

Een webpagina maken

2.1 Inleiding

In dit hoofdstuk leer je een webpagina te maken. Je leert uit welke onderdelen een webpagina bestaat, hoe je tekst kunt weergeven en hoe je hyperlinks maakt.





2.2 Structuur van een HTML-document

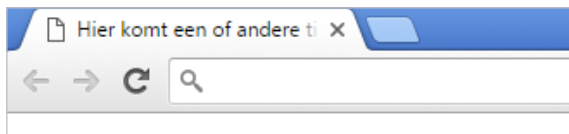
2.2.1 Elementen en tags

Een HTML5-bestand bestaat niet alleen uit tekst. Je kunt in een HTML5-bestand namelijk ook beschrijven welke objecten, zoals hyperlinks en afbeeldingen, er voorkomen. Voor het weergeven van hyperlinks en afbeeldingen gebruik je **elementen**.

Een HTML5-bestand is volledig opgebouwd uit elementen. Dat betekent dat alle onderdelen van de webpagina, dus je inhoud, tot een bepaald element behoren. Een voorbeeld van zo'n element is het title-element. Dat element vertelt de webbrowser wat de naam van de webpagina is. Bijvoorbeeld:

```
<title>Hier komt een of andere titel</title>
```

Dit ziet er zo uit in de webbrowser:



Zoals je kunt zien bestaat dit element uit drie onderdelen:

- een **begintag**, namelijk <title>
- inhoud, namelijk 'Hier komt een of andere titel'
- een **eindtag**, namelijk </title>

De meeste elementen hebben een begin- en eindtag. Elementen zonder inhoud bestaan meestal uit slechts één tag. Voorbeelden hiervan komen later aan de orde. De inhoud van het element moet altijd tussen de begin- en eindtag in staan, anders behoort het niet tot het element. Belangrijk is dat je alle elementen met kleine letters schrijft.

2.2.2 De opbouw van een HTML5-bestand

Zo bouw je een HTML5-bestand op:

- **DOCTYPE-declaratie**

Je begint een HTML-bestand altijd met een tag waarin een **DOCTYPE**-declaratie zit. Daarmee vertel je de webbrowser welke versie van HTML je gebruikt. Voor HTML5 is dat:

```
<!DOCTYPE html>
```

Dit lijkt een heel algemene omschrijving, want er staat 'html', en niet 'html 5', maar deze DOCTYPE-declaratie geldt echt alleen voor HTML versie 5. De code om aan te geven dat je HTML versie 4 gebruikt is heel anders, namelijk:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
```

- **html-element**

Na de DOCTYPE-declaratie zet je het **html-element** neer. Dit is het hoofdelement van je bestand waarin je al je verdere code plaatst. Het html-element mag maar één keer voorkomen in je HTML5-bestand.

```
<!DOCTYPE html>
<html>
</html>
```

Je kunt het html-element als volgt uitbreiden:

```
<html lang="nl">
```

Door deze toevoeging zien zoekmachines en webbrowsers dat de webpagina is geschreven in de Nederlandse taal.

**Head-element****Title-element****Body-element**

In dit voorbeeld staat er nog niets binnen het html-element. Dat is uiteraard wel de bedoeling. Je plaatst altijd twee elementen in het html-element: het head-element en het body-element. Alle andere elementen plaats je in het head- óf in het body-element.

- **head-element**

Het head-element bevat informatie over de website die je niet op de pagina zelf ziet, zoals de titel (voor in de webbrowserbalk of op het titelblad), gegevens over de auteur van de website, informatie voor zoekmachines en gegevens over een eventueel opmaakbestand (dit komt in dit hoofdstuk nog niet aan bod).

- **title-element**

De titel van het tabblad (in dit voorbeeld 'Hier komt een of andere titel') kun je in het head-element opgeven met het title-element:

```
<head>
  <title>Hier komt een of andere titel</title>
</head>
```

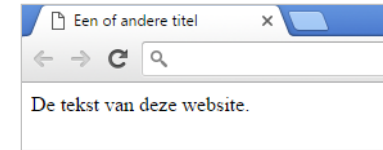
- **body-element**

Zoals gezegd plaats je het body-element, net als het head-element, in het html-element. Het body-element bevat de werkelijke inhoud van de pagina, dus alle tekst, afbeeldingen enzovoort die de gebruiker te zien krijgt. Dit is een voorbeeld van een HTML-bestand met de head- en body-elementen:

```
<!DOCTYPE html>
<html lang="nl">
  <head>
    <title>Een of andere titel</title>
  </head>

  <body>
    De tekst van deze website.
  </body>
</html>
```

Als je dit in de webbrowser Chrome bekijkt, ziet het er zo uit:



- de inhoud van het head-element op het tabblad
- de inhoud van het body-element op de pagina zelf.

- **meta-tags**

In het head-element kunnen naast de titel van de website ook gegevens staan over de auteur van de website, informatie voor zoekmachines en gegevens over een eventueel opmaakbestand. Dit zijn de meta-tags. Een meta-tag heeft twee **attributen**, namelijk 'name' en 'content'.

Dit zijn de meest gebruikte meta-tags:

Name	Toepassing
author	Naam van de auteur
keywords	Zoekwoorden voor zoekmachines, gescheiden door een komma
description	Omschrijving van de website voor zoekmachines

Bijvoorbeeld:

```
<meta name="author" content="Uitgeverij Instruct">

<meta name="keywords" content="HTML5, head, body, meta-tags">

<meta name="description" content="Op deze pagina leer je de basis van HTML5!">
```

Naast deze drie meest gebruikte meta-tags is er een bijzondere meta-tag, waarmee je kunt opgeven van welk letterstelsel je website gebruik maakt. Hiermee voorkom je dat speciale karakters, zoals é, ï of Å verkeerd worden weergegeven. Deze meta-tag is als volgt:

Meta-tags**Attributen**



Genest

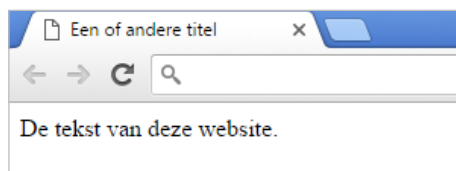
```
<meta charset="utf-8">
```

In dit voorbeeld zijn de meeste meta-tags gebruikt:

```
<!DOCTYPE html>
<html lang="nl">
  <head>
    <title>Een of andere titel</title>
    <meta charset="utf-8">
    <meta name="author" content="Uitgeverij Instruct">
    <meta name="keywords" content="HTML5, head, body, meta-tags">
    <meta name="description" content="Op deze pagina leer je de
    basis van HTML5!">
  </head>

  <body>
    De tekst van deze website.
  </body>
</html>
```

Als je dit in de webbrowser Chrome bekijkt, ziet het er nog steeds zo uit:



Voor de overzichtelijkheid van de voorbeeldcode die in dit boek gegeven is, zijn de meta-tags weggelaten.

Genest

In de bovenstaande code zie je dat het title-element en de meta-elementen in het head-element staan. We zeggen ook wel dat deze elementen **genest** zijn in het head-element. De elementen head en body zijn op hun beurt weer genest in het html-element.

Het is gebruikelijk om geneste elementen door middel van tabs te laten inspringen, zoals ook is gedaan in bovenstaand voorbeeld. Dit maakt je code overzichtelijk.

2.3 Tekst opmaken

Vergelijk de onderstaande code eens met het resultaat in de webbrowser:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Koppen en alinea's</title>
  </head>

  <body>
    <h1>Koppen en alinea's</h1>
    <p>In een HTML-document zijn koppen en alinea's op een duidelijke
    manier van elkaar gescheiden.</p>

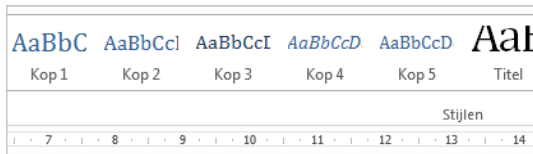
    <h2>Koppen</h2>
    <p>Koppen kun je gebruiken om je tekst een titel te geven. Op die
    manier weet de lezer meteen waar de over gaat.<br>
    Je kunt ook subkoppen gebruiken om stukjes in je tekst titels te
    geven. HTML ondersteunt koppen tot en met het zesde niveau.</p>
  </body>
</html>
```



