

HTML: de basis van iedere website

http://www.websitemaken.be/index.php?page=show_item&id=95

Als je nog nooit eerder een website hebt gemaakt, is het een goed idee om te beginnen met deze html-cursus. Ook als je je wat meer wilt verdiepen in HTML (je hebt bijvoorbeeld alleen nog maar met Frontpage gewerkt) is dat een goed plan. HTML is namelijk de absolute basis van een website en iedere webmaster zou eigenlijk verstand van HTML moeten hebben.

In deze cursus leer je de basisprincipes van webdesign. Het is natuurlijk onmogelijk om alles in één cursus te leren, webdesign leer je vooral door het te doen. We zullen je dus alleen de basis leren. De rest zul je langzamerhand zelf ontdekken

De basis van iedere website is de structuurtaal HTML, dat staat voor Hyper Text Markup Language. Met deze taal kun je het uiterlijk van de website in de browser beïnvloeden.

De opbouw van een html bestand

Zoals je waarschijnlijk weet is HTML de basis van iedere website. Deze structuurtaal omschrijft hoe je pagina eruit komt te zien. In deze cursus leer je de basis van HTML. Als je een beetje je best doet kun je binnen 5 minuten je eigen website maken.

HTML 'schrijf' je in een tekstverwerker. Je kunt het programmaatje Notepad of Kladblok, dat standaard bij Windows zit gebruiken, maar wij adviseren om Editpad te gebruiken. Hierin heb je iets meer mogelijkheden.

Je hoeft tijdens het typen geen rekening te houden met allerlei lettertypen of lettergrootten. Het maakt totaal niet uit hoe je iets typt, als je de tekst een andere kleur of opmaak wil geven, doe je dat in HTML en niet door de broncode een ander uiterlijk te geven.

Je eerste 'site'

Aan de hand van de volgende stappen maak je je eerste site.

Open een nieuw document in Editpad of Notepad.

Typ het volgende:

```
<html> hiermee vertel je de browser dat hij met een HTML-bestand te maken heeft)
</html> hiermee geef je het einde van het HTML-bestand aan)
```

De bovenstaande regeltjes noemen we 'tags'. Veel tags hebben ook een eindtag(niet allemaal). De tag met de slash erin geeft altijd het einde van een setje tags aan. In dit geval dus het einde van het HTML-document. De browser weet nu alleen dat hij met een HTML-bestand te maken heeft, daar kan hij natuurlijk geen website van maken.

Het volgende dat we willen is een setje head-tags. Dat geeft letterlijk het hoofd van het bestand aan. Wat er in de head staat wordt wel door de browser herkend, maar je ziet het niet direct terug. Wat je in de head typt blijft dus grotendeels onzichtbaar voor je bezoekers, maar toch is het niet onbelangrijk. Met head-tags ziet het bestand er zo uit:

```
<html>
<head> (hiermee geef je het begin van de head aan)
</head> (hiermee geef je het einde vande head aan)
</html>
```

Maar wat hebben we nou precies aan die head? Eerst zullen we de titel erin zetten. De bezoekers zien die titel in de statusbalk en ook voor zoekmachines is de titel belangrijk.

```
<html>
  <head>
    <title>Welkom op mijn website!</title>
  </head>
</html>
```

Het grootste deel van je website staat tussen de tags `<body>` en `</body>`. Het gedeelte tussen deze tags, de body, is het lichaam van de website. Dit is het resultaat met de body-tags.

```
<html>
  <head>
    <title>Welkom op mijn website!</title>
  </head>
  <body> (het begin van de body)
</body> (het einde van de body)
</html>
```

Het enige wat je nu nog moet doen is je pagina opslaan. Klik op **Opslaan als...** en sla je pagina op als **index.htm** of **index.html**. De extensie htm of html geeft aan dat het een html-bestand is. De meeste homepages (de eerste pagina die je ziet) heten index.htm of index.html. Open nu je pagina in de browser. In de meest gebruikte browser Internet Explorer doe je dat met **Bestand -> Openen --> Bladeren** en blader naar de zojuist gemaakte index.htm.

Je ziet dan... helemaal niks! Alleen de titel staat in de statusbalk, maar omdat er verder niets in de body staat, is er verder niets te zien. Dat gaan we in de volgende les veranderen.

Tekst op je website zetten

We gaan nu tekst op je website zetten. Dat doen we als volgt: Typ tussen de tekst body: Dit is mijn eerste website. Resultaat:

```
<html>
  <head>
    <title>Welkom op mijn website!</title>
  </head>
  <body>
    Dit is mijn eerste website.
  </body>
</html>
```

Sla je pagina nu op en bekijk het resultaat. Het ziet er allemaal nog erg saai uit, maar daar komt later verandering in. Nu we toch met de tekst bezig zijn, leren we je gelijk hoe je het uiterlijk van de tekst kunt veranderen. Eerst zullen we de tekst vet maken. Dat doen we met de tag ``. Deze tag is een *gesloten* tag, je sluit hem dus af met ``.

```
<html>
  <head>
    <title>Welkom op mijn website!</title>
  </head>
  <body>
    <b>Dit is mijn eerste website.</b>
  </body>
</html>
```

Sla op en bekijk het resultaat. Je ziet dat de tekst nu vet staat afgedrukt. Natuurlijk kun je tekst niet alleen vet afdrukken. Ook cursief is mogelijk, met de tag `<i>` In plaats van de b zet je nu dus een i (talic) neer. Dat doe je ook bij de afsluittag. Ook kun je de tekst onderstrepen door in plaats van b (old) een u (nderscore) neer te zetten. Door een s neer te zetten wordt de tekst doorgestreept. Dit waren de belangrijkste tekstopmaak-tags, meer heb je nog even niet nodig als beginner.

Het is handig om, bij het begin van een nieuwe alinea (paragraph in het Engels) een <p>-tag neer te zetten. De browser laat dan een witregel open tussen die tekst en de volgende alinea. Je mag de <p>-tag afsluiten, het hoeft niet. Het begin van een nieuwe alinea is logischerwijs het einde van de vorige, dat snapt zelfs je browser.

