

GVR Soft's HTMLcursus

Index en snelzoek pagina

Les 1: De basis

- De basis tags: `<html>`, `<head>`, `<title>` en `<body>` tags
- Het nesten van tags
- Achtergrond- en tekstkleur
- De basiskleuren
- Kleurenkiezer
- Een achtergrondafbeelding
- Commentaar in HTML code

Les 2: Tekst (deel 1)

2.1. Positionering van tekst

- 2.1.1. De `<div>` en `` tags
- 2.1.2. Witruimte in tekst, de `<pre>` tag

2.2. Invoeging van witregels

- 2.2.1. Linebreak, de `
` tag
- 2.2.2. Paragraaf, de `<p>` tag
- 2.2.3. Horizontale regel, de `<hr>` tag

Les 3: Tekst (deel 2)

2.3. Hoofdingen

- 2.3.1. De `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>` en `<h6>` tags

2.4. Benadrukking

- 2.4.1. De `` tag
- 2.4.2. De `` tag

2.5. Tekststijl

- 2.5.1. Italic (schuin), de `<i>` tag
- 2.5.2. Bold (vet), de `` tag
- 2.5.3. "Groot", de `<big>` tag
- 2.5.4. "Klein", de `<small>` tag

- 2.5.5. Monospaced, de `<tt>` tag
- 2.5.6. Sub- en Superscript, de `<sub>` en `<sup>` tags

Opmerkingen: tekst en tekststijl in CSS!

- font-style (tekststijl)
- font-weight (tekstdikte)
- color (tekstkleur)
- font-family (lettertype)
- font-size (tekstgrootte)
- text-decoration (tekstdecoratie)

Les 4: Tekst (deel 3)

2.6. Opsommingen

- 2.6.1. De ``, `` en `` tags
 - 2.6.1.1. Ongeordende lijst, de `` tag
 - 2.6.1.2. Geordende lijst, de `` tag
- 2.6.2. Definitielijst, de `<dl>`, `<dt>` en `<dd>` tags

Les 5: Afbeeldingen

- 3.1. De `` tag
 - > Het `src` attribuut
 - > De `width` en `height` attributen
 - > Het `alt` attribuut
- Opmerkingen: Directorystructuur!

Les 6: Link

- 4.1. De `<a>` tag
 - > `<a>` in CSS
 - > E-mail link
 - > Een hyperlink
 - > Een afbeelding als link
 - > Een afbeeldingsmap (imagemap)
 - > Het `target` attribuut

Les 7: Tabellen

- 5.1. De `<table>`, `<tr>` en `<td>` tags
 - > Het `width` attribuut
 - > Het `border` attribuut
 - > Tabel- en celkleur
 - > De `cellpadding` en `cellspacing` attributen
 - > Het centreren van een tabel
 - > Het centreren van tekst in een tabel
 - > Links en rechts uitlijnen
 - > Verticaal uitlijnen

- 5.2. De `<caption>` tag
- 5.3. De `<thead>`, `<tfoot>` en `<tbody>` tags
- 5.4. De `<colgroup>` en `<col>` tags
- 5.5. De `<th>` tag
 - > Het `rowspan` attribuut
 - > Het `colspan` attribuut
 - > Een tabel in een tabel plaatsen

Les 8: Formulieren

- 6.1. Het formulier, de `<form>` tag
 - 6.1.1. Ingave, de `<input>` tag
 - > Het `type="text"` attribuut
 - > Het `type="password"` attribuut
 - > Het `type="checkbox"` attribuut
 - > Het `type="radio"` attribuut
 - > Het `type="file"` attribuut
 - > Het `type="hidden"` attribuut
 - > De `type="submit"` en `type="reset"` attributen
 - > Het `type="image"` attribuut
 - 6.1.2. De `<select>`, `<optgroup>` en `<option>` tags
 - > Het `multiple` attribuut
 - > Het `size` attribuut
 - > Het `selected` attribuut
 - 6.1.3. Tekstveld, de `<textarea>` tag
- Een voorbeeld: GVR Soft's enquête

Les 9: Frames

- 7.1. De `<frameset>` en `<frame>` tags
 - > De `cols` en `rows` attributen
 - > Het `frameborder` attribuut
 - > De `marginwidth` en `marginheight` attributen
 - > Het `scrolling` attribuut
 - > Het `noresize` attribuut
 - > Het `name` attribuut
 - > Het `target` attribuut
- 7.2. De `<noframes>` tag

Les 10: Goed om weten

- 8.1. Het `<lang>` attribuut
- 8.2. De `<meta>` tag
 - > Het `keywords` attribuut
 - > Het `description` attribuut
 - > Het `author` attribuut
 - > Het `robots` attribuut
 - > Het `charset` attribuut
 - > Het `refresh` attribuut

- 8.3. De [<link>](#) tag
- 8.4. De [<base>](#) tag
- 8.5. De [<script>](#) tag

--> Opmerking: HTML karakters

© 2002, GVR Soft's HTMLcursus.

GVR Soft's HTMLcursus

--> les 1:<--

1. De basis tags:

Een geldig basis HTML document bestaat eigenlijk maar uit een paar tags. Het zijn de `<html>`, `<head>`, `<title>` en `<body>` tags. Een voorbeeldje:

<code><html></code>	<-- optioneel
<code><head></code>	<-- optioneel.
<code><title></code>	<-- <u>moet</u> aanwezig zijn in de code.
<code></title></code>	<-- <u>moet</u> aanwezig zijn in de code.
<code></head></code>	<-- optioneel.
<code><body></code>	<-- optioneel, maar wel aangeraden
<i>De inhoud komt hier...</i>	
<code></body></code>	<-- optioneel, maar wel aangeraden
<code></html></code>	<-- optioneel

De volgorde van deze tags moet gerespecteerd worden maar men mag ook deze vorm gebruiken:

```
<html> <head> <title> </title> </head> <body> </body> </html>
```

nota: Ook hoofdletters zijn toegestaan.

nota: optioneel betekent: mag je gebruiken maar het hoeft niet.

Merk op dat er voor deze 4 tags een begintag en een eindtag is!

begintag: <code><html></code>	eindtag: <code></html></code>
begintag: <code><head></code>	eindtag: <code></head></code>
begintag: <code><title></code>	eindtag: <code></title></code>
begintag: <code><body></code>	eindtag: <code></body></code>

Ik raad je aan deze 4 basistags te gebruiken alhoewel enkel de `<title>` tag verplicht is in de code.

ERG BELANGRIJK!

Tags moeten steeds genest zijn!!! Een voorbeeld:

GOED: (1-2-3-3-2-1)

```
<html><head><title></title></head></html>
```

GOED: (1-2-3-3-2-4-4-1)

```
<html><head><title></title></head><body></body></html>
```

FOUT: (1-2-3-1-2-3)

```
<html><head><title></html></head></title>
```

Het nesten van de tags is erg belangrijk. Web pagina's kunnen er vreemd uitzien of helemaal niet meer werken als je de verkeerde volgorde neemt!

Laat ons nu eens een web pagina maken. Ik gebruik alle tags ook al zijn er sommige optioneel. Typ de volgende gegevens in je kladblok in:

```
<html>
<head>
<title>les 1
</title>
</head>
<body>
Mijn eerste web pagina
</body>
</html>
```

Maak een map op jouw desktop of waar je wil met bijv. de naam "website" en sla daar het bovenstaande document in op als **pagina1.html**.

Open nu het bestand **pagina1.html** in jouw browser. (dubbelklik op de file in de map)

Ziedaar, je eerste werkende web pagina! Hoezo, een web pagina maken is moeilijk? Laat ons nu nog wat verder gaan, ik ben er zeker van dat je wel wat meer wil zien dan $\omega\omega\omega$ zinnetje. In deze cursus volg ik de specificaties van W3C zoals je wellicht gelezen hebt in de introductie. Sommige tags zijn ondertussen verouderd en aangezien men deze binnen afzienbare tijd niet meer zal gebruiken lijkt het mij nutteloos ze nog te vermelden of te gebruiken. Met de introductie van CSS (Cascading Style Sheets) is het maken van web pagina's een stuk flexibeler geworden. Natuurlijk is CSS een hoofdstuk apart waar ik in deze cursus niet dieper op in ga tenzij ze worden voorgeschreven door de W3C standaard.

Een voorbeeldje van een verouderd document:

```
<html>
<head>
<title>les 1
</title>
</head>
<body bgcolor="#800080" text="#ffffff">
Mijn eerste web pagina
</body>
</html>
```

Vrees niet als je plots tags ziet die nog niet ter sprake zijn gekomen. Ze zullen wel degelijk uitgelegd worden.

Je ziet in de `<body>` tag --> `bgcolor="#800080" text="#ffffff"`

D.w.z.: Een donker purpere achtergrond en witte letters. Typ dit voorbeeld maar eens in in jouw kladblok, sla het op, dubbelklik op de file en zie wat ik bedoel. Je ziet dit:

Mijn eerste web pagina

Zowel `bgcolor` en `text` zijn verouderd en zullen in de toekomst niet meer worden gebruikt (er zijn er nog meer maar later meer daarover). In plaats daarvan gebruikt men ze nu in Cascading Style Sheets (CSS). Alles wat in het rood staat is de CSS:

```
<html>
<head>
<title>les 1
</title>
<style type="text/css">
body { background: #800080; color: #ffffff }
</style>
</head>
<body>
Mijn eerste web pagina
</body>
</html>
```















Vanaf nu gebruiken we deze vorm !!!


Probeer deze vorm ook maar eens uit in jouw browser. Zoals je ziet betekent:

`background: #800080` --> achtergrondkleur is donkerpurper.
`text: #ffffff` --> tekstkleur is wit.

Hoe bepaal je nu een kleur? Volgend schema maakt \color{red} en ander duidelijk.

Dit zijn de basiskleuren:

 Black = "#000000"	 Green = "#008000"
 Silver = "#C0C0C0"	 Lime = "#00FF00"
 Gray = "#808080"	 Olive = "#808000"
 White = "#FFFFFF"	 Yellow = "#FFFF00"
 Maroon = "#800000"	 Navy = "#000080"
 Red = "#FF0000"	 Blue = "#0000FF"
 Purple = "#800080"	 Teal = "#008080"

 Fuchsia = "#FF00FF"  Aqua = "#00FFFF"

Je mag ook deze vorm gebruiken:

```
<style type="text/css">  
body { background: purple; color: white }  
</style>
```

Ik raad je wel aan de hexadecimale code (vb: #800080) te gebruiken. Per slot van rekening is het gebruik van deze code veel flexibeler. Bovenstaande kleuren zijn natuurlijk niet de enige kleuren, er zijn verschillende gradaties mogelijk.

bijvoorbeeld:

```
mijn eerste web pagina ---> achtergrondkleur=#800080  
mijn eerste web pagina ---> achtergrondkleur=#c000c0  
mijn eerste web pagina ---> achtergrondkleur=#f000f0
```

Je kan ook de [kleurenkiezer](#) gebruiken om een achtergrondkleur voor je pagina te bepalen. Je ziet er ook de hexadecimale waarde die je moet gebruiken.

Je kan ook een afbeelding als achtergrond gebruiken:

```
<style type="text/css">  
body { background-image: url("http://.../jouwachtergrond.gif") }  
</style>
```

Je moet natuurlijk de URL (je web adres) invullen tussen de aanhalingstekens en zorgen dat de afbeelding op dat web adres te vinden is. Een web adres kan bijvoorbeeld zijn: *http://www.GVRSoft.be*. De afbeelding noemt in ons voorbeeld *jouwachtergrond.gif*, dit kan ook *jouwachtergrond.jpg* zijn. Meer over de verschillende types afbeeldingen vind je in de [les over afbeeldingen](#).

Een voorbeeld:

We nemen aan dat jouw web adres (URL) <http://www.mijnadres.com> is en dat jouw afbeelding [achtergrond.gif](#) noemt.

Dit wordt dan de te gebruiken vorm:

```
<style type="text/css">  
body { background-image:  
url("http://www.mijnadres.com/achtergrond.gif" )  
</style>
```

Laat ons dit eens testen, zelfs zonder dat we een URL hebben moet dit lukken.



Klik met de rechtermuistoets op de afbeelding en sla ze op als mijnafbeelding (GIF formaat) in dezelfde map waar jouw HTML documenten zich bevinden. (*Kies figuur opslaan als...of save image as...*).

Typ het volgende in je kladblok in en sla het op als **achtergrond.html**

```
<html>
<head>
<title>
les 1
</title>
<style type="text/css">
body { background-image: url("mijnafbeelding.gif") }
</style>
</head>
<body>
</body>
</html>
```

Open **achtergrond.html** door er op te dubbelklikken. Ziet het er leuk uit of niet? Natuurlijk zal je, als je je eigen web adres hebt, de juiste URL moeten invullen zoals eerder vermeld.

Ziezo, we zijn bijna aan het einde van de eerste les. Nog even herhalen:

Dit is de aanbevolen vorm om een HTML document te maken:

```
<html>
<head>
<title>
De titel van je pagina komt hier
</title>
<style type="text/css">
body { background: #kleur; text:#kleur }
ofwel body { background-image:
url("http://.../jouwachtergrond.gif") }
</style>
</head>
<body>
De inhoud komt hier
</body>
</html>
```

Nog een kleine maar zeker niet onbelangrijke opmerking!

Soms zie je dit in een broncode (HTML code):

```
<style type="text/css">
<!--
body { background: #kleur; text: #kleur }
-->
```

`</style>`

In het voorbeeld hierboven wordt het commentaar attribuut gebruikt. Oudere browser versies kennen de Cascading Style Sheets niet en zullen dit gewoon vertalen als tekst, niet als code. Om dit uit te sluiten wordt het commentaar attribuut toegevoegd.

`<!-- Alles hiertussen is commentaar en wordt door de browser genegeerd -->`

Dit attribuut is ook erg handig om je eigen code te voorzien van commentaar!

Einde van de eerste les!

