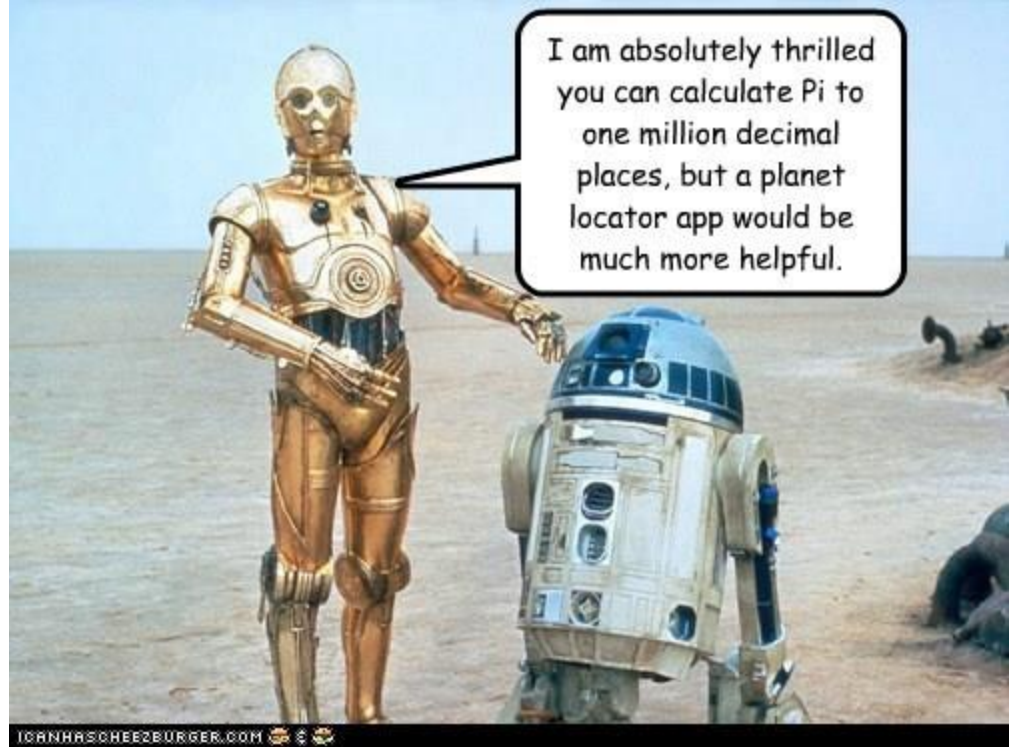


Javascript 2

DOM

Variables

Functions



Vorige les...

- beginnen met javascript
 - een **script-element** opnemen in een HTML-pagina
 - beginnen met het programmeren in **javaScript**
 - een **button** in HTML opnemen
 - deze button binden aan een **javascript-variabele**
 - een **click-event** toevoegen aan de javascript-variabele
- het Date-object
 - hoe te gebruiken
 - zelf verder onderzoeken
 - de opdracht uitvoeren en inleveren

