

# Intro JS

## 1 Wat is JavaScript?

[datasource](#)

*In deze les leer je wat JavaScript is, waarom we het gebruiken en hoe je jouw eerste stukje code uitvoert in de browser.*

### ? Leerdoelen

- Je weet wat JavaScript is en waarvoor het wordt gebruikt
- Je kunt een script uitvoeren in de browserconsole
- Je kunt een script toevoegen aan een HTML-pagina met `<script>`

### ? Wat is JavaScript?

- **HTML** zorgt voor de inhoud (tekst, koppen, knoppen...)
- **CSS** zorgt voor de opmaak (kleuren, marges, lettertypes...)
- **JavaScript** zorgt voor de interactie (reageren op klikken, invullen, bewegen...)

Bijvoorbeeld, een **knop**

- **HTML**: een **kale knop** zonder opmaak
- **CSS**: dezelfde knop, maar nu **blauw** met **afgeronde hoeken**
- **JavaScript**: als je op de knop klikt, **verschijnt er een melding** ("popup")

### Wat kun je met JavaScript doen?

- Een knop laten reageren als je erop klikt
- Een rekenmachine bouwen
- Invoer van een formulier controleren

- Spelletjes maken in de browser

## ?Video

In deze video wordt goed uitgelegd waarom je als Web Developer, JavaScript moet leren.

Niet alles wat hier in de voorbeelden wordt getoond wordt niet allemaal in deze lessen behandeld. In [blok 5](#) wanneer we het over DOM gaan hebben zal dit allemaal uitvoerig worden behandeld. We houden het voor nu even bij de basics.

<https://www.youtube.com/watch?v=zofMnllkVfI>

<https://www.youtube.com/embed/zofMnllkVfI>

## ? Oefenen in de browserconsole

1. Open je browser (bijv. Chrome of Firefox)
2. Rechtermuisklik op een lege plek op een website > Kies **Inspecteren** > Ga naar het tabblad **Console**

3. Typ daar:

```
console.log("Hallo wereld!");
```

en druk op Enter

4. Je ziet in de console de tekst `Hallo wereld!`

## ? JavaScript in een HTML-bestand gebruiken

1. Open Phoenix Code
2. Maak een nieuw HTML-bestand aan, noem het `script.html`
3. Typ of plak deze code in het bestand:

```
<!DOCTYPE html>
<html>
<head>
  <title>Eerste script</title>
```

```
</head>
<body>

  <h1>Welkom op mijn site</h1>
  <p>Open de console om het bericht te zien.</p>

  <script>
    console.log("Dit bericht komt uit de HTML!");
  </script>

</body>
</html>
```

4. Sla op en open het bestand in de browser
5. Open opnieuw de **console** en zie het bericht verschijnen

Zie je meer berichten of fouten?  
Verwijder alle bestanden uit je project zodat je alleen script.html overhoud.

## ?? Opdracht – Jouw eerste script

1. Maak een HTML-bestand in Phoenix Code met een eigen titel en één paragraaf
2. Voeg onderaan het bestand een script toe met console.log()-regels
3. Laat in de console bijvoorbeeld jouw naam en je favoriete dier zien

### Voorbeeld:

```
console.log("Ik heet Yasmin");
console.log("Mijn lievelingsdier is een rode panda");
```

## ? Reflectie

- Wat is het verschil tussen wat je in de browser ziet en wat je in de console ziet?
- Waarom gebruiken programmeurs de console?

## ? Inleveren

1. Maak een screenshot van jouw HTML-bestand én de console-output.

## 2 Variabelen en de console

In deze les leer je hoe je informatie kunt opslaan in een variabele, hoe je dat zichtbaar maakt in de console, en waarom de console zo belangrijk is bij het programmeren.

### ? Leerdoelen

- Je begrijpt wat een variabele is
- Je kunt een variabele maken met `let` of `const`
- Je gebruikt `console.log()` om informatie weer te geven
- Je weet wat debuggen is en waarom de console daarbij helpt

### ? Wat is een variabele?

Een variabele is een soort doosje waarin je iets bewaart, zoals tekst of een getal.

```
let naam = "Jasper";  
const leeftijd = 17;
```

- `let` = variabele die later nog mag veranderen
- `const` = variabele die niet meer mag veranderen

Je gebruikt `console.log()` om de inhoud van een variabele te bekijken:

```
console.log(naam);
```

### ? Wat is debuggen?

**Debuggen** betekent: je code controleren op fouten (bugs) en begrijpen wat er gebeurt.

- De **console** is een soort gereedschapskist voor programmeurs.
- Je kunt er controleren of variabelen kloppen, of een functie wordt uitgevoerd, of ergens een fout zit.
- Een programmeur kijkt vaak in de console tijdens het schrijven van code.