GVR Soft's HTMLcursus

--> les 2: <--

2. Tekst (deel 1):

2.1. Positionering van tekst

2.1.1. de <div> en tag:

Centreren van tekst is dikwijls nodig. Vroeger werd de <center> tag hiervoor veel gebruikt maar deze tag is verouderd, nu gebruikt men deze vorm:

```
<div class="center">
```

<div> tags en tags zijn eigenlijk niet meer dan lege 'containers', ze bevatten eigenlijk geen 'echte' informatie. Ze worden veelvuldig gebruikt om stijlen in te voeren in bepaalde regels of blokken en om tekst te centreren.

Opmerking: Voor beide tags is er een eindtag namelijk </div> en . Je MOET ze steeds gebruiken!

Bijvoorbeeld:

...	...
<body>	<body>
lijn 2	lijn 2
<div>	
lijn 3	lijn 3
</div>	
lijn 4	lijn 4
</body>	</body>
...	...

Geeft dit resultaat:

lijn 2	
lijn 3	lijn 2 lijn 3 lijn 4
lijn 4	

Zoals je ziet forceert de <div> tag een linebreak (een nieuwe regel) en is dit niet het geval voor de tag. Hou deze tags goed in de gaten, ze worden meer en meer gebruikt!

Als we nu tekst willen centreren gebruiken we deze vorm:

```
...  
<body>
```

```
<div class="center">
gecentreerde tekst
</div>
</body>
```

...

Een **class** definiëren we met CSS (Cascading Style Sheets). Dat doen we zo:
(Alles in het rood is de CSS)

```
<html>
<head>
<title>
je titel komt hier
</title>
<style type="text/css">
.center { text-align:center }
</style>
</head>
<body>
<div class="center">
gecentreerde tekst
</div>
</body>
</html>
```

Ziet er zo uit in je browser:

gecentreerde tekst

.center definieert de klasse 'center'. Je mag dit ook een andere naam geven maar vergeet het **.** niet voor de klassenaam.

Als je de tekst links wil uitlijnen dan gebruik je: **.klassenaam { text-align:left }**

Rechts uitlijnen doe je zo: **.klassenaam { text-align:right }**

2.1.2 de <pre> tag:

De **<pre>** tag laat het gebruik van witruimte toe. Een voorbeeldje:

```
...
<body>
<pre>
    Dit is witruimte
</pre>
</body>
...
```

Ziet er als volgt uit:

Dit is witruimte

2.2. Invoeging van witregels

2.2.1. de
 tag:

Een veel gebruikte tag die niet meer doet dan een *linebreak*.

```
...  
<body>  
Dit is lijn 1  
<br>  
Dit is lijn 2  
</body>
```

...

Zonder de
 tag krijgen we dit te zien:

Dit is lijn1 Dit is lijn2

Met de
 tag krijgen we dit te zien:

Dit is lijn1
Dit is lijn2

Opmerking: Er is geen eindtag voor de
 tag!

2.2.2. de <p> tag:

De <p> tag staat voor het paragraaf element.

```
...  
<body>  
Dit is lijn 1  
<p>  
Dit is lijn 2  
</body>
```

...

Zonder de <p> tag krijgen we dit te zien

Dit is lijn1 Dit is lijn2

Met de <p> tag krijgen we dit te zien:

Dit is lijn1

Dit is lijn2

Merk op dat er bij het gebruik van de <p> tag een extra witregel tussen beide zinnen is!

Opmerking: Er is een eindtag voor de <p> tag, namelijk de </p> tag maar deze is optioneel.

Een voorbeeld:

```
...  
<body>  
<p>  
Dit is lijn 1  
</p>  
<p>  
Dit is lijn 2  
</p>  
</body>  
...
```

Dit geeft hetzelfde resultaat als het voorgaande voorbeeld met de enkele `<p>` tag.

2.2.3. de `<hr>` tag:

De `<hr>` tag staat voor 'horizontal rule'.

```
...  
<body>  
Dit is lijn 1  
<hr>  
Dit is lijn 2  
</body>  
...
```

Bovenstaand voorbeeld geeft als resultaat:

```
Dit is lijn 1  
-----  
Dit is lijn 2
```

Met andere woorden: de `<hr>` tag vult de linebreak op met een lijn.

Opmerking: Er is geen eindtag voor de `<hr>` tag!

Laat ons nu eens al het bovenstaande in een web pagina gieten. Neem je kladblok en begin een nieuw document (sla de vorige gegevens op en kies 'nieuw' uit het menu 'bestand'). Ok, hier gaan we:

```
<html>  
<head>  
<title>  
Je titel komt hier  
</title>  
<style type="text/css">  
<!--
```

```
body { background: #005382; color: #ffff00 }
.center { text-align:center }
-->
</style>
</head>
<body>
<div class="center">
Lijn nummer 1 is gecentreerd
</div>
<br>
lijn nummer 2 met hieronder een lijn
<hr>
<p>
een extra witregel boven lijn nr.3
</p>
<pre>
    witruimte voor lijn nr.4
</pre>
</body>
</html>
```

Sla het bestand op als bijv. **tekst1.html**, kijk daarna in je map en dubbelklik op het bestand **tekst1.html** om het in je browser te openen. Als alles goed is zie je een donkerblauwe achtergrond met zinnen in gele letters.

Einde van de tweede les! In de volgende les meer over tekstgrootte, tekstkleur, tekststijl, lettertype en meer ...

GVR Soft's HTMLcursus

--> les 3: <--

2. Tekst (deel 2):

2.3. Hoofdingen

2.3.1. de <h1>, <h2>, <h3>, <h4>, <h5> en <h6> tags:

Deze tags worden gebruikt voor het maken van titels (hoofdingen) en ondertitels.

Opmerking: Voor elke <h> tag is er een eindtag namelijk een </h> tag.

Een voorbeeld:

```
...  
<body>  
<h1>  
zeer groot  
</h1>  
<h2>  
groot  
</h2>  
<h3>  
groter  
</h3>  
<h4>  
klein  
</h4>  
<h5>  
kleiner  
</h5>  
<h6>  
zeer klein  
</h6>  
</body>  
...
```

Bovenstaand voorbeeld geeft dit als resultaat:

zeer groot

groot

groter

klein

kleiner

zeer klein

2.4. Benadrukkingselementen:

2.4.1. de tag:

Deze tag wordt gebruikt om tekstelementen te benadrukken.

Opmerking: Voor de tag is er een eindtag:

```
...  
<body>  
Dit is de <em>eerste</em> lijn!  
</body>  
...
```

In een webpagina vertaalt zich dit als volgt:

Dit is de *eerste* lijn!

2.4.2. de tag:

Deze tag wordt eveneens gebruikt om tekstelementen te benadrukken, zij het nog meer dan bij de tag.

Opmerking: Voor de tag is er een eindtag:

```
...  
<body>  
Dit is de <strong>eerste</strong> lijn!  
</body>  
...
```

Ziet er als volgt uit:

Dit is de **eerste** lijn!

2.5. Tekststijl

2.5.1. de <i> tag:

Deze tag staat voor "italic" en betekent "schuine" tekst. Heeft hetzelfde effect als de tag.

Opmerking: Voor de <i> tag is er een eindtag: </i>

```
...  
<body>  
Dit is de <i>eerste</i> lijn!  
</body>  
...
```

Resulteert in het volgende:

Dit is de *eerste* lijn!

2.5.2. de tag:

Deze tag staat voor "bold" en betekent "vette" tekst. Heeft hetzelfde effect als de tag.

Opmerking: Voor de tag is er een eindtag:

```
...  
<body>  
Dit is de <b>eerste</b> lijn!  
</body>  
...
```

Resultaat:

Dit is de **eerste** lijn!

2.5.3. de <big> tag:

Deze tag geeft de tekst in een "groter lettertype" weer.

Opmerking: Voor de <big> tag is er een eindtag: </big>

```
...  
<body>  
Dit is de <big>eerste</big> lijn!
```

```
</body>
```

...

Ziet er zo uit:

Dit is de eerste lijn!

2.5.4. de <small> tag:

Deze tag geeft de tekst in een "kleiner lettertype" weer.

Opmerking: Voor de <small> tag is er een eindtag: </small>

...

```
<body>
```

Dit is de <small>eerste</small> lijn!

```
</body>
```

...

In je browser zie je dit:

Dit is de eerste lijn!

2.5.5. de <tt> tag:

Deze tag geeft de tekst 'monospaced' weer. D.w.z. alle letters van het woord staan op dezelfde afstand van elkaar.

Opmerking: Voor de <tt> tag is er een eindtag: </tt>

...

```
<body>
```

Dit is de <tt>eerste</tt> lijn!

```
</body>
```

...

Zo ziet monospaced er uit:

Dit is de eerste lijn!

2.5.6. de <sub> en <sup> tag:

Deze tags betekenen respectievelijk subscript en superscript.

Opmerking: Beiden hebben een eindtag namelijk </sub> en </sup>

...

```
<body>
```

```
H<sub>2</sub>O  
E = mc<sup>2</sup>  
</body>  
...
```

Resulteert in het volgende:

```
H2O  
E = mc2
```

Opmerkingen !

De `<i>`, ``, `<big>`, `<small>` en `<tt>` tags zijn nog niet verouderd (je mag ze dus gebruiken) maar het gebruik ervan wordt afgeraden. Men raad aan gebruik te maken van Cascading Style Sheets (CSS)!

Weet je het nog? Dit is een voorbeeld van een Cascading Style Sheet: (de rode tekst is de CSS)

```
<html>  
<head>  
<title>  
je titel komt hier  
</title>  
<style type="text/css">  
body { background:#000000; color:#ffffff }  
.center { text-align:center }  
</style>  
</head>  
<body>  
<div class="center">  
gecentreerde tekst  
</div>  
</body>  
</html>
```

Hoe gebruik je nu de `<i>`, ``, `<big>`, `<small>` en `<tt>` en andere tekst tags in Cascading Style Sheets (CSS) ? Alles wat in het rood staat is de CSS:

```
<html>  
<head>  
<title>  
je titel komt hier  
</title>  
<style type="text/css">  
body { background:#ffffff; color:#000000 }  
.center { text-align:center }  
.schuin { font-style: italic }  
.vet { font-weight: bold }  
.kleur { color: #00c091 }  
</style>  
</head>  
<body>  
<div class="center">  
gecentreerde tekst  
</div>  
</body>  
</html>
```

```
.lettertype { font-family: Verdana, 'Comic Sans Ms', Arial }
.onderlijn { text-decoration: underline }
.grootte { font-size: 20px }
</style>
</head>
<body>
<div class="center">
gecentreerde tekst
</div>
<div class="schuin">
schuine tekst
</div>
<div class="vet">
vette tekst
</div>
<div class="kleur">
gekleurde tekst
</div>
<div class="lettertype">
een ander lettertype
</div>
<div class="onderlijn">
onderlijnde tekst
</div>
<div class="grootte">
grotere tekst
</div>
</body>
</html>
```

Geeft je dit resultaat:

gecentreerde tekst

schuine tekst
vette tekst
gekleurde tekst
een ander lettertype
onderlijnde tekst
grotere tekst

Dit is een hele boterham dus zet je schrap:

Bij het gebruik van CSS kunnen we bepaalde 'klassen' of 'identiteiten' toewijzen. Laten we de derde lijn van de CSS eens bekijken.

```
.schuin { font-style: italic }
```

Het eerste wat opvalt is het `.` voor het woord schuin.

Dit `.` definieert een 'class' ('klasse').

```
.schuin betekent class="schuin" (klasse="schuin")
{ font-style: italic } betekent zoiets als { tekststijl : schuin }
```

Deze 'classes' (klassen) worden meestal gebruikt met de `<div>` tag of de `` tag.

Als we een klasse gedefinieerd hebben kunnen we ze aanroepen in ons HTML document. Dit doen we als volgt:

```
<div class="schuin">  
schuine tekst  
</div>
```

Belangrijk! We moeten de naam gebruiken die we in onze CSS na het `.` hebben gespecificeerd!

Vervang de `<div>` tag ook eens door de `` tag en bemerk het verschil.

Ik ga zeker niet uitgebreid in op de vraag hoe men CSS moet gebruiken (dit is een cursus apart) maar aangezien men steeds vaker CSS gaat gebruiken voor tekst en tekststijlen zal ik de meest gebruikte "stijlen" een beetje nader beschrijven. Niet al de eigenschappen van een stijl komen aan bod, enkel de belangrijkste of meest gebruikte vormen.

1. font-style: (tekststijl)

```
.klassenaam { font-style: normal of italic }
```

normal = normaal, italic = schuin

2. font-weight: (tekstdikte)

```
.klassenaam { font-weight: normal, bold, bolder, lighter }
```

normal = normaal, bold = vet, bolder = vetter, lighter = lichter

3. color: (tekstkleur)

```
.klassenaam { color: hexadecimaal (#.....) of tekst (white) }
```

4. font-family: (lettertype)

```
.klassenaam { font-family: Arial, Verdana, 'Comic Sans MS', ..., serif  
}
```

Lettertypes met meer dan $\omega\pi\pi$ woord moeten beginnen en eindigen met een `'`

--> `'New Century Schoolbook'`

Eindig altijd met **serif** groep. Als de browser geen lettertype vindt dan zal hij de 'serif' familie gebruiken. (Je hoeft dit niet te doen maar ik raad het je toch aan).

Er zijn verschillende lettertypes geïnstalleerd op jouw computer maar ik kan je niet precies vertellen waar jouw computer zijn lettertypes heeft opgeslagen (op mijn computer is dit: `c:\windows\fonts`). Tracht dit uit te zoeken met de help functie op jouw computer.

GVR's Tip: Gebruik steeds meer dan $\omega\pi\pi$ lettertype en test degene die je gebruikt eerst uit in jouw browser. Een pagina kan er goed uitzien in een bepaald lettertype en dan weer niet in een ander lettertype! En niet iedereen heeft dezelfde lettertypes die jij hebt! De

browser zal het eerste lettertype nemen dat hij tegenkomt in de klasse en als de lezer van jouw pagina dit lettertype niet op zijn computer heeft geïnstalleerd neemt hij het volgende enz... Het lettertype 'Arial' is bijna aanwezig op elke computer dus is het handig om er mee te beginnen of te eindigen.