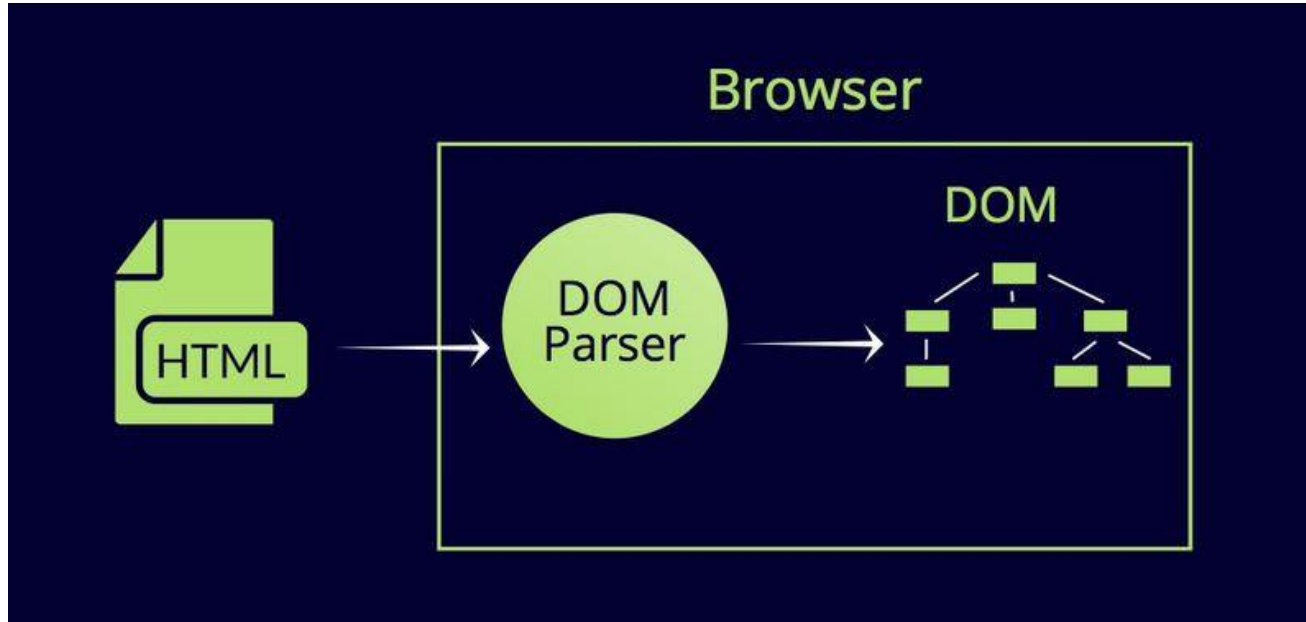
Deze les...

- Document Object Model **DOM**
- javascript
 - een extern **script** linken aan een HTML-pagina
 - Javascript **variables** declareren
 - Let const var
 - Scope van een variable
 - **Functions** in javascript
 - Een **text input** vanuit HTML gebruiken in javascript
- Kennis vorige les toepassen
 - deze button binden aan een **javascript-variabele**
 - een **click-event** toevoegen aan de javascript-variabele
 - Output 'schrijven' naar HTML

