

# Front End

- [Front End - 1 HTML](#)
- [Front End 2 - CSS](#)
- [Front End 4 - DOM Manipulation](#)
- [Flex Boxes](#)

# Front End - 1 HTML

python reformatterCanvas.py 18665 52539 <https://www.roc.ovh/books/2024/page/front-end-1-html>

## 01 - Intro + wat is HTML

Welkom bij de Frontend Module 1!

In deze module leer je hoe je je eerste webpagina maakt met behulp van HTML.

Laten we beginnen met een overzicht van wat HTML is en wat je daarvan zult leren in deze module.

sta je.html.jpg type unknown

## HTML

HTML staat voor **HyperText Markup Language** .

Het is de taal voor het maken van websites. Een website gemaakt in HTML bestaat uit bouwblokken die jouw internet browser vertellen hoe alles op de website eruit komt te zien.

Untitled.png type unknown

Gebruik eens de toets combinatie (CTRL + U) op de pagina van de opdracht, dan zie je precies hoe de structuur van de website in elkaar zit!

Untitled 1.png type unknown

Er staan vast dingen tussen waarvan je niet precies weet wat het doet, `<a>`, `<h1>`, `<div>`, `<span>` en wat zijn de teksten die tussen de `<>` tekens staan? Bijvoorbeeld

`class="dupe-option" role="presentation".` En wat doen überhaupt die “<>”tekens?? En hoe werkt dit allemaal samen????

Het doel van deze module is dat jij aan het eind voldoende basis kennis hebt van HTML en een redelijk idee hebt hoe bijvoorbeeld, de bovenstaande 2 voorbeelden werken! :)

Laten we beginnen met het maken van jouw eerste HTML pagina!

## Opdracht

- Ga naar de "Voortgangsoverzicht pagina" van je Canvas Monitor en zoek op de structuur van de websitepagina naar op welke regel je naam voorkomt.
- Wat staat er op de eerste en laatste regel?
- Hoeveel regels telt de code?
- Noteer op welke regel je naam staat en welke HTML-code er vermeld staat.
- Zoek uit wat deze code doet.

## Inleveren

- Screenshot van de code waar je naam staat.
- pdf bestand met antwoord op de vragen.

Voorbeeld:

[image.1720177943220.png](#)

## 02 - Jouw eerste HTML pagina

Oke, nu we een kleine introductie hebben gehad van HTML weten we dat het gebruikt wordt voor het maken van website.

Als het goed is hebben jullie Visual studio code geïnstalleerd. Zo niet, je kan het gemakkelijk downloaden in de windows store :)

[Untitled.png](#)

en kan dit gemakkelijk openen via de zoekbalk

Untitled 1.png  
image not found or type unknown

Maak een nieuwe map genaamd “Frontend1” en open deze map (“Folder”) in visual studio code

Untitled 2.png  
image not found or type unknown

Untitled 3.png  
image not found or type unknown

Nu je de map geopend hebt, kan je in deze map een nieuw HTML bestand maken met “new file”.  
Druk weer op “File” links boven en daarna “New File”. “New file” = Nieuw bestand

Untitled 4.png  
image not found or type unknown

Untitled 5.png  
image not found or type unknown

Je kan dit net gemaakte HTML bestand ook openen door in de map er op te klikken. Je ziet al een chrome icoontje of het icoontje van een andere browser.

Untitled 6.png  
image not found or type unknown

Laten we iets zetten in dit HTML bestand.

# Opdracht

Voeg het volgende toe aan je HTML bestand en herlaad de website

```
<h1> Dit is een titel! </h1>  
<p> en dit een ondertekst!</p>
```

## Inleveren

1. Screenshot van d gehele browser waarop deze pagina te zien is
2. het HTML bestand (mijnWebsite.html).

**WAARSCHUWING:** Aan het einde van deze module is er een eindopdracht die mondeling besproken wordt. Als blijkt dat je onvoldoende kennis hebt over de onderwerpen van deze module, gaan we ervan uit dat er gefraudeerd is (bijvoorbeeld door het gebruik van ChatGPT of het overnemen van werk van anderen). Zorg ervoor dat dit niet gebeurt!

# 03 - De basics

In dit hoofdstuk leer je de basisprincipes van HTML-elementen en tags, met voorbeelden en enkele best practices.

## Tag

Tags zijn de bouwstenen van HTML. Ze worden gebruikt om het begin en het einde van een HTML-element aan te geven. Tags zelf doen niets anders dan een structuur aangeven.

**Starttag** : *in het onderstaande voorbeeld de `<p>` ,*

**Sluittag** : *in het onderstaande voorbeeld de `</p>`*

```
<p> </p>
```

(`<p>` = alina, een manier om tekst te weergeven in een element)

## Element

Een element is hetzelfde als een tag maar het heeft inhoud! Elementen bestaan daarom dus ook uit een **starttag** , en meestal een **eindtag** . Maar ook **content** (De inhoud).

**Starttag** : *in het onderstaande voorbeeld de `<p>` ,*

**Content** : *in het onderstaande voorbeeld de tekst; "Dit is een alinia"*

**Sluittag** : *in het onderstaande voorbeeld de `</p>`*

```
<p>Dit is een alinea.</p>
```

(`<p>` = alina, een manier om tekst te weergeven in een element)

## Attribuut

Een attribuut is een extra stukje tekst dat je toevoegt aan een HTML-element om meer informatie toe te voegen.

Attributen worden gebruikt om eigenschappen van een element te veranderen en of toe te voegen. Bijvoorbeeld bij een link.

```
<a href="https://www.talnet.nl">Bezoek Example</a>
```

Het href attribuut wordt gebruikt om de locatie aan te geven waar de link naar toe verwijst.

# Basis HTML Elementen

Je hebt een heel veel elementen hier beneden staan een aantal voorbeelden.

**Koppen** : Kan je zien als titels, je hebt verschillende niveaus van koppen met `<h1>` tot `<h6>` .  
Bijvoorbeeld:

```
<h1>Hoofdtitel</h1>  
<h2>Subtitel</h2>
```

**Paragrafen** : Gebruik de `<p>` tag voor tekstblokken (alinia's). Bijvoorbeeld:

```
<p>Dit is een paragraaf tekst.</p>
```

**Links** : Gebruik de `<a>` tag voor links en het `href` label voor waar de link naar toe verwijst.  
Bijvoorbeeld:

```
<a href="https://www.example.com">Bezoek Example</a>
```

**Afbeeldingen** : Voeg afbeeldingen toe met de `<img>` tag en het `src` attribuut. Bijvoorbeeld:

```

```

De HTML `img` tag zoekt naar een afbeelding door te kijken in dezelfde map als het HTML-bestand waarin het staat. Als de afbeelding daar niet te vinden is, kan de browser deze niet laten zien.

[Lees dit artikel om een goede basis te krijgen over pathing](#) : dit is erg belangrijk ook naast deze module!

Zo heb je er nog een hoop die je gebruikt bij specifieke doeleinden.

## EXTRA: Commentaar

*In HTML kan je commentaar schrijven in de code, deze code wordt niet uitgevoerd en kan je gebruiken om code te beschrijven en of uit te leggen.*

```
<!-- Dit is een stukje commentaar -->
```

```
<!--
```

```
En
```

```
Dit
```

```
Is
```

```
Veel
```

```
Commentaar
```

```
-->
```

```
<!-- Deze tag is belangrijk! Niet verwijderen -->
```

```
<p> </p>
```

## Opdracht

- Maak een nieuw html bestand “opdracht4.html” en voeg daar een aantal html elementen aan toe en lever deze in.
  - een <h1> element
  - een <h2> element
  - een <p> tag
  - een <a> tag met de link naar <https://www.google.nl>
  - een <img> tag met een foto die je hebt gedownload

## Inleveren

1. Screenshot van d gehele browser waarop deze pagina te zien is

**WAARSCHUWING:** Aan het einde van deze module is er een eindopdracht die mondeling besproken wordt. Als blijkt dat je onvoldoende kennis hebt over de onderwerpen van deze module, gaan we ervan uit dat er gefraudeerd is (bijvoorbeeld door het gebruik van ChatGPT of het overnemen van werk van anderen). Zorg ervoor dat dit niet gebeurt!

## 04 - Semantic elements

Nu we de basics onder de knie hebben gaan we de pagina indelen! Het indelen wordt gedaan met **semantische elementen**. Semantische elementen in HTML zijn elementen die een duidelijke betekenis geven aan de inhoud die ze bevatten.

**Semantiek** : “de betekenis van een woord of een zin ”

<nav> = bevatten de navigatie links

structuur.png  
structure.png or type unknown

Zulke elementen maken de structuur van een webpagina begrijpelijker voor zowel de browser als de ontwikkelaars. Hier zijn wat belangrijke semantische elementen en hun betekenissen:

```
<header> </header>
```

Bevat inleidende inhoud of navigatie links voor een deel van de site of pagina.

```
<nav> </nav>
```

Geeft een deel van de pagina aan die navigatie links bevat.

```
<main> </main>
```

Bevat de belangrijkste inhoud van het document.

```
<article> </article>
```

Een element waar je informatie kan neerzetten dat niet.

```
<section> </section>
```

Geeft een groep van inhoud aan die bij elkaar komen, meestal met een kop.

```
<aside> </aside>
```

Bevat inhoud die hoort bij de hoofdinhoud, zoals zijbalken.

```
<footer> </footer>
```

Bevat informatie zoals auteursrechtinformatie of contactgegevens.