5. text-decoration: (tekstdecoratie)

```
.klassenaam { text-decoration: none, underline }
```

none = geen, underline = onderlijn

Onderlijnt een woord of zin.

6. font-size: (tekstgrootte)

```
.klassenaam { font-size: bijv. 12pt of 12px of 150% of 1.5em }
```

Test de verschillende waarden eens uit dan zie je het verschil. (pt of px worden het meest gebruikt)

! Dit zijn niet alle CSS eigenschappen of vormen die men kan gebruiken voor tekst behandeling maar je komt er al een heel eind mee! Geloof me maar.

! Men kan ook verschillende eigenschappen aan `u` en dezelfde klasse geven!

(de volgorde speelt geen enkele rol!)

Een voorbeeld:

```
.tekst { color:#FF7901; text-decoration:underline; font-family: Verdana, 'Comic Sans Ms', Arial; font-size: 18pt; font-weight:normal; font-style:italic }
```

een tag zoals deze in je code: `<div class="tekst">I love this!</div>`

resulteert dan in dit:

I love this!

Experimenteer gerust maar eens met al deze CSS stijlen en eigenschappen. Ze zullen in de toekomst meer en meer gebruikt worden! Ik raad je aan om steeds deze vorm te gebruiken:

```
<html>
<head>
<title>
je titel hier
</title>
<style type="text/css">
body { background: #..... ; color: #..... }
.klassenaam { color:#.....; text-decoration: ... ; font-family: ... ;
font-size: ... ; font-weight: ... ; font-style: ... }
```

(je hoeft ze natuurlijk niet allemaal te gebruiken!)

```
</style>  
</head>  
<body>  
<div class="klassenaam"> (of <span class="klassenaam">)  
inhoud komt hier  
</div>  
</body>  
</html>
```

Einde van deze les! In de volgende les meer over opsommingen en lijsten.

GVR Soft's HTMLcursus

--> les 4: <--

2. Tekst tags (deel 3)

2.6. Opsommingen

2.6.1. de , en tags:

Deze tags gebruikt men voor het maken van een lijst (opsommingen).

 staat voor 'unordered list' (ongeordende lijst)

 staat voor 'ordered list' (geordende lijst)

 staat voor 'list item' (lijst item)

Opmerking: De eindtag voor de tag is optioneel. De en tags MOETEN afgesloten worden met respectievelijk en

2.6.1.1. de tag:

```
...  
<body>  
<ul>  
<li>  
een item  
<li>  
een item  
<li>  
een item  
</ul>  
</body>  
...
```

Merk op dat de tag gebruikt wordt tussen de en tag (of tussen de en tag, zie later) en geen tag vereist!

Bovenstaand voorbeeld geeft dit als resultaat:

- een item
- een item
- een item

2.6.1.1. de tag:

```
...  
<body>  
<ol>  
<li>  
een item  
</li>  
een item  
</li>  
een item  
</ol>  
</body>  
...
```

resultaat van de `` tag zoals hierboven gebruikt:

1. een item
2. een item
3. een item

Men kan deze tags ook combineren. Een voorbeeld:

```
...  
<body>  
<ul>  
<li>nummer 1  
  <ol>  
    <li>een item  
    <li>een item  
  </ol>  
</li>nummer 2  
</ul>  
</body>  
...
```

Merk op hoe de tags genest zijn! Weet je het nog? **1-2-2-1**
In dit geval ` ` aangezien de `` tag niet
moet worden afgesloten.

Bovenstaand voorbeeld ziet er in je browser als volgt uit:

- nummer 1
 1. een item
 2. een item
- nummer 2

2.6.2. de `<dl>`, `<dt>` en `<dd>` tags:

Deze taags gebruikt men voor het bepalen van een definitie lijst.

<dl> staat voor 'definition list' (gedefinieerde lijst)
<dt> staat voor 'definition term' (definitie term)
<dd> staat voor 'definition description' (definitie beschrijving)

Opmerking: De eindtag voor de <dt> en de <dd> tag is optioneel.

De <dl> tag MOET worden afgesloten met de </dl> tag!

```
...
<body>
<dl>
  <dt>
  Hacker
  <dd>
  iemand die bestanden op het internet kraakt.
  <dt>
  browser
  <dd>
  software, nodig om te kunnen surfen.
</dl>
</body>
...
```

Zoals je ziet is er geen </dt> of </dd> tag nodig.

Zo zal het eruit zien in je browser:

```
Hacker
    iemand die bestanden op het internet kraakt.
browser
    software, nodig om te kunnen surfen.
```

Ook deze vorm is mogelijk:

```
...
<body>
<dl>
  <dt>
  Pad
  <dt>
  Pad
  <dd>
  een smal weggetje
  <dd>
  een kikkerachtig dier
</dl>
</body>
...
```

Ziet er in je browser als volgt uit:

```
Pad
Pad
    een smal weggetje
    een kikkerachtig dier
```

Einde van les 4. Volgende les --> Afbeeldingen.

GVR Soft's HTMLcursus

--> les 5: <--

3. Afbeeldingen

3.1. de tag:

Deze tag wordt heel veel gebruikt voor het plaatsen van een afbeelding op een web pagina!

De meest voorkomende types voor afbeeldingen op het web zijn GIF types of JPG types.

Opmerking:

Er wordt ook veel gebruik gemaakt van 'animated gifs' (geanimeerde afbeeldingen). Dit is ook een GIF type, het enige verschil is dat dit een aaneenschakeling is van verschillende aparte GIF's.
--> allemaal aparte plaatjes die men aaneenschakelt zoals bij een tekenfilm.

Visueel is er meestal weinig of geen verschil tussen een GIF en JPG type, het is belangrijker te weten dat ze beiden veel worden gebruikt.

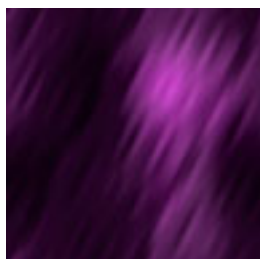
Een voorbeeld:



<-- dit is een GIF type



<-- dit is een animated GIF type



<-- dit is een JPG type

Dit zijn enkel voorbeelden, de afbeeldingen kunnen natuurlijk groter of kleiner zijn.

Hoe bepaal je het type van een afbeelding? Als je geen grafisch

programma zoals bijvoorbeeld Paint Shop Pro of Photoshop hebt dan kan je, als je werkt met Windows, het programmaatje Paintbrush gebruiken. Dit is normaliter standaard bij Windows bijgevoegd. Dit programma ondersteunt verschillende types, waaronder GIF en JPG. Zoek het programma op, start het en open de afbeelding die je wenst te gebruiken in het programma. Om te weten welke extensie (.GIF, .JPG of andere) bij de afbeelding hoort zoek je de eigenschappen van de afbeelding op (ergens in het Paintbrush menu).

Heb je een grafische editor (programma) dan kan je eventueel zelf je afbeeldingen maken.

Opmerking: Elke afbeelding bestaat uit een aantal bytes, klik maar eens met de rechtermuistoets op een afbeelding hierboven en kies eigenschappen, mac gebruikers houden de muistoets ingedrukt. Elke afbeelding dient door de browser ingeladen te worden. Hoe groter het aantal bytes dat een afbeelding beslaat, hoe langer het zal duren eer de pagina (en de afbeelding) geladen zal worden! Hou daarom rekening met het aantal en/of de grootte (aantal bytes) van de afbeeldingen. Iemand die een web pagina bezoekt en erg lang moet wachten om hem te kunnen bekijken geeft er al vlug de brui aan en surft gewoon verder!

een "persoonlijke" hint: gebruik afbeeldingen kleiner dan 20 K (20000 bytes) en verdeel afbeeldingen over verschillende pagina's indien mogelijk.

Het gebruik van de `` tag:

Ik gebruik het eerste plaatje:

```
...  
<body>  
  
<br>  
Dit is cupido  
</body>  
...
```

Laten we de `` tag nu eens van dichterbij bekijken.

```

```

--> `img` staat voor image (afbeelding).

--> `src` staat voor source (bron): source (`src`) is in dit geval "cupido.gif" omdat

- 1) ik de afbeelding "cupido" heb genoemd
- 2) de afbeelding een GIF formaat is

Indien ik een JPG formaat had gebruikt dan had de `src` er zo uitgezien:

```

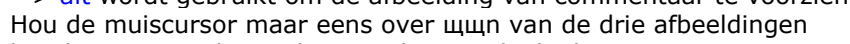
```

--> `width` staat voor de breedte van de afbeelding.

--> `height` staat voor de hoogte van de afbeelding.

Opmerking bij `width` en `height`: Deze 2 attributen zijn optioneel en

hoeven niet in de `` tag vermeld te worden. Ik raad je toch aan ze te gebruiken. Als je de breedte en hoogte opgeeft voor een bepaalde afbeelding dan weet de browser onmiddellijk hoeveel ruimte hij moet reserveren om de afbeelding op de pagina te plaatsen. Dit versnelt het inladen en zal er voor zorgen dat de pagina netjes zal worden ingeladen zonder al teveel bokkesprongen!

--> `alt` wordt gebruikt om de afbeelding van commentaar te voorzien. Hou de muiscursor maar eens over  van de drie afbeeldingen hier boven. Je zal zien dat er zich een tekstkadertje vormt met een kleine beschrijving van de afbeelding. Commentaar in de `` tag moet steeds tussen aanhalingstekens worden geplaatst. De waarde voor de width en height mag in beide vormen, met of zonder aanhalingstekens!

Opmerking bij `alt`: Dit attribuut is optioneel, dus je hoeft hem niet te gebruiken. Ik raad je aan hem toch te gebruiken. Als er zich een probleem voordoet, bijvoorbeeld, de browser kan de afbeelding op jouw internet adres (server) niet vinden, dan zal de lezer van je pagina de afbeelding niet kunnen zien maar alleszins een idee hebben van wat er had moeten staan.

Ik wil bijvoorbeeld een wereldbol tonen maar ik vergat deze afbeelding op mijn server te plaatsen:

Ik heb in mijn code het `alt` attribuut gebruikt zodat de afbeelding van een tekst is voorzien bij eventuele problemen.

Erg belangrijk!

Het is erg belangrijk de browser (Internet Explorer, Netscape, ...) te vertellen waar de afbeelding zich juist bevindt!

Bijvoorbeeld

Mijn internet adres is --> `http://www.GVRSoft.be`
Alle pagina's die je nu bekijkt (deze cursus) heb ik moeten transfereren van mijn computer naar mijn server. De server is de instantie die ervoor zorgt dat mijn pagina's op het internet te zien zijn op mijn eigen persoonlijk adres (zie hierboven). Ik gebruik WS FTP (File Transfer Protocol) (er zijn nog meer FTP programma's en sommige zijn gratis te downloaden op het web) om mijn documenten naar de server te verplaatsen en omgekeerd. Van de server krijg ik een adres toegewezen naar waar ik de documenten moet transfereren (verplaatsen). Dit is te vergelijken met een map op de desktop van je computer. Ik zal trachten een vergelijking te maken:

<u>Mijn computer</u>	<u>Mijn server</u>
Mijn c: (harddisk)	www.GVRSoft.be

Mijn werkmap is: c:\website	Mijn internet map is: www.GVRSoft.be/website
Mijn afbeeldingen c:\website\afbeeldingen	Mijn afbeeldingen vind je op www.GVRSoft.be

Merk op dat je op je eigen computer \ moet gebruiken om onderverdelingen te maken, op het internet gebruik je /. Vergeet dit niet!

Laat ons deze map veronderstellen: c:\website
In deze map bevinden zich het document [introdunctie.html](#) en ook de afbeelding [wereldbol.gif](#).

Als de afbeelding [wereldbol.gif](#) in mijn document [introdunctie.html](#) voorkomt en zowel de afbeelding als het document zich in dezelfde map bevinden dan mag men deze vorm gebruiken:

```

```

Als nu de afbeelding [wereldbol.gif](#) zich in een andere map bevindt dan waar het document [introdunctie.html](#) zich bevindt dan moet men dit specificeren (aanduiden).

Veronderstel dat ik in mijn map c:\website nog een andere map onderbreng waar zich al mijn afbeeldingen zullen bevinden en dat ik deze map *afbeeldingen* zal noemen (een veel gebruikte techniek om alles overzichtelijk te houden). Dan krijg ik volgende structuur:

c:\website (hierin bevindt zich [introdunctie.html](#))
c:\website\afbeeldingen\ (hierin bevindt zich [wereldbol.gif](#))
(Je kan nog verder onderverdelen indien je dat wenst natuurlijk)

Als ik nu de afbeelding wil oproepen dan MOET ik deze vorm gebruiken:

```

```

D.w.z. haal de afbeelding [wereldbol.gif](#) uit de map c:\website\afbeeldingen\.

Let op: Als je een web adres gebruikt dan zal het er zo uitzien:

```

```

Je kan ook linken naar een afbeelding die niet op jouw server staat:

```

```

Jouw browser zal dan de afbeelding opzoeken op de andere server (in dit geval: [www.tekoop.com](#)) en in je pagina laden.

Als je exact weet weer alle documenten zich op de server bevinden dan is het echt 'een makkie'!

Einde van deze les! Op naar de volgende les --> Hoe maak ik een link?

GVR Soft's HTMLcursus

--> les 6: <--

4. Link

4.1. de <a> tag:

Wat zou het internet zijn zonder links? Men zou niet eens kunnen surfen! Gelukkig zijn ze er wel, laat ons de verschillende mogelijkheden eens van dichterbij bekijken.

```
...
<body>
Ga naar <a href="http://www.yahoo.com">Yahoo!</a>
</body>
...
```

Zoals je ziet heeft de <a> tag een verplichte eindtag namelijk
Bovenstaand voorbeeld resulteert in je browser als volgt:
Ga naar [Yahoo!](http://www.yahoo.com)

Zoals je ziet staat het woord yahoo! in blauwe letters (of paarse letters als je deze site al eens bezocht hebt) en is het onderlijnt.
Alles wat tussen en staat vormt de link.
Het href attribuut bevat het internet adres (URL) naar waar de link moet verwijzen.