Wil je je tekst centreren? Doe dat dan als volgt:

```
<html>
  <head>
    <title>Welkom op mijn website!</title>
  </head>
  <body>
    <p align="center"><b>Dit is mijn eerste website.</b> </p>
  </body>
</html>
```

Je ziet dat de tag <p> nog meer info bevat. align="center" geeft aan dat de tekst in het midden uitgelijnd moet worden. Zo'n extra stuk info in een tag noemen we voortaan een 'attribuut', maar dat terzijde. Andere manieren van uitlijnen zijn left en right. Wil je links uitlijnen, dan hoef je left niet per se te specificeren, omdat de browser automatisch links uitlijnt als er verder niets aan staat gegeven.

Koppen

In HTML kun je ook verschillende kopjes gebruiken. Dat doe je door de <h>-tags. Alleen <h> is niet genoeg, er moet ook een getal van 1 t/m 6 achter. 1 geeft de grootste kop, 6 de kleinste. Ook de <h>-tags worden bij voorkeur wel afgesloten met </h>.

Tag	Resultaat
<h1>	Grootste kop
<h2>	Grote kop
<h3>	Gewone kop
<h4>	<u>Kleinere kop</u>
<h5>	<i>Kleine kop</i>
<h6>	Noem je dit nog een kop?

Kleuren op je website

Wat kunnen we verder nog met de tekst doen? Een kleurtje zou niet verkeerd zijn. Kleuren definiëren in HTML doe je met zogenaamde hexadecimale codes. Wil je de code bij een kleur weten? Gebruik dan RGB-Hex. De meest gebruikte kleuren vind je hieronder.

Code	Kleur
#000000	Zwart
#FFFFFF	Wit
#FF0000	Rood
#00FF00	Groen
#0000FF	Blauw
#FFFF00	Geel

#00FFFF

Lichtblauw(cyaan)

#FF00FF

Paars(magenta)

Dit zijn enkele veelgebruikte waarden. Er zijn nog veel meer variaties mogelijk en zo'n beetje iedere kleur is dan ook wel te krijgen in de HTML-code.

Je kunt dus rustig gaan experimenteren met de verschillende codes. In plaats van de nullen kun je ook andere cijfers van 1 tot en met 9 invullen.

Zet altijd een hekje voor de kleur, de browser herkent de kleur anders niet! Eerst willen we de achtergrond van de website zwart maken. Omdat de hele site zwart moet worden, is het logisch dat je dit in de body doet. Je HTML-code voor de body wordt dan:

```
<body bgcolor="#000000">
  <p align="center">Dit is mijn eerste website.</p>
</body>
```

Zo, die achtergrond is zwart! Nu moet de tekst natuurlijk nog een andere kleur krijgen, want een zwarte tekst op een zwarte achtergrond werkt niet echt. Wil je alle tekst op de site bijvoorbeeld rood maken, dan doe je dat in de <body>-tag, het geldt immers voor het hele document. We geven dus in de body aan dat alle tekst in het document de kleur #ff0000(rood) moet krijgen. Dat doen we zo:

```
<html>
  <head>
    <title>Welkom op mijn website!</title>
  </head>
  <body bgcolor="#000000"
    text="ff0000">
    <p align="center">Dit is mijn eerste website.</p>
  </body>
</html>
```

Let op: de </p>sluitingstag hoeft er dus niet te staan! Het mag wel. Zo, we kunnen de tekst nu een kleur geven. Maar wat als we nu een regel op de site groen willen hebben en de andere regel weer rood. Dit kunnen we niet in de body aangeven, want het geldt niet voor alle tekst. Het moet dus per zin aangegeven worden. Wil je dit bereiken...

Welkom op mijn website!

Deze regel is rood.

...dan gebruik je deze HTML-code:

```
<html>
  <head>
    <title>Welkom op mijn website!</title>
  </head>
  <body bgcolor="#000000">
    <p><font color="#00ff00">Welkom op mijn website!</font>
    <p><font color="#ff0000">Deze regel is rood</font>
  </body>
</html>
```

Meer informatie over het instellen van de achtergrond van je website(zoals afbeeldingen) vind je [hier](#).

Lettertypen

De grootte van de tekst bepaal je met het attribuut "size". Dit gebruik je in het element . De grootte kan een waarde van 1 t/m 7 aannemen. Verder kun je de grootte zowel absoluut als relatief bepalen. Een voorbeeld van een absolute grootte is:

```
<font size="1">Dit is lettergrootte één</font>.
```

Een relatieve grootte geeft de grootte ten opzichte van de standaardgrootte op. Een voorbeeld van een relatieve lettergrootte is:

```
<font size="-1">Deze tekst wordt één niveau kleiner dan het  
standaardlettertype</font>.
```

Met de relatieve grootte geef je dus de verhouding tot het standaardlettertype op. Om je website grafisch wat aantrekkelijker te maken, is het ook mogelijk om van verschillende lettertypen gebruik te maken. Net als met het typen van gewone Word-bestanden heb je erg veel keuze, uit bijvoorbeeld Arial, Times New Roman, Verdana etc.

Het lettertypen geef je op met het attribuut "face". Wil je een een tekst bijvoorbeeld in *arial* hebben, dan zet je dit neer:

```
<p><font face="arial">Deze tekst staat in Arial</font>
```

Natuurlijk kun je alle tags ook combineren, dus bijvoorbeeld een rode tekst, lettertype arial, gecentreerd en in lettergrootte 3. De tag ziet er dan zo uit:

```
<p align="center"><font face="arial" color="#ff0000"  
size="3">Dit is de eigenlijke tekst</font></p>
```

Wanneer je op de hele pagina dezelfde kleur tekst? Dan kun je dat al in het body-element opgeven. Je hoeft dan niet meer bij ieder stuk tekst aan te geven in welke kleur je de tekst wilt zien. Dit plaats je in de body:

```
<body bgcolor="ffffff"  
text="#ff0000">
```

Zo krijg je dus een witte achtergrond en een rode tekst.

