

Labo 13 – Verslag JavaScript | Deel 1

Opleidingsonderdeel: **Web Development 1**

Bachelor in de: **Toegepaste informatica**

Afstudeerrichting: **Applicatieontwikkeling**

Specialisatie: **Cybersecurity & infrastructure**

Leerling: **Jari Opsomer**

Docent: **Pim Debaere**

Academiejaar: **2025-2026**

2 JavaScript: Hoofdstuk 1 & 2

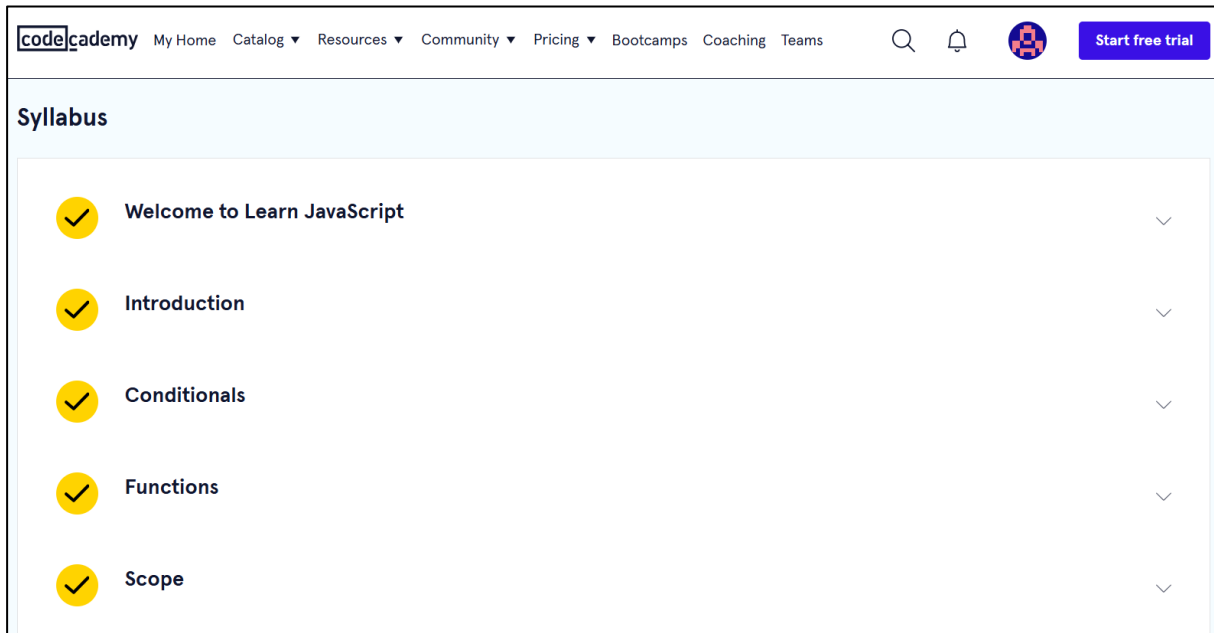
Hoofdstuk 1: Introductie

- **Evolutie:** Van statische HTML naar dynamische client-side scripting, gestandaardiseerd als ECMAScript.
- **Werking:** Toevoegen van functionaliteit via events, DOM-manipulatie en data-uitwisseling.
- **Integratie:** Koppeling via `<script>`-tags, bij voorkeur onderaan de `<body>` voor performantie.
- **Debugging:** Foutopsporing en logging via browser console.
- **Interactie:** Gebruik van blokkerende dialoogvensters:
 - `alert()`: Melding weergeven.
 - `confirm()`: Bevestiging vragen (Ok/Annuleren).
 - `prompt()`: Tekstinvoer vragen.

Hoofdstuk 2: Functies en properties

- **Puntnotatie:**
 - Opvragen eigenschappen (properties) via `.` (bv. `.length`).
 - Uitvoeren acties (functies) via `()` (bv. `.toUpperCase()`).
- **Arrow Notation:**
 - Moderne schrijfwijze ter vervanging van verouderde function-keyword.
 - Opslag in `const` variabelen.
 - Syntax: `const naam = (params) => { ... }`.
- **Verkorte syntax:**
 - Weglaten `{}` en `return` bij single-line functies.
 - Weglaten `()` bij exact één parameter.

2.1 Opdracht 1: Codecademy



The screenshot shows the Codecademy website's syllabus for the 'Learn JavaScript' course. The navigation bar at the top includes the Codecademy logo, 'My Home', 'Catalog', 'Resources', 'Community', 'Pricing', 'Bootcamps', 'Coaching', and 'Teams'. There are also search, notification, and user profile icons, along with a 'Start free trial' button. The syllabus section is titled 'Syllabus' and lists five completed topics, each with a yellow checkmark icon and a dropdown arrow:

- ✓ Welcome to Learn JavaScript
- ✓ Introduction
- ✓ Conditionals
- ✓ Functions
- ✓ Scope

2.2 Globale vs. lokale variabelen

Hoofdstuk 3: Variabelen

• 3.1 Soorten (Scope)

- **Lokale variabelen:** Enkel gekend binnen de functie van declaratie.
- **Globale variabelen:** Toegankelijk in het hele programma, inclusief gekoppelde scripts.
- **Risico:** Gedeelde scope over scripts heen leidt tot naamconflicten en ongewenst overschrijven.