Zoals je kunt zien wordt sommige tekst groter en dikker weergegeven dan andere tekst. Bekijk de HTML-code eens en bedenk hoe dit komt.

Koppen

Je kunt gebruik maken van **koppen** en **subkoppen** om een titel of een subtitel in je tekst aan te geven, net zoals in Microsoft Word.



De grootste kop kun je invoegen in je website door middel van de elementen:

```
<h1>
<h2>
```

Webbrowsers passen dan automatisch de tekstgrootte aan en maken de tekst dikgedrukt. Ook zal de kop op een nieuwe regel komen te staan. Een wat kleinere kop voeg je in met:

```
<h3>
```

In totaal zijn er zes niveaus van (sub)koppen. De webbrowser maakt de tekstgrootte van deze koppen automatisch steeds kleiner. De kleinste kop is dus:

```
<h6>
```



Paragrafen

Tekst op een website moet, net als bijvoorbeeld een kop, binnen een element staan. Een alinea met tekst plaats je op je website door middel van het **paragraph-element**:

```
<p>
```

Een webbrowser plaatst automatisch een lege regel tussen twee alinea's als die met dit element zijn gemaakt.

Kijk hier eens naar:

```
<p>In een HTML-document zijn koppen en alinea's op
een duidelijke
manier
van elkaar gescheiden.</p>
```



Zoals je ziet, staan hier heel veel spaties en een paar witregels. Als je deze code zou uitvoeren in een webbrowser, dan zie je van die extra spaties en witregels niets. Dit komt omdat je webbrowser een enter als een spatie beschouwt, en extra spaties tussen woorden simpelweg negeert. Als je toch naar een volgende regel wilt gaan, gebruik je het **break-element**:

```
<br>
```

Dit element bestaat maar uit één tag: het is begin- en eindtag ineen.

Opmaak

Er zijn enkele elementen die je voor de opmaak van de tekst kunt gebruiken. De mogelijkheden van HTML-elementen voor de opmaak zijn vrij beperkt. De 'echte' opmaak van je website, qua kleuren en dergelijke, gebeurt met CSS-code. Dit wordt uitgewerkt in hoofdstuk 4.

Begincode	Eindcode	Resultaat
<u>	</u>	<u>onderstreept</u>
		vet
<i>	</i>	<i>cursief</i>
		benadrukt

Speciale karakters

Heb je in je HTML-tekst een dubbele spatie, dan zie je die normaal gesproken niet. Als je die zichtbaar wilt maken, moet je hiervoor een 'speciaal karakter' gebruiken. Er zijn verschillende speciale karakters, bijvoorbeeld:

Code	Resultaat	Naam
 		spatie
"	"	dubbele aanhalingstekens
&	&	ampersand
<	<	kleiner dan
>	>	groter dan
©	©	copyright
€	€	euro

