HTML

Tim Quax (tquax@novacollege.nl)

Contents

HTML

Deze reader is een introductie en behandelt de meest eenvoudige basisbegrippen van HTML en CSS. Zodra je de reader helemaal hebt doorgewerkt, ben je in staat om zelfstandig een eenvoudige website te bouwen.

De reader heeft zeker niet als doel om alles over HTML te behandelen, daarvoor is het gebied veel te uitgebreid. Als je méér wilt weten over HTML of als je bepaalde stof van de reader niet helemaal begrijpt, dan is Google er altijd om je te helpen. Als je er dan nog niet uitkomt staan er natuurlijk docenten voor je klaar.

Benodigdheden

- Editor (PHPStorm bijvoorbeeld)
- Browser (Firefox of Chrome bijvoorbeeld).
- Google (Begrijp je iets niet? Dit is je beste vriend!)

Geschiedenis van HTML

HTML staat voor HyperText Markup Language. Vrij vertaald is een markup language een opmaaktaal. Het wordt niet gebruikt om je website op te maken (mooi maken), maar het is bedoeld om tekst automatisch te kunnen laten verwerken. Door een browser, bijvoorbeeld.

HTML werd in 1991 bedacht en gebouwd door Sir Tim Berners-Lee om wetenschappelijke documenten van het CERN makkelijker te delen. Hij heeft ook de eerste webbrowser gebouwd, die heette WorldWideWeb.

Figure 1: De eerste web server ooit, met een sticker zodat collega's de computer niet uitzetten.

In 1994 is het World Wide Web Consortium opgericht (W3C), een non-profit organisatie die de nieuwe standaarden voor het web ontwerpt. Inmiddels heeft W3C meer dan 60 werknemers en wordt het gesteund door bedrijven zoals Google en Microsoft. Tim Berners-Lee is de directeur van W3C.

Alle browsers proberen de web standaarden van W3C zo nauw mogelijk te volgen, zodat websites op alle browsers er hetzelfde uit zien. Ongeacht wat de veranderingen en toevoegingen zijn, het doel blijft altijd het zogenoemde backwards-compatibility; oftewel zelfs websites gemaakt in 1991 moeten altijd blijven werken.

Hoofdstuk 1: Kennis maken met HTML

HTML bestaat uit elementen. Je kan het zien als een blokkendoos. Elke link, afbeelding, paragraaf, etc, is een blok (oftewel een /element/). Vervolgens kan je dat mooi maken met CSS, daar komen we later op terug.

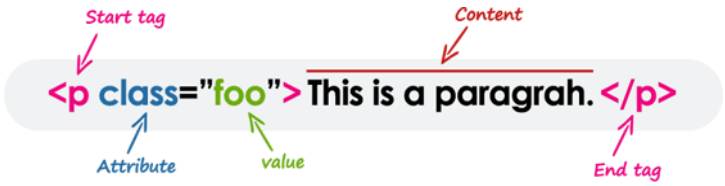
1.1 HTML elementen

Een element bestaat uit meerdere onderdelen. Het heeft een begin en een eind, dat noem je `tags`. Tussen de begin- en eind-tag staat jouw content - bijvoorbeeld de tekst die je wilt laten zien. Je kan in de begin-tag extra informatie meegeven. Dat kan je zien als instellingen, met een naam en de werkelijke instelling. Dat noem je een `attribuut`, en daarin geef je een `waarde` mee.

Er zijn in principe twee soorten HTML elementen. Eigenlijk zes, maar we houden het even simpel:

Normale elementen (normal elements)

Als voorbeeld pakken we een paragraaf (een alinea aan tekst). Dat begint met de tag `<p>`. Tags staan altijd in die chevron tekens (< en >). De tag om te eindigen is dan `</p>`; de slash (/) vertelt dat dit het einde van het element is. Tussen die twee tags staat dus je paragraaf, de werkelijke tekst.



De paragraaf ziet er zo uit:

```
<p>Dit is een paragraaf, en wordt ook zo getoond in de browser.</p>
```

Een link (naar een ander HTML bestand bijvoorbeeld) ziet er zo uit:

```
<a href="https://google.com">Dit is een link naar Google</a>
```

Die ziet er wel heel raar uit, dus laten we die even ontleden:

- Een link werd vroeger een `anchor` genoemd, daarom heet het element dus `<a>`
- De plek waar je de browser heen wilt wijzen, werd de `hypertext reference` genoemd. Oftewel een `href`. Dat is dus het attribuut waar je je link in zet.
- Een link kan wijzen naar een compleet andere website (b.v. `href="https://google.com/"`) of naar een HTML bestand in dezelfde map (b.v. `href="test.html"`) of zelfs een HTML bestand in een andere map (b.v. `map/nog_een_map/test.html`).
- Tussen de open- en sluitende tag (`<a>` en `</a>`) is er wel degelijk inhoud; de tekst waar je op moet klikken om naar de link te gaan.

