



Javascript

# Table of contents

- Objecten creëren



JavaScript

# OOP in Javascript

# Object aanmaken

1. Object initializer
2. Factory functie
3. Constructor

# Object initializer

= Onmiddellijk creëren van objecten

```
var obj = {  
  prop1: value1,  
  prop2: value2,  
  method1: function1,  
  method2: function2  
};
```

# Demo samen



# Factory functie

= Nuttig indien we meerdere objecten nodig hebben.

```
function createRectangle(width, height, x, y) {  
    var rect = { };  
    rect.width = width;  
    rect.height = height;  
    rect.x = x;  
    rect.y = y;  
    rect.area = function() {  
        return (this.height * this.width);  
    };  
    rect.perimeter = function() {  
        return 2 * (this.height + this.width);  
    };  
    return rect;  
}
```

# Constructor

```
function Rectangle(width, height, x, y) {  
  this.width = width;  
  this.height = height;  
  this.x = x;  
  this.y = y;  
  this.area = function() {  
    return (this.height * this.width);  
  };  
  this.perimeter = function() {  
    return 2 *(this.height + this.width);  
  };  
}
```



# Constructor

```
function Rectangle(width, height, x, y) {  
  this.width = width || 0;  
  this.height = height || 0;  
  this.x = x || 0;  
  this.y = y || 0;  
  this.area = function() {  
    return (this.height * this.width);  
  };  
  this.perimeter = function() {  
    return 2 * (this.height + this.width);  
  };  
}  
  
var rect = new Rectangle();
```

# Prototype v/e object

Objecten worden dynamische gemaakt en voorzien van eigenschappen en methoden.

→ Eigenschappen en methoden kunnen in loop van programma veranderen.

# HAS-A relation

DEMO

# IS-A relation

Overerving? BESTAAT NIET!

→ Prototype chain

→ DEMO