DOM

- Document Object Model

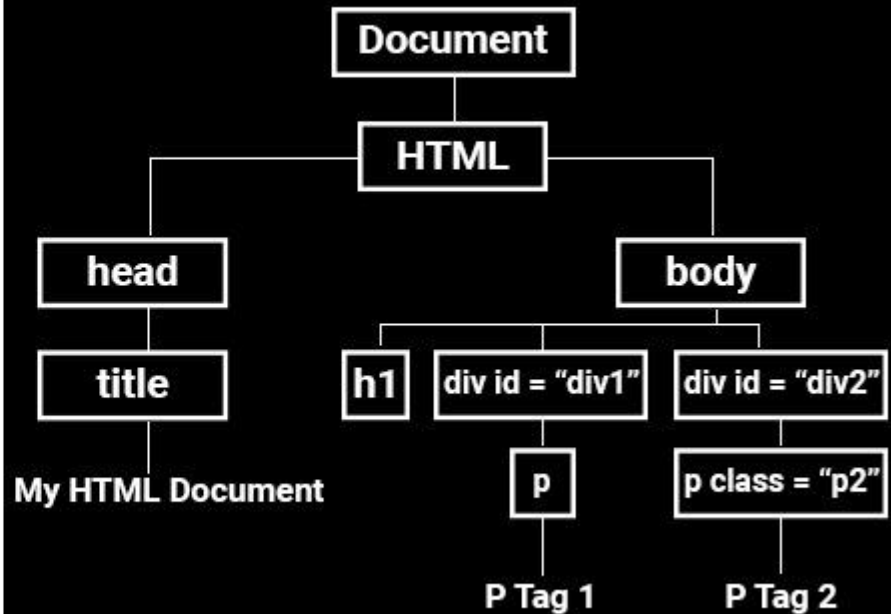
DOM Document Object Model



HTML Document

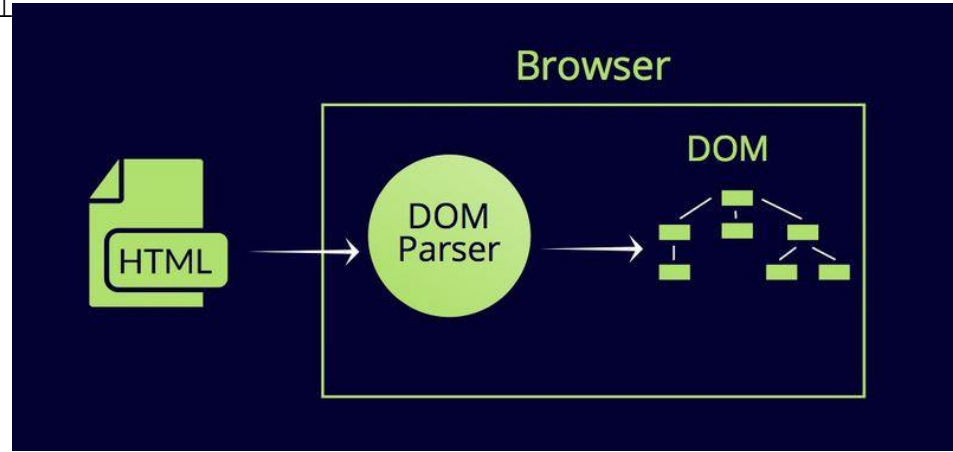
```
index.html x
1 <html>
2   <head>
3     <title>My HTML Document</title>
4   </head>
5
6   <body>
7     <h1>Heading</h1>
8     <div id="div1">
9       <p>P Tag 1</p>
10    </div>
11    <div id="div2">
12      <p class="p2">P Tag 2</p>
13    </div>
14  </body>
15 </html>
```

Document Object Model (DOM)



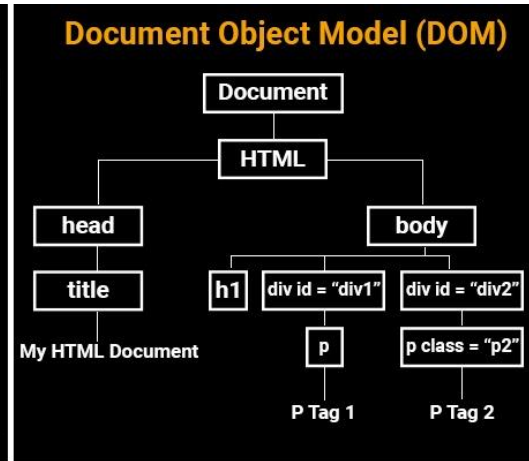
DOM Document Object Model

- 1) HTML elements worden 'geparsed'
- 2) Komen terecht in een DOM Tree
- 3) Javascript kan via de DOM Tree HTML elements benaderen, maar ook genereren



HTML Document

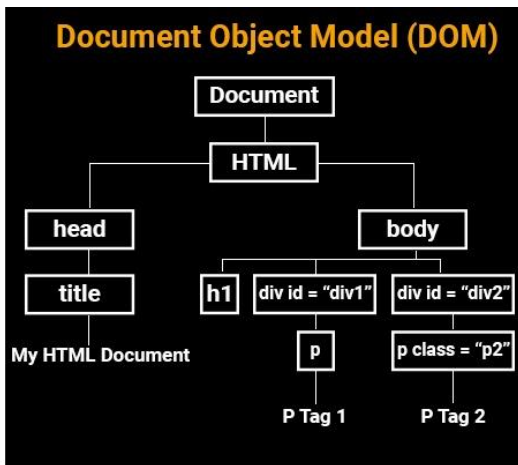
```
index.html *
1  <html>
2    <head>
3      <title>My HTML Document</title>
4    </head>
5
6    <body>
7      <h1>Heading</h1>
8      <div id="div1">
9        <p>P Tag 1</p>
10     </div>
11     <div id="div2">
12       <p class="p2">P Tag 2</p>
13     </div>
14   </body>
15 </html>
```



DOM Document Object Model

Javascript kan via de DOM Tree HTML elements benaderen, maar ook genereren (appendChild createElement)

```
HTML Document
index.html x
1 <html>
2 <head>
3 <title>My HTML Document</title>
4 </head>
5
6 <body>
7 <h1>Heading</h1>
8 <div id="div1">
9 <p>P Tag 1</p>
10 </div>
11 <div id="div2">
12 <p class="p2">P Tag 2</p>
13 </div>
14 </body>
15 </html>
```