Lege elementen (void elements)

Sommige elementen, zoals een horizontale lijn op je scherm tekenen of een enter in je tekst (een newline), die hebben geen inhoud nodig. Dat noem je lege elementen, of zelf-sluitende elementen.

Een horizontale streep (Horizontal Rule) typ je bijvoorbeeld zo:

```
<hr>
```

Een enter (oftewel een BReak) typ je zo:

```
<br>
```

Een afbeelding (image -> img) typ je zo:

```
<img src="naam-van-je-afbeelding.jpg">
```

Je kan geen afbeelding direct in tekst stoppen. Dus er is geen inhoud in het `<img>` element, maar wél een attribuut om te vertellen welk bestand (welke afbeelding) daar moet komen te staan. Dat is de source, oftewel de bron waar de afbeelding vandaan komt.

1.2 Waar zet je HTML elementen neer?

Je kan HTML elementen in andere HTML elementen stoppen (bijvoorbeeld een afbeelding in een paragraaf), of meerdere elementen naast elkaar hebben (bijvoorbeeld een link en een afbeelding onder elkaar). Het zetten van een HTML element in een ander HTML element noem je *nesten*.

Je kan bijvoorbeeld tekst in een paragraaf zetten:

```
<p>Voorbeeld tekst</p>
```

Maar je kan er ook in zetten. En enters (newlines). We schrijven het iets anders, zodat je duidelijk ziet welke tekst en HTML elementen waar staat:

```
<p>
    Voorbeeld tekst.
    <br>
    <a href="https://google.com">Ga naar Google</a>
    <br>
    <a href="https://www.w3.org/">Ga naar W3C</a>
</p>
```

Misschien is het eerste wat opvalt dat we heel veel enters, spaties en tabs gebruiken om het overzichtelijker te maken. HTML filtert dat er automatisch uit. Zelfs in een paragraaf zal het alle enters, spaties en tabs vervangen naar 1 spatie.

In de paragraaf staat dus 1 regel tekst, een link naar Google op een nieuwe regel en vervolgens een link naar W3C op nog een nieuwe regel. Zo kan je eigenlijk eindeloos HTML elementen blijven plaatsen in andere HTML elementen. Dat is handig als je bijvoorbeeld een balk bovenin je pagina wilt hebben met een logo én een menu. Dat zijn dan meerdere blokken van HTML elementen die wel bij elkaar staan.

1.3 HTML pagina

Een HTML pagina bestaat uit iets meer dan alleen maar paragrafen en links. Het bestaat uit vier delen:

Doctype

Een `doctype` is exact wat het zegt: een document type. Het geeft aan wat voor soort document jouw bestand is. Het is niet zomaar een tekst document, het is HTML dus toon het ook als HTML.

Dit is altijd de eerste regel in een HTML bestand!

HTML

Na de doctype is het `html` element altijd het eerste wat je ziet. Daar wordt vervolgens alle HTML ingezet. Het is gebruikelijk om daar nog een `lang` attribuut tegen te komen; oftewel de gebruikte taal.

Head

De `head` is een apart gedeelte van de pagina, met informatie /over/ je pagina. Hier staan zaken die je niet gaat zien op je pagina maar wel heel belangrijk zijn. Denk bijvoorbeeld aan opmaak, een titel in de tab bovenin, etc.

De head zie je niet maar is zeker wel deel van de html. Deze staat in het `html` element, altijd als eerste.

Body

De `body` is waar de magie gebeurt. Al je HTML en alle tekst komen hierin te staan.

De body staat ook in het `html` element, direct na de `head` .

Voorbeeld

Als deze vier onderdelen samen komen komt het er zo uit:

```
<!doctype html>
<html lang="en">
  <head>
    <title>HTML hoofdstuk 1</title>
  </head>
  <body>
    <p>Dit is een paragraaf met een mooi brok tekst.</p>
  </body>
</html>
```

1.4 Opdrachten hoofdstuk 1

Maak een nieuw project aan in `PHPStorm` . Maak een bestand `hoofdstuk1a.html` en een bestand `hoofdstuk1b.html` . Zet het bovenstaande voorbeeld erin en maak twee paragrafen in elk bestand, elk met een link naar het andere bestand.

