



# INTERACTIEVE WEBSITES

## JavaScript LES 5

*Objecten*

Graduaat Informatica Programmeren

# OPDRACHTEN

## Opdracht: Pinguïns

Maak 3 pinguïns aan via een object literal. Deze geef je de namen **gunter**, **ramon** en **vreemdePenguin**. Ze hebben onderstaande **properties**. Merk op dat de laatste geen naam heeft, maar character!

- **name**: "Gunter"
- **origin**: "Adventure Time"
- **canFly**: false
- **name**: "Ramon"
- **origin**: "Happy Feet"
- **canFly**: true
- **character**: "pinguïn McPenguin"
- **origin**: "Donald Duck"
- **canFly**: true

Geef deze ook een **methode** sayHello. Deze schrijft het volgende naar de **console**:

- gunter => "Hoi! Ik ben <name> en ik zeg KWAK"
- ramon => "Ik ben blij je te ontmoeten."
- vreemdePenguin => "Hola, Ik ben een pinguin zonder naam. Mijn naam is <name>!"

De vierde pinguïn, **fred**, maak je aan via het gebruik van een object constructor function (blueprint).  
Name = Fred, origin = Sitting Ducks, canFly = true

Geef de functie de naam **Penguin** en als parameters kun je name, origin & canFly meegeven.  
Voeg ook de methode sayHello aan **fred** toe die afdrukt naar de console "Hallo daar!"

**Voer onderstaande opdrachten uit en log deze steeds in de console.**

**Zie achteraan als bijlagen voor een schermafbeelding van het resultaat!**

- Creëer een nieuwe variabele pinguïns met een lijst (array) met de objecten
- Vraag het eerste object in de array op en print alle properties en methods van dit object.
- Creëer een nieuwe variabele nogEenPenguin en stel deze gelijk aan de tweede pinguïn in de array.
- Print de naam van de laatste. Zorg dat de code nog werkt indien er meer of minder objecten in de array zitten.
- We hebben één vreemde pinguïn. Voeg deze toe aan het einde van onze array!
- Toon de lengte van onze array.
- Zet voor de eerste pinguïn de property canFly op true en druk deze af.
- Roep de sayHello methode op voor de eerste pinguïn in de array!
- Schrijf een lus om iedere pinguïn op te vragen en print de namen naar de console.
- Schrijf een lus om voor iedere pinguïn de methode sayHello te laten uitvoeren.
- Schrijf een lus om voor iedere pinguïn een nieuwe property toe te voegen: namelijk numberOfFeet en geef deze voor allemaal de waarde 2. Geef alle objecten weer in de console.
- Schrijf een lus en druk voor iedere pinguïn die kan vliegen (canFly = true) een melding naar de console af met de <naam van de pinguïn> + " kan vliegen!" We geven niets weer indien hij niet kan vliegen!

Bewaar deze oefening als **pingiun.html**

## Opdracht: Afstandsbediening

- A. Maak volgende html-pagina. De **waarden** van de radiobuttons zijn **tv**, **stereo** of **pc**.
- B. Definieer een **object** “toestel” met properties **volume** en **kanaal** via een constructor function.
- C. Maak **3 instanties** van dit object nl. **tv**, **stereo** en **pc**. Deze naamgeving moet overeenkomen met de **values** van de **radiobuttons**! De initiële waarden van het volume en kanaal is 0.
- D. De velden voor **volume** en **kanaal** zijn niet aanpasbaar.
- E. Maak een functie **setWaarden()**. Deze functie zet in de 6 velden de objectwaarden.  
Bijvoorbeeld in tekstveld **txtTvVolume** komt de waarde van het **volume** voor de **tv**. (tv.volume)  
Deze functie roepen we ook op bij het **laden** van het venster. Alles krijgt dus de waarde 0.
- F. Maak een functie **changeVolume()**. Deze functie wordt opgeroepen door de knoppen “**verhogen**” en “**verlagen**”. Je kan via “**this**” nagaan via het **id** (this.id) op welke knop geklikt is.  
Deze functie gaat eerst na welke radiobutton aangevinkt is. Vervolgens verhoog of verlaag je de waarde van de **eigenschap volume** van **tv/stereo/pc** van de aangevinkte radiobutton.
- G. Gebruik de **eval-functie** (zoek dit op!) om het **volume** met **1** te **vermeerderen/verminderen**.  
Bijvoorbeeld.  

```
let objectnaam = "tv"; // tv moet overeenkomen met de value van de geselecteerde radio-button  
eval(objectnaam.value + ".volume++");
```

  
Roep op het einde terug de functie **setWaarden()** op.
- H. Maak een functie **changeKanaal()**.  
Gebruik dezelfde werkwijze als hierboven. I.p.v. met 1 te vermeerderen/ verminderen, wordt het kanaal gelijkgesteld aan het ingevuld tekstveld.