# Extra informatie

Semantische elementen worden vooral gebruikt bij het verbeteren van de [SEO](#) van een pagina. Naast dat wordt het ook gebruikt voor de overzichtelijkheid van de HTML code.

" *SEO staat voor Search Engine Optimization, oftewel het verbeteren van een website met als doel hogere posities te krijgen in de organische zoekresultaten van zoekmachines. Hoe hoger jouw website scoort op relevante zoekwoorden, hoe groter de kans dat mensen jouw website bezoeken. En omdat SEO geen kosten per klik met zich meebrengt, in tegenstelling tot SEA, is het een duurzame manier om meer website bezoekers te realiseren.* "

## Opdracht

- Bovenaan de opdracht staat een foto met een voorbeeld van hoe de semantische elementen gecombineerd kunnen worden. Beantwoord de volgende vragen daarover
  1. Behoort de <nav> tag in de <header> tag? Leg uit waarom wel of waarom niet.
  2. Welke andere semantische elementen en hun betekenissen kan jij vinden? (w3schools)
  3. Waar kan je informatie over een bedrijf het beste neerzetten in een website?

## Inleveren

1. Een ".txt" bestand in met de antwoorden op de vragen

## 05 - Layout

Nu we het hebben gehad over de basics van HTML en een aantal semantische elementen is het belangrijk om te bespreken hoe de layout van een gemiddeld HTML bestand eruit ziet.

Een HTML bestand heeft altijd een basisstructuur en bestaat uit het volgende (HTML skeleton)

```
<!DOCTYPE html>
<html>
  <head>
    <title>Mijn website! </title>
```

```
</head>
<body>
<h1>Welkom op mijn webpagina!</h1>
<p>Dit is een paragraaf tekst.</p>
</body>
</html>
```

skeleton.png or type unknown

**<!DOCTYPE html>:** Geeft aan dat het een HTML 5 document is, dit is belangrijk omdat browsers verschillende HTML-versies op verschillende manier kunnen weergeven. DOCTYPE html zorgt ervoor dat het altijd op dezelfde manier wordt weergegeven ongeacht de browser.

**<html>:** Dit element vertelt de browser dat alles binnen dit element HTML code is.

**<head>:** Dit element bevat informatie over je html document, zoals titels, links naar stylesheets ( CSS ) en scripts

**<body>:** Bevat de zichtbare inhoud van de webpagina. Teksts, foto's, etc

# Containers

Wat ook belangrijk is voor de layout van html bestanden zijn containers. Containers kan je zien als dozen waar dingen in zitten zoals HTML elementen die bij elkaar horen. Bijvoorbeeld een aantal foto's. Een voorbeeld van zo'n "doos" is een <div> container.

Een **<div>** is een html tag dat gebruikt wordt om groter HTML elementen te groeperen. Zoals een aantal foto's.

h5-group-container.png or type unknown

Deze containers met foto's kunnen ook gewoon weer in een container gestopt worden!

container.png or type unknown

Voor kleinere elementen zoals stukken tekst kan je een **<span>** container gebruiken. In het volgende hoofdstuk vertellen we wat het verschil is.

Een div wordt ook wel een **"block-container"** genoemd omdat het gebruikt wordt om grote html elementen te bevatten waarbij een <span> een **"inline-container"** genoemd wordt omdat het

kleine html elementen bevat.

Je hebt meerdere block containers, kan je al raden welke?... De semantische elementen! Dit zijn ook containers wat HTML elementen die bij elkaar horen groeperen. Denk aan **<article>**, **<footer>**, **<nav>** etc. Dit zijn allemaal containers.

Voorbeeld:

```
<nav>
  <a href="home">Home</a>
  <a href="about">Over ons</a>
  <a href="contact">Contact</a>
</nav>
```

## Opdracht

- Maak een nieuw HTML bestand “opdracht6.html” en maak een website waar je een nieuwsartikel kan lezen.
- Om dit te doen, maak een HTML skeleton en gebruik de volgende containers
  - <header>
  - <nav>
  - <article>
  - <footer>

## Inleveren

1. Screenshot van d gehele browser waarop deze pagina te zien is
2. het HTML bestand.

**WAARSCHUWING:** Aan het einde van deze module is er een eindopdracht die mondeling besproken wordt. Als blijkt dat je onvoldoende kennis hebt over de onderwerpen van deze module, gaan we ervan uit dat er gefraudeerd is (bijvoorbeeld door het gebruik van ChatGPT of het overnemen van werk van anderen). Zorg ervoor dat dit niet gebeurt!

# 06 - Formatting

In het vorige hoofdstuk hebben we het kort gehad over inline-containers, containers die gebruikt worden voor kleine elementen voornamelijk tekst. In deze opdracht gaan we hier wat dieper op in.

## Inline Containers

Inline containers zijn HTML-elementen die worden gebruikt voor het stylen van tekst of kleine stukjes inhoud zonder een nieuwe regel te beginnen, wat block containers wel doen (*bijvoorbeeld: <div>*). Ze zijn handig wanneer je specifieke styling op delen van een tekst wilt doen, zoals het dik maken van een woord.

**Hieronder een aantal veel gebruikte tekst formatting tags:**

```
<span> </span>
```

`<span>` wordt gebruikt om een klein deel van de tekst of een groep inline elementen aan te geven. Het voordeel van een span vergeleken een `<div>` is dat een `<span>` geen nieuwe regel forceert op de tekst. (*probeer dit uit als je nieuwsgierig bent*)

```
<a> </a>
```

`<a>` wordt gebruikt om links te maken naar andere paginas of websites. Het kan tekst, afbeeldingen of andere tags bevatten. Het zorgt ervoor dat je erop kan klikken

```
<strong> </strong>
```

`<strong>` geeft aan dat de tekst extra belangrijk is en maakt deze vetgedrukt.

```
<em> </em>
```

`<em>` geeft aan dat de tekst extra nadruk heeft en maakt deze cursief.

```
<b> </b>
```

`<b>` maakt de tekst vetgedrukt.

```
<i> </i>
```

`<i>` maakt de tekst cursiefs.

```
<small> </small>
```

`<small>` maakt de tekst kleiner dan de rest van de tekst.

```
<mark> </mark>
```

`<mark>` markeert tekst met een achtergrondkleur om aan te geven dat deze belangrijk is. Zoals met een gele stift bij het studeren

## Opdracht

- Gebruik het HTML bestand van opdracht 6 en voeg wat tekst formatting toe gebruik de volgende elementen
  - `<i>`
  - `<strong>`
  - `<b>`
  - `<small>`
  - `<mark>`

## Inleveren

1. Screenshot van d gehele browser waarop deze pagina te zien is
2. het HTML bestand.

**WAARSCHUWING:** Aan het einde van deze module is er een eindopdracht die mondeling besproken wordt. Als blijkt dat je onvoldoende kennis hebt over de onderwerpen van deze module, gaan we ervan uit dat er gefraudeerd is (bijvoorbeeld door het gebruik van ChatGPT of het overnemen van werk van anderen). Zorg ervoor dat dit niet gebeurt!

## 07 - Lists

Lists zijn containers die je gebruikt om items die verband hebben met elkaar te weergeven. HTML heeft verschillende soorten lijsten ieder met zijn eigen doel en opmaak.

# Unorderd lists

Unorderd lists worden gebruikt om items weer te geven die geen specifieke volgorde hebben. Ze worden weergegeven met bullet points.

**Starttag** : in het onderstaande voorbeeld de `<ul>` (Unordered List)

**List-items** : in het onderstaande voorbeeld de `<li>` (List Items)

```
<ul>
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
</ul>
```

# Ordered lists

Ordered lists worden gebruikt om items weer te geven die in een specifieke volgorde moeten staan. Ze worden weergegeven met nummers.

**Starttag** : in het onderstaande voorbeeld de `<ol>` (Ordered List)

**List-items** : in het onderstaande voorbeeld de `<li>` (List Items)

```
<ol>
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
</ol>
```

# Opdracht

- Maak een nieuw HTML bestand "opdracht7.html" en maak een recept
  - Maak de instructies zodat ze in een lijst met een volgorde staan
  - Maak de ingrediënten zodat ze in lists staan zonder volgorde

# Inleveren

1. Screenshot van d gehele browser waarop deze pagina te zien is
2. het HTML bestand.

WAARSCHUWING: Aan het einde van deze module is er een eindopdracht die mondeling besproken wordt. Als blijkt dat je onvoldoende kennis hebt over de onderwerpen van deze module, gaan we ervan uit dat er gefraudeerd is (bijvoorbeeld door het gebruik van ChatGPT of het overnemen van werk van anderen). Zorg ervoor dat dit niet gebeurt!