Hoofdstuk 2: Tekst en afbeeldingen in HTML

Nu je wat meer weet over hoe HTML is opgebouwd en hoe het werkt, gaan we wat dieper in de stof van tekst en afbeeldingen. Je hebt al een beetje met paragrafen en links gewerkt, maar je kan er nog veel meer mee. Denk bijvoorbeeld aan dikgedrukte tekst, ondersteepte tekst, koppen bij teksten, enzovoorts.

2.1 Tekst

Zonder uberhaupt nog CSS aan te raken kan een paragraaf al best goed aangekleed worden.

Dikgedrukte tekst

Je kan tekst dikgedrukt maken met `<b>` (van bold).

Bijvoorbeeld: `Alleen dit <b>woord</b> is dikgedrukt.`

Schuingedrukte tekst

Schuin gedrukte tekst maak je met `<i>` (van italic).

Bijvoorbeeld: `Alleen dit <i>woord</i> is schuingedrukt.`

Ondersteepte tekst

Ondersteepte tekst maak je met `<u>` (van underline).

Bijvoorbeeld: `Alleen dit <u>woord</u> is ondersteept.`

Kop boven tekst

Een kop zet je boven de paragraaf, en die kop kan je in verschillende groottes neerzetten. Er zijn 6 groottes, en de grootte zit in de naam. `<h1>` (van header) is de grootste kop en `h6` is de kleinste. Bijvoorbeeld:

```
<h1>Een hele grote kop boven de tekst</h1>
<p>Met een mooie paragraaf eronder.</p>

<h6>Een hele kleine kop boven de tekst</h6>
<p>Met een iets minder mooie paragraaf eronder.</p>
```

2.3 Speciale tekens

Speciale karakters kan je laten zien op een webpagina door een bijhorende code in de HTML te zetten. Elk van de speciale karakters begint met een ampersand (`&`) en eindigt met een puntkomma (`;`). Als je deze code in je HTML zet, wordt het automatisch vertaald naar het speciale teken.

Het copyright symbool (`©`) kan je bijvoorbeeld neerzetten als `&copy;` .

Het euro teken (`€`) kan je neerzetten als `&euro;` .

We hebben het eerder gehad over hoe alle spaties, tabs en enters uiteindelijk worden getoond als 1 spatie. Als je toch meerdere spaties wilt tonen, kan je de “no break space” gebruiken: ` ` . Zoals de naam zegt wordt deze spatie niet opgebroken naar een andere regel en wordt het niet weggecijferd door HTML zelf.

2.4 Afbeeldingen

Afbeeldingen kan je overal in de HTML neerzetten, zelfs tussen tekst, en dan staat de afbeelding simpelweg tussen de tekst. Een afbeelding maak je met de `img` tag (image). De `<img>` tag heeft geen inhoud tussen de tags, want daar kan geen afbeelding in gezet worden. De HTML is alleen tekst, dus je zet daar de locatie van de afbeelding - de map en bestandsnaam waar het gevonden kan worden.

Bijvoorbeeld:

```
<img src="naam-van-je-afbeelding.jpg">
```

Als de afbeelding in een map `images` staat, dan doe je dit:

```
<img src="images/naam-van-je-afbeelding.jpg">
```

Je kan daar zoveel mappen neerzetten als je wilt. Je geeft de mappen aan vanaf de map waar je HTML bestand staat. Dus in dezelfde map van `voorbeeld.html` staat een map `images` . In die map `images` staat het bestand `naam-van-je-afbeelding.jpg` .

Je kan bij een afbeelding de hoogte of breedte (of allebei) aanpassen.

Een afbeelding die 400 pixels breed wordt, met automatisch de hoogte berekend:

```
<img src="naam-van-je-afbeelding.jpg" width=400>
```

Een afbeelding die 250 pixels hoog is, met de breedte automatisch berekend:

```
<img src="naam-van-je-afbeelding.jpg" height=250>
```

2.4 Opdrachten hoofdstuk 2

Maak een hoofdstuk2.html. Maak er iets leuks van, er moet minimaal in staan:

- Tekst - twee kopjes met elk een paragraaf;
- De tekst is opgemaakt - met dikgedrukte-, schuine-, onderstreepte tekst en enters;
- Een afbeelding - download een afbeelding en zet deze in de map bij je HTML bestand;
- Een footer (een regel onderin de pagina) met een copyright regel.