Als je in de console iets ziet wat je niet verwacht – dan kun je makkelijker de fout vinden. Daarom gebruiken we `console.log()` vaak tijdens het debuggen.

# ?? Opdracht – Variabelen gebruiken en debuggen

1. Maak een nieuw HTML-bestand in Phoenix Code: `variabelen.html`
2. Typ de volgende basiscode in je bestand:

```
<!DOCTYPE html>
<html>
<head>
  <title>Variabelen oefenen</title>
</head>
<body>

  <h1>Bekijk de console!</h1>

  <script>
    let voornaam = "Ali";
    const leeftijd = 18;

    console.log("Naam: " + voornaam);
    console.log("Leeftijd: " + leeftijd);
  </script>

</body>
</html>
```

3. Open je bestand in de browser en bekijk de console (rechtermuisklik → Inspecteren → Console)
4. Verander de naam en leeftijd naar jouw eigen gegevens
5. Voeg een derde variabele toe: `favorietDier` en toon die ook met `console.log()`

## Voorbeeld:

```
let favorietDier = "koala";  
console.log("Mijn favoriete dier is: " + favorietDier);
```

## ? Reflectie

- Wat gebeurt er als je de waarde van `const` probeert te veranderen nadat je deze eerst een waarde hebt gegeven?
- Waarom is het handig om even iets te loggen in de console?
- Heb je fouten gezien? Wat hielp je om ze op te lossen?

## ? Inleveren

1. Beantwoord de vragen uit de reflectie en lever die in (.txt. of .pdf bestand).

# 3 Functies

*In deze les leer je wat een functie is, waarom functies handig zijn en hoe je zelf een functie maakt in JavaScript. Je oefent met functies die tekst tonen of berekeningen uitvoeren.*

## ? Leerdoelen

- Je weet wat een functie is en wat het doel ervan is
- Je kunt een functie schrijven met `function`
- Je kunt een functie aanroepen (laten uitvoeren)
- Je begrijpt wat een parameter is en hoe je die meegeeft

## ? Wat is een functie?

Een **functie** is een blokje code dat je een naam geeft, zodat je het later makkelijk opnieuw kunt gebruiken.

Stel, je wilt iets drie keer doen, zoals een bericht tonen. Dan is het dus handig om dat in een functie te stoppen.

```
function zegHallo() {  
  console.log("Hallo!");  
}
```

```
}
```

Om de functie uit te voeren (aan te roepen), typ je:

```
zegHallo();
```

## ? Functies met parameters

Je kunt een functie ook informatie meegeven. Bijvoorbeeld een naam:

```
function begroet(naam) {  
  console.log("Hoi " + naam + "!");  
}  
  
begroet("Fatima"); // toont: Hoi Fatima!  
begroet("Jesse"); // toont: Hoi Jesse!
```

## ?? Opdracht – Bouw je eigen functies

1. Maak een nieuw HTML-bestand in Phoenix Code: `functies.html`
2. Typ deze code als basis:

```
<!DOCTYPE html>  
<html>  
<head>  
  <title>Functies oefenen</title>  
</head>  
<body>  
  
  <h1>Bekijk de console</h1>  
  
  <script>  
  
    function begroet(naam) {  
      console.log("Hallo " + naam + "!");  
    }  
  
    begroet("Maya");  
    begroet("Koen");
```

```
</script>

</body>
</html>
```

3. Maak nog een functie die een getal verdubbelt:

```
function verdubbel(getal) {
  console.log(getal * 2);
}
```

4. Roep die functie minstens twee keer aan met verschillende getallen
5. Voeg daarna zelf een nieuwe functie toe met een eigen idee, bijvoorbeeld:
  - Een functie die een leeftijd in hondenjaren berekent (× 7)
  - Een functie die een bericht toont in jouw stijl.

## ? Reflectie

- Waarom is het handig om herhaalbare code in een functie te zetten?
- Wat gebeurt er als je een functie oproept zonder de juiste parameter?

## ? Inleveren

1. Maak een screenshot van je code én de console-output

## 4 Voorwaardes met `if` en `else`

In deze les leer je hoe je met JavaScript keuzes kunt maken. Je gebruikt `if` en `else` om bepaalde code alleen uit te voeren als aan een voorwaarde is voldaan.