## 08 - Tables

Tabellen worden gebruikt om gegevens in een grid vorm te presenteren, zoals bijvoorbeeld in excel. HTML heeft specifieke tags om tabellen te maken.

### Standaard tabel

Een standaard tabel bestaat uit een `<table>` element met rijen ( `<tr>` ) en cellen ( `<td>` ).

**Starttag** : in het onderstaande voorbeeld de `<table>`

**Rijen** : in het onderstaande voorbeeld de `<tr>` (Table Row)

**Cellen** : in het onderstaande voorbeeld de `<td>` (Table Data)

**“border=1”**: zorgt ervoor dat je de tabel goed kan zien, anders zie je niet de onderstaande zwarte lijnen

### Table Header

De tableheader wordt gebruikt om kolom koppen te maken. Het wordt gemaakt met de `<th>` tag.

**Starttag** : in het onderstaande voorbeeld de `<th>` (Table Header)

```

<table border=1>
  <tr>
    <th>kolom 1</th>
    <th>kolom 2</th>
  </tr>
  <tr>
    <td>Cel 1</td>
    <td>Cel 2</td>
  </tr>
  <tr>
    <td>Cel 3</td>
    <td>Cel 4</td>
  </tr>
</table>

```

table 3.png  
 background or type unknown

## Tussenopdracht

- **Maak een tabel naar keuze, deze tabel gaan we hierna uitbreiden.**

## Tabellen Met Header, Body en Footer

Nu je een tabel hebt gaan we deze uitbreiden. Een uitgebreide tabel kan een kop ( `<thead>` ), een lichaam ( `<tbody>` ) en een voettekst ( `<tfoot>` ) hebben.

Dit helpt grote tabellen overzichtelijk te maken. Net zoals een HTML pagina, je kan het dus vergelijken met de semantische HTML elementen in een pagina. `<header>` , `<body>` , `<footer>`

```

<table border=1>
  <thead>
    <tr>
      <th>Kop 1</th>
      <th>Kop 2</th>
    </tr>
  </thead>

```



table 1.png  
 language not found or type unknown

Met de attributen `colspan` en `rowspan` kun je cellen maken die uit meerdere columns of rows bestaan.

```
<table border=1>
  <tr>
    <th>Kop 1</th>
    <th>Kop 2</th>
    <th>Kop 3</th>
  </tr>
  <tr>
    <td>Cel 1</td>
    <td colspan="2">Cel 2 en 3</td>
  </tr>
```

```
<tr>
  <td rowspan="2">Cel 4 en 5</td>
  <td>Cel 6</td>
  <td>Cel 7</td>
</tr>
<tr>
  <td>Cel 8</td>
  <td>Cel 9</td>
</tr>
</table>
```

## Opdracht

- Maak een nieuw HTML bestand "opdracht9.html" en voeg daar een tabel aan toe met de volgende eisen:
  - Een table header met twee columns.
  - Drie rows met data.
  - Een voettekst voor de tabel.
  - Een cel die twee columns beslaat.
  - Een cel die twee rows beslaat.

## Inleveren

1. Screenshot van d gehele browser waarop deze pagina te zien is
2. het HTML bestand.

**WAARSCHUWING:** Aan het einde van deze module is er een eindopdracht die mondeling besproken wordt. Als blijkt dat je onvoldoende kennis hebt over de onderwerpen van deze module, gaan we ervan uit dat er gefraudeerd is (bijvoorbeeld door het gebruik van ChatGPT of het overnemen van werk van anderen). Zorg ervoor dat dit niet gebeurt!

## 09 - Forms

Forms zijn een manier om gegevens van de gebruiker te verzamelen en naar een andere plek te sturen. HTML heeft verschillende elementen om een formulier te bouwen.

# Standaard Formulier

Een formulier bestaat uit een `<form>` element met verschillende invoerelementen, [Met deze link](#) kan je ze allemaal bekijken het zijn er 11 maar wij bespreken er maar een aantal hiervan.

**Starttag** : *in het onderstaande voorbeeld de* `<form>`

**Invoerelementen** : *in het onderstaande voorbeeld de* `<input>`

```
<form action="submit.html" method="post">
  <input type="text" name="naam" placeholder="Voer je naam in">
  <input type="submit" value="Verzenden">
</form>
```

 found or type unknown

Het `action` attribuut verteld de form waar de data naar toe moet worden gestuurd wanneer er op de `submit` input word gedrukt. De `method` beschrijft hoe de data wordt verstuurd **(in deze module is dat niet belangrijk)**

## Input types

Het input element wordt het meest gebruikt in forms en hebben verschillende types. Dit kan je aangeven met het `type=` attribuut in de input. De `name` attribuut geeft de naam van de variabele aan waar de data in komt te staan van het desbetreffende element in de form.

```
<!-- tekst -->
<input type="text" name="naam" placeholder="Voer je naam in">

<!-- wachtwoord -->
<input type="password" name="wachtwoord" placeholder="Voer je wachtwoord in">

<!-- email-->
<input type="email" name="email" placeholder="Voer je e-mail in">
```

```
<!--radio button-->
<input type="radio" name="gender" value="man"> Man

<!-- checkbox -->
<input type="checkbox" name="hobby" value="lezen"> Lezen

<!-- Form submit(verstuur) button -->
<input type="submit" value="Verzenden">

<!-- Form reset button -->
<input type="reset" value="Reset">
```

input fields.png or type unknown

Zoals je misschien al opvalt hebben deze input elements geen sluit tag, dit hoeft inderdaad niet.

## Tussenopdracht

- ***Maak eens een eigen formulier! Gebruik de bovenstaande voorbeelden, deze gaan we later in de opdracht verder gebruiken***

## Andere invoerelementen

Naast `<input>` zijn er nog een aantal elementen die gebruikt worden in een form om data te verzamelen hier nog 2 voorbeelden.

### Select

Wordt gebruikt om een enkele optie te selecteren uit een dropdown menu.

```
<select name="land">
  <option value="nederland">Nederland</option>
  <option value="belgie">België</option>
  <option value="duitsland">Duitsland</option>
</select>
```

select.png or type unknown

## Textarea

Wordt gebruikt om grotere hoeveelheden tekst in te voeren.

```
<textarea name="opmerkingen" placeholder="Voer je opmerkingen in"></textarea>
```

textarea.png  
text area tag and or type unknown

# Opdracht

- Maak een nieuw HTML bestand "opdracht10.html" en voeg daar een formulier toe waarmee je informatie ophaalt van een persoon. We willen de volgende informatie verzamelen
  - Voornaam
  - Achternaam
  - Email
  - Wachtwoord
  - Welke klas (dropdown menu)
  - Feedback voor de module (Grote hoeveelheid tekst)
  - Een checkbox of we de gegevens mogen gebruiken

## Inleveren

1. Screenshot van d gehele browser waarop deze pagina te zien is
2. het HTML bestand.

**WAARSCHUWING:** Aan het einde van deze module is er een eindopdracht die mondeling besproken wordt. Als blijkt dat je onvoldoende kennis hebt over de onderwerpen van deze module, gaan we ervan uit dat er gefraudeerd is (bijvoorbeeld door het gebruik van ChatGPT of het overnemen van werk van anderen). Zorg ervoor dat dit niet gebeurt!

# 10 - Eindopdracht

Het doel van deze eindopdracht is om al je kennis van HTML toe te passen door een volledige webpagina te maken die gebruikmaakt van verschillende HTML-elementen

en technieken die je hebt geleerd. Deze opdracht omvat elementen uit alle hoofdstukken van de module. **Deze wordt uiteindelijk mondeling besproken.**

## Het gaat niet om het uiterlijk van de website maar het gaat om de code!

**Je gaat een blog maken. Blog →** “Een blog is een informatief of inspirerend artikel dat online staat, regelmatig verschijnt, een persoonlijk haakje heeft met een informele tone-of-voice en waarop lezers kunnen reageren.”

### de blog bevat:

1. Een blogpost met onderwerp naar keuze en bevat een table (*Uitgebreide table met header body en footer*) en een list met data
2. De blogpost heeft minsten 4 vormen tekst formatting
5. De blogpost heeft minstens 2 afbeelding(en)
6. De blogpost heeft een logische layout/ structuur (alles is goed gegroepeerd met containers <div> <span> en logische semantische elementen)
7. Review formulier over de blog

***Naast de eindopdracht worden er ook theorie vragen gesteld over de module dus verwerk zoveel mogelijk in je eindopdracht en zet er commentaar bij des te minder je hoeft te onthouden!***

### Tips:

- gebruik steekwoorden als commentaar in je html! Dit helpt bij de mondeling.
- Gebruik w3schools voor extra hulp met HTML elementen die niet besproken zijn in deze module. **LET OP, je moet deze wel kunnen uitleggen**
- Het gaat niet om de uiterlijk van de website dit leer je later in het schooljaar, alleen het uiterlijk van je code!

## Inleveren

1. Een schermafdruck van je hele browser waarop je blog te zien is.

## 2. De HTML code

# Front End 2 - CSS

## 01 - Introductie

### Wat is CSS?

CSS is een taal die wordt gebruikt om de stijl van een website te beschrijven. Met CSS kun je bepalen hoe HTML-elementen eruitzien, zoals de kleuren, lettertypen, marges en afstanden.