Hyperlinks maken

Een van de dingen die het web zo handig maken zijn de *hyperlinks*. Hyperlinks zorgen ervoor dat je met één muisklik naar een andere locatie kunt 'surfen'. Die andere pagina kan zich op dezelfde server bevinden, maar ook aan de andere kant van de wereld. Je kunt zowel linken naar html-bestanden als mp3-bestanden, software, afbeeldingen enzovoorts.

Hoe maak ik zo'n hyperlink? We gebruiken hiervoor de tag <a>, van *anchor*. Deze tag op zich zegt niet veel. Een belangrijk attribuut in deze tag is "href". Deze tag vertelt de browser welk document er geopend moet worden. Neem je bijvoorbeeld een link op naar pagina.htm, dan ziet de broncode er zo uit:

```
<a href="pagina.htm">Een link naar het bestand pagina.htm</a>
```

Zet het bovenstaande in de body van je pagina. Sla de pagina op en open hem in je browser. Wat gebeurt er nu als je op de hyperlink klikt? Je krijgt een foutmelding! Logisch natuurlijk, want het bestand pagina.htm bestaat nog niet. Maak daarom eerst een pagina die je pagina.htm noemt:

```
<html>
  <head>
    <title>Dit
is dus de pagina waar je naartoe linkt</title>
  </head>
  <body bgcolor="#000000" text="ffffff">
    <p>Dit is mijn
eerste website. Met <a href="index.htm">deze link </a>
ga je weer terug naar de vorige pagina
  </body>
</html>
```

Je kunt nu dus heen en weer surfen tussen de twee pagina's. Je begrijpt dat we hier heel veel mee kunnen doen.

Zoals je al gezien hebt, is datgene wat tussen `` en `` staat, de tekst die uiteindelijk als link op je site verschijnt. Vaak komt er dan een streepje onder en verandert die tekst van kleur.

Relatief en absoluut

Wat je hierboven ziet is een *relatieve* link. Nog een voorbeeld van een relatieve link is

```
<a href="mijnpagina.htm">Naar mijn pagina</a>
```

Relatieve links gebruik je om te linken naar pagina's binnen jouw eigen website.

Verder zijn er ook nog absolute links. Daarmee geef je een link aan die niet per se binnen jouw web valt. Link je bijvoorbeeld naar deze pagina, dan doe je dat zo:

```
<a href="http://www.websitemaken.be">Naar Websitemaken</a>
```

Verder kun je links opnemen naar bijvoorbeeld een e-mailadres. Klik je op zo'n hyperlink, dan opent de browser een nieuw e-mailbericht. In het veld waar het adres moet komen staat dan al het adres dat jij hebt opgegeven:

```
<a href="mailto:mijnadres@mijnprovider.nl">Stuur mij
een e-mail!</a>
```

Tenslotte kun je ook het onderwerp wat in de subject-regel moet komen te staan al aangeven in je link. Dat doe je met 'subject':

```
<a href="mailto:mijnadres@mijnprovider.nl ?subject=Website">Stuur mij een e-mail!</a>
```

Tabellen

Bij tabellen denk je waarschijnlijk aan saaie overzichten van cijfers en andere informatie. Dan heb je echt een verkeerd beeld van wat je allemaal met tabellen kunt doen. Er bestaan maar weinig websites die geen gebruik maken van tabellen. Probeer het onderstaande maar eens te bereiken zonder tabellen(en css, maar dat komt later).

Probeer
dit
maar

eens
zonder
tabellen

Dat wordt moeilijk inderdaad. Hierboven zie je dus een tabel. Dat is op de volgende manier duidelijker:

Probeer	eens
dit	zonder
maar	tabellen

Het zal je duidelijk worden dat je met deze tabellen veel kunt doen. Zo kun je bijvoorbeeld tekst in twee kolommen zetten:

Dit is de eerste kolom van de tekst. Deze tekst gaat eigenlijk nergens over, maar laat je wel zien dat je met tabellen...

...tekst in **twee** kolommen kunt zetten. Natuurlijk is het mogelijk om de tekst in 3, 4 of meer kolommen te zetten!

Zo zijn er nog veel meer belangrijke toepassingen van tabellen in een webpagina. Voor een goede webpagina zijn tabellen bijna onmisbaar. We gaan je nu leren hoe je zo'n tabel maakt. Zoals je weet bestaat een tabel uit rijen en kolommen. Hieronder zie je een tabel met 5 rijen en 2 kolommen:

Dit is een kolom met 2 rijen en 5 kolommen:

Nu willen we een tabel opnemen in onze html-code. Het begin van een tabel markeren we met de tag **<table>**. We eindigen een tabel altijd met de sluittag **</table>**. Maar met alleen die twee codes kan de browser niets. We moeten opgeven hoeveel rijen en kolommen er in moeten komen. Het begin van een nieuwe rij geef je aan met de tag **<tr>**, van **Table Row**(tabel rij). Je geeft hiermee aan dat je de tabelrij begint. Het einde van een rij geef je natuurlijk aan met **</tr>**.

Zo een rij hebben we, maar ook hier kan de browser niet veel mee. Een rij zonder kolommen bestaat namelijk niet, iedere rij heeft minstens 1 kolom. Onze tabel krijgt 2 kolommen. Het begin van een kolom geef je aan met de tag **<td>**, van **Table Data**. Deze tag sluit je natuurlijk af met **</td>**. De broncode van onze tabel met 2 rijen en 2 kolommen ziet er zo uit:

```
<table>
  <tr>
    <td>Inhoud van kolom 1 in rij 1</td>
    <td>Inhoud van kolom 2 in rij 1</td>
  </tr>
  <tr>
    <td>Inhoud van kolom 1 in rij 2</td>
    <td>Inhoud van kolom 2 in rij 2</td>
  </tr>
</table>
```

Het resultaat ziet er zo uit:

Inhoud van kolom 1 in rij 1
Inhoud van kolom 1 in rij 2

Inhoud van kolom 2 in rij 1
Inhoud van kolom 2 in rij 2

Dit stelt nog niet veel voor, maar je kunt nu wel een tabel maken. Als je het bovenstaande probeert en je laadt de pagina in de browser, valt het je op dat de breedte van de tabel niet overeenkomt met wat je hierboven ziet. Daarom geef je in de tag **<table>** de breedte op. Dat doe je met het attribuut **width**. We raden je aan om de breedten in procenten op te geven. Een tabel die de 90% van de breedte moet bedekken ziet er zo uit:

```
<table width="90%">
De rest van de tabel
</table>
```

Om de breedte van de kolommen geef je op met die tag. Een tabel die 90% van de pagina bedekt en kolommen bevat die allebei de helft van de tabel(dus 45% van de pagina) innemen, komt er zo uit te zien:

```
<table width="90%">
  <tr>
    <td width="45%">Een
tabel van 90%</td>
    <td width="45%">met een celbreedte
van 45%</td>
  </tr>
</table>
```

Behalve de breedte kun je ook de hoogte van de tabel en de cellen opgeven. Dat doe je met het attribuut `height="hoogte"`. Dit kan ook weer in pixels of procenten.

Een ander belangrijk attribuut is "**border**". Dit geef je in de tag `<table>` aan. Zoals je waarschijnlijk al verwacht geeft het de dikte van de rand van je tabel aan. Wil je een onzichtbare rand, dan zet je deze op 0.

Verder kennen we nog de attributen `cellpadding` en `cellspacing`. `Cellspacing` vertelt de browser de ruimte tussen de twee cellen. Kijk hieronder maar.

Eerst zie je een tabel met dikte 1 en cellspacing 1.	

en deze tabel heeft dikte 1 en cellspacing 5	

Ook is er nog het attribuut `cellpadding`. Dat maakt ruimte vrij tussen de tekst in de cel en de rand. Kijk maar naar de verschillen:

Deze cellpadding is 1	

en deze cellpadding is 15	

Zo, dat verschil is duidelijk. Verder kun je nog kleuren aan je tabel toevoegen. Dat kan op 2 manieren. Je kunt de border een kleur geven, met de tag `bordercolor="#3366cc"`, bijvoorbeeld. Een cel kan ook een achtergrondkleur krijgen, dat doen we met het attribuut `bgcolor="#000080"`. Je kunt zowel per cel(dat doe je in de `<td>` tag) of per tabel(in de `<table>` tag) de achtergrondkleur opgeven

Het is ook mogelijk om het volgende resultaat te krijgen:

Cel 1	Cel 2
	Cel 3

Je ziet hier een tabel met 2 rijen. De eerste rij bestaat uit 1 cel, de tweede rij bestaat uit 2 cellen. Met de tags die je al kent was je dit niet gelukt. Daarom leren we je twee nieuwe tags: `colspan` en `rowspan`.

```

<table width="90%" border="1">
<tr>
<td rowspan="2">Cel 1</td>
<td>Cel 2</td>
</tr>
<tr><td>Cel 3</td>
</tr>
</table>

```

Je geeft zo dus aan dat de eerste <td> tag 2 <tr> tags om **spant**. Je kunt zoveel rows opspannen als je wilt. Ook verticaal is dit mogelijk:

Cel 1	
Cel 2	Cel 3

De broncode voor deze tabel ziet er zo uit:

```

<table width="90%" border="1" bordercolor="#000000"
cellspacing="0" cellpadding="0">
<tr>
<td colspan="2">Cel 1</td>
</tr>
<tr>
<td>Cel 2</td>
<td>Cel 3</td>
</tr>
</table>

```

Frames

Veel website bevatten frames. Je hebt er ongetwijfeld zelf gebruik van gemaakt, maar waarschijnlijk wist je toen nog niet dat het frames waren. Daarom leggen we hier eerst maar even uit wat frames zijn en...dat is best lastig.

Stel je voor: je hebt een pagina, met links een soort menu. Dit menu bestaat bijvoorbeeld uit home, wie ben ik, links, e-mail me, enzovoorts. Als de browser de pagina opent, begin je in het rechtervlak in home. Klik je vervolgens op wie ben ik, dan verschijnt rechts een pagina met info over de maker, en links blijft de inhoud staan. Even een voorbeeld om dat te verduidelijken:

home wie ben ik links e-mail	<p>Welkom op mijn website. Je vindt hier onder andere informatie over mijzelf, links, andere onzin etc....</p> <p>Je bevindt je nu op de pagina genaamd home</p>
---	--

Zo, dat is dus 'home'. Als de bezoeker nu op 'wie ben ik' zou klikken, ziet hij het volgende:

home wie ben ik links e-mail	<p>Wie ben ik? Ik ben de maker van deze website, maar dat wist je al. Mijn hobby's zijn:.....enzovoorts</p> <p>Je bevindt je nu op de pagina genaamd wie ben ik</p>
---	---

Wat gebeurt er dus? De linkerkant blijft hetzelfde, de rechterkant verandert door in het linkervlak op een hyperlink te klikken. Volg je het nog? Vast wel. Je ziet dat het gebruik van frames erg veel tijd kan sparen: ten eerste hoef je de inhoudsopgave maar een keer maken, maar belangrijker is misschien wel dat de bezoeker deze maar een keer in z'n browser hoeft te laden. Dat gaat dus weer sneller!

De pagina hierboven bestaat uit 3 verschillende pagina's:

- ◆ De eerste is de framepagina, in dit geval index.htm. Deze deelt het blad op in 2 stukken.
- ◆ De tweede is de pagina met daarop de inhoud, hier is dat inhoud.htm.
- ◆ Als laatste is er nog home.htm. Er zijn er nog meer, die worden allemaal in het rechtervenster geopend.

In plaats van home.htm kun je natuurlijk ook een bestaande pagina gebruiken. De inhoud.htm kun zelf vast wel maken, zoniet, volg deze cursus dan even opnieuw. Het gaat ons nu even om index.htm, de pagina die het venster in tweeën deelt en zo van 2 verschillende pagina's 1 pagina maakt. **Split-screen**, om het zo maar even te noemen.