De standaard kleuren voor links zijn:

```
--> blauw voor een niet bezochte link
--> purper voor een bezochte link
--> rood als men op de link klikt
```

Men kan de kleuren van de links aanpassen:

verouderd (dus af te raden):

```
<body link="#0000ff" vlink="#800080" alink="#ff0000">
```

link staat voor niet bezochte link
vlink staat voor visited link (bezochte link)
alink staat voor active link (actieve link, de klik op de link)

Aan te raden manier: (CSS)

```
<html>
<head>
<title>...</title>
<style type="text/css">
body { background: #ffffff; color: #000000 }
a:link { color: #0000ff }
a:visited { color: #800080 }
a:active { color: #ff0000 }
```

```
</style>
</head>
<body>
...
</body>
</html>
```

- **a:link** staat voor niet bezochte link.
- **a:visited** staat voor een bezochte link.
- **a:active** staat voor de actieve (de klik op) link.

Men moet enkel de #..... waarde vervangen door de waarde die men als kleur wil.

Men kan ook een link maken om een e-mail bericht te versturen. Deze link opent de e-mail client (Outlook Express, of Messenger, of andere ...) en laat de lezer toe een e-mail bericht naar de geadresseerde te sturen. Dit is de vorm:

```
...
<body>
Stuur een <a href="mailto:tony@GVRSoft.be">e-mail</a> naar
Tony.
</body>
...
```

--> wordt weergegeven als: Stuur een [e-mail](#) naar Tony.

Als je op de link klikt wordt jouw e-mail client geopend en kun je me een mail sturen. Erg handig niet?

Je kan zelfs het onderwerp van de e-mail specificeren! Dat doe je zo:

Zend een [e-mail](mailto:tony@GVRSoft.be?subject=hulp!) bericht naar Tony.

Probeer maar eens! Klik op de volgende link:

Zend een [bericht](#) naar Tony.

Je zal zien dat het onderwerp reeds zal ingevuld zijn in je e-mail client.

Een hyperlink:

Stel je voor dat je een lange pagina hebt met veel tekst en ergens bovenaan op je pagina wil je verwijzen naar een deel van de tekst dat zich onderaan bevindt. Meer nog, je wil, als de bezoeker op de link klikt, dat de browser onmiddellijk naar beneden gaat en blijft staan daar waar de verwijzing zich bevindt. Bijvoorbeeld:

Klik [op deze link](#) en zie wat er gebeurt!

Je ziet dat de browser terug naar boven sprint en wel naar *Ga naar*

Yahoo!

Deze techniek wordt veel gebruikt om verwijzingen in lange teksten te maken of om snel van onder naar boven of omgekeerd te gaan. Dit bereik je als volgt:

1. We nemen de regel waar je naar wil verwijzen:

Ga naar `Yahoo!`
de *name* attribuut geeft deze link de naam "yahoo". Je kan natuurlijk eender welke naam kiezen.

2. We nemen de regel die de verwijzing zal doen:

Klik [op deze link](#) en zie wat er gebeurt!

De HTML code voor deze regel is:

Klik `op deze link` en zie wat er gebeurt!

#yahoo zal een link teweeg brengen naar de code waar zich *name="yahoo"* bevindt. **Vergeet het # teken niet in de code!**

Je kan ook een verwijzing maken naar een plaats in een ander document. Voorbeeld: [Klik hier](#) en zie wat ik bedoel!

Je kwam terecht op pagina 4 waar het hoofdstuk over de `<dl>` tag begint. Hoe doe je dit? (ik neem het voorbeeld van hierboven)

Stel dat ik wil verwijzen naar een hoofdstuk op pagina4.html. Meer precies naar het hoofdstuk handelend over de `<dl>` tag.

In de code van pagina 4 heb ik dit gebruikt:
`2.6.2. de <dl>, <dt> en <dd>`
tags:

de code voor de link: [Klik hier](#) en zie wat ik bedoel is:
`Klik hier` en zie wat ik bedoel!

Ik verwijs dus eerst naar de pagina waar de tekst staat, in dit geval *les4.html* met daaraan toegevoegd *#les4dldtdd*

opmerking: zoals je ziet mag je ook een 'lege' `<a>` tag gebruiken:
`--> ...`

Een afbeelding als link:

Je hebt ze waarschijnlijk al gezien. Een afbeelding die dienst doet als link. Weet je nog hoe je een afbeelding op je pagina plaatst? Dit doe je zo:

```

```



Laat ons nu eens een link maken van deze afbeelding:

```
<a href="http://www.liefde.com">  
  
</a>
```



Als je nu op deze afbeelding klikt dan zal je naar *http://www.liefde.com* gebracht worden (dit is een fictief adres).

Je ziet de blauwe (of paarse) kader als je het web adres al eens bezocht hebt) rond de afbeelding. Deze kader oogt meestal niet mooi op je pagina en daarom is het nuttig te weten dat je hem kan 'wegtoveren'. Gebruik dan deze vorm:

```
<a href="http://www.liefde.com">  
  
</a>
```



`border=0` zorgt ervoor dat de kader niet wordt getoond. Ga maar eens met de muis over de afbeelding, het is echt een link!

Opmerking: De volgorde binnen de `` tag speelt geen enkele rol. Dit is ook goed:

```

```

ook dit is OK:

```

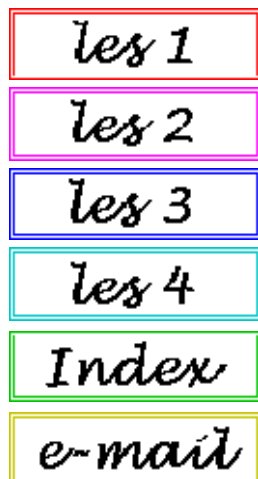
```

enz...

Een imagemap (afbeeldingsmap):

Stel dat je $\omega\pi\pi$ grote afbeelding wil maken die naar verschillende plaatsen kan verwijzen (linken). Je verdeelt dus als het ware de afbeelding in verschillende stukjes. Een soort van 'menu' dus. Geen probleem, ook dit kan.

Ik heb deze afbeelding gemaakt: (het is 1 afbeelding, geen 6 aparte)



Nu kan ik voor elk van de 6 'menu items' een link naar hun respectievelijk adres maken.

--> les 1 verwijst bv. naar les1.html

--> les 2 verwijst bv. naar les2.html

--> enz...

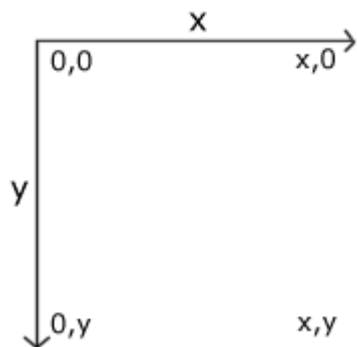
We vertellen de browser eerst hoe de afbeelding noemt en geven de map een naam:

```
...  
<body>  
  
</body>  
...
```

De afbeelding hier noemt *menumap.gif* en de map die we gaan gebruiken is *navigatiemap* (*usemap="#navigatiemap"*)

Zoals gezegd is dit slechts $\omega\pi\pi$ grote afbeelding die we gaan verdelen in 'kleinere' groepen. Er zijn 3 mogelijkheden: we kunnen *rectangles* (rechthoeken), *circles* (cirkels) of *polygons* (veelhoeken) gebruiken.

Hiervoor gaan we x en y coördinaten gebruiken.



Laten we er van uit gaan dat we *rectangles* (rechthoeken) gaan gebruiken.

Dit wordt dan de HTML code:

```

...
<body>

<map name="navigatiemap">
<area shape=rect coords="0,0,135,35" href="les1.html">
<area shape=rect coords="0,40,135,75" href="les2.html">
<area shape=rect coords="0,80,135,115" href="les3.html">
<area shape=rect coords="0,120,135,155" href="les4.html">
<area shape=rect coords="0,160,135,195"
href="htmlintro.html">
<area shape=rect coords="0,200,135,235"
href="mailto:naam@ergens.com">
</map>
</body>
...

```

Opmerking: er is een eindtag voor de `<map>` tag namelijk de `</map>` tag.

Opmerking: er is GEEN eindtag voor de `<area>` tag!

Een beetje uitleg over de code. We hebben de browser verteld gebruik te maken van de afbeelding *menumap.gif* en hij moet aan deze afbeelding de map *#navigatiemap* te koppelen. (*usemap="#navigatiemap"*)

Daarna vertellen we de browser: neem de map *#navigatiemap* --> `<map name="navigatiemap">` (het # teken wordt NIET gebruikt!) en maak zes rechthoeken (`area shape=rect`) met de aangeduide coördinaten.

Elke rechthoek heeft 4 coördinaten:

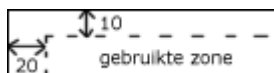
- de eerste coördinaat: x-links
- de tweede coördinaat: y-boven
- de derde coördinaat: x-rechts
- de vierde coördinaat: y-onder

--> biiv. `coords="0.0.135.35"`

Deze coördinaten maken eigenlijk deze rechthoek:

0,0	135,0
0,35	135,35

--> neem je bijv. coords="20,10,135,35" dan krijg je dit:



Je moet dus steeds de x of y (of beide) positie aanpassen. Bij een verticale figuur moet je meestal de y coördinaten aanpassen, voor een horizontale figuur de x coördinaten.

Bekijk de onderstaande afbeelding maar eens van dichterbij. Loopt traag over de afbeeldingen van boven naar onder. Je zult zien dat de rechthoeken die ik gemaakt heb mooi overeenstemmen met de 6 items van het menu. Ik heb enkel de y posities moeten aanpassen. (zie het voorbeeld van hierboven)

les 1

les 2

les 3

les 4

Index

e-mail

Het *href* attribuut specificeert het web adres voor de link.

Probeer zelf ook eens de twee andere mogelijkheden.

Voor de *circle* (cirkel) vorm gebruik je dit:

```
<area shape="circle" coords="center-x, center-y, straal" href="...">
```

Voor de *polygon* (veelhoek) vorm gebruik je dit:

```
<area shape="poly" coords="x1,y1,x2,y2,x3,y3,...." href="...">
```

Bij de *poly* vorm gebruikt je zoveel coördinaten als je zelf wil. Je kunt er allerhande vormen mee bouwen!

Goed om weten!

Er zijn verschillende manieren om een link in iemands browser te laten openen.

- **_top**: als je dit specificeert dan zal de browser elke nieuwe pagina die moet geladen worden (na het klikken van een link) laden in hetzelfde originele venster.
- **_blank**: als je dit specificeert dan zal de browser elke nieuwe pagina die moet geladen worden (na het klikken van een link) laden in een nieuw, apart venster.

er zijn nog **_self** en **_parent** maar die worden minder gebruikt tenzij met frames.

Deze vormen worden gebruikt in het **target** attribuut.

Dus als je een link wil laten openen in een nieuw, apart venster dan gebruik je dit:

```
<a href="http://je adres" target="_blank">jouw link</a>
```

Vergeet het **_** teken niet!

Klik maar eens op de volgende link dan zie je de bedoeling van **target="_blank"**:

--> Voor een cursus HTML [klik je hier!](#)

Ziezo, dit was de les 'Hoe maak ik een link?'

Volgende les --> tabellen! Zet je schrap!

GVR Soft's HTMLcursus

--> les 7: <--

5. Tabellen

5.1. de <table>, <tr> en <td> tag:

Tabellen worden veelvuldig gebruikt in web pagina's. Een tabel geeft je heel wat controle over de lay-out van een pagina. Je kan 'eenvoudige' en meer complexe tabellen gebruiken. Laten we beginnen met een eenvoudige tabel:

```
...
<body>
<table width="100%" border="1" cellpadding="1"
cellspacing="1">
<tr>
<td>
Dit is een erg eenvoudige tabel.
</td>
</tr>
</table>
</body>
...
```

Dit is een erg eenvoudige tabel.

Zoals je ziet worden er voor deze tabel 3 tags gebruikt. De `<table>`, `<tr>` en `<td>` tags. Elke tag heeft een eindtag respectievelijk `</table>`, `</tr>` en `</td>`. De eindtag `</table>` is verplicht, de andere twee, `</tr>` en `</td>`, zijn optioneel!

--> `<table>` betekent tabel.
--> `<tr>` betekent 'table row' en definieert een rij in de tabel.
--> `<td>` betekent 'table data cell' en definieert een cel in de tabel.

Deze volgende vorm is ook geldig en geeft hetzelfde resultaat als de eerste tabel:

```
...
<body>
<table width="100%" border="1" cellspacing="1"
cellpadding="1">
<td>
Dit is een erg eenvoudige tabel.
</table>
</body>
...
```

Toch raad ik je aan de volledige vorm te gebruiken! Je browser maakt

er soms een potje van als je de verkorte vorm gebruikt.

Erg belangrijk!!! Na de `<table>` tag moet een `<td>` tag volgen!

Laat ons eens een tabel maken met verschillende rijen en cellen. Hier gaan we:

```
...
<body>
<table width="100%" border="1" cellspacing="1"
cellpadding="1">
<tr>
<td>rij 1 cel 1</td><td>rij 1 cel 2</td><td>rij 1 cel 3</td>
</tr>
<tr>
<td>rij 2 cel 1</td><td>rij 2 cel 2</td><td>rij 2 cel 3</td>
</tr>
</table>
</body>
...
```

rij 1 cel 1	rij 1 cel 2	rij 1 cel 3
rij 2 cel 1	rij 2 cel 2	rij 2 cel 3

Voor elke cel gebruiken we een `<td>` tag en een `</td>` tag, van zodra we een nieuwe rij willen beginnen gebruiken we een `<tr>` tag.

Natuurlijk kunnen we de afmeting van onze tabel aanpassen aan onze pagina. Je hebt waarschijnlijk het `width` attribuut in de tabel opgemerkt. Hiermee kunnen we de afmeting van de tabel bepalen.

Als we `width="100%"` gebruiken dan zal de tabel over gans de breedte van de pagina worden weergegeven. Gebruiken we een waarde van bijv. `width="50%"` dan beslaat de tabel de helft van de pagina. Gebruiken we geen `width` attribuut dan neemt de tabel de nodige ruimte in, niet meer, niet minder.