## ? Leerdoelen

- Je begrijpt wat een voorwaarde is
- Je kunt werken met `if`, `else if` en `else`

- Je kunt eenvoudige beslissingen laten uitvoeren op basis van een getal

## ? Wat is een voorwaarde?

Met een voorwaarde bepaal je of een stukje code wél of niet uitgevoerd moet worden. Je gebruikt een `if`-statement:

```
let leeftijd = 17;

if (leeftijd >= 18) {
  console.log("Je bent volwassen");
} else {
  console.log("Je bent nog geen 18");
}
```

Je kunt meerdere keuzes maken met `else if`:

```
let punt = 6;

if (punt >= 8) {
  console.log("Super goed!");
} else if (punt >= 5.5) {
  console.log("Voldoende");
} else {
  console.log("Onvoldoende");
}
```

## ?? Opdracht – Leeftijdscontrole met voorwaarden

1. Maak een nieuw HTML-bestand in Phoenix Code: `ifelse.html`
2. Gebruik deze code als basis:

```
<!DOCTYPE html>
<html>
<head>
  <title>If-statement oefenen</title>
</head>
```

```
<body>

<h1>Bekijk de console</h1>

<script>

  let leeftijd = 15;

  if (leeftijd >= 18) {
    console.log("Je mag stemmen.");
  } else {
    console.log("Je bent nog te jong om te stemmen.");
  }

</script>

</body>
</html>
```

3. Pas de variabele `leeftijd` aan naar jouw eigen leeftijd en test het resultaat
4. Voeg een `else if`-blok toe dat controleert of iemand precies 17 is
5. Maak daarna zelf een voorbeeld met een `punt` (bijvoorbeeld een toetscijfer) en geef een boodschap: onvoldoende / voldoende / goed

## ? Inleveren

1. Maak een screenshot van je code én de console-output bij jouw leeftijd en punt
2. Lever dit screenshot in via Teams of Canvas

## ? Reflectie

- Wat gebeurt er als je geen `else` gebruikt?
- Wanneer gebruik je `else if` in plaats van meerdere `if`-regels?

## *5 Gebeurtenissen (events)*

In deze les leer je hoe je iets kunt laten gebeuren als een gebruiker op een knop klikt. Je leert werken met `onclick` waarmee je een **functie** kan aanroepen bij een gebeurtenis.

## ? Leerdoelen

- Je weet wat een event is in JavaScript
- Je kunt een knop laten reageren op een klik met `onclick`
- Je kunt een functie aanroepen wanneer een event plaatsvindt

## ? Wat is een event?

Een **event** is een gebeurtenis, zoals:

- een klik op een knop (`click`)
- iets typen in een tekstvak (`input`)
- de muis bewegen over een element (`mouseover`)

Met JavaScript kun je ervoor zorgen dat er iets gebeurt als zo'n event plaatsvindt. Het meest gebruikte event is `onclick`.

## Voorbeeld: klik op een knop

```
<button onclick="groet()">Klik hier</button>

<script>
  function groet() {
    console.log("Hoi! Je hebt op de knop geklikt.");
  }
</script>
```

Wordt je 'gek' van `console.log`? Je kunt ook `alert` gebruiken:

```
alert("Je hebt geklikt!");
```

## ?? Opdracht – Laat een knop iets doen

1. Maak een nieuw HTML-bestand in Phoenix Code: `knop.html`

2. Typ deze code als basis:

```
<!DOCTYPE html>
<html>
<head>
  <title>Gebeurtenis oefenen</title>
</head>
<body>

  <h1>Druk op de knop!</h1>

  <button onclick="toonBericht()">Klik hier</button>

  <script>
    function toonBericht() {
      console.log("Je hebt geklikt!");
    }
  </script>

</body>
</html>
```

3. Verander de tekst in de functie naar je eigen boodschap

4. Maak daarna nog een tweede knop met een andere functie, bijvoorbeeld:

```
toonLeeftijd()
```

5. Laat die tweede functie een leeftijd of getal tonen in de console (of via alert())

## Voorbeeld:

```
function toonLeeftijd() {
  console.log("Ik ben 16 jaar");
}
```

## ? Reflectie

- Wat is er nodig om een knop iets te laten doen?
- Wat gebeurt er als je een functie aanroept zonder dat die bestaat?

# ? Inleveren

1. Maak een screenshot van je HTML én console na het klikken op de knoppen

## *6 Formulierinvoer gebruiken met JavaScript*

### ? Leerdoelen

- Je weet hoe je met JavaScript input uit een formulier ophaalt met `getElementById()`.
- Je kunt een `if`-statement gebruiken op basis van invoer.
- Je toont het resultaat op het scherm, niet in de console.