We hebben hier te maken met een merkwaardig html-bestand. Ten eerste is er geen echte body. De pagina zelf laat niets zien, dus dat is ook niet nodig. Wel heeft hij een frameset. dat ziet er zo uit:

```
<html>
<head>
inhoud van je head, zoals titel en metatags
</head>
<frameset cols="35%,65%">
<frame src="inhoud.htm">
<frame src="home.htm">
</frameset>
</html>
```

Wat gebeurt er hierboven nou precies? Met frameset zeg je de browser dat er een framepagina aankomt. Vervolgens geef je op dat hij met kolommen(**cols**) te maken heeft. Je ziet staan: 35%,65%. Dit wil zeggen dat het linkerframe 35% van het venster vult, en het rechterframe krijgt 65%. Totaal is dat...inderdaad: 100%, dus het hele venster. De browser ziet dat jij maar 2 afmetingen hebt opgegeven, hieruit leidt hij af dat er maar 2 kolommen(pagina's) zijn in het frameset, en dat klopt. Daarnaast zie je 2 keer <frame src="paginanaam.htm"> staan. Deze tags vertellen de browser welke site er in de frames moeten komen. Behalve dat je de grootte in procenten op kunt geven, is het ook mogelijk om dat in pixels te doen. Je zet dan gewoon de grootte neer zonder het %-teken.

Wil je dat de kolommen gelijk over de pagina verdeelt worden, maak dan gebruik van het *-teken. Dat ziet er dan zo uit:

```
<frameset cols="*,*">
rest van je frame
</frameset>
```

In plaats van twee sterretjes, kun je ook de breedte van de ene pagina wel opgeven en de ander niet. Dan krijgt de ene pagina dat gedeelte wat je hem toezegt, terwijl de ander de overige ruimte krijgt.

Hierboven gebruikten we een voorbeeld met kolommen. De pagina werd hier horizontaal gesplitst. Als je wil kun je de splitsing ook verticaal maken. Je maakt dan geen gebruik van **cols** maar van **rows**. Verder is alles gelijk.

Het is niet alleen mogelijk om framepagina's bestaande uit twee frames te maken, je kunt er zo veel je wilt opgeven. Dat doe je door in plaats van 2 grootten in je frameset aan te geven, er bijvoorbeeld drie aan te geven. Ook is het goed mogelijk om kolommen(**cols**) en rijen(**rows**) te combineren. Dat kan er zo uit komen te zien:

Inhoud	Bovenkant
	Home

Dat doe je met deze html-code:

```
<frameset cols="20%,80%">
je geeft aan dat het eerste frameset uit twee delen bestaat, een van 20%(links)
en een van 80%(rechts)
<frame src="inhoud.htm">
de naam van de pagina in het linkervenster
<frameset rows="20%,80%">
het rechtervenster bestaat uit twee rijen, een van 20%(boven) en een van
80%(onder)
<frame src="kop.htm">
<frame src="home.htm">
de namen van de twee pagina's die in het rechtervenster worden geopend
</frameset>
</frameset>
omdat je twee framesets opent, sluit je er ook twee af!
```

Meer informatie over frames

Je kunt nu een aardige framespagina in elkaar draaien, maar er zijn toch nog een aantal erg belangrijke dingen die je zeker moet weten als je frames op je website gaat gebruiken. Een aantal van die dingen zullen we hieronder bespreken.

Je moet weten dat er nog steeds mensen zijn die wat oudere browsers gebruiken. Het zou goed kunnen dat die browsers nog geen gebruik maken van frames. Natuurlijk wil je ook deze mensen bereiken. Je wilt ze bijvoorbeeld zeggen dat je een framespagina gebruikt en dat daarom je website voor hen niet toegankelijk is. Omdat jouw index.htm een framespagina is, zouden zij niets te zien krijgen. Toch kunnen we ook een pagina op hun scherm tevoorschijn toveren. Dat doen we gewoon met de tag <body> en een nieuwe tag, de tag <noframes>.

```
<html>
<head>
inhoud van je head, zoals titel en metatags
</head>
<frameset cols="35%,65%">
  <frame src="inhoud.htm">
  <frame src="home.htm">

<noframes>
<body>
  <p>Helaas, uw browser ondersteunt geen frames. Ga naar de website
van <a href="http://microsoft.com">Microsoft</a>
om een nieuwe versie van Internet Explorer te downloaden</p>
</body>
</noframes>
</frameset>
```

Let op: Tegenwoordig surft bijna iedereen met een browser die frames ondersteunt. Het is dan ook niet echt meer nodig om voor die mensen een frameset te maken. Toch is de frameset erg belangrijk, en wel voor zoekmachines! Als een zoekmachine, bijvoorbeeld Google jouw pagina wilt opnemen, zal deze op zoek gaan naar waar jouw site over gaat. In een framespagina kan hij dat echter niet vinden! Neem daarom een beschrijving van je site op in het <noframes> gedeelte, zodat je deze informatie alsnog aan de zoekmachine kunt geven!

Let op: er zijn sites/mensen die beweren dat de <noframes>tag na de </frameset> moet komen. Dat is niet waar!!!

Er zijn echter nog een paar probleempjes: Ten eerste die lelijke rand tussen de verschillende frames. In veel gevallen zul je die weg willen hebben. Dat doen we op de volgende manier: In de tag `<frameset>` geef je aan dat je geen rand wilt. Dat kan op twee manieren. De ene browser accepteert het als je zegt `border="0"` en de ander wil zien staan `frameborder="0"`. Dan doen we het toch gewoon allebei:

```
<frameset row="60%,40%" border="0" frameborder="0">
```

Verder kunnen bezoekers als ze op de rand van twee frames gaan staan, zelf schuiven met de frames. Ze kunnen dus de grootte veranderen. Wil je dat voorkomen, dan zet je in de frameset-tag het volgende:

```
<frameset row="60%,40%" noresize>
```

Het laatste wat je moet kennen is het attribuut "target". Stel je voor, in de onderstaande pagina wil je, dat als iemand op de link Verhalen klikt, het bestand verhalen.htm geopend wordt in het venster rechtsonder. Hoe vertel je de browser dat verhalen.htm daar geopend moet worden en niet in die pagina er boven?

