: expression w. teruggeg.  
const square3 = (x) => x*x;
```



Optional arguments

Extra argumenten worden genegeerd

```
function square(x) {  
    return x * x;  
};  
console.log(square(2, "a", 12223));           // 4
```

Niet gespecificeerd bij aanroepen: undefined

```
function minus( a, b ) {  
    if ( b == undefined ) {  
        return -a;  
    } else {  
        return a - b;  
    }  
};  
console.log(minus(3));           // -3  
console.log(minus(3,4));        // -1
```



Recursion

Functie die zichzelf aanroept

Machtsverheffing (power)

$\text{base}^{\text{exponent}} = \text{base} * \text{base}^{(\text{exponent}-1)}$

$\text{base}^0 = 1$

```
function power(base, exponent) {  
  if (exponent == 0) {  
    return 1;  
  }  
  return base * power(base, exponent - 1);  
}  
console.log(power(2, 3));
```



Recursion

Functie die zichzelf aanroept

faculteit (factorial)

$$n! = n * (n-1) !$$
$$0! = 1$$

```
function factorial(number) {  
    if (number == 0) {  
        return 1;  
    }  
    return number * factorial(number - 1);  
}  
console.log(factorial(5));
```



Besluit

Declaratie van functions:

```
const twice = function (number) {  
    return number * 2;  
}
```

```
function theeTimes (number) {  
    return number * 3;  
}
```

```
const fourTimes = number => number*4;
```

```
const fiveTimes = (number) => {return number * 5;};
```

