

---

# Javascript Training



## Case 4: Objecten en functies

---

# Case 4

## Objecten en functies

---

- ▶ Intro objecten
- ▶ JSON
- ▶ Prototyping
  - Instance method
  - Instance variabele
  - Class method (static method)
  - Class variabele (static variabele)

# Case 4

## Intro objecten

---

```
var foo = "bar";  
var test = 1;  
var t = foo + test; -> ????
```

```
var foo = "0";  
var test = 0;  
  
foo == test -> ???  
foo === test -> ???
```

```
new Array(3) == ",,";
```

# Case 4

## Intro objecten

---

Bijna ALLES is een object in Javascript

Elk object is een associatieve array

```
var p = { "voornaam": "Bert", "achternaam": "Janssen" }
```

```
p["voornaam"] === "Bert";  
p.achternaam === "Janssen";
```

# Case 4

## First class functions

---

- ▶ Functies toewijzen aan variabelen
- ▶ Functies als parameters voor andere functies
- ▶ Functies als return waarde van een functie

# Case 4

## JSON

```
{
  "id": 2,
  "naam": "Jeroen Dadema",
  "gebruikersnaam": "jeroen",
  "todos": [
    {
      "id": 1,
      "titel": "Bouwen van app",
      "created": "31-10-2013",
      "priority": "low",
      "description": ""
    },
    {
      "id": 2,
      "titel": "Mail checken",
      "created": "24-10-2013",
      "priority": "low",
      "description": ""
    }
  ]
}
```

# Case 4

## JSON vs andere object notaties

---

```
var t = { "titel": "App bouwen", "priority": "high" }
```

```
var t = Object.create(Object.prototype);  
Object.defineProperty(t, 'titel', {  
  value: "App bouwen",  
  writable: true, enumerable: true, configurable: true  
});
```

```
Object.defineProperty(t, 'priority', {  
  value: "high",  
  writable: true, enumerable: true, configurable: true  
});
```

```
var Todo = function(titel, priority) {  
  this.titel = titel;  
  this.waarde = priority;  
}
```

```
var t = new Todo("App bouwen", "high");
```

# Case 4

## JSON vs andere object notaties

---

```
1  var t = { "titel": "App bouwen", "priority": "high" }

2  var t = Object.create(Object.prototype);
   Object.defineProperty(t, 'titel', {
       value: "App bouwen",
       writable: true, enumerable: true, configurable: true
   });

   Object.defineProperty(t, 'priority', {
       value: "high",
       writable: true, enumerable: true, configurable: true
   });

3  var Todo = function(titel, priority) {
   this.titel = titel;
   this.waarde = priority;
   }

   var t = new Todo("App bouwen", "high");
```



# Case 4

## Prototyping

---

Object.prototype

Key	Value
constructor	Object
toString	function() { [native code]}
toLocaleString	function() { [native code]}
valueOf	function() { [native code]}
hasOwnProperty	function() { [native code]}
isPrototypeOf	function() { [native code]}
propertyIsEnumerable	function() { [native code]}

# Javascript

## Instance variabele

---

```
var t = new Todo("App bouwen", "high");
```

```
var Todo = function(titel, priority) {  
    this.titel = titel;  
    this.waarde = priority;  
}
```

```
var todoInstance = new Todo("App bouwen", "high");  
todoInstance.toString(); → "[object Object]";
```

# Javascript

## Instance variabele / methode

---

```
var Todo = function(titel, priority) {  
  this.titel = titel;  
  this.waarde = priority;  
}
```

```
Todo.prototype.toString = function() {  
  return "titel [" + this.titel + "], priority [" + this.priority + "];"  
}
```

```
var todoInstance = new Todo("App bouwen", "high");  
todoInstance.toString(); → "titel [App bouwen], priority [high]"
```

```
console.log("Todo: " + todoInstance);  
→ "Todo: titel [App bouwen], priority [high]"
```

# Javascript

## Prototype

---

Object.prototype

Key	Value
toString	function() { [native code]}

Todo.prototype

Key	Value
constructor	Todo

todoInstance

Key	Value
titel	"App bouwen"
priority	"high"

# Javascript

## Prototype

---

### Object.prototype

Key	Value
toString	function() { [native code]}

### Todo.prototype

Key	Value
constructor	Todo
toString	function() { return "titel [" + this.titel + "], priority [" + this.priority + "]; }

Key	Value
titel	"App bouwen"
priority	"high"

# Javascript

## Static variabele/methode

---

```
var Todo = function(waarde) {  
    this.instanceVar = waarde  
}
```

```
Todo.STATIC_VAR = "staticWaarde";  
Todo.staticMethod = function() {  
    console.log("staticMethod");  
}
```

```
var todoInstance = new Todo("instanceWaarde");  
var todoInstance2 = new Todo("instanceWaarde");
```

```
todoInstance.instanceVar2 = "instanceWaarde2";  
  
todoInstance.instanceVar === todoInstance2.instanceVar  
todoInstance.instanceVar2 === todoInstance2.instanceVar2  
Todo.STATIC_VAR;  
todoInstance.STATIC_VAR;  
Todo.staticMethod();  
todoInstance.staticMethod();
```

# Javascript

## Return statement

---

```
function test() {  
    return  
    {  
        test: "waarde"  
    }  
}
```

```
function test() {  
    return {  
        test: "waarde"  
    }  
}
```