The following is a brief list of common APIs in web and

- `document.getElementById(id)`
- `document.getElementsByTagName(name)`
- `document.createElement(name)`
- `parentNode.appendChild(node)`
- `element.innerHTML`
- `element.style.left`
- `element.setAttribute()`
- `element.getAttribute()`
- `element.addEventListener()`
- `window.content`
- `window.onload`
- `window.scrollTo()`

JS



Variables

var

komt voor in legacy code,
voornaamste probleem 'hoisting'

let

- beschikbaar vanaf EcmaScript 6
- geeft controle over de scope
Block scope in een function of
een code block
- herdefinieren van een variable
binnen een scope geeft een
error

const

- zie let, waarde kan binnen een
scope niet gewijzigd worden

JavaScript Variable Example

~~var = 1;~~
let = 2;
const = 5;



De scope van een variable

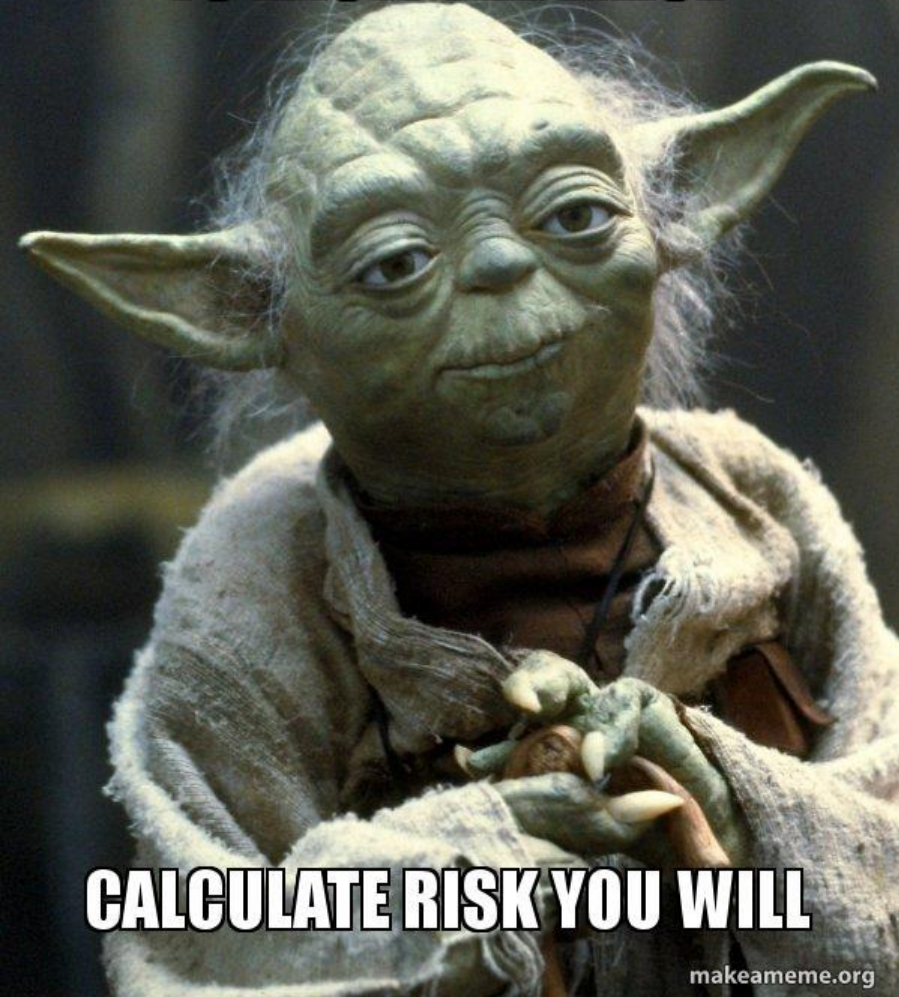
In regel 9 wordt `getal1` gedeclareerd de scope van `getal1` is beperkt tot binnen de curly brackets.