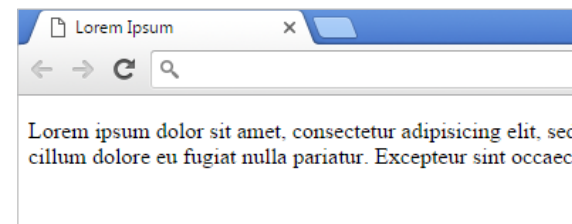


Lorem ipsum

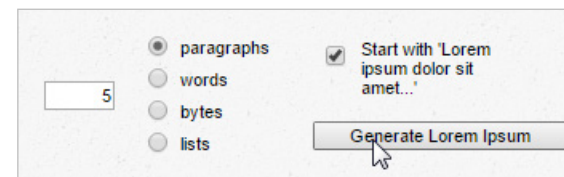
Om op een snelle en eenvoudige manier tekst op je website te krijgen kun je gebruik maken van **dummytekst**. Dit is tekst die door drukkers en grafisch ontwerpers wordt gebruikt om te kijken hoe een tekst of een lettertype eruit ziet. De meest gebruikte dummytekst is **lorem ipsum**. De eerste versie daarvan stamt al uit de 16e eeuw en begint als volgt:

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

De tekst van lorem ipsum lijkt op het eerste gezicht Latijn te zijn, maar in werkelijkheid heeft het volstrekt geen betekenis. Dat is de reden waarom het wordt gebruikt door drukkers en grafisch ontwerpers: als de tekst leesbaar zou zijn word je snel afgeleid, terwijl het nu om de vormgeving en niet om de inhoud gaat.



Op de website www.lipsum.com kun je zelf deze dummytekst genereren.



**Hyperlink**

In papieren boeken is het mogelijk om te verwijzen naar een ander boek. Als je die verwijzing wilt bekijken moet je dat andere boek ook in je bezit hebben en vervolgens bladeren naar de juiste bladzijde. Dit is best omslachtig. Op internet kan dit veel makkelijker en sneller, namelijk met een **hyperlink**. Een hyperlink is een stukje tekst of plaatje dat, als de gebruiker erop klikt, een andere pagina of website opent.

Er zijn drie typen hyperlinks:

Soort	Verwijst naar	Voorbeeld
Vaste URL	Een website	www.google.nl
Relatieve URL	Een subpagina	contact.html
Mailto URL	Een e-mailadres	mailto:iemand@aarde.nl

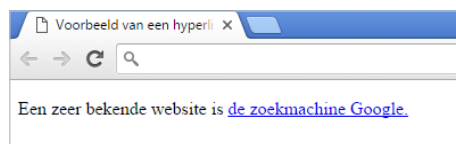
Hyperlink met een vaste URL

Vergelijk deze code eens met het resultaat eronder, zoals je het in de webbrowser Chrome ziet:

```
<!DOCTYPE html>

<html>
  <head>
    <title>Voorbeeld van een hyperlink</title>
  </head>

  <body>
    <p>Een zeer bekende website is
    <a href="http://www.google.nl">de zoekmachine Google</a>.</p>
  </body>
</html>
```



Je ziet dat 'de zoekmachine Google' tussen een begin- en eindtag staat:

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

Hierdoor maakt de webbrowser een hyperlink van die tekst. Als je geen opmaak aan een hyperlink geeft zal de webbrowser de hyperlink een blauwe tekstkleur en onderstreping geven.

De verwijzing van een hyperlink geef je op in de begintag met een **attribuut**. Dit attribuut heeft de naam **href** (hypertext reference). Achter een attribuut komt een =-teken, gevolgd door de waarde. Die waarde staat tussen dubbele aanhalingstekens (""). Je moet elke waarde apart tussen aanhalingstekens zetten.

De waarde die het href-attribuut krijgt is een **URL**. Dit is de afkorting van Universal Resource Locator.

Vergeet bij een hyperlink naar een vaste URL nooit het protocol (meestal http://), anders denkt de webbrowser dat je een hyperlink met een relatieve URL hebt gemaakt.

Hyperlink met een relatieve URL

De tegenhanger van een hyperlink met een vaste URL is de hyperlink met de relatieve URL. Vergelijk de code hieronder eens met de code in de vorige paragraaf.

```
<!DOCTYPE html>

<html>
  <head>
    <title>Relatieve URL</title>
  </head>

  <body>
    Klik <a href="hyperlink.html">hier</a>
    voor de pagina uit de vorige paragraaf.
  </body>
</html>
```

**Relatieve URL****Mailto-URL**

Hier is aan het attribuut *href* de waarde 'hyperlink.html' gegeven, zonder domeinnaam. Dit noemen we een **relatieve URL**. Een relatieve URL gaat uit van de URL van het document waarin hij staat. Vervolgens zoekt hij vanaf die locatie naar het opgegeven bestand.

