

Webontwerp voor beginners

Harry van Mosseveld

Er zijn vele manieren om een website te bouwen.

Via een Content Management Systeem (CMS), via een webbouwer van de Hosting Provider of zelf doen via HTML-codes.

Van mijn collega's van de groep 'Webontwerp' kreeg ik een tip dat het editprogramma KompoZer gratis te gebruiken is.

Ik ben bekend met Adobe Dreamweaver, maar dat is niet gratis.

Dus ben ik me nu aan het inleren hoe je met KompoZer kunt werken.

Menu Bar

Composition Toolbar

Format Toolbar 1

Format Toolbar 2

Page tab

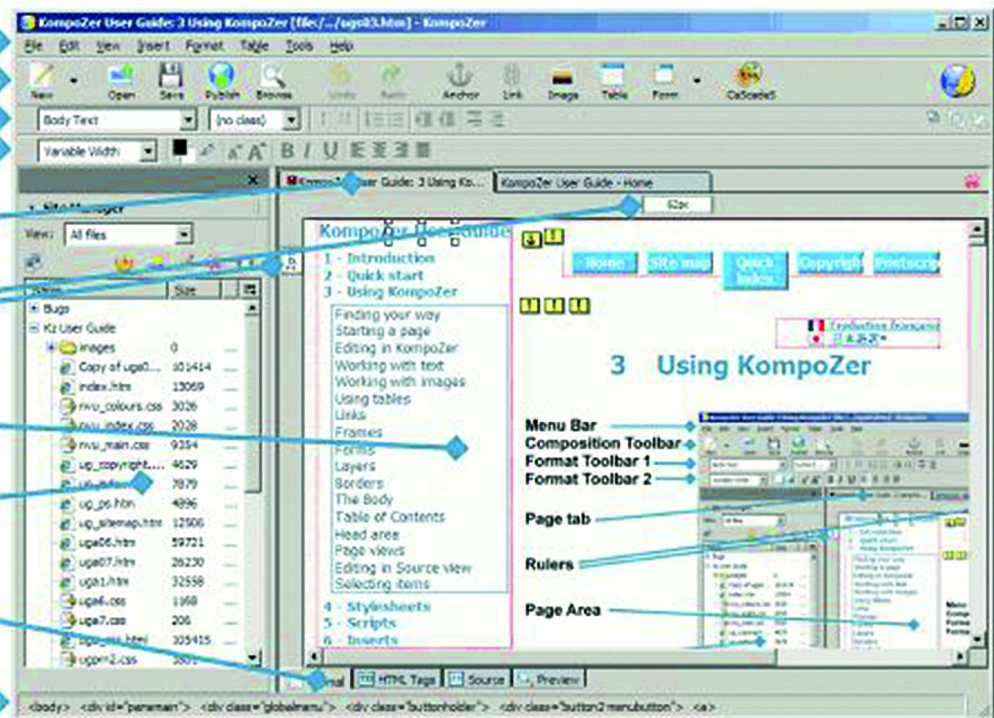
Rulers

Page Area

Site Manager

Edit mode toolbar

Status bar



HTML en CSS

Om met HTML-codes te kunnen werken is het nodig om kennis te hebben van de structuur van een website. In dit artikel wil ik het daar over hebben: het weten hoe een website eigenlijk in elkaar zit.

Een webbrowser, zoals Chrome, Edge, Safari, Firefox, enz., moet de inhoud van een webpagina kunnen weergeven zoals de auteur die bedoeld heeft. Daarvoor moeten alle gegevens voor de opmaak van de pagina aanwezig zijn.

De opmaakinstructies gebruiken twee 'talen'. De tekst moet geschreven zijn in codetaal 'Hyper Text Markup Language' (HTML). De visuele opmaak wordt in de codetaal 'Cascading Style Sheets' (CSS) geschreven.

In KompoZer, waarover ik later terugkom, kun je via de editor de tekst opnemen en die editor zet de codes automatisch in HTML, c.q. in CSS.

Eerst wil ik wat uitleggen over de codetalen om uit te leggen hoe de structuur van een webpagina in elkaar steekt. HTML bestaat uit code waarin onder andere de inhoud/tekst staat. Die code bestaan uit tags, elementen en attributen.