### Start Code

```
<!DOCTYPE html>
<html>
<head>
  <title>Leeftijd check</title>
  <style>
    body { font-family: sans-serif; }
    #output { font-weight: bold; margin-top: 15px; }
  </style>
</head>
<body>

  <h1>Voer je leeftijd in</h1>

  <input type="number" id="leeftijdInput" placeholder="Typ je leeftijd">
  <button>Check mijn leeftijd</button>

  <p id="output"></p>

  <script>
    function checkLeeftijd() {
```

```
let leeftijd = Number(document.getElementById("leeftijdInput").value);
let boodschap = "";

if (leeftijd >= 18) {
  boodschap = "Je bent volwassen.";
} else {
  boodschap = "Je bent een kind.";
}

document.getElementById("output").innerText = boodschap;
}
</script>

</body>
</html>
```

## ? Uitleg

(lees goed door!)

Op regel **14** wordt een leeftijd gevraagd.

Op regel **15** staat een knop. Deze doet nog niets maar die moet via het `onclick` event gekoppeld worden aan de JS functie `checkLeeftijd()`.

Op regel **17** staat een lege paragraaf. Deze paragraaf wordt (later) gevuld door de JS functie.

Op regel **21** wordt de waarde van het input veld met het id `leeftijdInput` gevraagd. De waarde wordt in de JS variabele `leeftijd` bewaard.

Op regel **30** wordt `document.getElementById("id").value` gebruikt om de waarde uit een inputveld op te halen. Het HTML element waarvan de tekst wordt opgehaald heeft als id `output`.

## ?? Opdracht – Leeftijd analyseren en tonen

1. Zorg ervoor dat de knop wordt gekoppeld aan de JavaScript functie `checkLeeftijd()`.
2. Bereid de if-then-else structuur uit: als je ouder dan 65 bent dan ben je senior, ben je jonger dan 2 dan ben je een baby, jonger dan 6 een peuter, jonger dan 12 een jong kind, en jonger dan 18 een kind.
3. Test op een leeftijd  $<0$  en laat dan een boodschap zien dat de invoer niet geldig is.

## ? Reflectie

- Wat gebeurt er als je niets invult of een negatieve waarde gebruikt?
- Waarom is `Number(...)` nodig?

## ? Inleveren

- Lever de geteste en werkende code als html bestand met daarin de JS code in.

# 7 Bereken een prijs met korting

*In deze les gebruik je JavaScript om een prijs te berekenen na korting. Je gebruikt een formulier met invoervelden voor het bedrag en de korting in procenten. Daarna laat je in de console zien wat de nieuwe prijs is.*

## ? Leerdoelen

- Je kunt meerdere waarden ophalen uit een formulier
- Je gebruikt JavaScript om een berekening te maken met die waarden
- Je toont het resultaat in de console

## ? Formule: prijs met korting

Als je een korting in procenten hebt, dan gebruik je de volgende formule:

```
kortingsbedrag = bedrag * (korting / 100);  
nieuwePrijs = bedrag - kortingsbedrag;
```

## Code

```
<!DOCTYPE html>  
<html>  
<head>  
  <title>Korting berekenen</title>
```

```
</head>
<body>

  <h1>Bereken de korting</h1>

  <p>Vul een bedrag en een kortingspercentage in:</p>

  <input type="number" id="bedragInput" placeholder="Bedrag in euro's">
  <input type="number" id="kortingInput" placeholder="Korting in %">
  <button onclick="berekenKorting()">Bereken</button>

  <script>
    function berekenKorting() {
      let bedrag = Number(document.getElementById("bedragInput").value);
      let korting = Number(document.getElementById("kortingInput").value);

      let kortingBedrag = bedrag * (korting / 100);
      let nieuwePrijs = bedrag - kortingBedrag;

      console.log("Origineel bedrag: €" + bedrag.toFixed(2));
      console.log("Korting: " + korting + "%");
      console.log("Nieuwe prijs: €" + nieuwePrijs.toFixed(2));
    }
  </script>

</body>
</html>
```

## ?? Opdracht

1. Test het formulier met verschillende bedragen en percentages
2. Controleer of het resultaat klopt in de console
3. In plaats van console.log maak je net als bij de vorige opdracht een lege paragraaf aan en plaats daarin het resultaat.  
De drie regels (regel 24-26 van de code) wordt dus op de pagina getoond, net als bij de vorige pagina.
4. Maak CSS stijl aanpassingen naar eigen inzicht zodat de pagina er netjes uitziet.

# ? Reflectie

- Wat gebeurt er als je niks invult? Wat kun je daaraan doen?
- Wat gebeurt er als je een korting invult die hoger is dan 100%?

# ? Inleveren

1. De complete geteste en werkende HTML/JS/CSS in één HTML bestand.

--

---

Revision #13

Created 2025-06-04 17:22:28 UTC by Max

Updated 2025-10-29 12:50:03 UTC by Max