

Programmeren P1

Javascript

Functions



WEEK 6





Leerdoelen

- Ik weet wat functies zijn
- Ik weet wanneer je een functie moet gebruiken
- Ik kan uitleggen waarom een functie tijdswinst oplevert tijdens programmeren
- Ik weet verschillende manieren hoe je een functie moet oproepen
- Ik kan bovenstaande kennis toepassen in de opdrachten



Wat is een function?

- Blok met code dat een bepaalde functie moet uitvoeren
- De code wordt pas uitgevoerd als het opgeroepen wordt

Nu je weet wat een functie is, waarom is het handig om functies te maken, in plaats van de code los te schrijven?



Hoe maak je een functie?

Een functie bestaat uit de “function” keyword gevolgd door:

- De naam van de functie
- Een lijst met brackets voor de functie, tussen haakjes en gescheiden door komma's.
- Code in de functie dat uitgevoerd moet worden, tussen de accolades (curly brackets)
- Dit noem je een *function declaration*

Hoe maak je een functie?



```
1 function changeColor() {  
2     //hier je de code te staan  
3 }
```

Hoe werkt een functie?

De code in je functie wordt alleen gelezen als je de functie oproept.

Zolang de functie niet opgeroepen wordt, leest je browser de code niet.

Je roept de functie op door de naam te schrijven van de functie + de () (brackets)

Deze brackets noem je *parameters*



```
1  function changeColor() {  
2      //hier komt je code te staan  
3  
4  }  
5  
6  changeColor();
```



Doe mee!

- Maak een functie genaamd **changeColor**
- In de functie plaats je: `console.log("de functie werkt")`
- Roep de functie op.
- Waar kun je nu controleren of de functie werkt?



Demo



Opdracht

Aan de hand van de bovenstaande theorie en de theorie/opdrachten van vorige week, maak je in Canvas **opdracht 6.1 – Maak een functie**.

Uiterlijk inleveren: zie Canvas



onclick

Je weet nu hoe je een functie moet oproepen.

Maar een functie oproepen in je code is eigenlijk nutteloos, het liefst wil je dat een functie opgeroepen wordt als een gebruiker iets op de website doet, bijvoorbeeld als je op een button klikt.

Dit doe je met een **onclick** attribute.

onclick

Recap:

- Een attribute plaats je in een html-tag.
- Een attribute is extra informatie voor je html-tag

OnClick is dus ook extra informatie voor je html-tag. Als er dan op je button wordt geklikt, wordt de functie gelezen en uitgevoerd.

```
1 <button onclick="changeColor()">Click me!</button>
```



Opdracht

Maak nu opdracht 6.2 – onClick & opdracht 6.3 – verdieping

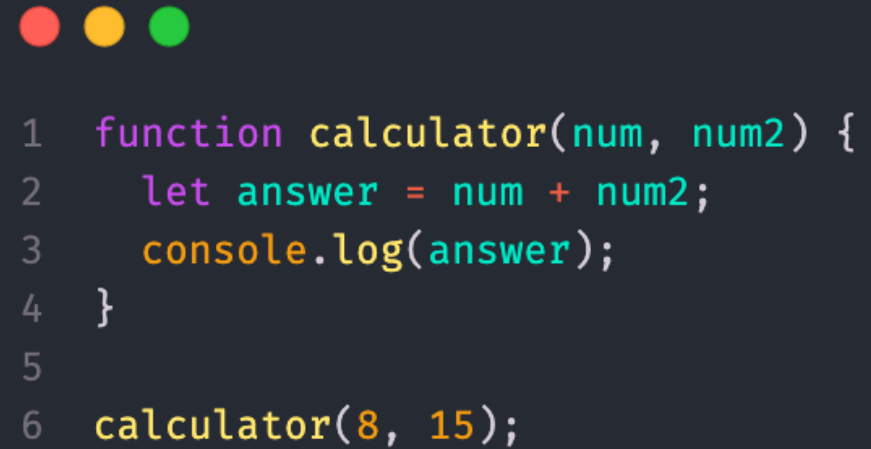
Uiterlijk inleveren: zie Canvas

Parameters

De parameters van een functie is een soort doorgeefluik voor de functie.

De functie krijgt bepaalde waarde binnen via de parameters en in de functie wordt daar wat mee gedaan.

Wat is het antwoord dat in je console komt te staan? Tip: type het over.



```
1 function calculator(num, num2) {  
2     let answer = num + num2;  
3     console.log(answer);  
4 }  
5  
6 calculator(8, 15);
```

Parameters

De twee getallen in de parameters (8,15) worden tijdelijk opgeslagen in de variabelen num en num2 in de parameters van de functie

Die geven het door aan de functie.

In de functie worden de twee variabelen opgeteld en opgeslagen in de variabele answer

Zonder variabele ziet het er zo uit →

Let op: onderaan is een voorbeeld!

```
1 function calculator(num, num2) {  
2     let answer = num + num2;  
3     console.log(answer);  
4 }  
5  
6 calculator(8, 15);
```

```
1 function calculator(8, 15) {  
2     let answer = 8 + 15;  
3     console.log(23)  
4 }  
5  
6 calculator(8, 15);
```



Demo



Oefeningen

In de map P1-Javascript maak je een nieuwe map aan genaamd "*opdracht_6.4*".

- Plaats in die map een index.html en een app.js
- In index.html koppel je de app.js.
- In app.js maak je oefening 1, 2, en 3



Oefening 1

Maak een optel-functie genaamd *calculate*.

- De functie kan twee random getallen optellen (parameters).
- De optelsom declareer je IN de functie.
- Met een `console.log()` kun je het antwoord zien.
- De random getallen bepaal jij wanneer je de functie aanroept.

Voorbeeld:

calculate(1,12);

antwoord: 13



Oefening 2

Deze oefening lijkt op oefening 1, maar niet helemaal!

Maak een functie genaamd *multiplyByFive*.

- De functie kan het random ingevoerde getal vermenigvuldigen met 5
- De som declareer je IN de functie.
- Met een `console.log()` kun je het antwoord zien.
- Het random getal bepaal jij wanneer je de functie aanroept.

Voorbeeld:

multiplyByFive(7);

antwoord: 35



Oefening 3

- In deze oefening ga je seconden omzetten in minuten.
- Maak een functie genaamd `calculateMinutes`. In de functie maak je een berekening dat ervoor zorgt dat de ingevoerde seconden omgezet worden in minuten.
- Seconden omzetten in minuten, hoe gaat dat ook alweer?

Voorbeeld:

`calculateMinutes(3600)`

antwoord: 60



Huiswerk

Opdracht 6.4 – functies met parameters (dat zijn de oefeningen uit de PowerPoint)

Uiterlijk inleveren: zie Canvas