Dat doen we zo: eerst geven we de verschillende frames een naam. Het frame rechtsonder noem je bijvoorbeeld rechtsonder. In onze html-tag ziet dat er zo uit:

```
<frameset cols="80,*">
  <frame src="inhoud.htm"
name="inhoud">
  <frame src="home.htm" name="main">
</frameset>
```

Zo, de frames hebben een naam. Stel, je wilt dat een hyperlink in pagina 'inhoud.htm' wordt geopend in het **inhoud-frame**. De hyperlink verwijst naar het bestand **verhalen.htm**. Om dit alles goed te laten verlopen, moeten we iets veranderen in de tag van de hyperlink. Dat doen we met het attribuut **target**. De hyperlink-tag ziet er nu zo uit:

```
<a href="verhalen.htm" target="inhoud">Verhalen</a>
```

Er zijn een aantal namen voor frames die 'gereserveerd' zijn. Al deze namen beginnen met een underscore, een liggend streepje: _

Naam	Resultaat
<code>_self</code>	Pagina wordt geopend in hetzelfde frame als de pagina van de hyperlink
<code>_blank</code>	Pagina wordt geopend in een nieuw browservenster
<code>_top</code>	Pagina wordt in hetzelfde browservenster geopend en vult het hele browservenster
<code>_parent</code>	Huidige frameset wordt vervangen door het frameset uit de nieuw te openen pagina

Als je op een pagina veel links hebt die bijna allemaal hetzelfde targetframe hebben, zet dan tussen `<head>` en `</head>` het volgende:

```
<base target="inhoud">
```

Gebruik je deze tag, dan hoef je niet meer per hyperlink aan te geven wat het **targetframe** is. Je hoeft niet bang te zijn dat je nu geen hyperlink kunt openen in een ander targetframe. Als je in de hyperlink een ander targetframe aan geeft dan het **base target**, dan krijgt het target dat je in de hyperlink aangeeft voorrang.

Lijsten maken

Het maken van lijsten is misschien wel een van de makkelijkste dingen in html, maar je moet het maar net even weten natuurlijk. Eerst zullen we je even laten zien wat een lijst precies is:

```
Punt 1
Punt 2
Punt 3
```

Dit noemen we een ongeordende lijst. Hier volgt een geordende lijst:

```
1. Punt 1
2. Punt 2
3. Punt 3
```

De ongeordende lijst maken we zo: We beginnen met ``. Dit betekent *unordered list*. De items uit de lijst geven we aan met ``, van *list item*. Deze moet je, net als de ``-tag weer beëindigen. Ons lijstje uit het voorbeeld komt er in html dus zo uit te zien:

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

Wil je een geordende lijst hebben, dan hoef je alleen maar `` om te zetten in ``, wat natuurlijk staat voor *ordered list*. Dan komt ons lijstje er dus zo uit te zien:

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

Ook is het natuurlijk mogelijk om een lijst in een lijst op te nemen, zo bijvoorbeeld:

```
1. Punt 1
2. Punt 2
   O Punt A
   O Punt B
   O Punt C
3. Punt 3
```

In HTML ziet dat er zo uit:

```
<ol>
<li>Punt 1</li>
<li>Punt 2</li>
  <ul>
<li>Punt A</li>
<li>Punt B</li>
  <li>Punt C</li>
</ul>
<li>Punt 3</li>
</ol>
```

Zoals we al vermeldden, is werken met lijsten erg simpel, maar toch onmisbaar wil je html goed kennen.

I-frames

Als je de cursus al helemaal gevolgd heb, weet je al dat je met HTML zogenaamde frames kunt maken. Een soort pagina in een pagina, als het ware. Erg handig, maar sommige dingen zijn wat lastiger met frames. Wil je dit maken:



...dan is een frame erg handig. Maar dit is al lastiger:



Gelukkig is hier ook iets op bedacht, het zogenaamde i-frame. Eerst maar even een voorbeeld:

Best handig, zo'n i-frame. We gebruiken er de `<iframe>` tag voor, die we natuurlijk afsluiten met `</iframe>`. Hierbinnen geven we de eigenschappen van het iframe op. De eerste is de bron van het iframe. Wil je bijvoorbeeld www.zomaareensite.be inframesen, dan doe je dat zo:

```
<iframe src="http://www.zomaareensite.be"></iframe>
```

Verder kun je de grootte opgeven. Wil je, zoals hierboven, een frame van 500 breed en 60 hoog, dan doen we:

```
<iframe src="http://www.zomaareensite.be" height="60"
width="500"></iframe>
```

Dan is er nog de rand. Want net zoals bij een 'gewoon' frame kunnen we ook hier de randen van het frame opgeven. Dit gebeurt gewoon weer met `border="x"`

```
<iframe src="http://www.zomaareensite.be" height="60"
width="500" border="3" frameborder="3"></iframe>
```

Vervolgens kun je opgeven of er gescrollt mag worden of niet. Wanneer je een iframe van 200 hoog maakt, en je laadt daarin een pagina van 250 hoog, zijn er schuifbalken nodig. Die krijgen we met `scroll="yes"`. Wil je ze liever niet? dan wordt die yes natuurlijk een no. Dit geldt natuurlijk ook voor scrollen in de breedte.

```
<iframe src="http://www.zomaareensite.be" height="60"
width="500" border="3" frameborder="3" scrolling="yes"></iframe>
```