Opmerking: in plaats van percentages te gebruiken kunnen we ook absolute waarden gebruiken zoals `width="250"`. Probeer dit maar eens uit.

Het `border` attribuut wordt gebruikt om een kader rond de tabel te plaatsen. Hoe groter de `border` waarde hoe dikker de kader. Een `border=0` waarde betekent geen kader.

Een voorbeeld:

```
...
<body>
<table width="70%" border="5">
<tr>
<td>rij 1 cel 1</td><td>rij 1 cel 2</td><td>rij 1 cel 3</td>
</tr>
<tr>
<td>rij 2 cel 1</td><td>rij 2 cel 2</td><td>rij 2 cel 3</td>
</tr>
</table>
</body>
...
```

```

</tr>
</table>
</body>
...

```

rij 1 cel 1	rij 1 cel 2	rij 1 cel 3
rij 2 cel 1	rij 2 cel 2	rij 2 cel 3

We kunnen ook de tabel of bepaalde cellen in de tabel een bepaalde achtergrond- of tekstkleur geven. Vroeger gebruikte men deze vorm: `<td bgcolor="#.....">` maar deze vorm is verouderd! We moeten nu gebruik maken van style sheets (CSS).

Dit doen we zo:

```

<html>
<head>
<title>...</title>
<style type="text/css">
td.groen { background: #00cc00; color: #cc00cc }
td.blauw { background: #0000ff; color: #ffff00 }
</style>
</head>
<body>
<table width="70%" border="1">
<tr>
<td class="groen">cel 1</td>
<td class="blauw">cel 2</td>
</tr>
</table>
</body>
</html>

```

cel 1	cel 2
-------	-------

In de style sheet van ons voorbeeld bepalen we twee *klassen* namelijk: `td.groen` en `td.blauw`.

Het *background* attribuut bepaalt de achtergrondkleur van de cel en het *color* attribuut bepaalt de tekstkleur.

Opgelet!

Om $\omega\omega\omega$ of andere reden moet in Netscape de `<td>` tag afgesloten worden met de `</td>` tag als je wilt werken met kleuren in een cel! Vergeet dit niet!

De attributen *cellpadding* en *cellspacing* zijn erg handig. Deze twee attributen bepalen de witruimte tussen de cellen onderling en in de cellen zelf.

Een voorbeeldje:

```

...
<body>
<table width="70%" border="5" cellpadding="15"
cellspacing="10">
<tr>
<td>rij 1 cel 1</td><td>rij 1 cel 2</td><td>rij 1 cel 3</td>
</tr>
<tr>
<td>rij 2 cel 1</td><td>rij 2 cel 2</td><td>rij 2 cel 3</td>
</tr>
</table>
</body>
...

```

rij 1 cel 1	rij 1 cel 2	rij 1 cel 3
rij 2 cel 1	rij 2 cel 2	rij 2 cel 3

De witruimte tussen de kader van de tabel (border) en de cellen noemen we [cellspacing](#).

De witruimte tussen de tekst in de cel en de celwand noemen we [cellpadding](#).

In alle voorbeelden die we tot nu toe gezien hebben zien we dat de tabel links op de pagina wordt geplaatst. We kunnen een tabel ook centreren. Vroeger gebruikte men deze tag: `<table align="center">`

--> Deze vorm is verouderd en dus af te raden!

Opmerking! *Internet Explorer en Netscape hanteren sommige tags op verschillende manieren. Zoals gezegd kunnen tabellen complex zijn en wordt er meer en meer gebruik gemaakt van style sheets. Ik heb geprobeerd zoveel mogelijk verouderde tags te 'vervangen' in de voorgeschreven style sheets maar in sommige gevallen kon ik geen $\omega\omega\omega$ vormige tabel krijgen. Internet explorer geeft soms andere vormen van tabellen weer dan Netscape en omgekeerd. Toch werkt alles in deze cursus $\omega\omega\omega$ vormig in beide browsers al gebruik ik soms mijn eigen techniek!*

Ik centreer mijn tabellen zo:

```

<html>
<head>
<title>...</title>
<style type="text/css">
.center { text-align:center }
</style>
</head>
<body>
<div class="center">

```

```
<table width="70%" border="5">
<tr>
<td>cel 1</td>
<td>cel 2</td>
</tr>
</table>
</div>
</body>
</html>
```

cel 1	cel 2
-------	-------

Ik plaats de HTML code tussen een `<div class="center">` tag zoals in bovenstaand voorbeeld beschreven.

Het centreren van de tekst in een cel doen we als volgt:

```
...
<body>
<table width="70%" border="1">
<tr>
<td align="center">cel 1</td>
<td align="center">cel 2</td>
</tr>
</table>
</body>
</html>
```

cel 1	cel 2
-------	-------

Het attribuut `align="center"` in de `<td>` tag zorgt er voor dat de tekst in de cel gecentreerd wordt. Men kan ook `align="left"` (links uitlijnen) of `align="right"` (rechts uitlijnen) gebruiken.

Opmerking: om de tekst (of afbeelding) **verticaal uit te lijnen** gebruik je dit:

```
--> valign="top" (bovenaan)
--> valign="middle" (in het midden)
--> valign="bottom" (onderaan)
```

De `<table>`, `<tr>` en `<td>` tag zijn eigenlijk de basistags van een tabel. Met de komst van HTML 4.0 zijn er meerdere tags bijgekomen om meer flexibiliteit aan tabellen te geven. Laten we ze eens van dichterbij bekijken:

5.2. de <caption> tag:

Deze (optionele) tag kun je gebruiken voor een beschrijving van je tabel.

Opmerking: voor elke `<caption>` tag moet je een `</caption>` tag gebruiken!

```
...
<body>
<table width="100%" border="1" cellpadding="1"
cellspacing="1">
<caption>Boven de tabel komt de beschrijving!</caption>
<tr>
<td>
Dit is een erg eenvoudige tabel.
</td>
</tr>
</table>
</body>
...
```

Boven de tabel komt de beschrijving!
Dit is een erg eenvoudige tabel.

5.3. de `<thead>`, `<tfoot>` en `<tbody>` tags:

- > `<thead>` staat voor "table header" (hoofding van de tabel)
- > `<tfoot>` staat voor "table footer" (voetnoot bij de tabel)
- > `<tbody>` staat voor "table body" (blok in een tabel)

Voor elke `<thead>`, `<tfoot>` en `<tbody>` bestaat er eindtag nl. `</thead>`, `</tfoot>` en `</tbody>`

Opmerking: Alle drie de tags zijn optioneel maar worden stilletjesaan meer en meer gebruikt.

De `<thead>` en de `<tfoot>` tags worden verondersteld informatie te bevatten over de kolommen in de tabel. De `<tbody>` tag moet rijen met tabel data bevatten!

Een voorbeeld:

```
...
<body>
<table>
<thead>
<tr>informatie over de hoofding
</tr>
<tfoot>
<tr>informatie over de voetnoot
</tr>
<tbody>
<tr>blok 1, eerste rij
<tr>blok 1, tweede rij
</tbody>
<tr>blok 2, eerste rij
<tr>blok 2, tweede rij
```

```
<tr>blok 2, derde rij
</table>
</body>
...
```

Merk op dat er na elke `<thead>`, `<tfoot>` en `<tbody>` een `<tr>` tag volgt en dat de `</thead>`, `</tfoot>` en `</tbody>` tag optioneel zijn!

5.4. `<colgroup>` en `<col>`:

A. De `<colgroup>` tag vertegenwoordigt een groep van kolommen in een tabel.
De eindtag `</colgroup>` is optioneel!

Er horen een paar attributen bij deze tag namelijk:

-- > `span` --> specificeert het aantal kolommen in de groep.
Dit moet een getal>0 zijn!

-- > `width` --> specificeert de breedte van elke kolom in de groep.
Dit kan een pixel waarde, percentage of relatieve waarde zijn.

Voorbeeld:

```
<colgroup span="10" width="15">
</colgroup>
```

--> 10 kolommen van elk 15 pixels breed.

B. De `<col>` tag vertegenwoordigt ущн kolom in een tabel.
De eindtag `</col>` bestaat **niet** en is dus verboden!

Volgende attributen zijn mogelijk:

-- > `span` --> standaardwaarde is 1 (ущн kolom)
Is de waarde bijv. 3 dan zullen kolom 1, 2 en 3 dezelfde eigenschappen bezitten.

-- > `width` --> specificeert de breedte van de kolom.
Dit kan een pixel waarde, percentage of proportionele waarde zijn.
Opgelet! Dit attribuut heeft dezelfde betekenis als het `width` attribuut van de `<colgroup>` tag en overschrijft het!

Een woordje uitleg bij proportionele waarde. Bekijk volgend voorbeeld:

```
...
<body>
<table>
<colgroup>
<col width="100"><-- kolom 1
<colgroup>
<col width="100"><-- kolom 2
<col width="0*"><-- kolom 3
```

```

<col width="2*"><-- kolom 4
<colgroup align="center">
<col width="1*"><-- kolom 5
<col width="3*"><-- kolom 6
<thead>
<tr>
<td>rij 1, cel 1
</td>
</tr>
</table>
</body>
...

```

Kolom 1 en 2 krijgen elk 100 pixel breedte toegewezen. Daarna wordt de minimale ruimte voor kolom 3 gereserveerd (0*). De overblijvende horizontale ruimte wordt verdeeld in 6 delen (2* + 1* + 3*) onder kolom 4, 5 en 6. Kolom 4 krijgt hiervan 2 delen, kolom 5 krijgt 1 deel en kolom 6 krijgt 3 delen.

Merk ook het attribuut `align="center"` op in de laatste `<colgroup>` tag. Dit zal ervoor zorgen dat alle cellen in elke kolom van deze groep zullen worden gecentreerd. Onthou wel dat alle attributen in de `<col>` tag die van de `<colgroup>` overschrijven!

5.5. de <th> tag:

NIET te verwarren met de `<thead>` tag!

Eigenlijk is deze tag verwant met de `<td>` tag maar de `<th>` tag is eerder nieuw. Er is een eindtag voor de `<th>` namelijk `</th>` maar deze is optioneel.

--> de `<th>` staat voor "table header cell" (hoofding van een cel in een tabel)

De `<th>` tag creëert een cel (**Het is dus eigenlijk een `<td>` tag!**) die hoofding informatie bevat!

Mogelijke attributen:

--> `rowspan` --> specificeert het aantal rijen dat een cel inneemt.

--> `colspan` --> specificeert het aantal kolommen dat een cel inneemt.

Deze 2 attributen mag men ook in een `<td>` tag gebruiken! Zie voorbeelden hieronder.

Er zijn nog een paar attributen die slechts sporadisch worden gebruikt dus beperk ik me tot deze 2 attributen.

Wat doet de `<th>` tag nu?

Een voorbeeld:

```

...
<tbody>

```

```

<div class="center">
<table width="65%" border=2>
<tr>
<th>Naam</th>
<th>Leeftijd</th>
<th>Aantal kinderen</th>
</tr>
<tr>
<td>Mark</td>
<td>41</td>
<td>2</td>
</tr>
<tr>
<td>Peter</td>
<td>27</td>
<td>0</td>
</tr>
</table>
</div>
</body>
...

```

Resulteert in je browser als volgt:

Naam	Leeftijd	Aantal kinderen
Mark	41	2
Peter	27	0

Zoals je ziet wordt de tekst gedefinieerd in de `<th>` tag automatisch gecentreerd in de cel en vetjes weergegeven. De tekst gedefinieerd in de `<td>` tag wordt **NIET** automatisch gecentreerd. Je kan hiervoor het `align="center"` attribuut gebruiken.

Het gebruik van het attribuut `rowspan`:

```

...
<body>
<div class="center">
<table width="65%" border=2>
<tr>
<td>1</td><td rowspan="2">2</td><td>3</td>
</tr>
<tr>
<td>4</td><td>6</td>
</tr>
<tr>
<td>7</td><td>8</td><td>9</td>
</tr>
</table>
</div>
</body>
...

```

Ziet er zo uit:

1	2	3
4	6	
7	8	9

De tweede cel neemt in bovenstaande tabel twee rijen in zoals gedefinieerd in:

```
<td rowspan="2">
```

Experimenteer maar eens met verschillende cellen!

Het gebruik van het attribuut `colspan`:

```
...
<body>
<div class="center">
<table width="65%" border=2>
<tr>
<td>1</td><td>2</td><td>3</td>
</tr>
<tr>
<td colspan="2">4</td><td>6</td>
</tr>
<tr>
<td>7</td><td>8</td><td>9</td>
</tr>
</table>
</div>
</body>
...
```

In je browser zie je dit:

1	2	3
4		6
7	8	9

De vierde cel neemt in bovenstaande tabel twee kolommen in zoals gedefinieerd in: `<td colspan="2">`

Experimenteer maar eens met verschillende cellen!

Opgel!

Definieer geen overlappende cellen! Als je dit toch doet krijg je ofwel niets ofwel heel rare constructies!

Een tabel in een tabel plaatsen

Deze techniek wordt veel gebruikt bij de constructie van complexe

tabellen. Ik ga hier niet te diep op in, experimenteer er zelf eens mee, het zal je verbazen wat je allemaal met tabellen kan doen!

Een klein voorbeeldje van een tabel in een tabel:

```

...
<body>
<div class="center">
<table width="85%" border="10" cellpadding="15">
<tr>
<td>tabel 1, rij 1, cel 1</td>
<td>tabel 1, rij 1, cel 2</td>
</tr>
<tr>
<td>tabel 1, rij 2, cel 1</td>
</tr>
<tr>
<td>
<table border="3">
<tr>
<td>tabel 2, rij 1, cel 1</td>
<td>tabel 2, rij 1, cel 2</td>
</tr>
<tr>
<td>tabel 2, rij 2, cel 1</td>
</tr>
</table>
</td>
</tr>
</table>
</div>
</body>
...