• 3.2 Declaratie

- **Dynamische typing:** Geen expliciet type (zoals int/String) nodig bij declaratie.
- **const:** Block-scoped, constante waarde (niet opnieuw toewijsbaar).
- **let:** Block-scoped, veilig en intuïtief.
- **var:** Function-scoped of globaal, **strikt te vermijden.**

• Cruciale richtlijnen & Debugging

- **Regel:** Gebruik **altijd** let (of const), **nooit** var.
- **Veiligheid:** let voorkomt potentiële programmeerfouten, vergelijkbaar met Java/C#.
- **Chrome Developer Tools (Scope):**
 - Lokale variabelen → sectie 'Local'.
 - Globale variabelen → sectie 'Script' (let op: niet 'Global!').
- **Gevaar globale variabelen:**
 - Alle globale variabelen (uit alle .js files) belanden op één hoop.
 - Groot risico op naamconflicten en overschrijving door code van derden/libraries.
 - Leidt tot subtiele, moeilijk te debuggen fouten.
- **Conclusie:** Minimaliseer het gebruik van globale variabelen absoluut.

2.3 Opdracht 2: Leeg project (template)

Settings > File and Code Templates

HTML-template:

```
<!DOCTYPE html>
<html lang="nl">
<head>
  <meta charset="UTF-8">
  <link rel="stylesheet" href="styles/style.css">
  <title>Title</title>
</head>
<body>

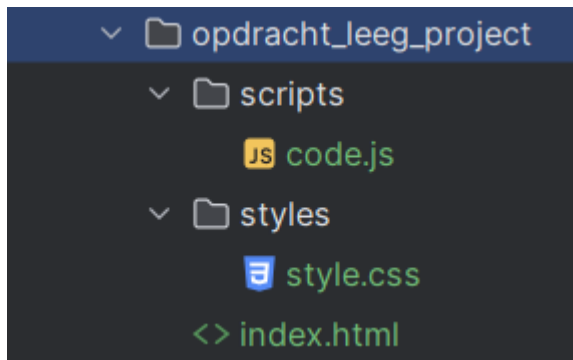
<script type="text/javascript" charset="utf-8"
src="scripts/code.js"></script>
</body>
</html>
```

JavaScript-template:

```
const setup = () => {
}
window.addEventListener("load", setup);
```

TIP: **ALT + ENTER** gebruiken om suggesties gemakkelijk te implementeren.

Overzicht van template-map:



2.4 Opdracht 3: Rekenmachine

HTML-code:

```
<!DOCTYPE html>
<html lang="nl">
<head>
  <title>Demo rekenmachine</title>
  <link rel="stylesheet" type="text/css" href="styles/index.css">
</head>

<body>
  <input type="number" id="txtLinks">
  <input type="button" id="btnOptellen" value="+">
  <input type="button" id="btnAftrekken" value="-">
  <input type="button" id="btnVermenigvuldigen" value="x">
  <input type="button" id="btnDelen" value=":">
  <input type="number" id="txtRechts">
  <span id="txtOutput">(geen output)</span>
  <script src="scripts/index.js"></script>
</body>
</html>
```

JavaScript-code:

```
const setup = () => {
  let btnOptellen=document.getElementById("btnOptellen");
  let btnAftrekken=document.getElementById("btnAftrekken");
  let btnVermenigvuldigen=document.getElementById("btnVermenigvuldigen");
  let btnDelen=document.getElementById("btnDelen");

  btnOptellen.addEventListener("click", optellen);
  btnAftrekken.addEventListener("click", aftrekken);
  btnVermenigvuldigen.addEventListener("click", vermenigvuldigen);
  btnDelen.addEventListener("click", delen);
}

const optellen = () => {
  let txtOutput=document.getElementById("txtOutput");
  let txtLinks=document.getElementById("txtLinks");
  let txtRechts=document.getElementById("txtRechts");

  let g1=parseInt(txtLinks.value, 10);
  let g2=parseInt(txtRechts.value, 10);
  let resultaat = g1+g2;

  let resultaatTekst=g1+" + "+g2+" = "+resultaat;
  txtOutput.innerHTML=resultaatTekst;
}
```

```

const aftrekken = () => {
  let txtOutput=document.getElementById("txtOutput");
  let txtLinks=document.getElementById("txtLinks");
  let txtRechts=document.getElementById("txtRechts");

  let g1=parseInt(txtLinks.value, 10);
  let g2=parseInt(txtRechts.value, 10);
  let resultaat = g1-g2;

  let resultaatTekst=g1+" - "+g2+" = "+resultaat;
  txtOutput.innerHTML=resultaatTekst;
}

const vermenigvuldigen = () => {
  let txtOutput=document.getElementById("txtOutput");
  let txtLinks=document.getElementById("txtLinks");
  let txtRechts=document.getElementById("txtRechts");

  let g1=parseInt(txtLinks.value, 10);
  let g2=parseInt(txtRechts.value, 10);
  let resultaat = g1*g2;

  let resultaatTekst=g1+" * "+g2+" = "+resultaat;
  txtOutput.innerHTML=resultaatTekst;
}

const delen = () => {
  let txtOutput=document.getElementById("txtOutput");
  let txtLinks=document.getElementById("txtLinks");
  let txtRechts=document.getElementById("txtRechts");

  let g1=parseInt(txtLinks.value, 10);
  let g2=parseInt(txtRechts.value, 10);
  let resultaat = g1/g2;

  let resultaatTekst=g1+" / "+g2+" = "+resultaat;
  txtOutput.innerHTML=resultaatTekst;
}

window.addEventListener('load',setup);

```

Resultaat:

5 * 4 = 20