Het is met een relatieve URL mogelijk om een hyperlink naar een bestand in een submap te maken. Als je bij een relatieve URL alleen een bestandsnaam opgeeft als waarde van het attribuut *href*, zal de webbrowser dit bestand in dezelfde map zoeken als waar het oorspronkelijke bestand staat. Geef je vóór de bestandsnaam een submap op, dan zal de webbrowser het bestand uit de hyperlink zoeken in deze submap. Zo wordt in dit voorbeeld het bestand `contact.html` in de map `pages` geopend als er op de hyperlink geklikt wordt:

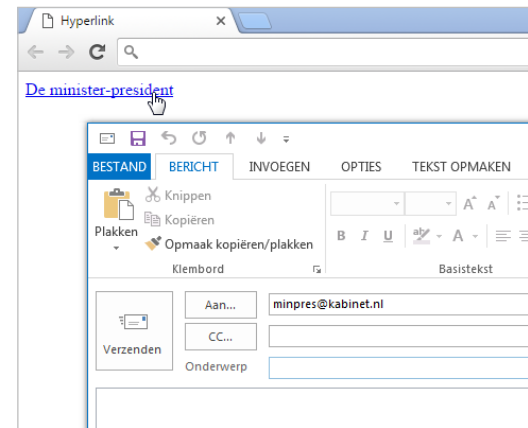
```
<a href="pages/contact.html">contact</a>
```

Hyperlink met een mailto-URL

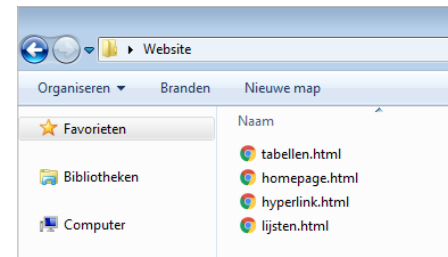
Naast het openen van een andere pagina is het ook mogelijk om bij het klikken op een hyperlink een e-mail te versturen met het standaard e-mailprogramma op de computer, zoals Microsoft Outlook. Hieronder zie je een voorbeeld van een **mailto-URL**:

```
<a href="mailto:minpres@kabinet.nl">
  De minister-president
</a>
```

Als je op de hyperlink klikt wordt je e-mailprogramma geopend met een mailtje naar het opgegeven adres.

**2.5 Subpagina's maken**

Een website bestaat over het algemeen uit meerdere pagina's die onderling met elkaar verbonden zijn door middel van hyperlinks met een relatieve URL. Om het beheren van deze bestanden en het maken van de hyperlinks overzichtelijk te houden is het aan te raden om alle HTML-bestanden van je website in één map te plaatsen.

**2.6 Commentaar**

Als websiteontwikkelaar kun je in je code aantekeningen maken, die alleen voor jou als ontwikkelaar zichtbaar zijn, dus niet voor de bezoekers van je website. Het is net zoiets als de post-it papiertjes die je in een boek kunt plakken.

**Handig**

Het gebruik van commentaar kan handig zijn als je een lastig stukje HTML-code hebt gemaakt waarvan je later de betekenis misschien niet meer goed weet.

Als je dan bij dit stuk code in een **commentaarregel** zet wat de werking is, kun je het later eenvoudig weer begrijpen. Ook tijdens het uitwisselen van code met andere ontwikkelaars is het handig om gebruik te maken van commentaar. Dingen die voor jou logisch zijn, zijn dat misschien voor anderen niet!