Dit helpt om je website mooier, aantrekkelijker en gebruiksvriendelijker te maken.

### HTML en CSS

**HTML** wordt gebruikt voor de **structuur** van een website.

**CSS** wordt gebruikt voor de **opmaak** en lay-out.

### Consistent

Als je CSS op de juiste manier gebruikt dan hoef je maar één keer een stijl aan te maken. Deze wordt dat netjes op de hele HTML toegepast. Op die manier zien bijvoorbeeld al je <h1> headers er hetzelfde uit. Dit heet **consistent**. Je website heeft een consistente vormgeving als de vormgeving uniform is.

CSS maakt het dus eenvoudig om een consistente stijl te behouden over meerdere pagina's.

### Responsive

Als je CSS goed toepast kan je er ook voor zorgen dat je website er op je laptop én op je telefoon goed uit ziet. Dit heet **responsive**. Met CSS kun je je website dus aanpassen aan verschillende schermformaten, zoals smartphones, tablets en desktopcomputers.

### Voorbeeld

Kijk nu naar het voorbeeld (EN) van hoe CSS wordt gebruikt.

<https://www.youtube.com/embed/0afZj1G0BIE?si=vhBdGRyxNvAzKfvn>



**WAARSCHUWING:** Aan het einde van deze module is er een eindopdracht die mondeling besproken wordt. Als blijkt dat je onvoldoende kennis hebt over de onderwerpen van deze module, gaan we ervan uit dat er gefraudeerd is (bijvoorbeeld door het gebruik van ChatGPT of het overnemen van werk van anderen).  
Zorg ervoor dat dit niet gebeurt!

# Opdracht

Maak je eigen webpagina mooi met CSS

**1:** Maak een nieuw HTML-bestand met visual studio code

bestandsnaam: **01-CSS-je\_eigen\_naam.html** en voeg de volgende code toe :

```
<!DOCTYPE html>
<html>
<head>
<title>Opdracht 1 - CSS Introductie</title>
<style>
p {
    font-family: Cambria;
    text-align: center;
}
h1 {
    font-size: 30px;
    color: rgb(218, 186, 235);
    text-align: center;
}
body {
    font-size: normal;
}
.intro {
    color: blueviolet;
}
</style>
</head>
<body>
    <h1 id="titel">Welkom op de website van .....!</h1>
```

```
<p class="intro">Dit is een introductie over mijzelf.</p>
<p class="intro">Ik woon in ..... en kom met de ..... naar school</p>
<p class="intro">Ik hou van programmeren en het maken van websites.</p>
</body>
</html>
```

Herken je de class in de paragraaf `<p class=....>` ? Nee, kijk dan nog eens naar de video of zoek op wat class precies inhoud!

**2:** Vul de puntjes in.

**3:** Zoek uit met welke CSS code de achtergrond van je website een andere kleur krijgt. Geef je website je favoriete kleur door in de CSS van de body een regel toe te voegen onder de regel `font-size: normal;`

**4:** Open het HTML-bestand in een webbrowser en zie hoe de CSS de stijl van je webpagina verandert.

Experimenteer met verschillende CSS eigenschappen en waarden om de opmaak verder aan te passen.

Bekijk en probeer de [Try it Yourself >>](#) op: <https://www.w3schools.com/css/default.asp>

### Inleveren:

1. HTML bestand
2. Screenshot  
(van je browser waarin je webpagina hebt geopend, zorg dat de URL van je pagina zichtbaar is.)

#### Voorbeeld:

image-1717574533140.png

## 02 - Basisconcepten van CSS

CSS staat voor **C**ascading **S**tyle **S**heet en wordt gebruikt om een website vorm te geven. Als je met HTML de structuur bouwt dan is CSS de opmaak. We kunnen nu dus ook wat meer creativiteit laten zien.

### Selectors

Een CSS-selector is een patroon dat wordt gebruikt om de HTML-elementen te selecteren die je wilt stylen. Hier zijn enkele veelvoorkomende selectors:

- **Element** selector: selecteert alle elementen van een bepaald type.

```
p {  
  color: blue;  
}
```

Dit verandert de tekstkleur van alle <p>-elementen naar blauw.

- **Class** selector: selecteert elementen met een specifieke class.

```
.intro {  
  font-size: 20px;  
}
```

Dit stelt de lettergrootte in van alle elementen met de class "intro" op 20 pixels.

- **ID** selector: selecteert het element met een specifiek ID.

```
#header {  
  background-color: yellow;  
}
```

Dit verandert de achtergrondkleur van het element met het ID "header" naar geel.

## Eigenschappen en waarden

CSS gebruikt eigenschappen en waarden om de stijl te definiëren. Enkele veelvoorkomende eigenschappen zijn:

**color:** de kleur van de tekst.

**background-color:** de achtergrondkleur van een element.

**font-size:** de grootte van de tekst.

**margin:** de ruimte buiten een element.

**padding:** de ruimte binnen een element.

Bekijk en probeer de [Try it Yourself >>](https://www.w3schools.com/css/css_selectors.asp) op: [https://www.w3schools.com/css/css\\_selectors.asp](https://www.w3schools.com/css/css_selectors.asp)

# Opdracht

1. Maak een nieuw HTML-bestand met visual studio code en voeg de onderstaande code bij 4. toe

bestandsnaam: **02-CSS-je\_eigen\_naam.html**

2. Vul in de puntjes de ontbrekende gegevens.

3. Bekijk je werk.

Open het HTML-bestand in een webbrowser en zie hoe de CSS de stijl van je webpagina verandert.

Experimenteer met verschillende CSS-eigenschappen en waarden om de opmaak verder aan te passen naar wat je zelf mooi vindt.

4. Code

```
<!DOCTYPE html>
<html>
<head>
<title>Opdracht2 - CSS - ...</title>
<style>
body {
    background-color: ....;
}
#header {
    background-color: ....;
}
h1 {
    font-family: ....;
    font-size: ...;
    color: ...;
    text-align: ...;
}
h2 {
    font-family: ...;
    font-size: ...;
    color: ...;
    text-align: ...;
}
.intro {
    font-size: ...;
```

```

    color: rgb(...);
    text-align: ...;
}
.alinea {
    font-size: ...;
    color: ...;
    text-align: ...;
}
p {
    font-family: verdana;
    color:blueviolet;
    text-align: left;
}
</style>

</head>
<body>
    <div id="header">
        <h1 id="titel">Welkom op de website van .....!</h1>
    </div>
    <h2>Dit is een introductie over mijzelf.</h2>
    <p class="intro">Ik woon in .... en kom met de .... naar school.</p>
    <p class="alinea">Ik hou van .... en .....
        In mijn vrije tijd .... en ....<br>
        Als sport ....<br>
        Ik probeer iedere dag .....
    </p>
</body>
</html>

```

## Inleveren:

1. HTML bestand
2. Screenshot  
van je browser waarin je webpagina hebt geopend, zorg dat de URL van je pagina zichtbaar is!

Voorbeeld:

## 03 - CSS Linken aan bestand

Je merkt al dat je code stuk langer gaat worden. Het is het netste om jouw CSS-code apart in een speciaal ".css" bestand te schrijven. Hierdoor wordt de html overzichtelijker en kan je css voor meerdere HTML bestanden gebruikt worden.

Er zijn ook nadelen waar je rekening mee moet houden. Voor een beginner is het lastig om met meerdere bestanden rekening te houden en de juiste naamgeving te gebruiken. Bij inline of interne CSS zie je direct de stijlen direct de stijlen op de HTML van toepassing zijn.

Als je met grote projecten met meerdere pagina's gaat werken is het overzichtelijker, onderhoudsvriendelijker en efficiënter om CSS code te linken aan een bestand.

```
<link rel="stylesheet" type="text/css" href="03-CSS-Yildiz-Style.css" />
```

Zorg dat je als het href attribuut de naam van jouw css bestand staat, dit is **hoofdletter gevoelig!!!**

## Opdracht

1. Kopieer het html bestand van de vorige opdracht en geef het de, bestandsnaam: **03-CSS-je\_eigen\_naam.html**
2. Maak in de zelfde map een nieuw bestand met de naam en extensie "**03-CSS-je\_eigen\_naam-Style.css**".
3. Knip uit je html bestand het gedeelte wat tussen de <style> tags staat en plak het in je css bestand.
4. Voeg aan je html bestand ergens tussen de <head> tags de link naar je css bestand toe. Dit doen we door de volgende regel van het HTML-bestand te zetten:

Laat je werk aan je vakdocent zien.

### Inleveren 3 bestanden

- HTML bestand
- CSS bestand

- Screenshot HTML code  
Zorg dat de URL van je pagina zichtbaar is!

Voorbeeld:

image-1717586844812.png

# 04 - Kleuren in CSS

## Wat zijn Kleuren in CSS?

Kleuren zijn een belangrijk onderdeel van webdesign. Ze helpen om de aandacht van gebruikers te trekken, emoties over te brengen, en de leesbaarheid van tekst te verbeteren. In CSS kun je kleuren op verschillende manieren specificeren: met namen, hexadecimale waarden, RGB, RGBA, HSL, en HSLA.