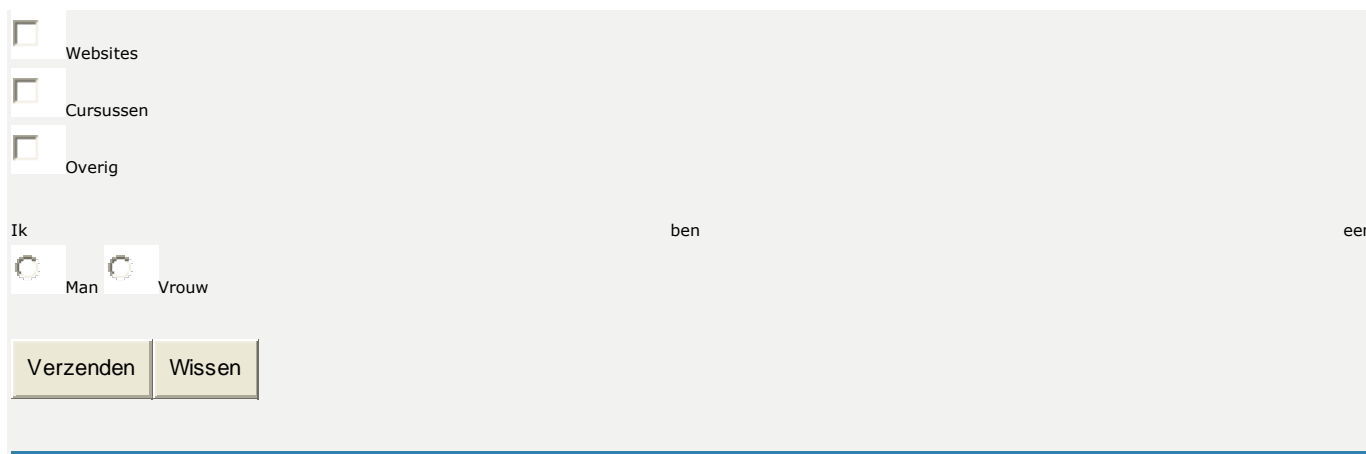
Formulieren maken

Iedereen die regelmatig op internet surft, kent ze: formulieren. Je vult een aantal gegevens in, en dit wordt, nadat je op een knop drukt, verzonden naar de mensen achter die site. Zo'n formulier is best handig. Op deze site vind je er ook eentje: klik maar eens op 'add url'. Na het invullen van dit formulier, worden de gegevens naar ons verzonden. In dit deel van de cursus leren we je zelf zo'n formulier te maken en te gebruiken.

Er zijn grofweg twee typen formulierverwerking:

- ◆ Server-based: het formulier wordt verzonden via een script op de server
- ◆ Client-based: het formulier wordt verzonden via e-mail.

Wij zullen ons gaan bezighouden met de client-based formulerverking. Dit is een eenvoudige manier om gegevens van je bezoekers te krijgen. We gaan je in deze les leren hoe je het volgende formulier moet maken:



The image shows a simple web form. It has three checkboxes labeled 'Websites', 'Cursussen', and 'Overig'. Below these is a text input field with 'Ik' on the left and 'een' on the right. Underneath the text field are two radio buttons labeled 'Man' and 'Vrouw'. At the bottom of the form are two buttons: 'Verzenden' and 'Wissen'.

Daar gaan we...nu jij!

Dit ziet er behoorlijk ingewikkeld uit, maar het valt allemaal best mee. Ten eerste goed nieuws: een formulier bestaat uit puur html, dus nog even geen andere talen zoals javascript of php. Je leert hier dus gewoon weer iets meer van html, er is dus niks 'engs' aan een formulier. Het is geen speciale pagina, maar gewoon een deel van een pagina. Je kunt zoveel formulieren op een pagina zetten als je wilt. Ook is het mogelijk om binnen het formulier allerlei andere html-tags op te nemen, zoals tabellen, afbeeldingen en tekst.

We beginnen bij het begin: een formulier staat tussen de tags `<form>` en `</form>`. Deze twee tags geven het begin en het einde van een formulier aan. In deze tags moet je ook aangeven wat je met het formulier willen doen. In ons geval willen we de gegevens per e-mail opgevoerd krijgen. Dat doen we zo:

```
<form method="post" action="mailto:jouwemail@jeprovider.nl">  
de rest van het formulier(dat leer je straks)  
</form>
```

Invoerelementen

Het eerste element dat we je leren is een invoerveld voor tekst, zoals je dat ook hierboven in het formulier als eerste tag ziet. Eerst willen we natuurlijk aangeven wat er in dit tekstvlak moet komen te staan, de naam in dit geval. Dit gebeurt gewoon met de bekende `<p>`-tag. Daarna geven we aan dat er een tekstvlak moet komen:

```
<form method="post" action="mailto:jouwemail@jeprovider.nl">  
  
<p>naam:<input type="text" size="30" maxlength="50"  
name="naam">  
</form>
```

Je ziet na het tekstvlak een paar belangrijke attributen staan. We bespreken die eerst even:

type="text"

Hiermee geef je aan dat het een tekstvlak is.

size="30"

Hiermee geef je aan dat de lengte van het tekstvlak 30 tekens is.

maxlength="50"

De bezoeker kan maximaal 50 tekens invoeren. Dit aantal hoeft niet gelijk te zijn aan de waarde van het attribuut "size", 30 in dit geval.

name="naam"

Je geeft het tekstvlak een naam, zodat je het bij de verwerking kunt herkennen.

Dat was het tekstvlak. Het tweede formulierelement dat we tegenkomen is het tekstvlak **e-mail**. dit werkt precies hetzelfde als bij het zojuist behandelde tekstvlak naam. In plaats van **name="naam"** zou je hier kunnen kiezen voor **name="email"**.

We gaan verder in het formulier en komen zogenaamde checkboxen tegen. Je ziet een lijstje, en je kunt aanvinken wat je wilt. Er zijn meerdere opties mogelijk. Die checkboxen maak je zo:

```
<input type="checkbox" name="websites">
<input type="checkbox" name="cursussen">
  <input
type="checkbox" name="overig">
```

Ook dit is weer behoorlijk simpel, hier hoeven we verder niets aan uit te leggen.