Vergelijk de onderstaande code eens met het resultaat in de webbrowser:

```
<!DOCTYPE html>

<html>
  <head>
    <title>Voorbeeld met commentaar</title>
  </head>

  <body>
    <!--Dit is een regel commentaar-->
    <p>In deze pagina is een regel commentaar geplaatst,
    <br>maar die is voor een bezoeker niet zichtbaar.</p>
  </body>
</html>
```



De eerste coderegels in de body is:

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

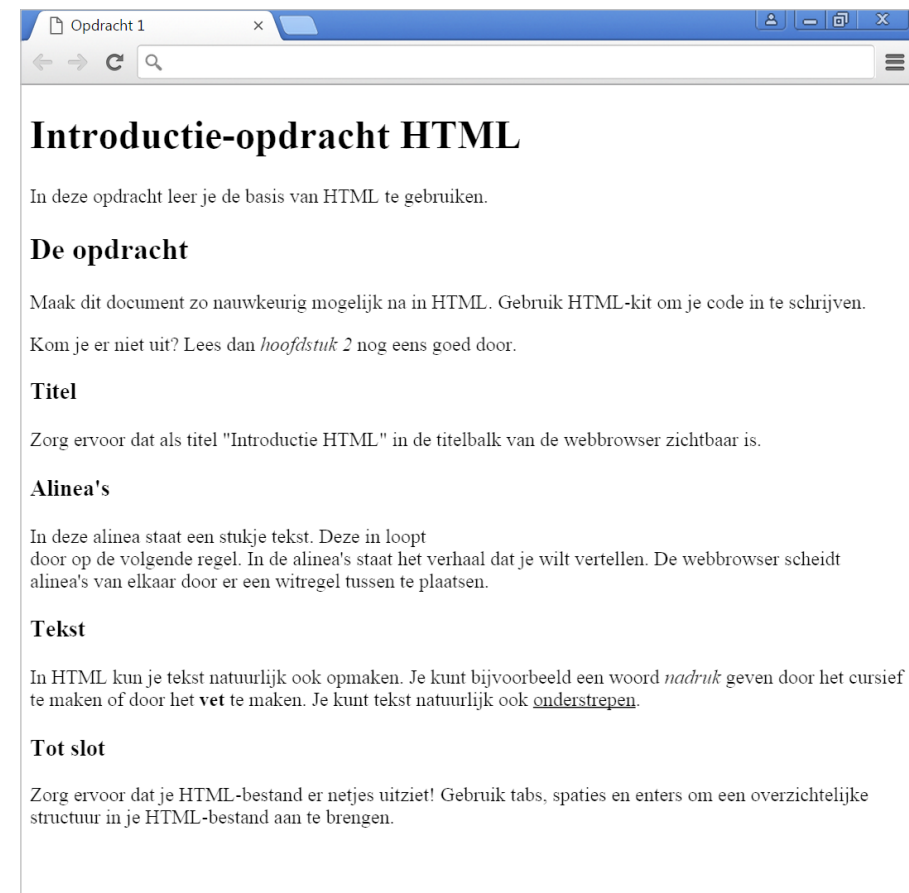
Dit is een commentaarregel, die alleen voor jou als ontwikkelaar zichtbaar is. Je begint een commentaarregel met: `<!--`

Je sluit een commentaarregel af met: `-->`

Je ziet niets van wat je tussen de begin- en eindcode zet, in de webbrowser. Dus gebruik gerust de enter-toets als je je tekst over meerdere regels wilt laten lopen om het overzichtelijk te houden.

2.7 Opdrachten**2.7.1 Opdracht 1: Introductie HTML**

Maak de onderstaande afbeelding zo nauwkeurig mogelijk na in HTML en volg de instructies in de tekst op.



2.7.2 Opdracht 2: Hyperlinks gebruiken

Deze opdracht lijkt op de vorige opdracht, alleen ga je nu ook gebruik maken van hyperlinks. Maak de onderstaande afbeelding zo nauwkeurig mogelijk na in HTML en volg de instructies in de tekst op.