## Kleuren Specificeren in CSS

### 1. Kleurnamen

CSS ondersteunt een lijst van 147 standaardkleurnamen zoals `red`, `blue`, `green`, enz.

```
p {  
  color: red;  
}
```

### 2. Hexadecimale Kleuren

Hexadecimale kleuren gebruiken een combinatie van zes letters en cijfers om een kleur te definiëren, voorafgegaan door een `#`. Het formaat is `#RRGGBB`, waarbij `RR`, `GG`, en `BB` de rode, groene, en blauwe componenten van de kleur zijn.

```
p {  
  color: #ff5733; /* Een tint oranje */  
}
```

### 3. RGB Kleuren

RGB staat voor rood, groen en blauw. Kleuren worden gedefinieerd door de intensiteit van deze drie kleuren op een schaal van 0 tot 255.

```
p {  
  color: rgb(255, 87, 51); /* Dezelfde tint oranje als hierboven */  
}
```

#### 4. RGBA Kleuren

RGBA is hetzelfde als RGB, maar met een extra parameter voor de alfa (transparantie) waarde, die gaat van 0 (volledig transparant) tot 1 (volledig ondoorzichtig).

```
p {  
  color: rgba(255, 87, 51, 0.5); /* Dezelfde tint oranje met 50% transparantie */  
}
```

#### 5. HSL Kleuren

HSL staat voor tint (hue), verzadiging (saturation) en lichtheid (lightness). De tint gaat van 0 tot 360 (kleurenspectrum), verzadiging en lichtheid gaan van 0% tot 100%.

```
p {  
  color: hsl(14, 100%, 60%); /* Dezelfde tint oranje */  
}
```

#### 6. HSLA Kleuren

HSLA is hetzelfde als HSL, maar met een extra parameter voor alfa (transparantie).

```
p {  
  color: hsla(14, 100%, 60%, 0.5); /* Dezelfde tint oranje met 50% transparantie */  
}
```

## Opdracht

**1.** Maak een nieuw HTML-bestand met de naam **04-CSS-je\_eigen\_naam.html**

**2.** Voeg de volgende code met inline css toe ( vul de puntjes in )

**3.** Wijzig de kleuren naar persoonlijke smaak.

Open het HTML-bestand in een webbrowser en zie hoe de CSS-kleuren de uitstraling van je webpagina veranderen. Experimenteer met verschillende kleurmethoden (kleurnamen, hexadecimaal, RGB, RGBA, HSL, HSLA) en waarden om de opmaak verder aan te passen.

```
<!DOCTYPE html>  
  
<html>  
  
<head>
```



```
<title>Opdracht4 - CSS kleuren - ....</title>
<style>
body {
  background-color: #....; /* AliceBlue achtergrondkleur */
  font-family: Arial, sans-serif;
}
h1 {
  color: hsl(240, ...%, ...%); /* Helder blauw */
  text-align: center;
}
.intro {
  color: rgb(..., ..., 0); /* DarkOrange tekstkleur */
  font-size: ..px;
  text-align: center;
}
.box {
  background-color: rgba(75, 0, 130, 0.5); /* Indigo met 50% doorsichtigheid */
  color: #.....; /* Wit */
  padding: 20px;
  margin: 20px auto;
  text-align: center;
  width: 50%;
}
.outro {
  color: #ff69b4; /* HotPink tekstkleur */
  text-align: ....;
}
</style>
</head>
<body>
  <h1>Welkom op mijn gekleurde website van .....!</h1>
  <br>
  <hr>
  <p class="intro">CSS kleuren zijn heel leuk om mee te werken.</p>
  <div class="box">Dit is een gekleurde box.</div>
  <p class="outro">Bedankt voor je bezoek en tot ziens!</p>
  <hr>
</body>
</html>
```

## Inleveren:

1. HTML bestand
2. Screenshot browser  
Zorg dat de URL van je pagina zichtbaar is!

## Voorbeeld:

image 1717588788333.png

# 05 - Box Model en Layout

**Het Box Model:** Het CSS box model beschrijft de opbouw van elementen op een webpagina. Het bestaat uit:

- **Content:** De inhoud van de box, waar tekst en afbeeldingen worden weergegeven.
- **Padding:** Ruimte binnen de box, tussen de content en de border.
- **Border:** De rand om de padding en content heen.
- **Margin:** Ruimte buiten de border, tussen de box en andere elementen.

```
<!DOCTYPE html>
<html>
<head>
  <title>Opdracht 5 - CSS Box Model - .....</title>
  <style>
    .box {
      width: 200px;
      padding: 20px;
      border: 5px solid blue;
      margin: 10px;
    }
  </style>
</head>
<body>
  <h1 id="titel">Welkom op de website van .....!</h1>
  <div class="box">
```

Dit is een voorbeeld van het CSS box model.

```
</div>  
</body>  
</html>
```

## Opdracht:

1. Maak een nieuw HTML-bestand met de naam 05-CSS-je\_eigen\_naam.html
2. Voeg de bovenstaande code toe en experimenteer met de waarden voor padding, border en margin.
3. Maak onderstaande voorbeeld (met eigen naam).

### Inleveren:

1. HTML bestand
2. Screenshot browser  
Zorg dat de URL van je pagina zichtbaar is!

### Voorbeeld:

[image 1717590065334.png](#)

## 06 - Flexbox Basis

### Wat is Flexbox?

Flexbox is een CSS3 layout module die een efficiënte manier biedt om items in een container te verdelen en uit te lijnen.

[image 1717591361877.png](#)

## Opdracht

1. Maak een nieuw HTML-bestand met de naam 06-CSS-je\_eigen\_naam.html
2. Voeg de bovenstaande code toe en pas de waarden van justify-content aan om te zien hoe de items zich verplaatsen.
3. Maak een overzicht met verschillende waarden.

## Inleveren:

1. HTML bestand
2. Screenshot browser  
Zorg dat de URL van je pagina zichtbaar is!

## Voorbeeld:

[image 1718105546995.png](#)

# 07 - Flexbox Geavanceerd

Uitgebreid gebruik van Flexbox:

```
<!DOCTYPE html>
<html>
<head>
  <title>Opdracht 7 - Flexbox Geavanceerd - Yildiz</title>
  <style>
    .container {
      display: flex;
      flex-wrap: wrap;
      justify-content: space-around;
      align-items: center;
      height: 400px;
    }
    .item {
      background-color: lightcoral;
      padding: 20px;
      margin: 10px;
      width: 100px;
      height: 100px;
      text-align: center;
    }
  </style>
</head>
<body>
  <div class="container">
    <div class="item">Item 1</div>
```

```
<div class="item">Item 2</div>
<div class="item">Item 3</div>
<div class="item">Item 4</div>
<div class="item">Item 5</div>
</div>
</body>
</html>
```

## Opdracht

1. Maak een nieuw HTML-bestand met de naam 07-CSS-je\_eigen\_naam.html
2. Voeg de bovenstaande code toe en experimenteer met flex-wrap en align-items.
3. Lever in: HTML-bestand, screenshot van je webpagina.

### Inleveren:

1. HTML bestand
2. Screenshot browser  
Zorg dat de URL van je pagina zichtbaar is!

### Voorbeeld:

[image-1718105173530.png](#)

## 08 - CSS Grid Basis

### Wat is CSS Grid?

CSS Grid is een layout systeem dat een rasterstructuur biedt voor het plaatsen van items op een webpagina.

Basisgebruik:

[image-1718105883522.png](#)

### Opdracht:

1. Maak een nieuw HTML-bestand met de naam 08-CSS-je\_eigen\_naam.html.
2. Voeg de bovenstaande code toe en pas de grid-template-columns aan om meer kolommen toe te voegen of te verwijderen.
3. Lever in: HTML-bestand, screenshot van je webpagina.

**Inleveren:**

1. HTML bestand
2. Screenshot browser  
Zorg dat de URL van je pagina zichtbaar is!

**Voorbeeld:**

[image 1718106256699.png](#)

## 09 - CSS Grid Geavanceerd

Geavanceerd gebruik van CSS Grid:

[image 1718106729521.png](#)

## Opdracht:

1. Maak een nieuw HTML-bestand met de naam 09-CSS-je\_eigen\_naam.html.
2. Voeg de bovenstaande code toe en pas de grid-template-areas aan om een andere layout te creëren
3. Lever in: HTML-bestand, screenshot van je webpagina.