In regel 13 wordt ook `getal1` gedeclareerd, dit is dus niet dezelfde variable als `getal1` in regel 9

```
7  mijnButton.addEventListener('click', function(){
8
9      let getal1 = document.getElementById('getal1').value;
10
11  })
12
13  function telOp(getal1, getal2){
14
15      let antwoord = getal1 + getal2;
16
17  }
```

De curly brackets
bepalen de scope
van variable
'getal1'

TO KNOW THE VALUE



CALCULATE RISK YOU WILL

makeameme.org



HandsOn



Amsterdam
TECH ACADEMY

HandsOn

- **use case**

maak een rekenmachine met
2 invoervelden, een
button bereken, geeft
antwoord voor 4 soorten
bewerkingen

- **Demo** http://schw.hosts1.ma-cloud.nl/javascript/les_2/les2.html

Functions Javascript

bereken	20	30
---------	----	----

Optellen 50

Aftrekken

Vermenigvuldigen 600

Delen

HTML

Functions Javascript

Optellen

Aftrekken

Vermenigvuldigen

Delen

```
1  <!doctype html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Functions</title>
6          <style>body{ font-family: monospace;} </style>
7      </head>
8
9      <body>
10         <h3>Functions Javascript</h3>
11         <!-- input -->
12         <input type="button" value="bereken" id="mijnButton">
13         <input type="text" id="getal1" placeholder="getal 1">
14         <input type="text" id="getal2" placeholder="getal 2">
15         <!-- output -->
16         <br><br>
17         Optellen <span id = "optellen">....</span>
18         <br><br>
19         Aftrekken <span id = "aftrekken">....</span>
20         <br><br>
21         Vermenigvuldigen <span id = "vermenigvuldigen">....</span>
22         <br><br>
23         Delen <span id = "delen">....</span>
24         <!-- javascript -->
25         <script src="script1.js"></script>
26     </body>
27 </html>
```

Structuur Javascript

Bind HTML
elements aan
JS via DOM

eventlistener
met function

function

```
'use strict';  
  
let mijnButton = document.getElementById('mijnButton');  
let optellen = document.getElementById('optellen');  
let vermenigvuldigen = document.getElementById('vermenigvuldigen');  
  
mijnButton.addEventListener('click', function(){ ...  
  
    > function telOp(getal1, getal2){ ...  
        }  
  
    function vermenigvuldig(getal1, getal2){ ...  
        }
```

Event listener

bereken

20

30

Optellen 50

Aftrekken

Vermenigvuldigen 600

Delen

eventlistener
met function

```
9  
10  
11 mijnButton.addEventListener('click', function(){  
12     // lees de waarden van de text input velden in  
13     // let op DIT ZIJN STRINGS  
14     let getal1 = document.getElementById('getal1').value;  
15     let getal2 = document.getElementById('getal2').value;  
16  
17     // zet de ingelezen strings om in getallen  
18     getal1 = parseFloat(getal1);  
19     getal2 = parseFloat(getal2);  
20  
21     // roep de function aan met arguments  
22     telOp(getal1,getal2);  
23     vermenigvuldig(getal1,getal2);  
24 })
```

Call naar
function telOp()
en
vermenigvuldig()

Function telOp()

function
name

arguments

De argumenten worden in de functie definitie gedeclareerd.

De scope van getal1 en getal2 is de function zelf {tussen de curly brackets}

```
function telOp(getal1, getal2){  
    let antwoord = getal1 + getal2; // doe de bewerking  
    console.log("optellen " + antwoord); // laat ook in de console zien  
    optellen.innerHTML = antwoord; // schrijf naar HTML  
}
```

