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

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

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

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

`class="dupe-option" role="presentation".` En wat doen uberhaupt 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!

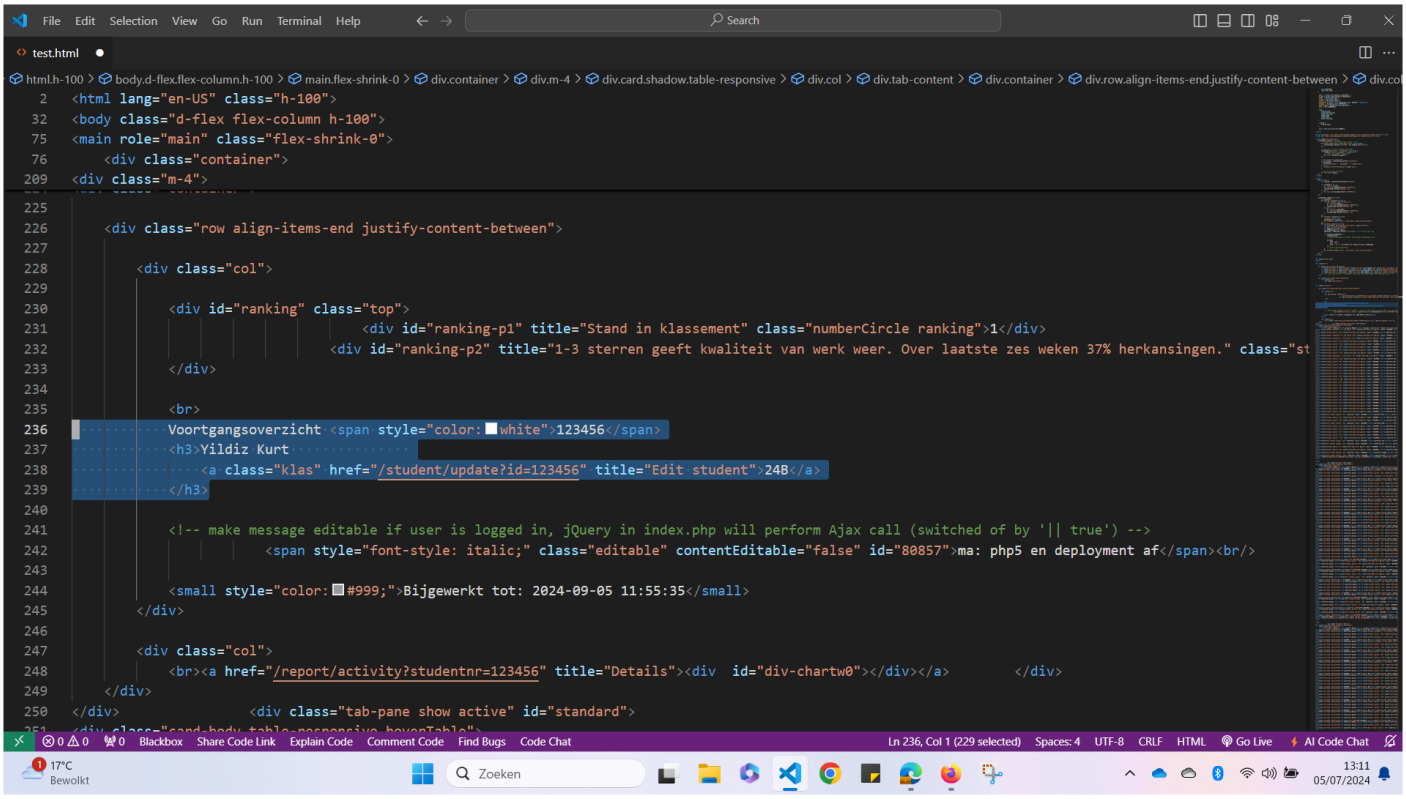
Opdracht

- 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:



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

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

Untitled 2.png

Untitled 3.png

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

Untitled 5.png

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

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 broswer 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 * ,
Sluittag : *in het onderstaande voorbeeld de *

```
<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 * ,
Content : *in het onderstaande voorbeeld de tekst; “Dit is een alinia”*
Sluittag : *in het onderstaande voorbeeld de *

```
<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:

```
<img src="afbeelding.jpg" alt="Beschrijving van de afbeelding">
```

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 broswer 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

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

<!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 weergeven 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 container.

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

h5-group container.png

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

container.png

Voor kleinere elementen zoals stukken tekst kan je een **** 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 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 broswer 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:)* . 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 is dat een 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>
  -
  - <b>
  - <small>
  - <mark>

Inleveren

1. Screenshot van d gehele broswer 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.

In dit hoofdstuk laten we er 2 zien

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 broswer 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
```

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>
    <tbody>
        <tr>
            <td>Cel 1</td>
            <td>Cel 2</td>
        </tr>
        <tr>
            <td>Cel 3</td>
```

```
            <td>Cel 4</td>
        </tr>
    </tbody>
    <tfoot>
        <tr>
            <td>Voettekst 1</td>
            <td>Voettekst 2</td>
        </tr>
    </tfoot>
</table>

table 1.png
```

Gecombineerde Cellen

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

(`colspan` betekent dat een cel meerdere kolommen beslaat, en `rowspan` betekent dat een cel meerdere rijen beslaat.)

```
<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 broswer 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>
form.png
```

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
```

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 latern in de opdracht verder gebruiken***

Andere invoerelementen

Naast `<input>` zijn er nog een aantal elemtenen 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
```

Textarea

Wordt gebruikt om grotere hoeveelheden tekst in te voeren.

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

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 broswer 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 gegroupeerd met containers   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 schermafdruk van je hele browser waarop je blog te zien is.
2. De HTML code

Revision #9
Created 2024-06-10 08:36:55 UTC by Auke
Updated 2024-07-08 12:20:41 UTC by Max