LEERDOELEN

Leerdoelen:

* Ik kan een variabele gebruiken in mijn JavaScript code √

* Ik maak de juiste keuzen tussen consten let.

Let = variabel, waarde kan wijzigen  
const = constant, waarde wijzigd niet

* Ik gebruik camelCaseNotation

jaDatDoeIk

* Ik ken de primitive types van JavaScript, met als belangrijkste:
  + String 'Dit is een String'
  + Boolean true / false
  + Undefined
  + Null Null
  + Number 1, 2, 3
* Ik kan achterhalen wat voor een type een variable is.

In console typeof + naam

* Ik weet wat Dynamically Typed betekent.

Kan je veranderen

Leerdoelen:

* Ik kan uitleggen wat een command line is. **check**
* Ik kan uitleggen wat een GUI is. **check**
* Ik kan uitleggen waarom sommige taken handiger uit te voeren zijn met de command line ten opzichte van een Graphic User Interface (GUI). **check**
* Ik kan een terminal-programma starten en afsluiten **check**
* Ik kan commando's gebruiken om de volgende dingen te doen:
  + bepalen in welke folder ik sta **check**
  + navigeren naar andere folders **check**
  + een nieuwe file aanmaken **check**
  + een file en een folder kopiëren **check**
  + een file en een folder verwijderen **check**
  + zoeken in een grote verzameling bestanden **check**

Leerdoelen:

* Ik kan inline comments schrijven. **Check //**
* Ik kan blok comments schrijven. **Check /\* \*/**
* Ik weet de shortcut in mijn editor om code uit te "commenten". **Shift alt A**

**Leerdoelen:**

* Ik weet wat een comparison operator is.

**<, >, ==, ===, <= , >=, !=**

* Ik kan een == gebruiken

**controleert de waarde**

* Ik kan een === gebruiken   
  **controleert de waarde en het type (string, number etc)**
* Ik kan uitleggen wat het verschil is tussen == en ===

**controleert de waarde en het type (string, number etc)**

* Ik weet wat een logical operator is.

**&& and, || or**

* Ik kan && gebruiken **check**
* Ik kan || gebruiken **check**
* Ik begrijp waarom de titel van deze module geen typefout is.   
  **JS: Comparison && Logical Operators =** JS: Comparison **AND** Logical Operators

**Leerdoelen:**

* Ik kan een object maken in JavaScript **check**

let person = {

name: 'Robin',

age: '12',  
 sport: true

};  
 console.log(person);  
  
 ***=> person is het object, name = property, 'Robin' = value***

* Ik weet hoe ik een property van het object kan aanspreken en gebruiken

1 DOT notation –-- ***person.name = 'Jan'***

2 BRACKET notation –-- ***person['name'] = 'Jan'***

* Ik kan een array maken in JavaScript

**let soortenSport = [**

**'voetbal',**

**'tennis',**

**'hardlopen',**

**'boxen',**

**'icehockey'**

**];**

* Ik kan een element aan een array toevoegen

**let soortenSport [**

**'voetbal',**

**'tennis',**

**'hardlopen',**

**'boxen'**

**];**

**soortenSport ['icehockey']; of**

**soortenSport [14];**

* Ik kan 1 element van de array aanspreken en gebruiken

***console.log(soortenSport[3]); dit is boxen***, want voetbal = 0  
 tennis = 1  
 hardlopen = 2

**Leerdoelen:**

* Ik kan een functie schrijven

|  |  |
| --- | --- |
| function greet () {    console.log('Hello World');  }  greet(); | Function drive () {  console.log('Hello Driver');  } drive(); |
|  |  |
| Function greet (name, lastName) {  console.log('Hello' + ' ' name + ' ' + lastName);  }  greet(Robin, ten Barge); |  |

* Ik kan een functie aanroepen (calling a function)

zie boven  
 greet(); en drive();

Leerdoelen:

* Ik kan functies op een abstract niveau uitschrijven.

**function zetKoffie () {**

**vul koffiezetter met water;  
 doe filter in koffiezetter;**

**doe koffie in filter;**

**}  
 zetKoffie()**

* Ik kan een functie schrijven met parameters.

**function zetKoffie (***aantalSchepjesKoffie***) {**

**vul koffiezetter met water;  
 doe filter in koffiezetter;**

**doe** *aantalSchepjesKoffie***;**

**}  
 zetKoffie(8)**

* Ik kan een functie aanroepen.

z**etKoffie()**

* Ik kan een functie aanroepen met een argument.

**ZetKoffie(8)**

* Ik kan het resultaat van een functie gebruiken.
* Ik weet wat return statements zijn en waarom ik ze wil gebruiken.