```

Ziet er als volgt uit:

tabel 1, rij 1, cel 1	tabel 1, rij 1, cel 2				
tabel 1, rij 2, cel 1					
<table border="1"> <tr> <td>tabel 2, rij 1, cel 1</td> <td>tabel 2, rij 1, cel 2</td> </tr> <tr> <td>tabel 2, rij 2, cel 1</td> <td></td> </tr> </table>	tabel 2, rij 1, cel 1	tabel 2, rij 1, cel 2	tabel 2, rij 2, cel 1		
tabel 2, rij 1, cel 1	tabel 2, rij 1, cel 2				
tabel 2, rij 2, cel 1					

Zorg er steeds voor dat de tags **correct genest** zijn!

Dit is al iets complexer maar laat je niet afschrikken. Het gebruik van echt complexe tabellen komt weinig voor. Vraag je ook niet teveel af of je nu wel of niet de optionele (eind)tags moet gebruiken maar wees wel consequent en bepaal voor jezelf een goede werkwijze.

GVR's Tips:

- Oefen maar wat met simpele constructies en je komt al een heel eind!
- Gebruik alle 3 de hoofdtags, t.t.z. `<table>`, `<tr>` en `<td>` en hun eindtags `</table>`, `</tr>` en `</td>`.

Oef! Einde van deze les!

Eventjes op adem komen en dan naar les 8 --> formulieren.

GVR Soft's HTMLcursus

--> les 8: <--

6. Formulieren

6.1. de <form> tag:

In sommige gevallen kan het nuttig zijn om gegevens van je bezoekers te kunnen ontvangen. Zoals bijvoorbeeld *feedback* (commentaren van je bezoekers) of een *enquête* of een ander soort informatie. Hiervoor kunnen we de `<form>` tag gebruiken. Deze tag wordt meestal gecombineerd met CGI (Common Gateway Interface). CGI doet heel wat, waaronder het verwerken van gegevens afkomstig van een web pagina, maar dit is een cursus apart. Vrees niet, er zijn andere manieren om gegevens van formulieren te verwerken (zonder kennis van CGI) en deze gaan we eens bekijken.

Als we een formulier willen maken dan moeten we beginnen met de `<form>` tag en eindigen met de `</form>` tag! Zo ziet de `<form>` tag er uit:

```
<form action="http://..." method="post" enctype="...">
of
<form action="mailto:..." method="post" enctype="...">
of
<form action="http://..." method="get" enctype="...">
```

Zoals je ziet zijn er drie belangrijke attributen in de `<form>` tag.

1) --> `action="..."`

Twee mogelijkheden:

- "http://..." --> Dit wil zeggen dat het formulier gebruik zal maken van een web adres (URL).
- "mailto:..." --> dit wil zeggen dat het formulier gebruik zal maken van een e-mailadres.

Opmerking: Ik ga niet dieper in op de "http://..." optie omdat deze eerder met CGI te maken heeft of iets moeilijker ligt. We gaan ons beperken tot de "mailto:..." optie. Let wel, alle voorbeelden werken zowel met de "http://..." als de "mailto:..." optie!

2) --> `method="..."`

Twee mogelijkheden:

- "get" --> de informatie in (van) het formulier wordt aan het web adres (URL) toegevoegd (en gepost). Dit zijn speciale tekens (ASCII) die bijvoorbeeld door zoekmachines worden gebruikt om te zoeken naar een 'henaald' woord (of zin)

- zonder verdere actie.
- "post" --> de informatie in (van) het formulier kan in verschillende vormen worden behandeld en gepost.

Opmerking: Ik ga niet verder in op de "get" optie, deze optie wordt meestal gebruikt in combinatie met CGI. Let wel, alle voorbeelden werken zowel met de "get" als "post" optie!

3) --> `enctype="..."`

Er zijn verschillende mogelijkheden maar we beperken ons tot `enctype="text/plain"`.

Dit stelt ons in staat gegevens te verzenden in 'een leesbare vorm'.

Misschien is dit allemaal een beetje verwarrend maar ik beperk me tot de meest simpele vorm van een formulier. Dit wil zeggen een formulier dat we zonder de hulp van CGI kunnen verwerken en dat ons in staat stelt gegevens te verzamelen van onze bezoekers. We zullen als voorbeeld op het einde van deze les een formulier maken dat onze bezoekers toe laat commentaar, opmerkingen of iets dergelijks vanaf de bezochte pagina rechtstreeks naar ons te sturen.

Natuurlijk hebben we meer nodig dan de `<form>` tag alleen.

6.1.1. de `<input>` tag:

Deze tag laat ons toe verschillende soorten van gegevens in te voeren.

Opgelet! Er is **GEEN** eindtag voor de `<input>` tag!

Dit zijn de mogelijke attributen:

`type="text"`

Geeft je de mogelijkheid om tekst in te geven.

Voorbeeld:

```
...  
<body>  
<form action="mailto:..." method="post"  
enctype="text/plain">  
Voornaam: <input type="text">  
</form>  
</body>  
...
```

Dit is het resultaat:

Voor naam:

Maar er is meer natuurlijk ... Bekijk volgend voorbeeld:

```
...
<body>
<form action="mailto:..." method="post"
enctype="text/plain">
Voor naam: <input type="text" size="15" value="dit is de tekstbox"
maxlength="20" name="voornaam">
</form>
</body>
...
```

Resulteert in:

Voor naam:

Opmerking: Deze 4 attributen zijn optioneel dus NIET verplicht!

- `size="..."` --> bepaalt de grootte van de tekstbox.
- `value="..."` --> plaatst de opgegeven tekst in de tekstbox.
- `maxlength="..."` --> bepaalt het maximum aantal karakters dat in de box mag gebruikt worden. Probeer maar eens in bovenstaande box meer dan 20 karakters in te tikken.
- `name="..."` --> geeft een specifieke naam aan de box.
Belangrijk! Deze naam zal getoond worden in de e-mail zodat je weet, zoals in bovenstaand voorbeeld, wat de voor naam van je bezoeker is! Maak er een goede gewoonte van dit attribuut te gebruiken!

[type="password"](#)

Eigenlijk hetzelfde als tekstingave. Het enige verschil is dat de tekst in de vorm van asterisks (*) wordt ingegeven zodat men niet kan zien wat er ingetikt wordt.

Opgelet! Dit betekent niet dat de tekst onleesbaar zal zijn als de tekst verstuurd wordt. De server (het gegeven dat instaat voor de transfer van de gegevens van en naar verschillende locaties) beschouwd dit als gewone tekst.

Een voorbeeld:

```
...
<body>
<form action="mailto:..." method="post"
enctype="text/plain">
Voor naam: <input type="password">
</form>
```

```
</body>
```

...

Ziet er in je browser als volgt uit:

Voornaam:

Typ in bovenstaand voorbeeld maar eens een paar karakters in.

[type="checkbox"](#)

Deze vorm kun je eigenlijk vergelijken met een schakelaar die je aan of uit zet.

Een voorbeeld:

...

```
<body>
<form action="mailto:..." method="post"
  enctype="text/plain">
Goed: <input type="checkbox" name="goed">
Slecht: <input type="checkbox" name="slecht">
</form>
</body>
```

...

Goed: Slecht:

Ga maar eens met de muiscursor in een vierkantje staan en klik met de muistoets. Er verschijnt een 'v-tje' in het vierkantje dat je koos. Klik je er nog eens op dan verdwijnt het 'v-tje'.

Je kan ook al op voorhand een 'v-tje' in een vierkantje zetten. Dat doe je zo:

...

```
<body>
<form action="mailto:..." method="post"
  enctype="text/plain">
Goed: <input type="checkbox" checked name="goed">
Slecht: <input type="checkbox" name="slecht">
</form>
</body>
```

...

Goed: Slecht:

Merk op dat het vierkantje naast 'Goed' reeds is aangekruist.

[type="radio"](#)

Era verwant met 'checkbox'.

Een voorbeeld:

```
...
<body>
<form action="mailto:..." method="post"
enctype="text/plain">
Goed: <input type="radio" name="goed">
Slecht: <input type="radio" name="slecht">
</form>
</body>
...
```

Ziet er als volgt uit:

Goed: Slecht:

Opmerking: Als je eenmaal een keuzerondje hebt aangestipt dan kun je het niet meer ongedaan maken. Test het maar eens uit.

MAAR!! Je kan dit omzeilen door, in dit geval, beide keuzes dezelfde *'name'* eigenschap te geven!

Bekijk volgend voorbeeld:

```
...
<body>
<form action="mailto:..." method="post"
enctype="text/plain">
Goed: <input type="radio" name="keuze">
Slecht: <input type="radio" name="keuze">
</form>
</body>
...
```

Goed: Slecht:

Selecteer eens het eerste keuzerondje en daarna selecteer je het tweede (of omgekeerd) en zie wat er gebeurt.

NOTA: Ook bij het *type="radio"* attribuut kan je al op voorhand een keuzerondje selecteren met de *checked* waarde.

type="file"

Deze vorm van input laat je toe een bestand op te halen dat je kan verzenden.

```
...
<body>
<form action="mailto:..." method="post"
enctype="multipart/form-data">
<input type="file" name="file">
</form>
```

```
</body>
```

...

Klik maar eens op de knop naast de tekstbox. Je kan dan een bestand van je harde schijf opzoeken om eventueel te versturen.

Merk op: Bij het `type="file"` attribuut wordt best `"enctype=multipart/form-data"` gebruikt in plaats van `"enctype=text/plain"`. Je e-mail client zal een nieuw bericht aanmaken waar je handmatig het bestand moet aan toevoegen.

`type="hidden"`

Als je het `hidden` attribuut gebruikt kan je mechanismen of informatie in het formulier onzichtbaar maken.

Let op! De mechanismen of informatie werken wel, je ziet ze alleen niet.

Een voorbeeld:

...

```
<body>
<form action="mailto:..." method="post"
enctype="text/plain">
Jouw vraag: <input type="text" name="tekst">
<input type="hidden" name="onderwerp" value="cursus">
</form>
</body>
```

...

Jouw vraag:

Je ziet dus de tekstbox (met 'Jouw vraag' ervoor) en dat is het. Het `hidden` attribuut creëert eigenlijk een 'onzichtbare' regel in het formulier MAAR zal de informatie wel uitvoeren, d.w.z. als het formulier zou verzonden worden via e-mail dan zal de ontvanger twee regels informatie ontvangen, namelijk (we nemen aan dat de lezer iets heeft ingevuld in de tekstbox):

1. Jouw vraag: ...
2. onderwerp=cursus

Het `hidden` attribuut wordt veel gebruikt in CGI toepassingen. Een veel gebruikt mechanisme is het maken van een regel die, nadat de informatie is verstuurd, de lezer naar een 'bedankings' pagina leidt.

`type="submit"` `type="reset"`

Deze twee attributen maken twee knoppen. Een knop die het mogelijk maakt de informatie in het formulier te verzenden en een andere knop die het mogelijk maakt alle informatie in het formulier te wissen.

Voorbeeld:

```

...
<body>
<form action="mailto:..." method="post"
enctype="text/plain">
Jouw vraag: <input type="text" name="tekst">
<p>
<input type="submit" name="verzenden" value="VERZEND
NU!">
<input type="reset" name="wissen" value="OPNIEUW">
</form>
</body>
...

```

Jouw vraag:



- `type="submit"` creëert een knop die het mogelijk maakt de informatie te verzenden.
- `type="reset"` creëert een knop die het mogelijk maakt de informatie te wissen.

Typ in de tekstbox bijvoorbeeld jouw naam eens in en klik dan met de muistoets op de OPNIEUW knop. Je zal zien dat de informatie in de tekstbox wordt gewist. Klik je op de VERZEND NU ! knop dan zal de informatie in de tekstbox worden verzonden. Probeer maar eens. Het zal niet echt lukken om de informatie te verzenden aangezien er geen emailadres in het `action="mailto:..."` attribuut is gespecificeerd. Maar geloof me, het werkt echt.

De tekst die verschijnt op de knoppen kan je zelf bepalen in het `value="..."` attribuut. Maar er is nog een manier om een knop te maken ...

`type="image"`

Erg handig! Stelt je in staat een afbeelding te gebruiken in plaats van een grijze knop.

```

...
<body>
<form action="mailto:..." method="post"
enctype="text/plain">
Jouw vraag: <input type="text" name="tekst">
<p>
<input type="image" src="verzendknop.gif" name="verzend">
</form>
</body>

```

...

Ziet er als volgt uit:

Jouw vraag:



- `src="..."` geeft aan welke afbeelding je wil gebruiken voor de knop. In dit geval is de naam van de afbeelding `verzendknop.gif` maar dit kan eender welke naam zijn. De eigenschappen voor de afbeeldingen vind je in [de les over afbeeldingen](#).

Opmerking: Je kan GEEN afbeelding gebruiken voor de *reset* (wis) knop!

6.1.2. de `<select>`, `<optgroup>` en `<option>` tag:

Laat ons toe een selectielijst te maken die $\omega\omega\omega$ of meerdere keuzes toelaat.

Opgelet! De eindtags `</select>` en `</optgroup>` zijn VERPLICHT!
De eindtag `</option>` is optioneel.

Opgelet! De `<select>` en `<option>` tags zijn een minimum om een lijst te maken, de `<optgroup>` tag is optioneel!

Een voorbeeld:

```
<form action="mailto:..." method="post">
<select name="selectielijst">
<optgroup label="computer">
<option value="PC">PC
<option value="MAC">MAC
<option value="Playstation">Playstation
<option value="Deep Blue">Deep Blue
</optgroup>
</select>
</form>
```



Je ziet een tekstbox met een pijl naar beneden. Klik je op deze pijl dan krijg je een lijst (menu) waaruit je kan kiezen.

Hoe maak ie zo'n lijst?

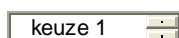
- Begin met de `<select>` tag en geef hem een naam (niet verplicht maar wel handig). Vergeet ook niet de `</select>` tag op het einde!!!
- zoals gezegd is de `<optgroup>` tag optioneel. Ik heb hem in bovenstaand voorbeeld gebruikt om de werking ervan te laten zien. Deze tag maak een lijst overzichtelijk. Het *label* attribuut dient voor de beschrijving van een groep in de lijst en mag ook gebruikt worden in de `<option>` tag. Als je de `<optgroup>` tag gebruikt vergeet dan niet de `</optgroup>` tag!
- Om de lezer van je pagina een keuze in de lijst aan te bieden gebruik je de `<option>` tag. Elke `<option>` tag definieert een nieuwe keuze. De `<option>` tag HOEFT NIET afgesloten te worden met de `</option>` tag! Het *value* attribuut heeft dezelfde betekenis als in de `<input>` tag.