HTML-tags

Een stukje tekst op een webpagina bestaat uit de tekst die we lezen en een HTML-code, die om de tekst heen staat. Bij een alinea staat vóór de tekst de HTML-tag <p>. De 'p' is van het Engelse 'paragraph', dat alinea betekent.

<p>Dit is een stukje tekst.</p>

HTML-elementen

De begin- en eindtag vormen samen het HTML-element. In het vorige voorbeeld heb je dus het element p en de begintag <p> en de eindtag </p>.

Een ander belangrijk element zijn headings, zoals de h1. De 'h' staat hier voor heading, ofwel kopregel. Het cijfer erachter geeft de belangrijkheid van de kopregel aan. Des te hoger het cijfer, hoe lager het belang.

Er is niet altijd een begintag en een eindtag. Sommige elementen hebben geen eindtag. Dit is bijvoorbeeld het geval bij het element voor een afbeelding, het element img:

Dit element verzorgt een nieuwe regel in je tekst:

Postbus 12
1224 AA AALSMEER

HTML-attributen

Een element kan ook 'attributen' hebben. In het voorbeeld hierboven heeft het img-element het attribuut 'src', dat de directory en naam van de afbeelding aangeeft.

Elk element heeft bepaalde attributen. Zo heeft een afbeelding het attribuut voor een alternatieve tekst: het alt-attribuut:

```

```

Commentaar in HTML

In HTML kun je tekst ook markeren als commentaar. Die tekst wordt dan niet op de website getoond, maar is alleen te zien in de broncode.

Hiervoor gebruik je '`<!--`' aan het begin en '`-->`' aan het eind van de tekst die je niet wilt tonen. De code hiervoor is:

```
<!-- Deze tekst is even uitgezet. -->
```

Tot zover het begin over HTML. Nu de lay-out van je website regelen.

CSS

In de CSS is de opmaak van de teksten en andere onderdelen van de site vastgelegd. Ook regel je er eventueel responsive lay-outs mee.

In de CSS is vastgelegd welke opmaak er hoort bij welke tag, bijvoorbeeld bij de h1:

```
h1 {
font-size: 1.3em;
font-weight: bold;
}
```

In bovenstaande CSS-code staat:

- De grootte van de tekst van de h1 is 1,3 x de standaardgrootte.
- De tekst is **vetgedrukt**.

De browser voegt de HTML samen met de CSS tot een leesbare en opgemaakte titel.

Het werken met CSS is heel handig, want je definieert met de CSS de huisstijl van de website. Ook leg je bijvoorbeeld posities van blokken tekst en afbeeldingen vast.

De CSS regelt onder andere de opmaak van:

- Lettertipes
- Tekstkleur
- Achtergrondkleur
- Alinea-afstand
- Positie van tekst en andere elementen, zoals afbeeldingen
- Weergave knoppen in formulier

Tot slot: met CSS regel je ook de lay-out op de verschillende schermen die zoal in gebruik zijn. Zo maak je je website dus **responsive**: de site past zich aan aan de grootte van het beeldscherm van de gebruiker. (Desktop, Laptop, Tablet of Telefoon).

Opnemen van de CSS

Je kunt CSS op meer manieren opnemen in je site; de meest voor de hand liggende is:

- CSS als apart bestand (Externe style sheet), dit bestand staat ook in de website, net zoals de bestanden van je afbeeldingen.

Zoals gezegd, dé manier om de CSS te koppelen aan de pagina is de CSS op te nemen in een apart bestand, en in het head-gedeelte van de pagina te verwijzen naar dit bestand.

In code ziet dat er bijvoorbeeld zo uit:

```
<link rel="stylesheet" href="../styles/default.css"
type="text/css" media="screen">
```

Als je nu een alinea op een bepaalde manier opgemaakt wilt hebben, hoeft je alleen in de CSS de opmaak voor p-element op te nemen:

```
p {
margin-bottom: 1.3em;
}
```

Ook kun je een class of id bij het element opnemen. Via de CSS kun je hier dan opmaak aan toevoegen. Dus bijvoorbeeld in de HTML voeg je de class 'summary' toe:

```
<p class="summary">Stukje tekst</p>
```

In de CSS staat dan de volgende code:

```
.summary {
margin-bottom: 1.3em;
font-weight:bold;
}
```

- CSS-code opgenomen in de HTML-tag (Inline style)

De CSS-code is geplaatst in het style-attribuut:

```
<p style="margin-bottom:1.2em">Stukje tekst</p>
```

De nadelen van het opnemen van CSS via een inline style zijn:

- De opmaak staat toch in de HTML, het aanpassen van de opmaak moet in elke pagina.
- De opmaak is niet responsive. De gebruiker kan geen eigen opmaak kiezen.