Dan zie je nog twee cirkeltjes. In tegenstelling tot de checkboxen kun je er hier maar eentje selecteren. Logisch, want je bent man of vrouw, terwijl je bijvoorbeeld wel over websites én cursussen info aan kunt vragen. Als je er slechts één kunt selecteren heet het geen **checkbox** maar een **radiobutton**. Alles werkt hetzelfde als bij de checkboxen, alleen in plaats van `<input type="checkbox">` zeggen we nu natuurlijk `<input type="radio">`, de overige dingen, zoals name, blijven gelden (natuurlijk geef je wel een andere name op)

We zijn er bijna, het laatste invoerelement van ons formulier is een selectielijst. Dit werkt iets anders dan je gewend bent, en wel zo:

```
<select>
<option>Nederland</option>
  <option>België</option>
  <option>Duitsland</option>
  <option>Frankrijk</option>
</select>
```

Verder kun je hier nog in het element `<select>` de **size** invoeren. Voer je die niet in, dan krijg je een selectielijst. Zet je hem bijvoorbeeld op 3, `<select name="land" size="3">` dan is het geen dropdown menu, maar een schuifmenuutje.

Dan zie je verder nog een groot leeg veld waar de bezoeker teksten kan invoeren. Dat noemen we een **textarea**. Je kunt bij een textarea de hoogte opgeven in rows en de breedte in cols. Verder geef je hem een naam met "name", bijvoorbeeld de naam jouwopmerkingen. Ons veld ziet er dus zo uit in de html:

```
<textarea name="jouwopmerkingen"
rows="5" cols="30">
</textarea>
```

We zijn zover: het kan verzonden worden. Daarvoor hebben we twee knoppen: Verzenden (**submit**) en Wissen (**reset**). De betekenis hiervan snap je natuurlijk wel. De broncode hiervan ziet er zo uit.

```
<input type="Submit" name="verzenden" value="Verzenden">
<input type="Reset" name="wissen" value="Wissen">
```

De Value mag je zelf bedenken. Dit zijn natuurlijk de meest logische, maar in principe kun je erin zetten wat je wilt. De value is de tekst die op de knop verschijnt.

Verder nog één tip: het beste kun je ook het volgende nog in de begintag <form> opnemen:

```
enctype="text/plain"
```

Dan wordt het totaal dus:

```
<form name="form1" method="post" action="mailto:r.haasteren@planet.nl"
enctype="text/plain">
```

Het is een kleinigheidje, maar het zorgt er wel voor dat het een beetje fatsoenlijk leesbaar op je mail binnenkomt.

Metatags

Metatags zijn een speciaal soort tags: ze zorgen niet voor een zichtbaar resultaat op de pagina zelf, maar vertellen dingen over de pagina. Een voorbeeld daarvan is de keywords-metatag: hierin geef je sleutelwoorden over jouw site op. Vroeger werd deze tag vooral gebruikt door zoekmachines om te kijken naar waar jouw pagina over ging. Inmiddels is die tag zo vaak misbruikt dat zoekmachines er niet eens meer naar kijken.

Een lijst met andere metatags vind je onderaan deze pagina. Let wel: lang niet alle metatags zijn nodig, sommigen zijn zelfs erg nutteloos. Toch zijn er ook nuttige invullingen van deze tag.

```
<META NAME="keywords" CONTENT="de trefwoorden die jouw site beschrijven">
```

Er zijn nog veel meer metatags. Hier volgen er enkele.

meta-tag	Betekenis
meta name="author" content="naam"	Wie heeft de site gemaakt?
meta name="publisher" content="naam"	Wie publiceert de site? Als je een site voor een bedrijf maakt, maak jij hem, maar publiceren zij hem.
meta name="copyright" content="(c)2001 naam"	Copyrights die op de site van toepassing zijn.
meta name="language" content="nl"	De taal van je website, Nederlands in dit geval. Dit is belangrijk voor zoekmachines die op taal kunnen zoeken, zoals Google.
meta name="date" content="2000-11-06T08:49:37+00:00"	De datum waarop de pagina voor het laatst gewijzigd is.
meta name="robots" content="all"	Zoekmachines werken met een robot die jouw pagina doorzoekt. Zo geef je aan dat de zoekmachine jouw pagina mag indexeren en verder mag zoeken in pagina's die gelinkt worden.

<code>meta name="robots" content="index"</code>	De zoekmachine indexeert jouw pagina: je komt dus in de lijst met zoekresultaten te staan.
<code>meta name="robots" content="noindex"</code>	De zoekmachine indexeert jouw pagina niet. Handig als je een proefversie van je site maakt die je eerst grondig online wilt testen zonder dat andere hem al zien.
<code>meta name="robots" content="follow"</code>	De links op de pagina worden gevolgd door de zoekmachine.
<code>meta name="robots" content="nofollow"</code>	De links op een pagina worden niet door de zoekmachine weer gevolgd.
<code>meta http-equiv="charset" content="ISO-8859-1"</code>	Site is HTML bestand, karakterset is ISO-8859-1 standaard. Dit betekent dat je op je site gebruik maakt van westerese leestekens (andere charsets zijn: Arabisch, Chinees, etc.).
<code>meta http-equiv="Content-style-type" content="text/css"</code>	Je gebruikt Cascading Style Sheets oftewel CSS. Zie voor meer info onze CSS-cursus.
<code>meta http-equiv="content-script-type" content="javascript"</code>	Je gebruikt Javascripts.
<code>meta http-equiv="reply to" content="je e-mailadres"</code>	Hoe kunnen bezoekers reageren?
<code>meta http-equiv="expires" content="dag, datum, tijdstip"</code>	Na genoemde datum is de pagina verlopen.
<code>meta http-equiv="pragma" content="no-cache"</code>	Site wordt niet opgenomen in cachegeheugen. Dit zorgt dat je bezoekers wat langer op je site moeten wachten, maar dan wel steeds de nieuwste versie zien. Alleen handig als je je site erg vaak update (meerdere keren per dag).
<code>meta http-equiv="refresh" content="x;URL=http://..."</code>	Na "x" seconden gaat de browser naar de opgegeven URL. Een doorverwijzing naar een andere pagina dus.
<code>meta http-equiv="refresh" content="x"</code>	Na "x" seconden wordt de pagina ververs, en wordt dezelfde pagina weer getoond.

Je beheerst nu de basis van HTML. Hiermee kun je een leuke website maken. Heb je nog vragen over HTML? Stel ze dan in ons [forum](#)! Verder kun je nog de andere cursussen op deze site volgen om meer over Webdesign te leren. Succes!