**Inleveren:**

1. HTML bestand
2. Screenshot browser  
Zorg dat de URL van je pagina zichtbaar is!

**Voorbeeld:**

[image 1718107350939.png](#)

# 10 - Oefenen met flex en grid

## Opdracht:

1. Speel het spel: <https://flexboxfroggy.com> (minimaal tot oefening 5, kom je verder dan krijg je extra punten)
2. Bekijk de 3 hoofdstukken over flexbox <https://www.roc.ovh/w3schools.com/css> (doe de "try it yourself oefeningen.)
3. Speel het spel: <https://cssgridgarden.com> (minimaal tot oefening 5, kom je verder dan krijg je extra punten)
4. Bekijk de 3 hoofdstukken over grid op [https://www.w3schools.com/css/css\\_grid](https://www.w3schools.com/css/css_grid) (doe de "try it yourself oefeningen)

## Inleveren:

1. Screenshot totale scherm van flexboxfroggy om te laten zien dat je minimaal tot oefening 5 gekomen bent.
2. Screenshot totale scherm van van cssgridgarden om te laten zien dat je minimaal tot oefening 5 gekomen bent.
3. Kom je in aanmerking voor bonuspunten, geef dat dan even aan in commentaar.

# 11 - Responsief Design met Media Queries

Wat zijn Media Queries?

Media Queries maken het mogelijk om CSS-regels toe te passen op basis van de eigenschappen van het apparaat waarop de site wordt bekeken.

## Code:

[image1718108159159.png](#)

## Opdracht:

1. Maak een nieuw HTML-bestand met de naam 11-CSS-je\_eigen\_naam.html.
2. Voeg de bovenstaande code toe en experimenteer met verschillende max-width waarden en stijlen.
3. Laat je opdracht aan de docent zien en lever een screencast van je browser in.

**Inleveren:**

1. HTML bestand
2. Screencast browser  
Zorg dat de URL van je pagina zichtbaar is!

**Voorbeeld:**

[image 1718109222228.png](#)

## 12 - Animaties met CSS

CSS animaties maken het mogelijk om overgangen tussen verschillende CSS-stijlen te animeren.

code:

[image 1718109952546.png](#)

## Opdracht:

1. Maak een nieuw HTML-bestand met de naam 12-CSS-je\_eigen\_naam.html.
2. Voeg de bovenstaande code toe en pas de animation-duration en @keyframes waarden aan om andere animaties te creëren.
3. Laat je opdracht aan de docent zien en lever een screencast van je browser in.

**Inleveren:**

1. HTML bestand
2. Screencast browser  
Zorg dat de URL van je pagina zichtbaar is!

**Voorbeeld:**



image 1718110369379.png

---

# Eind Project Ontwerp

Deze eindopdracht zal dienen als een afsluitend project waarin je al je kennis van CSS en HTML toepast. Je gaat een website maken die aan specifieke criteria voldoet. In deze website toont je vaardigheden en interesses. Kijk naar het CSS introductievideo : <https://youtu.be/0afZj1G0BIE> en bouw je eigen website zoals in de video.

Voordat je gaat maken, ontwerp je eerst hoe het gaat worden. Zonder ontwerp kan je niet bouwen. Wees creatief en maak er iets moois van.

Daarna ga je met het realiseren van je website aan de slag.

Als je klaar bent

Veel succes.

## Ontwerp

De site als geheel heeft vaak een ontwerp (template). Je hebt een soort pagina structuur die vaak terugkomt op alle pagina's.

Dit kun je weergeven in een wireframe, mag iemand anders je website op een

bijvoorbeeld:

image 1688328258540.png

Wireframes kun je bijvoorbeeld maken met [Lucidchart](#) of Powerpoint.

Screenshot 2023-07-14 143000.png

Wireframe in Powerpoint

Een wireframe kun je in principe gebruiken om een schets te maken voor alle pagina's niet alleen voor de template-pagina.

### Site map

In een site map laat je de structuur van je gehele website zien. Dus hoe je navigeert van de ene naar de andere pagina. Bijvoorbeeld:

Screenshot 2023-07-14 142428.png

Een website sitemap geeft de structuur van je website visueel weer. Daarom is het bepalen en vastleggen van een sitemap een belangrijke eerste stap tijdens de ontwikkeling van je website. Alle informatie die je op je website wil aanbieden zal ergens op een web pagina in de sitemap een plek krijgen.

(bron: <https://onlineexpert.be/online-tips/website-ontwikkeling/website-sitemap-maken> )

Laat je ontwerp goedkeuren en ga dan verder met het realiseren.

### **Inleveren 1 bestanden**

- Ontwerp van je website.

### **Voorbeeld:**

[image1718195684524.png](#)

## *Eind Project Realisatie*

Maak een website met een header, footer, navigatie en tekst gedeelte. De navigatie gedeelte staan minimaal 4 links naar andere pagina's of website's.

## Testen

Ben je klaar met het realiseren van je website dan laat je het door twee andere personen testen met een test rapport. Zij geven feedback over de werking en lay-out van je website.

Heb je goedkeuring dan kan je het testrapport en website inleveren

### **Inleveren 2 bestanden**

- Testrapport
- Software (zip bestand)

## *Eind Project Presenteren*

Bij het beroep hoort het laten zien wat je hebt gemaakt. Nu is het de tijd om met trots je product te tonen aan belangstellende en rond je deze module af.

## Eisen

Later in je examen moet je ook preenteren. Daarbij zijn dit de belangrijkste punten:

- Je presenteert overtuigend (positief, met passie, trots en met een goede energie)?
- Reageer je op de juiste manier op vragen/feedback?
- Is film bestand kleiner dan 400 MB?

Vraag in de les om uitleg.

### **Inleveren 2 bestanden**

1. Powerpoint met de presentatie
2. Kort filmpje (ongeveer 2 à 3 minuten) waarin je een indruk geeft van je presentatie.

# Front End 4 - DOM Manipulation

## Inleiding

Neem eerste de theorie door op de [deze](#) pagina!

De opdrachten van dit blok sluiten op elkaar aan. Je moet ze dus in de juiste volgorde maken.

Waarschuwing: als je de opdrachten met ChatGPT maakt en je snapt niet wat je doet dan loop je met de laatste opdrachten vast. Deze opdrachten zijn getest en kunnen niet met behulp van ChatGPT 4.0 integraal worden opgelost.

Kom je ergens niet verder, ga dan **niet** verder met de volgende opdracht maar vraag hulp!

## Opdracht 1, kleur aanpassen (via id)

Je hebt de volgende HTML/JS-code.

```
<!DOCTYPE html>
<html lang="nl">
<head>
  <meta charset="UTF-8">
  <title>Tekstkleur Veranderen</title>
</head>
<body>
  <p id="tekst">Dit is een voorbeeldtekst.</p>
  <button onclick="veranderKleur()">Verander Kleur</button>

  <script>
    function veranderKleur() {
      // maak de code af en plaats hieronder de code die de kleur van de tekst verandert
      ...
    }
  </script>
</body>
</html>
```

```
</script>
</body>
</html>
```

De JavaScript-functie wordt aangeroepen vanuit de `onclick` attribuut van de HTML-knop.

Wanneer de knop wordt ingedrukt, moet de kleur van de tekst in het paragraafelement veranderen. Maak hiervoor het script af.

Zorg dat je in het script zoekt naar het element met het id "tekst". Als je het element hebt gevonden dan kan je de daarna de kleur aanpassen.

## Inleveren

Zet jouw naam als commentaar in de JS-code en lever het geteste en werkende HTML-pagina in.

## Opdracht 2, kleur aanpassen (via class)

Je hebt de volgende HTML/JS-code:

```
<!DOCTYPE html>
<html lang="nl">
<head>
  <meta charset="UTF-8">
  <title>Achtergrondkleur Veranderen</title>
</head>
<body>
  <p class="kleurVerander">Dit is een voorbeeldtekst 1.</p>
  <p class="kleurVerander">Dit is een voorbeeldtekst 2.</p>
  <p class="kleurVeranderNiet">Dit is een voorbeeldtekst 3.</p>
  <p class="kleurVerander">Dit is een voorbeeldtekst 4.</p>

  <button onclick="veranderAchtergrondKleur()">Verander Achtergrondkleur</button>

  <script>
    function veranderAchtergrondKleur() {
      // maak de code af en plaats hieronder de code die de kleur van de tekst met de class kleurVerander
```

```
verandert
    ...
}
</script>
</body>
</html>
```

De JavaScript-functie wordt aangeroepen vanuit de `onclick` attribuut van de HTML-knop.

Wanneer de knop wordt ingedrukt, moet de kleur van de tekst in alle paragraafelement met de class *kleurVerander* veranderen. Maak hiervoor het script af.

Maak gebruik van `document.getElementsByClassName` en gebruik een for-loop.

Denk erom dat `document.getElementsByClassName` een array met elementen terug geeft en dat je met de loop door het array moet heenlopen.

Bedek zelf een mooie kleur.

## Inleveren

Zet jouw naam als commentaar in de JS-code en lever het geteste en werkende HTML-pagina in.