Hoofdstuk 3: Opsomminglijsten, tabellen en divs

Teksten en afbeeldingen zijn belangrijk, maar het tonen van data ook. HTML heeft opsomminglijsten en tabellen om op een overzichtelijke manier data te laten zien.

3.1 Opsomminglijsten

- Een opsomminglijst kan handig zijn.
- Als je een lijst van items wilt noteren.
- Denk aan b.v. boodschappen of taken.
- Maar vooral geen alinea aan tekst, verworven in een lijst.

Een lijst geef je aan met `<ul>` (unordered list) of `<ol>` (ordered list). Een ordered list is genummerd waar een unordered list stipjes heeft. In zo'n lijst heb je vervolgens list items (`<li>`) staan.

Een unordered list ziet er zo uit:

```
<ul>
  <li>List item nummer</li>
  <li>List item nummer, nog eentje</li>
  <li>List item nummer, de laatste</li>
</ul>
```

En een ordered list ziet er dan zo uit:

```
<ol>
  <li>List item nummer 1</li>
  <li>List item nummer 2</li>
  <li>List item nummer 3</li>
</ol>
```

3.2 Tabellen

Als je meer dan een lijst wilt, dan kan je de data neerzetten in rijen en kolommen. Oftewel, tabellen.

Tabel

Je start een tabel met `<table>` en sluit deze weer af met `</table>` .

Rij

Je geeft een rij aan met `<tr>` (table row) en daarin komen vervolgens de kolommen. Het enige wat ooit in een `<tr>` mag staan zijn kolommen. Elke `<tr>` wordt een nieuwe rij in de tabel.

Kolommen

Je geeft een kolom aan met `<td>` (table data). Des te meer kolommen je neerzet in een rij, des te meer vakken de rij wordt opgedeeld. Dit bepaalt dus hoeveel kolommen de tabel krijgt.

Bijvoorbeeld:

```
<table>
  <tr>
    <td>Rij 1. Kolom 1.</td>
    <td>Rij 1. Kolom 2.</td>
  </tr>
  <tr>
    <td>Rij 2. Kolom 1.</td>
    <td>Rij 2. Kolom 2.</td>
  </tr>
</table>
```

Dat komt er dan zo uit te zien:

```
Rij 1. Kolom 1. Rij 1. Kolom 2.
Rij 2. Kolom 1. Rij 2. Kolom 2.
```

Kopjes

Het is altijd wel wenselijk om kopjes boven alle kolommen te zetten, zodat je weet waar je precies naar kijkt. In plaats van een normale kolom (`<td>` , table data) zet je deze neer als een `<th>` (table header).

Bijvoorbeeld:

```
<table>
  <tr>
    <th>Kop 1</th>
    <th>Kop 2</th>
  </tr>
  <tr>
    <td>Rij 1. Kolom 1.</td>
    <td>Rij 1. Kolom 2.</td>
  </tr>
  <tr>
    <td>Rij 2. Kolom 1.</td>
    <td>Rij 2. Kolom 2.</td>
  </tr>
</table>
```

Dat ziet er dan zo uit:

Kop 1	Kop 2
Rij 1. Kolom 1.	Rij 1. Kolom 2.
Rij 2. Kolom 1.	Rij 2. Kolom 2.

3.3 div

Regelmatig zal je je HTML willen opsplitsen, puur voor het stijlen. Meerdere kopjes met paragrafen onder elkaar hoeven helemaal niet bij elkaar horen. Misschien wil je dat tonen met ruimte ertussen, of een andere achtergrondskleur. Daar komt `<div>` (division) bij kijken.

Het heeft van zichzelf nauwelijks opmaak. Het zal de volledige breedte innemen dat het kan innemen, verder zie je de hele `<div>` niet. Maar de verschillende divs kan je vervolgens wel anders stijlen.

Je kan alle soorten HTML in een div zetten, en zoveel divs gebruiken als je maar wilt. Een voorbeeld:

```
<div>
  <h1>Kopje 1</h1>
  <p>Korte paragraaf 1</p>
</div>

<div>
  <h1>Kopje 2</h1>
  <p>Korte paragraaf 2</p>
</div>
```

3.4 Opdrachten hoofdstuk 3

- Een boodschappenlijst met prijzen en euro tekens, zonder sortering (unordered list);
- Een stappenlijst van hoe je iets koopt in de winkel, met sortering (ordered list);
- Een tabel met een aantal producten uit de supermarkt, met een naam, merk, prijs, etc, inclusief kopjes.