Javascript compleet

```
1  use strict';
2  let mijnButton = document.getElementById('mijnButton');
3  let optellen = document.getElementById('optellen');
4  let vermenigvuldigen = document.getElementById('vermenigvuldigen');
5  mijnButton.addEventListener('click', function(){
6      // lees de waarden van de text input velden in
7      // let op DIT ZIJN STRINGS
8      let getal1 = document.getElementById('getal1').value;
9      let getal2 = document.getElementById('getal2').value;
10     // zet de ingelezen strings om in getallen
11     getal1 = parseFloat(getal1);
12     getal2 = parseFloat(getal2);
13     // roep de function aan met arguments
14     telOp(getal1, getal2);
15     vermenigvuldig(getal1, getal2);
16 })
17 function telOp(getal1, getal2){
18     let antwoord = getal1 + getal2; // doe de bewerking
19     console.log("optellen " + antwoord); // laat ook in de console zien
20     optellen.innerHTML = antwoord; // schrijf naar HTML
21 }
22 function vermenigvuldig(getal1, getal2){
23     let antwoord = getal1 * getal2; // doe de bewerking
24     console.log("vermenigvuldigen " + antwoord); // laat ook in de console zien
25     vermenigvuldigen.innerHTML = antwoord; // schrijf naar HTML
26 }
```

Nu
j i j



Doen 1

- Maak de code
- Run de code in jouw browser, check de console output

Functions Javascript

bereken

12

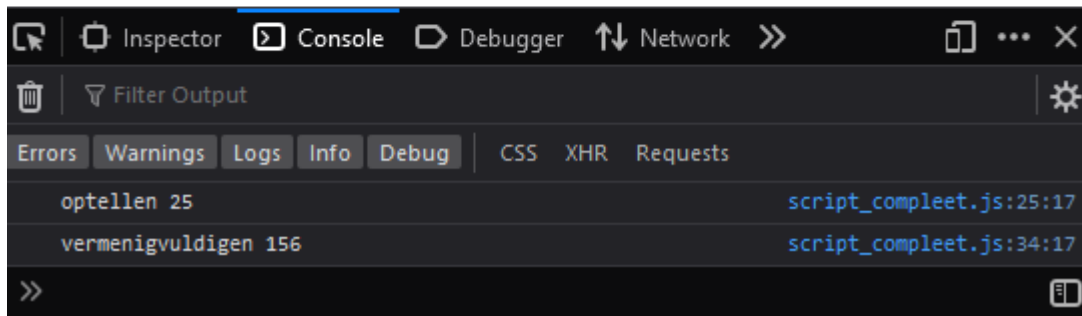
13

Optellen 25

Aftrekken

Vermenigvuldigen 156

Delen



Doen 2 opdracht

- Voeg code toe waarmee je de bewerking Aftrekken en Delen ook uitvoert

Functions Javascript

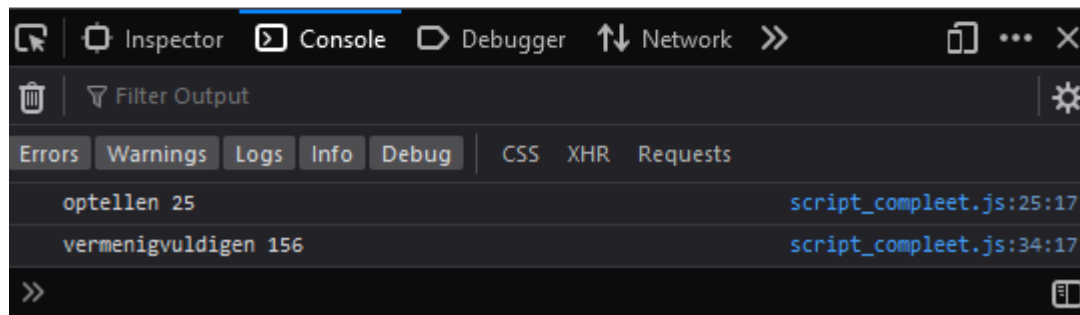
bereken

Optellen 25

Aftrekken

Vermenigvuldigen 156

Delen



Challenge

- Als je bij de bewerking 'delen' als 2e getal een 0 invoert krijg je als resultaat "Infinity".
- Zorg dat jouw code in dit geval weergeeft "delen door 0 kan niet, geef een ander getal"

Vragen

- Wat is het voordeel/nadeel van `"use strict"`
- Wat wordt bedoeld met `"hoisting"` binnen javascript
- Wat is camelCase
- Wat wordt bedoeld met `"Scope"` van een variable

thanks
folks!