## Opdracht 3, stijl toggle

Je hebt de volgende HTML/JS-code:

```
<!DOCTYPE html>
<html lang="nl">
<head>
  <meta charset="UTF-8">
  <title>Tekststijl Veranderen</title>
</head>
<body>
  <h1>Dit is een koptekst</h1>
  <p>Dit is een gewone paragraaf.</p>
  <p class="speciaal">Dit is een speciale paragraaf.</p>
```

```
<div>Dit is een div element.</div>

<p class="speciaal">Nog een speciale paragraaf.</p>
<p id="speciaal">Nog een speciale paragraaf.</p>

<button onclick="veranderTekstStijl()">Verander Tekststijl</button>

<script>
  function veranderTekstStijl() {
    var elementen = document.querySelectorAll(.....);
    for (.....) {
      // Maak de elementen green, bold, and italic (schuingedrukt)
    }
  }
</script>
</body>
</html>
```

Vul de code aan op de puntje en zorg ervoor dat zodra op de knop wordt gedrukt:

- alle elementen met de class *speciaal* **groen, vet en schuingedrukt** worden.

Als er weer op de knop wordt gedrukt dan worden:

- alle elementen met de class *speciaal* worden weer zwarte, niet vet en niet schuingedrukt.

De knop wisselt dus telkens de stijl alle tekst met de class *speciaal*.

## Inleveren

Zet jouw naam als commentaar in de JS-code en lever het geteste en werkende script in.

Zet jouw naam als commentaar in de JS-code en lever het geteste en werkende HTML-pagina in.

## Opdracht 4, counter

Bekijk het volgende voorbeeld.

```
<div id="nummerDiv">123</div>
<button onclick="toonGetal()">Toon Getal</button>

<script>
  function toonGetal() {
    var divElement = document.getElementById('nummerDiv');
    var getal = divElement.innerText; // Lees de tekst (in dit geval een getal) uit de div
    alert("Het getal in de div is: " + getal);
  }
</script>
```

Probeer te begrijpen wat er gebeurt en test de code om te kijken of je snapt wat er gebeurt!

## Opdracht

1. maak een extra button
2. als je op de button druk dan roep je een javascript funcite aan (bedenk zelf een naam voor de functie).
3. met een console.log druk je de waarde af die in de div staat; in dit voorbeeld dus 123.
4. tel 1 op bij het getal en druk dit getal ook af met een console.log. In dit voorbeeld is dat dus 124.

Als je in JS een getal bij een getal optelt met + dan 'denkt' JS standaard dat je een string hebt. Dan wordt 1+1 dus 11 en niet 2! Om dit te voorkomen moet je aangeven dat je geen string maar een integer hebt.

Dit doe je met parseInt(string).

Dus parseInt(getal) + 1 wordt dan 2.

Dus, stel getal = 1

getal + 1 wordt 11 en  
parseInt(getal) + 1 wordt 2

5. dan zet je het getal (in dit voorbeeld dus 124) terug in de div.

Als jet goed is, heb je nu een div met een getal en telkens als je op de knop druk wordt het getal opgehoogd.



# Inleveren

Zet jouw naam als commentaar in de JS-code en lever het geteste en werkende HTML-pagina in.

## Opdracht 5, Ronde teller

Maak een interactieve webpagina met een tabel die schaatsters en hun rondeteller weergeeft.

image 1712404637053.png

De gebruiker kan het aantal ronden voor elke schaatser verhogen door op de teller in de tabel te klikken.

Begin met deze HTML/CSS-code:

```
<!DOCTYPE html>
<html lang="nl">
<head>
  <meta charset="UTF-8">
  <title>Schaatser Ronde Teller</title>
  <style>
    table {
      width: 50%;
      border-collapse: collapse;
      margin: 20px auto;
    }
    th, td {
      border: 1px solid #dddddd;
      text-align: left;
      padding: 8px;
    }
    th {
      background-color: #f2f2f2;
    }
    td.rondes {
      cursor: pointer;
      background-color: #e6ffe6;
```

```
}
td.rondes:hover {
    background-color: #ccffcc;
}
</style>
</head>
<body>

<table>
  <tr>
    <th>Schaatser</th>
    <th>Aantal Rondes</th>
  </tr>
  <tr>
    <td>Mo Mot</td>
    <td class="rondes" onclick="verhoogRondes(this)">0</td>
  </tr>
  <tr>
    <td>Ayoub Allemaal</td>
    <td class="rondes" onclick="verhoogRondes(this)">0</td>
  </tr>
  <tr>
    <td>Klaas Kaas</td>
    <td class="rondes" onclick="verhoogRondes(this)">0</td>
  </tr>
  <tr>
    <td>Goedele Goedgeluk</td>
    <td class="rondes" onclick="verhoogRondes(this)">0</td>
  </tr>
  <tr>
    <td>Roos Rommelhuis</td>
    <td class="rondes" onclick="verhoogRondes(this)">0</td>
  </tr>
  <tr>
    <td>Lars Leeghoofd</td>
    <td class="rondes" onclick="verhoogRondes(this)">0</td>
  </tr>
</table>

</body>
```

</html>

Maak de code af en maak een script waarbij als je op het ronde nummer drukt, de ronde met een wordt verhoogd. Denk aan het gebruik van `parseInt` (vorige opdracht).

## Inleveren

Zet jouw naam als commentaar in de JS-code en lever het geteste en werkende HTML-pagina in.

## Opdracht 6, eventListener

In plaats van een functie aan heel veel rijen toe te voegen, kan je ook een zogenaamde **eventListener** maken.

Een event is een gebeurtenis en een **eventListener** luistert dus of er een bepaalde gebeurtenis plaatsvindt.

We gaan onze code uit de vorige opdracht (5) aanpassen.

1. haal alle onclick attributen uit de `<td>`'s weg.
2. geef de tabel een id `'rondeTellerTabel'`

Bekijk de volgende code:

```
document.addEventListener('DOMContentLoaded', function() {  
    var rondesCellen = document.querySelectorAll('#rondetellerTabel .rondes');  
    rondesCellen.forEach(function(cel) {  
        cel.addEventListener('click', verhoogRondes);  
    });  
});
```

## Uitleg

Regel	Uitleg
1	Zodra de DOM (HTML-document) is geladen voer de volgende code uit.
2	Laad alle cellen uit de tabel met het id <code>rondeTellerTabel</code> en de class <code>rondes</code> .

3	Itereer door alle gevonden cellen ( <i>iteren</i> betekent 'doorlopen').
4	Laat alle gevonden cellen luisteren naar een click en als er een click plaatsvindt, roep dan de functie <i>verhoogRondes</i> aan.

De functie *verhoogRondes* moeten we iets aanpassen.

```
function verhoogRondes() {  
    var huidigeRondes = parseInt(this.innerText);  
    this.innerText = huidigeRondes + 1;  
}
```

Hier gebruiken we *this*, *this* is het huidige element waarop is gelicked.

Zorg ervoor dat de ronde teller weer werkt maar nu met de eventListener zoals hierboven uitgelegd.

Lees goed en aandachtig en als het moet lees het nog eens stap voor stap. Als je alle stappen volgt en begrijpt dan zul je de opdracht kunnen maken.

## Inleveren

De aangepaste code waarin je ronde kan tellen van de vorige opdracht, maar nu maak je gebruik van de eventlistener zoals uitgelegd.

## Opdracht 7, shift key

We nemen de code van onze vorige opdracht en gaan functionaliteit uitbreiden.

Soms wordt er namelijk per ongeluk op de verkeerde kolom geclicked. Je wilt dan het rondenummer verlagen.

Als de shif-key wordt ingedrukt terwijl we clicken moet de het rondenummer worden verlaagd.

Je kun in JS testen of de shift-key wordt ingedrukt:

```
if (event.shiftKey) {
```

Gebruik dit en bereid de JS code uit zodat:

- als je clicked, wordt het rondenummer één hoger (dat hadden we al).
- als je shift-clicked, wordt het rondenummer één lager (nieuw).

We hebben nog één extra eis:

- het rondenummer mag mag niet negatief worden.

## Inleveren

De aangepaste code (volledig werkende HTML/CSS/JS) van de vorige opdracht aangevuld met de functionaliteit zoals hierboven beschreven.

Plaats commentaar in de code met uitleg én plaats commentaar met jouw naam.

## Opdracht 8, shift key en kleur

We maken nog een laatste aanpassing aan de ronde teller. Zorg dat je de vorige opdracht helemaal af hebt.

Als je op de rondeteller klikt wordt de rondeteller verhoogd, doe je dat met shift dan wordt de ronde teller verlaagd.

De achtergrondkleur van de ronde teller is groen. Om aan te geven dat de rondeteller is verlaagd wordt de achtergrondkleur rood. Als daarna de teller weer wordt verhoogd, wordt de achtergrond kleur weer groen.

image1712404975032.png

De eisen:

- standaard staan alle ronde tellers op 0 en zijn ze groen.
- als je op een ronde teller clicked dan, wordt die verhoogd en is/wordt de achtergrondkleur groen.
- als je op een rondeteller shift-clicked, dan wordt die verlaagd en/is wordt de achtergrondkleur rood.
- de achtergrondkleur mag neit negatief worden (lager dan 0), dus als je op een 0 shift-clicked dan gebeurt er niets; de achtergrondkleur blijft hetzelfde en de rondeteller wordt

niet verlaagd.

# Inleveren

De aangepaste code (volledig werkende HTML/CSS/JS) van de vorige opdracht aangevuld met de functionaliteit zoals hierboven beschreven.

Plaats commentaar in de code met uitleg én plaats commentaar met jouw naam.

