

HTML

Tim Quax (tquax@novacollege.nl)



Contents

HTML	3
Benodigdheden	3
Geschiedenis van HTML	3
Hoofdstuk 1: Kennis maken met HTML	4
1.1 HTML elementen	4
Normale elementen (normal elements)	4
Lege elementen (void elements)	5
1.2 Waar zet je HTML elementen neer?	5
1.3 HTML pagina	6
Doctype	6
HTML	6
Head	6
Body	6
Voorbeeld	6
1.4 Opdrachten hoofdstuk 1	6
Hoofdstuk 2: Tekst en afbeeldingen in HTML	7
2.1 Tekst	7
Dikgedrukte tekst	7
Schuingedrukte tekst	7
Onderstepte tekst	7
Kop boven tekst	7
2.2 Opsomminglijsten	7
2.3 Afbeeldingen	7

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 ``) 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:

```

```

Je kan geen afbeelding direct in tekst stoppen. Dus er is geen inhoud in het `` 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, onderstepte tekst, koppen bij teksten, opsomminglijsten, enzovoorts.

2.1 Tekst

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

Dikgedrukte tekst

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

Bijvoorbeeld: Alleen dit `woord` is dikgedrukt. .

Schuingedrukte tekst

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

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

Onderstepte tekst

Onderstepte 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>
```

2.2 Opsomminglijsten

Asd

2.3 Afbeeldingen

Asd