Opmerking! Bovenstaand voorbeeld is niet altijd erg praktisch. Je kan maar $\omega\omega\omega$ keuze maken uit de lijst en soms is het maken van meerdere keuzes nodig.

Oplissing:

```
<form action="mailto:..." method="post">
<select multiple name="selectielijst">
<option value="keuze1">keuze 1
<option value="keuze2">keuze 2
<option value="keuze3">keuze 3
<option value="keuze1">keuze 1
</select>
</form>
```

Resultaat:



Je ziet nu een grotere tekstbox met 4 keuzes (1, 2, 3, 1 --> ik heb de laatste keuze in de lijst terug keuze 1 genoemd zodat we eventueel ook keuze 3 EN 1 kunnen nemen). Als je op een keuze klikt en dan een andere keuze wenst te doen dan zal de vorige keuze geannuleerd worden! Probeer maar eens. Als je op een keuze klikt dan zal de tekst in een blauwe achtergrond geplaatst worden. Neem je een volgende (andere) keuze dan gaat de blauwe achtergrond van de vorige keuze weg en wordt de huidige keuze op een blauwe achtergrond geplaatst. Hoe kun je nu meer dan $\omega\omega\omega$ keuze maken??? Zet de muiscursor op een bepaalde keuze, klik er op, en beweeg de muis naar boven of onder (terwijl je de muisknop ingedrukt houdt!) voor meer keuzes.

We kunnen ook de grootte van de tekstbox (daar waar de keuzes in geplaatst worden) aanpassen naar onze zin. Dit doen we als volgt:

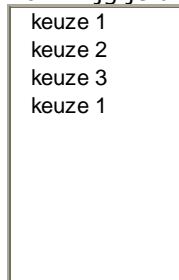
```
<form action="mailto:..." method="post">
<select size="10" multiple name="selectielijst">
<option value="keuze1">keuze 1
<option value="keuze2">keuze 2
```

```

<option value="keuze3">keuze 3
<option value="keuze1">keuze 1
</select>
</form>

```

Dan krijg je dit:



Het `size="..."` attribuut definieert het aantal rijen in de lijst. De waarde van het `size="..."` attribuut moet een geheel nummer zijn, kleiner of groter dan het aantal keuzes.

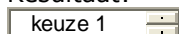
Opmerking: We kunnen een keuze al op voorhand aanduiden. De methode is erg gelijkend op die van de `<input type="checkbox">` of `type="radio">` tag.

```

<form action="mailto:..." method="post">
<select multiple name="selectielijst">
<option value="keuze1">keuze 1
<option value="keuze2" selected>keuze 2
<option value="keuze3">keuze 3
<option value="keuze1">keuze 1
</select>
</form>

```

Resultaat:



Zoals je ziet is de tweede keuze reeds in een blauwe achtergrond geplaatst zonder dat we zelf al een keuze hebben moeten te maken.

Het `selected` attribuut zal op voorhand een keuze aanduiden zoals `checked` dit doet bij de `<input type="checkbox" of type="radio">` tag.

Opmerking: De attributen `checked` en `selected` worden NIEt tussen aanhalingstekens geplaatst!

6.1.3. de <textarea> tag:

Hiermee kunnen we een tekstveld maken dat uit meerdere lijnen kan bestaan.

Opgelet! De eindtag `</textarea>` MOET je gebruiken!

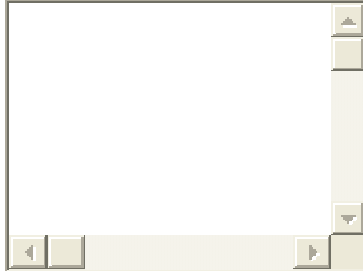
```

<form action="mailto:..." method="post">

```

```
<textarea name="tekstveld" rows="8" cols="30">
</textarea>
</form>
```

Geeft je deze vorm:



In dit tekstveld kun je nu bijvoorbeeld commentaar intypen. Klik maar eens in de box en typ maar wat, je kunt zelfs meerdere regels intypen.

- `rows="..."` bepaalt het aantal rijen (horizontaal)
- `cols="..."` bepaalt het aantal kolommen (verticaal)

Laat ons nu eens een formulier maken met verschillende tags. Het volgende formulier kun je echt naar mij vanaf deze pagina via e-mail zenden (tenminste, als je online bent). Ik zou het eigenlijk wel tof vinden. OK, hier gaan we ... dit is het formulier:

GVR Soft's Enquete

Voornaam:

Achternaam:

E-mail adres:

Heb je een web pagina? Indien ja:

Hoe heb je van deze cursus gehoord?

van een vriend

toevallig gevonden

een link op een pagina

zoekmachine

Andere specificeer

Wat vind je van deze cursus:

ik versta er niets van!

matig

OK, ik heb wel iets geleerd

goed en leerrijk!

Fantastisch!!!

Zou je deze cursus aan iemand anders aanbevelen?

Maak uw keuze

Typ hier je vragen, opmerkingen, suggesties of commentaar:

Dit is de code van het formulier: (uitgezonderd de CSS die ik gebruikt heb)

```
<form action="mailto:tony@GVRSoft.be" enctype="text/plain"
method="post">
<input type="hidden" name="onderwerp" value="cursus">
Voornaam: <input type="text" name="voornaam"
size="15"><br>
Achternaam: <input type="text" name="achternaam"
size="15"><br>
E-mail adres: <input type="text" name="e-mail"
size="15"><br>
Heb je een web pagina? Indien ja: <input type="text"
name="URL" value="http://" size="20"><br><br>
Hoe heb je van deze cursus gehoord?<br>
<input type="checkbox" name="van een vriend">van een
vriend<br>
<input type="checkbox" name="toevallig">toevallig
gevonden<br>
<input type="checkbox" name="een link">een link op een
pagina<br>
<input type="checkbox"
name="zoekmachine">zoekmachine<br>
```

```
<input type="checkbox" name="andere">Andere
specifieer <input type="text" name="anderespec"
size="15"><br> <br>
Wat vind je van deze cursus?<br>
<input type="radio" name="1">ik versta er niets van!<br>
<input type="radio" name="2">matig<br>
<input type="radio" name="3">OK, ik heb wel iets
geleerd<br>
<input type="radio" name="4">goed en leerrijk!<br>
<input type="radio" name="5">Fantastisch!!!<br><br>
Zou je deze cursus aan iemand anders aanbevelen?<br><br>
<select name="aanbeveling">
<option>Maak uw keuze
<option>Ja!
<option>Neen
</select><br><br>
Typ hier je vragen, opmerkingen, suggesties of
commentaar:<br><br>
<textarea name="commentaar" rows="8" cols="35">
</textarea><br><br><br>
<input type="submit" name="verzend" value="OK,
verzend!">
<input type="reset" name="reset" value="Mag ik opnieuw?">
</form>
```

Einde van deze les! Volgende les: frames

GVR Soft's HTMLcursus

--> les 9: <--

7. Frames

7.1. de <frameset> en <frame> tags:

Frames, sommigen houden ervan anderen dan weer niet. Ik persoonlijk hou er wel van en het lijkt er op dat steeds meer en meer mensen ze gaan gebruiken. Laten we beginnen met het maken van een paar bestandjes die we zullen nodig hebben om een constructie met frames te maken. Dit is enkel een voorbeeld natuurlijk. In ons voorbeeld maken we een frames constructie bestaande uit 3 frames, t.t.z. 2 kolommen en 2 rijen.

Maak achtereenvolgens:

- een map met de naam 'frames' (of een naam die je zelf verkiest)
- in deze map zullen we 4 HTML documenten opslaan:
 1. index.html
 2. frame1.html
 3. frame2.html
 4. frame3.html

Laat ons eerst de bestanden "frame1.html, frame2.html en frame3.html" aanmaken. Het zijn eenvoudige HTML bestanden:

<u>frame1.html</u>	<u>frame2.html</u>	<u>frame3.html</u>
<html>	<html>	<html>
<head>	<head>	<head>
<title>	<title>	<title>
</title>	</title>	</title>
</head>	</head>	</head>
<body>	<body>	<body>
Dit is frame 1	Dit is frame 2	Dit is frame 3
</body>	</body>	</body>
</html>	</html>	</html>

Sla deze bestandjes in je map op als respectievelijk "frame1.html, frame2.html en frame3.html".

Nu maken we ons hoofdbestand 'index.html' aan. Dit doen we als volgt (de rode tekst is enkel mijn commentaar, dus die moet je niet typen):

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
```

```

Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd">
<html>
<head>
<title>
</title>
</head>
<frameset cols="25%,75%">
<frame src="frame1.html"> <-- inhoud kolom 1
  <frameset rows="15%,85%"> <-- maak twee rijen in de
tweede kolom
  <frame src="frame2.html"> <-- inhoud rij 1, kolom 2
  <frame src="frame3.html"> <-- inhoud rij 2, kolom 2
  </frameset>
</noframes>
  Uw browser ondersteunt geen frames, klik hier om ....
</noframes>
</frameset>
</html>

```

Typ dit bestand in je kladblok in en sla het in je map op als 'index.html'.

In je map 'frames' (of de map die je zelf hebt gekozen) moeten nu 4 bestanden aanwezig zijn. Dubbelklik op het bestand 'index.html' om je eerste frames constructie te bekijken.

Laten we nu de tags eens van dichterbij bekijken:

We kunnen onze pagina verdelen in kolommen en rijen.



Als we een frames constructie willen maken dan beginnen we met de `<frameset>` tag.

Hierbij kunnen we de keuze maken tussen *cols* (kolommen) en *rows* (rijen). In ons voorbeeld zeggen we tegen de browser: 'Maak twee kolommen':

```
--> <frameset cols="25%,75%">
```

De eerste kolom zal 25% van de pagina in beslag nemen, de tweede kolom krijgt 75%.

Opmerking: In ons voorbeeld maken we gebruik van percentages maar er zijn nog andere mogelijkheden:

- absolute waarden: `<frameset cols="100,300">`
- of deze vorm:
bijvoorbeeld `<frameset cols="25%,*">` of `<frameset cols="150,*">`
Het * vult de rest van de ruimte aan. Dus als we bijvoorbeeld

deze tag zouden zien:

`<frameset cols="25%,*,5%">` dan wil dit zeggen:
kolom 1 krijgt 25% van de pagina, kolom 3 krijgt 5% en
kolom 2 krijgt de rest, in dit geval, 70% want
 $25\%+70\%+5\%=100\%$. Het * is erg handig als je gebruik
maak van complexe frame constructies.

Dus we hadden tegen de browser gezegd: 'Maak twee kolommen'. Nu
geven we aan dat de eerste kolom 'frame1.html' moet bevatten door
te zeggen:

```
<frameset cols="15%,75%">
--> <frame src="frame1.html">
src betekent 'bron' en geeft aan welk document moet gebruikt
worden.
```

Nu hebben we dus al twee kolommen gedefinieerd en weten we dat
kolom 1 document 'frame1.html' zal bevatten. Onze tweede kolom
gaan we nu verdelen in 2 rijen. Dit doen we als volgt:

```
<frameset cols="15%,75%">
<frame src="frame1.html">
--> <frameset rows="15%,85%">
```

De eerste rij in onze tweede kolom krijgt 15% van de ruimte
toebedeeld en de tweede rij krijgt 85% van de ruimte. Nu moeten we
enkel de browser nog vertellen welke inhoud de twee rijen zullen
bevatten:

```
<frameset cols="15%,75%">
<frame src="frame1.html">
<frameset rows="15%,85%">
-> <frame src="frame2.html">
-> <frame src="frame3.html">
```

De eerste rij van de tweede kolom bevat de informatie van
'frame2.html', de tweede rij van de tweede kolom bevat de informatie
van 'frame3.html'.

De tag `<noframes>` kan je gebruiken (maar hoeft niet) als
alternatief voor een browser die om $\omega\omega\omega$ of andere reden geen
frames ondersteund.