## Opdracht 9, uitklappen

Bekijk en test de volgende code.

```
<!DOCTYPE html>
<html lang="nl">
<head>
  <meta charset="UTF-8">
  <title>Elektronica Producten</title>
  <style>
    .detailRow {
      display: none;
      background-color: #f9f9f9;
    }
    .clickableRow:hover {
      background-color: #f1f1f1;
      cursor: pointer;
    }
    table {
      width: 100%;
      border-collapse: collapse;
    }
    th, td {
      border: 1px solid #dddddd;
      text-align: left;
      padding: 8px;
    }
    th {
```

```
        background-color: #f2f2f2;
    }
</style>
</head>
<body>

<table id="infoTabel">
    <tr>
        <th>Product</th>
        <th>Categorie</th>
    </tr>
    <tr class="clickableRow">
        <td>iPhone 13</td>
        <td>Smartphone</td>
    </tr>
    <tr class="detailRow">
        <td colspan="2">Merk: Apple | Opslag: 128GB | Kleur: Zwart</td>
    </tr>
    <tr class="clickableRow">
        <td>Samsung Galaxy S21</td>
        <td>Smartphone</td>
    </tr>
    <tr class="detailRow">
        <td colspan="2">Merk: Samsung | Opslag: 256GB | Kleur: Phantom Gray</td>
    </tr>
    <tr class="clickableRow">
        <td>Dell XPS 13</td>
        <td>Laptop</td>
    </tr>
    <tr class="detailRow">
        <td colspan="2">Merk: Dell | Scherm: 13.3 inch | CPU: Intel i7</td>
    </tr>
    <tr class="clickableRow">
        <td>MacBook Air</td>
        <td>Laptop</td>
    </tr>
    <tr class="detailRow">
        <td colspan="2">Merk: Apple | Scherm: 13.3 inch | CPU: M1</td>
    </tr>
    <tr class="clickableRow">
```

```

        <td>Sony WH-1000XM4</td>
        <td>Hoofdtelefoon</td>
    </tr>
    <tr class="detailRow">
        <td colspan="2">Merk: Sony | Type: Over-ear | Noise Cancelling: Ja</td>
    </tr>
    <tr class="clickableRow">
        <td>Google Nest Hub</td>
        <td>Smart Home</td>
    </tr>
    <tr class="detailRow">
        <td colspan="2">Merk: Google | Scherm: 7 inch | Assistant: Google Assistant</td>
    </tr>
</table>

<script>
    document.querySelectorAll('.clickableRow').forEach(row => {
        row.addEventListener('click', function() {
            var detailRow = this.nextElementSibling;
            detailRow.style.display = 'table-row';
        });
    });
</script>

</body>

```

Deze code laat een tabel zien en als je op de regel clicked dan worden details getoond.

### **clickableRow**

De hoofdregels waar je op kan clicken hebben de class *clickableRow*.

### **detailRow**

De detail regels die zichtbaar/onzichtbaar worden door op de hoofdregels te clicken heben de class *detailRow*.

# Opdracht

## Stap 1



Pas de code aan zodat als de detailregel wordt getoond en er weer op de *clickableRow* wordt gedrukt de detailregel weer wordt 'weggehaald' (`style.display = 'none'`).

Het klikken op de *clickableRow* wordt dus een toggle: het laat de details zien en als er weer wordt geclicked worden de details weer weggehaald.

## Stap 2a

We willen een andere vormgeving; de detail-regels moeten inspringen. Jou lead developer gevraagd om de code op te bouwen en om geen table te gebruiken maar div. Je krijgt dus een div met een class *clickableDiv* en daaronder een div met een class *detailRow*

Zorge ervoor dat de twee kolommen in de *clickableDiv* apart in de div staan. Je hebt telkens de naam van het product (bijvoorbeeld iPhone 13) en de categorie (bijvoorbeeld Smartphone), Maak hier twee divs van die naast elkaar staan (tip: gebruik flexbox). Je hebt dus twee divs die in een div staan:

```
<div class="clickableRow">
  <div class="product">iPhone 13</td>
  <div class="category">Smartphone</td>
</div>
<div class="detailRow">
  .....
```

## Stap 2b

Als de table is omgebouwd in divs, zorg er dan voor dat de *clickableRows* en *detailRow* beter te onderscheiden zijn:

- De grootte van het lettertype van de *clickableRows* is meer dan die van de *detailsRows*.
- De *detailsRows* krijgen een margin waardoor het lijkt dat deze inspringen (iets naar rechts gaan)

## Stap 3

Pas de kleuren van de tabel aan zodat de tabel mooi is en goed leesbaar is

## Stap 4

Plaats commentaar in de code waarin je uitlegt hoe de code werkt. Zet ook je naam in ergens in het commentaar.

# Voorbeeld

Als je klaar bent dan zou het er zo kunnen uitzien (kleuren zelf kiezen).

image-1712409645862.png

# Inleveren

De aangepaste code (volledig werkende HTML/CSS/JS) van de vorige opdracht aangevuld met de functionaliteit zoals hierboven beschreven.

# Opdracht 10, Winkelmandje

Je hebt nu een soort productpagina gemaakt bij opdracht 9.

We gaan een begin maken met een winkelmandje. Hiervoor gebruiken we de code van de vorige opdracht en breiden deze uit.

1. Op elke detailRow komt een knopje *Bestellen* - maak deze knop en geef deze knop een mooi CSS stijl.

Voorbeeld:

image-1712409789855.png

2. Bovenin de pagina maak je een nieuwe div, dit is het winkelmandje.
3. Als de knop *Bestel* wordt ingedrukt dan wordt het product toegevoegd als een regel in het winkelmandje. Als je bijvoorbeeld op iPhone 13 clicked dan verschijnt er een regel in het winkelmandje met iPhone 13.
4. Als je dezelfde knop weer indrukt dan komt er een tweede regel iPhone13 in het winkelmandje.
5. Maak vervolgens een knop in de het winkelmandje waarmee je het winkelmandje leeg kunt maken.

# Voorbeeld

image-1712418445270.png

# Inleveren

De aangepaste code (volledig werkende HTML/CSS/JS) van de vorige opdracht aangevuld met de functionaliteit zoals hierboven beschreven.

## Optie, Teller in winkelmandje

Zet in het winkelmandje achter elk product een teller waarmee je klan aangeven hoeveel je wilt bestellen. Gebruik een (deel van de) code van de rondeteller.

- Zorg ervoor dat het aantal niet lager dan 0 en niet hoger dan 10 kan worden.
- (optioneel) zorg ervoor dat er geen dubbele regels in het winkelmandje kunnen ontstaan.

# Inleveren

De aangepaste code (volledig werkende HTML/CSS/JS) van de vorige opdracht aangevuld met de functionaliteit zoals hierboven beschreven.

# Flex Boxes

## Basisstappen om Flexbox te gebruiken

- Voeg `display: flex;` toe aan de container.
- Gebruik eigenschappen zoals:
  - **justify-content**: Hiermee regel je hoe de items horizontaal worden uitgelijnd.  
Bijvoorbeeld: `justify-content: center;` zet de items in het midden.
  - **align-items**: Dit regelt de verticale uitlijning van de items.
  - **flex-wrap**: Hiermee kun je ervoor zorgen dat de items naar een volgende rij gaan als er te weinig ruimte is.

```
.container {  
  display: flex;  
  justify-content: center;  
  align-items: center;  
}  
  
.item {  
  width: 100px;  
  height: 100px;  
  background-color: lightblue;  
  margin: 10px;  
}
```

## flex-basis

Stel, je wilt dat een container de hele breedte van de pagina in beslag neemt dan pas je het item aan

```
.container {  
  display: flex;  
}  
  
.item {
```

```
flex-basis: 100%;  
}
```

More on: <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

## Basis structuur

image.png  
image not found or type unknown

```
<html>  
<head>  
  <style>  
  
    .container {  
      display: flex;  
      flex-direction: column;  
      height: 98vh;  
    }  
  
    .header {  
      background-color: salmon;  
      flex: 0 0 100px;  
      text-align: center;  
      padding: 20px;  
    }  
  
    .content {  
      display: flex;  
      flex: 1;  
    }  
  
    .menu {  
      background-color: lightblue;  
      flex: 1;  
      padding: 20px;  
      text-align: center;  
      margin: 10 10 10 0;  
    }  
  
    .main {
```

```

        background-color: lightgreen;
        flex: 4;
        padding: 20px;
        text-align: center;
        margin: 10 0 10 0;
    }

    .footer {
        background-color: lightgray;
        flex: 0 0 100px;
        text-align: center;
        padding: 20px;
    }
</style>
</head>
<body>
    <div class="container">
        <div class="header">Header</div>
        <div class="content">
            <div class="menu">Menu</div>
            <div class="main">Main</div>
        </div>
        <div class="footer">Footer</div>
    </div>
</body>
</html>

```

## flex: 0 0 100px;

- The item will not grow ( flex-grow: 0 ).
- The item will not shrink ( flex-shrink: 0 ).
- The item has a fixed size of 100 pixels ( flex-basis: 100px ).