### Afstandsbediening

Toestel	Volume	Kanaal
<input type="radio"/> Tv	<input type="text" value="0"/>	<input type="text" value="0"/>
<input checked="" type="radio"/> Stereo	<input type="text" value="7"/>	<input type="text" value="12"/>
<input type="radio"/> PC	<input type="text" value="0"/>	<input type="text" value="0"/>
volume:	<div><div>verhogen</div><div>verlagen</div></div>	
kanaal:	<div><input type="text" value="12"/><div>veranderen</div></div>	

Bewaar deze oefening als **afstandbediening.html**

## Opdracht: Objecten in een array

Maak een array **arrUsers** met **7 objecten** in.

Dit object heeft 1 property **user**. Deze bevat de **namen** van de users. (Zelf ingeven!)

Via de **map methode** geven we ieder object in de array een nieuwe property **punten** en vullen deze automatisch op met een random waarde van 0 tot 100.

Toon de array **arrUsers** in de console.

```
▼ (7) [...]
  ▶ 0: Object { user: "Jef", punten: 3 }
  ▶ 1: Object { user: "Anja", punten: 94 }
  ▶ 2: Object { user: "Roos", punten: 77 }
  ▶ 3: Object { user: "Didier", punten: 87 }
  ▶ 4: Object { user: "Louis", punten: 53 }
  ▶ 5: Object { user: "Emma", punten: 9 }
  ▶ 6: Object { user: "Marnix", punten: 49 }
    length: 7
  ▶ <prototype>: Array []
```

Via de **methode filter** halen we er alle objecten eruit waar de punten groter zijn dan 50.

Bewaar deze in de array **arrSelected** en toon deze in de console.

```
▼ (7) [...]
  ▶ 0: Object { user: "Jef", punten: 3 }
  ▶ 1: Object { user: "Anja", punten: 94 }
  ▶ 2: Object { user: "Roos", punten: 77 }
  ▶ 3: Object { user: "Didier", punten: 87 }
  ▶ 4: Object { user: "Louis", punten: 53 }
  ▶ 5: Object { user: "Emma", punten: 9 }
  ▶ 6: Object { user: "Marnix", punten: 49 }
    length: 7
  ▶ <prototype>: Array []

▼ (4) [...]
  ▶ 0: Object { user: "Anja", punten: 94 }
  ▶ 1: Object { user: "Roos", punten: 77 }
  ▶ 2: Object { user: "Didier", punten: 87 }
  ▶ 3: Object { user: "Louis", punten: 53 }
    length: 4
  ▶ <prototype>: Array []

Het gemiddelde van de users boven de 50 is: 77.75
```

Via de **methode reduce** bereken je de som van alle punten die in de **arrSelected** zitten.

Toon het **gemiddelde** d.m.v. een zin in de console.

Opgelet: zorg ervoor wanneer we de objecten in het begin uitbreiden of verminderen dat het gemiddelde nog steeds correct is!

Bewaar deze oefening als **users\_boven\_50.html**

# BIJLAGEN

Schermafruk console opdracht pinguïns.

```
// A
▼ (3) [...]
  ▶ 0: Object { name: "Gunter", origin: "Adventure Time", canFly: true, ... }
  ▶ 1: Object { name: "Ramon", origin: "Happy Feet", canFly: true, ... }
  ▶ 2: Object { name: "Fred", origin: "Sitting Ducks", canFly: true, ... }
  ▶ 3: Object { character: "pinguïn McPenguin", origin: "Donald Duck", canFly: true, ... }
    length: 4
  ▶ <prototype>: Array []

// B
▶ Object { name: "Gunter", origin: "Adventure Time", canFly: false, sayHello: sayHello() ↵ }

// C
▶ Object { name: "Ramon", origin: "Happy Feet", canFly: true, sayHello: sayHello() ↵ }

// D
Fred

// E

// F

4

// G
true

// H
Hallo daar!

// I
Gunter
Ramon
Fred
undefined

// J
Hoi! Ik ben Gunter en ik zeg KWAK
Ik ben blij je te ontmoeten.
Hallo daar!
Hola, Ik ben een pinguïn zonder naam. Mijn naam is undefined!

// K
▶ Array(4) [ {}, {}, {}, {} ]

// L
0
Gunter kan vliegen!
1
Ramon kan vliegen!
2
3
undefined kan vliegen!
```