```
--> <noframes> Plaats hier de inhoud </noframes>
```

`</noframes>` is **verplicht** als je de `<noframes>` tag gebruikt!

Opgelet! Er is een eindtag voor de `<frameset>` tag namelijk de
`</frameset>` tag en die **MOET** je gebruiken voor elke `<frameset>`
tag die je gebruikt! Let er ook steeds op dat je de tags correct NEST
zoals je dit ook moet doen bij andere tags. Weet je het nog?

GOED: (1-2-3-3-2-1)

```
<html><head><title></title></head></html>
```

GOED: (1-2-3-3-2-4-4-1)

```
<html><head><title></title></head><body></body></html>
```

FOUT: (1-2-3-1-2-3)

```
<html><head><title></html></head></title>
```

Als je dit niet doet dan zul je rare of helemaal geen constructies krijgen!

Nu we alles netjes verdeeld hebben kunnen we eender welk document aan eender welk deel van de frames constructie toekennen. Handig niet?

Er zijn verschillende attributen die we aan de `<frame>` kunnen toevoegen. Laten we ze eens bekijken:

- `frameborder=...` Dit attribuut kun je gebruiken om de kaders in de frames constructie weg te laten of om ze te laten zien.

Typ in het bestand 'index.html' dit maar eens in:

```
--> <frame src="frame1.html" frameborder="0"> (ook  
in de andere 2 <frame> tags) en herlaad de pagina in je  
browser. Zoals je ziet zijn de 'metalen' kaders verdwenen.  
Gebruik je frameborder="1" dan zijn ze wel te zien.
```

- `marginwidth=...` en `marginheight=...` Deze twee attributen definiëren de marges in een bepaald frame.

`marginwidth=...` definieert de marge (links en rechts) tussen de tekst en het frame.

`marginheight=...` definieert de marge (boven en onder) tussen de tekst en het frame.

Gebruik steeds absolute waarden groter dan 1. Voorbeeld:

```
--> <frame src="frame1.html" marginwidth="15"  
marginheight="20">
```

- `scrolling="auto/yes/no"`. Dit attribuut vertelt de browser een 'scrolling' mechanisme te gebruiken of niet. Een 'scrolling' mechanisme is de schuifbalk die je gebruikt om door een pagina heen en weer te bewegen (meestal verticaal). Standaard staat dit attribuut ingesteld op 'auto' wat automatisch betekent, de browsers ziet zelf of een schuifbalk nodig is of niet. Wil je geen 'scrolling' (geen schuifbalk) dan gebruik je 'no'. Wil je wel een schuifbalk dan gebruik je 'yes'. Een voorbeeld:

```
--> <frame src="frame1.html" scrolling="no">
```
- `noresize`. Gebruik dit attribuut als je niet wil dat de lezer de vensterlengte of vensterbreedte kan veranderen. Het werkt net zoals op je desktop. Als je een map op je desktop wilt vergroten of verkleinen dan kun je dit doen door met je muis op de rand van de de map te gaan staan totdat er pijltjes verschijnen. Met deze pijltjes kun je dan de mapgrootte aanpassen. Hetzelfde kan je doen met frames. Probeer maar eens. Zet het `frameborder=...` attribuut op 1 en herlaad 'index.html' zodat je terug de metalen kaders te zien krijgt. Zet de muiscursor op \updownarrow van de kaders (zodat je pijltjes ziet) en probeer de lengte of breedte van het frame aan te passen. Als je niet wil dat de lezer de vensterlengte of vensterbreedte kan veranderen dan gebruik je `noresize`. Voorbeeld:

```
--> <frame src="frame1.html" scrolling="no" noresize>
```
- `name="..."`. Dit is een belangrijk attribuut! Als je bijvoorbeeld

wil dat een bepaald document in een bepaald frame (venster) opent dan moet de browser weten in welk frame. Dit doe je door een bepaald frame een bepaalde naam te geven. Als de lezer bijvoorbeeld een link in frame 1 aanklikt en je wil dat de inhoud tevoorschijn komt in frame 3 dan doen we dit zo:

In ons 'index.html' document typen we dit:

```
--> <frame src="frame3.html" name="frame3">
```

We maken nog een bestandje aan (sla dit op in je map als 'welkom.html'):

```
<html>  
<head>  
<title>  
</title>  
</head>  
<body>
```

Hallo, welkom surfer. Dit is mijn webpagina, met 'cool' stuff en ...

```
</body>  
</html>
```

In ons 'frame1.html' bestand creëren we een link:

```
<a href="welkom.html" target="frame3">klik hier!</a>
```

Herlaad nu terug je 'index.html' bestand en klik op de link 'Klik hier!' en zie wat er gebeurt. Als je alles correct hebt uitgevoerd dan zal de inhoud van 'welkom.html' in frame 3 openen. Is dat 'cool' of niet? We kunnen nu eender document in eender welk frame (venster) laten openen. We moeten enkel de `name="..."` attribuut gebruiken in ons hoofddocument (in ons geval 'index.html') en er later naar verwijzen met het `target="..."` attribuut in onze link.

Met frames kun je eenvoudige en complexe constructies bouwen. Meestal volstaat een eenvoudige constructie wel, soms is een iets complexere structuur nodig. Maak het jezelf niet moeilijker dan nodig. Een eenvoudige constructie geeft je al heel wat controle over de layout van je pagina! Een goede raad: probeer eerst een eenvoudige constructie en ga zo verder naar een iets moeilijkere constructie (indien nodig natuurlijk).

O ja, er is nog zoiets als de `<iframe>` tag (inline frame tag) maar die werkt voorlopig niet universeel. Ik hou je op de hoogte ...

Op naar de volgende les! Les 10: Interessant om weten.

GVR Soft's HTMLcursus

--> les 10: <--

8. Interessant om weten

8.1. Het <lang> attribuut:

Specificeert de gebruikte taal in een document.

Een voorbeeld:

```
<html lang="fr">
<head>
<title>
</title>
</head>
<body>
Dit is Frans ...
<p lang="en">
Engels ...
</body>
</html>
```

Dit attribuut wordt voor verschillende doeleinden gebruikt: om zoekmachines te helpen bij het klasseren van pagina's, bij spraakcomputers, spelling en grammatica onderzoek, enz... De verschillende taalcodes (zoals "fr" voor Frans) kun je vinden op de [W3C](#) website.

8.2. de <meta> tag:

Veelvuldig gebruikt om web pagina's beter of hoger te 'ranken' (rangschikken) in zoekmachines, zodat je meer bezoekers kan aantrekken.

Opmerkingen:

- Er is GEEN eindtag voor de <meta> tag!
- De <meta> tag staat tussen de </title> en de </head> tag.

Er zijn er heel veel, ik haal enkel de belangrijkste aan.

- `name="keywords" content="..."`. Dit attribuut laat je toe een aantal keywords (sleutelwoorden) over jouw pagina te specificeren. In het geval van deze cursus zouden we dit kunnen gebruiken:
--> `<meta name="keywords" content="cursus, HTML, html 4.0, style sheets, web pagina, ...">`
- `name="description" content="..."`. Dit attribuut laat je toe een beschrijving over jouw webpagina te specificeren.

--> `<meta name="description" content="Maak je eigen web pagina met GVR's HTML cursus. Alles wat je nodig hebt om eenvoudige of complexe web pagina's te maken ... ">`

- `name="author" content="..."`. Specificeert de auteur van de web pagina.
--> `<meta name="author" content="Tony">`
- `name="robots" content="index, follow"`. 'Robots' zijn mechanismen die een web adres (URL) met de daarbij horende links indexeren. 'index, follow' wil zeggen: je mag het web adres (index) en de daaropvolgende links (follow) indexeren. Je kan ook 'noindex, nofollow' specificeren als je dit niet wil.
- `meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-5"`. Specificeert de inhoud en karakterset van de web pagina.
- `meta http-equiv="refresh" content="3,http://www.mijnpagina.com">`. Dit is een interessante tag! In dit voorbeeld zal de bezoeker na 3 seconden naar het adres `http://www.mijnpagina.com` geleid worden. Erg handig als je bijvoorbeeld een openingspagina wil tonen om daarna de bezoeker mee te nemen naar de eigenlijke pagina. Gebruik deze tag ook als je site van adres (URL) is veranderd om de bezoeker naar je nieuwe stek te leiden!

Een voorbeeldje:

```
<html>
<head>
<title>
Mijn web pagina
</title>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-
8859-5"> <meta name="keywords" content="cursus, HTML, html 4.0,
style sheets, web pagina>
<meta name="description" content="Maak je eigen web pagina met
GVR's HTML cursus. Alles wat je nodig hebt om eenvoudige of
complexe web pagina's te maken">
<meta name="robots" content="index, follow">
</head>
<body>
Hier komt de inhoud
</body>
</html>
```

Zoals je ziet mag je verschillende 'meta' tags tegelijkertijd gebruiken. Of je web site er beter door gerangschikt wordt blijft nog maar de vraag. Sommige zoekmachines gebruiken de meta tags om je site te indexeren, andere dan weer niet. Er is gewoon geen garantie dat je site binnen de top tien van een zoekmachine komt te staan, laat je niks wijsmaken. Ik persoonlijk gebruik sommige meta tags, het kost niets en het is een kleine moeite om ze te gebruiken.

8.3. de <link> tag:

Deze tag creëert een link **maar** je ziet ze niet en je mag ze enkel in de `<head>` sectie gebruiken! Ik heb deze tag niet aan [les 6](#) (over links) toegevoegd om niet voor verwarring te zorgen. Hoe moet je deze tag dan wel gebruiken? Een woordje uitleg:

De `<link>` tag wijst eerder op een 'relatie' tussen de link en het document. In deze cursus maak ik gebruik van Cascading Style Sheets (CSS) voor het weergeven van tekst, zoals de kleur, de grootte, enz... Ik heb daarvoor een apart CSS bestand aangemaakt dat alle style elementen bevat die ik gebruik in deze cursus. Dit bestand heb ik 'cursus.css' genoemd (dit mag ook een andere naam zijn). Het zijn eigenlijk style elementen die ik gewoon in de HTML code zou kunnen gebruiken maar dan zou ik ze telkens moeten hertypen en dit zou veel tijd kosten. In plaats daarvan heb ik ze verzameld in `css` bestand en maak ik steeds een 'link' naar het bestand zodat de browser weet waar hij de style elementen kan vinden. In de code van elke pagina van deze cursus staat in de HTML code deze lijn:

```
<link href="cursus.css" rel="stylesheet" type="text/css">
```

- `href="cursus.css"` vertelt de browser dat hij het bestand "cursus.css" aan het hoofddocument moet linken (koppelen). Telkens de browser nu een style element (CSS) nodig heeft weet hij dat ze zich bevinden in "cursus.css".
- `rel="stylesheet"` specificeert de 'relatie' tussen het document en de link (href). In dit geval 'stylesheet'.
- `type="text/css"` specificeert het inhoudstype dat gebruikt wordt (in dit geval CSS).

Mijn cursus.css bestand ziet er als volgt uit: (ingekort)

```
...
body { background:#000000; color:#ffffff }
.rood { background:#ffffff; color:#ff0000; }
.titel { background:#ffffff; color:#c00000; text-align:center; text-decoration:underline; font-family: 'Comic Sans Ms', Verdana, Arial, serif; font-size: 32px; font-weight:bold }
.subtitel { background:#ffffff; color:#01955c; text-align:left; text-decoration:underline; font-family: Verdana, 'Comic Sans Ms', Arial, serif; font-size: 20px; font-weight:normal }
...
```

Opmerking: Een `<link>` tag zoals hierboven beschreven wordt steeds vaker gebruikt omdat het gebruik van CSS in de lift zit!

Opmerking: Er is GEEN eindtag voor de `<link>` tag!

Je kan de `<link>` tag ook gebruiken om zoekmachines (vb. Altavista) te vertellen waar alternatieve versies van een document te vinden zijn. Neem aan dat deze cursus ook bestaat in het Engels. Dan kan ik de zoekmachines vertellen dat er naast deze Nederlandstalige versie ook een Engelstalige versie op het web te vinden is.

```
<link title="The HTML tutorial in English" type="text/html"
rel="alternate" hreflang="en" href="http://daar waar de Engelstalige
cursus zich bevindt">
```

- `title="..."` specificeert de titel van de 'link'.
- `type="text/html"` specificeert het inhoudstype dat gebruikt wordt. (in

- dit geval HTML code)
- `rel="alternate"` specificeert de 'relatie' tussen het document en de link (href). (in dit geval 'alternate' --> 'alternatief')
 - `href="..."` vertelt de browser waar hij het bestand kan vinden (een web adres of URL).
 - `hreflang="..."` specificeert de landstaal van het document. (in dit geval 'en' --> 'English')

8.4. de <base> tag:

Een handige tag. Deze tag vergemakkelijkt het linken naar een bepaald iets. Neem onderstaand voorbeeld:

```
<html>
<head>
<title>
Onze producten
</title>
<base href="http://www.hier.com">
</head>
<body>
Bekijk onze nieuwe <a
href="../afbeeldingen/computer.gif">computer</a>nu!
</body>
</html>
```

In onze 'hoofd' sectie (tussen '/title' en '/head') maken we een *hoofdlink*. Deze is in bovenstaand voorbeeld <http://www.hier.com> genoemd. Daar zorgt de `<base>` tag voor.

In de 'body' sectie zien we een nog een andere link namelijk:

```
<a href="../afbeeldingen/computer.gif">computer</a>
```

Dit noemt men een *relatief* (../) web adres (URL).

De ../ zal het adres <http://www.hier.com> aannemen omdat we dit gespecificeerd hebben in de `<base>` tag.

Dus:

```
<a href="../afbeeldingen/computer.gif">computer</a>
IS GELIJK AAN
<a
href="http://www.hier.com/afbeeldingen/computer.gif">computer</a>
```

Erg handig als je met lange web adressen moet werken!!

Opmerking: Er is GEEN eindtag voor de `<base>` tag!

8.5. de <script> tag:

Zoals je wellicht weet is HTML niet de enige 'web'taal die gebruikt wordt. Er zijn er nog andere, waaronder CSS, javascript, vbscript en nog een paar. Soms is het nodig om een andere taal in een HTML document te integreren (die perfect samenwerken). Stel dat je met javascript (ook een programmeertaal) een programmaatje gemaakt hebt dat een mooie klok op je web pagina kan tonen. Je kan dit in je HTML document als volgt inpassen:

```
<html>
<head>
<title>
</title>
<script type="text/javascript" src="http://www.hier.com/klok.js">
</script>
</head>
<body>
</body>
</html>
```

Hier wordt de `<script>` tag gebruikt in de 'hoofd' (head) sectie en zal de browser het javascript programma ophalen op het `src` attribuut, in dit geval: `http://www.hier.com`

Je kan de `<script>` tag ook gebruiken in de 'body' sectie als het nodig is de taal in het HTML document zelf te schrijven.

```
<html>
<head>
<title>
</title>
</head>
<body>
<script type="text/javascript">
Hier komt de javascript code
</script>
</body>
</html>
```

Merk op dat er in de 'body' sectie geen `src` attribuut gebruikt wordt omdat de javascript code in het HTML document zelf wordt geschreven.

Opmerking: Er is een eindtag voor de `<script>` tag namelijk de `</script>` tag en die MOET je gebruiken bij iedere `<script>` tag!

Voor we deze cursus afsluiten toch nog een belangrijke opmerking! Soms is het nodig dat we aparte HTML karakters kunnen gebruiken. Een paar voorbeelden:

- Als je in je HTML code meer dan ` ` spatie ruimte wilt creëren (tussen 2 woorden bijvoorbeeld) dan zal dit niet gaan door in de code zelf de twee woorden te typen en daartussen een paar keer de spatiebalk te drukken! Probeer maar eens. Maak een klein bestandje en in je 'body' sectie maak je bijvoorbeeld een lijn met twee woorden. Je zal zien dat, hoe vaak je ook een spatie maakt met de spatiebalk, je browser maar ` ` enkele spatie zal gebruiken en tonen! Dit kunnen we oplossen door volaend HTML karakter in