Hoofdstuk 4: CSS

Nu je weet hoe HTML in grote lijnen gebouwd wordt, gaan we het mooi maken. Het stylen van de HTML doen we met CSS (Cascaded Style Sheets). Wat het “cascaded” gedeelte uit CSS betekent wordt later duidelijk.

Het is een specifieke taal (geen programmeertaal) om styling aan te kunnen geven aan HTML elementen. Specifiek aan 1 html element, of aan alle paragrafen (`<p>`) of alle afbeeldingen (`img`), of zelfs alle afbeeldingen die in paragrafen staan.

Daarom zijn het cascaded style sheets; cascading betekent "trapsgewijs": Je zet bijvoorbeeld een tekstgrootte op de hele `<body>` en alle paragrafen (`<p>`) in de body krijgen deze stijl mee. Als het water van een waterval, gaan de stijlen ook trapsgewijs naar de HTML elementen eronder.

4.1 CSS regels

CSS bestaat uit rulesets, of rule / regel voor kort, om aan te geven welke stijl aan welk element moet hangen. Zo'n regel bestaat uit deze onderdelen:

Selector

Als eerst geef je aan waar je de stijlen op wilt toepassen. Op een paragraaf? Een afbeelding? Een specifieke afbeelding in een paragraaf? Dat is de selector.

Bijvoorbeeld `p` zal de stijlen toepassen op alle paragrafen op de hele pagina. `img` zal de stijlen toepassen op alle images op de hele pagina.

`p img` zal de stijlen toepassen op alle afbeeldingen die in een paragraaf staan. En `ul li` zal alleen de stijlen toepassen op list items in een unordered list.

`p, img` zal de stijlen toepassen op zowel paragrafen EN afbeeldingen. Een subtiel verschil.

Declaratie (of enkele regel, single rule)

Elke stijl is hetzelfde opgebouwd. Alle stijlen hebben een vaste naam (een property) en daar kan je een waarde in stoppen (property value). Dat geef je aan zoals dit: `background-color: red` . In dit geval is `background-color` de vaste naam voor de stijl van de achtergrondkleur en de waarde die je meegeeft is rood, dus de achtergrondkleur wordt dan ook rood.

Elke stijl (regel) wordt afgesloten met een puntkomma (`;`).

Al deze stijlen (regels) voor een selector staat bij elkaar tussen curly braces (`{` en `}`).

De bovenstaande onderdelen komen dan zo samen:

```
p {
  color: green;
  background-color: black;
}
```

Dit maakt de kleur van de tekst in alle paragrafen groen, en de achtergrond van alle paragrafen worden zwart.

4.2 Specifieke CSS selectors

Een heel veel voorkomende situatie is het willen stylen van een specifiek HTML element. Niet alle afbeeldingen, of alle paragrafen, maar 1 specifieke paragraaf. Op die ene pagina, daar. En niks anders. Of je wilt misschien een aantal paragrafen een style geven - maar niet allemaal.

ID

Je kan elk HTML element een ID meegeven, een HTML attribuut dat een unieke waarde moet hebben.

HTML voorbeeld:

```html
<h1 id="unieke_naam_van_deze_kop">Kopje</h1>
```

In CSS kan je HTML met een ID stylen met een hekje (`#`) ervoor. Bijvoorbeeld:

```css
#unieke_naam_van_deze_kop {
  color: red;
}
```

Classes

Je kan een groep van HTML elementen ook stijlen geven. Dan kies je zelf exact welk HTML element de stijlen wel krijgt, als een groepering. Dat is het `class` attribuut die je op alle HTML elementen kan toevoegen.

HTML voorbeeld:

```html
<ul>
  <li>Geen speciale styling</li>
  <li class="speciale_styling">WEL speciale styling!</li>
  <li>Saaie list item</li>
  <li class="speciale_styling">Hele toffe list item!</li>
</ul>
```

In CSS kan je HTML met een class stylen met een punt (`.`) ervoor. Bijvoorbeeld:

```css
.speciale_styling {
  background-color: black;
  color: yellowgreen;
}
```

Dat ziet er dan zo uit:

- Geen speciale styling
- WEL speciale styling!
- Saaie list item
- Hele toffe list item!

Je kan dit nog steeds gebruiken in combinatie met andere selectors. Bijvoorbeeld alleen elementen met de class `speciale_styling` in unordered lists:

```css
ul .speciale_styling {
  ...
}
```