Dynamic Web Development

Syntax

Javascript gebruiken

Javascript

- javascript is een interpreted language.
  - Dit wil zeggen dat de code niet gecompileerd wordt. De code wordt door de browser geïnterpreteerd zoals ze geschreven is.
- Belangrijk : javascript heeft weinig tot niets te maken met java
  - (hoewel de naam anders doet vermoeden)
- Wordt gebruikt om onze website dynamischer te maken

ecmascript

- Javascript ≈ ecmascript
- Javascript => is de programmeertaal en implementeert ecmascript
- Ecmascript => is een standaard waaraan voldaan moet worden.

Meer info op ⇒ freecodecamp

Javascript gebruiken

- Embedden
  - In de html file
- Linken
  - In een aparte JS file
  - Geniet de voorkeur

Javascript embedding

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Demo 1 </title>
</head>

<body>

  <script>
    console.log("demo")
  </script>
</body>

</html>
```

Javascript linken

```html
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Demo 1 </title>
</head>

<body>

  <script src="index.js"></script>
</body>

</html>
```

```js
console.log("test");
```

Script tags

- Je kan de script tag zowel in head tag als in de body tag plaatsen
- HTML wordt van boven naar onder gelezen.
  - Dit wil ook zeggen dat eerder komende js code eerder wordt uitgevoerd.
- Denk dus goed na over de positie van je js code.
- Meestal is vlak voor het einde van het body element een goede plaats.

Variabelen & Constanten

Dynamische types

- Een variabele heeft geen vast data type
- Een variabele kan het ene moment een string zijn en het andere moment een integer
- We hoeven variabelen en constanten niet vooraf te voorzien van een type
- Dit zorgt voor flexibiliteit MAAR is heel fout gevoelig

Const

- Constanten
- Moeten gedeclareerd worden bij initialisatie
- Een variabele die achteraf niet meer gewijzigd kan worden
- Properties van een constante kunnen wel nog wijzigen
- Items van een constante array kunnen ook wijzigen

[codepen]

Let

- Voor variabelen
- Function scoped, global scoped, block scoped
  - De variabele bestaat enkel binnen de { }
- Kan niet opnieuw aangemaakt worden

[codepen]

Var

- Vooral in oudere code
- Function en global scoped (niet blocked)
- Kan wel opnieuw gemaakt worden
  - Foutgevoeliger dan let
- Voorkeur gaat naar let

[codepen]

Datatypes

Datatypes in javascript zijn beperkt:

- String
- Number
- Boolean
- Array
- Object
- Undefined
- NULL

[codepen]

Naamgeving

We hanteren onderstaande regels voor het benoemen van variabelen:

- Namen kunnen letters, cijfers, underscores en dollar tekens bevatten
- Namen moeten starten met een letter, $ of _
- Namen zijn hoofdletter gevoelig.
- Je mag geen reserved keywords gebruiken
- Bij voorkeur camelCasing gebruiken.

Objecten

- Objecten bestaan net zoals in andere talen
  - Objecten hebben eigenschappen
  - Objecten hebben functies
- Eigenschappen van objecten kunnen aangesproken worden a.d.h.v
  - Dictionary notatie
  - Property notatie (dotnotatie)
- Objecten aanmaken
  - Via new Object()
  - Later via Classes of json notatie

[codepen]

Arrays

Met Array kunnen we collecties van data bijhouden

- In een javascript array moeten niet alle elementen hetzelfde type hebben
- JS arrays hebben geen vaste lengte
  - De lengte van een array wordt bepaald door het laatste element
- Elementen kunnen aangesproken worden aan de hand van [ ]

[Codepen]

Array methodes

- push
  - Voegt een item achteraan de array toe
- pop
  - Geeft het laatste item terug + verwijdert dit
- length
  - Geeft de lengte terug van de array
- shift
  - Geeft het eerste item terug + verwijdert dit
- unshift
  - Voegt een item vooraan de array toe

[codepen]

Controlestructuren

Controlestructuren

Zoals in andere programmeertalen, hebben we in JS ook controlestructuren.

Deze kunnen we standaard opsplitsen in

- Selecties
- Iteraties

Selecties

- If structuur => [codepen]
- Switch structuur => [codepen]

Iteraties

- For structuur => [codepen]
- While structuur => [codepen]
- Do/while structuur => [codepen]
- Foreach structuur => [codepen]
- for/in structuur => [codepen]
- for/of structuur => [codepen]

Functies

Functies

Functies/methodes kennen we ook al uit andere talen.

In tegenstelling tot strongly typed programmeertalen (≈ geen dynamic typing) geven we geen types mee.

Voorbeeld functies

- Zonder parameters & zonder return => [codepen]
- Met parameter & return => [codepen]

Self invoking function

Een self invoking functie, is een functie die zichzelf direct uitvoert.

Het is aan te raden je code hierin te plaatsen

[codepen]

Voordelen self invoking function

- Variabelen zijn beperkt in scope
- Variabelen gedeclareerd in de function scope kunnen niet beïnvloed worden door andere js files
- We kunnen niet per ongeluk variabelen van andere files (libraries) overschrijven.