- CSS als stijlblok in de head (Interne style sheet)

Bij deze manier neem je de CSS in een stijlblok op in de head. Een voorbeeld:

```
<head>
[... ]
<style>
p {margin-bottom:1.2em;}
</style>
[... ]
</head>
```

Ook nu weer de nadelen:

- niet-responsief
- aanpassen van de opmaak moet in elke pagina.

Samengevat: de voordelen van het gebruik van een externe stylesheet zijn:

- De opmaak kan eenvoudig worden aangepast, slechts eenmaal: en wel in het CSS-bestand.
- De opmaak kan gevarieerd worden afhankelijk van de beeldschermgrootte (responsive design).

Nu we dit weten over HTML en CSS kunnen we verder met de structuur van de website.

Basisopbouw

De basisopbouw van 'n webpagina bestaat uit Head en Body.

De HTML-code opent en sluit met het HTML-element:

```
<html>
[... ]
</html>
```

Daartussen staan twee delen:

- head-gedeelte met de meta-informatie
- body-gedeelte met alle content

Deze delen hebben ook hun eigen element. De opbouw is als volgt:

```
<html>
<head>
[allerlei meta-informatie]
</head>
<body>
[de content van de pagina]
</body>
</html>
```

Vóór het HTML-element staat nog de zogenaamde DOCTYPE-declaratie. Daarin is aangegeven welke HTML-versie de website gebruikt. Voor HTML5 is dat:

```
<!DOCTYPE html>
```

Het HTML-element bevat ook de taalkeuze ('lang') van de pagina; bv. voor een Nederlandstalige site is dat 'nl':

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

Head-gedeelte

In de Head staat meta-informatie: de titel van je pagina is geen onderdeel van de content van de pagina, maar wordt getoond in de browserbalk.

De titel van de site-pagina is belangrijk. Google gebruikt de titel bij het tonen van het zoekresultaat. Daarom is het ook heel belangrijk dat de titel betekenisvol is en uniek binnen de site.

Het is onwenselijk dat twee verschillende pagina's exact dezelfde titel hebben, want voor Google is het dan onduidelijk welke pagina bij welke titel hoort. Niet goed is het om een lange 'titelnaam' te gebruiken.

Metatags

In het head-gedeelte staan vaak **metatags**. De bekendste metatags zijn:

- description: beschrijving van de pagina
- robots: instructies of zoekmachines die de pagina moeten indexeren

Metatag description

In code ziet deze er bijvoorbeeld als volgt uit:

```
<meta name="description" content="Het weekmenu toont vijf menu's voor alle werkdagen.">
```

(Een website die elke week elke werkdag een menu toont.)

Deze metatag is niet belangrijk voor de vindbaarheid, maar wel belangrijk voor de zichtbaarheid van jouw site, want zoekmachines presenteren de inhoud van deze tag soms bij de zoekresultaten.

Metatag robots

Deze tag gebruik je als je wilt dat een zoekmachine iets niet doet. Hieronder een voorbeeld van de code.

```
<meta name="robots" content="noindex, nofollow">
```

In dit voorbeeld is de wens dat zoekmachines de pagina NIET indexeren (**noindex**) en de links in de pagina NIET volgen (**nofollow**).

Het nadeel van zo'n metatag is dat je deze moet toevoegen op elke pagina die je niet wilt indexeren. Wil je het voor je hele site of een hele directory laten gelden, dan kan het handig zijn om de HTTP-header **X-Robots-Tag** te gebruiken.

Links naar bestanden

In het head-gedeelte staan ook enkele links naar bestanden. Bij het opragen van de pagina worden deze bestanden ook opgehaald. Het zijn onder andere links naar CSS-bestanden, bijvoorbeeld:

```
<link rel="stylesheet" href="/styles/screen.css" media="all">
```

en naar JavaScript-bestanden, bijvoorbeeld:

```
<script src="/scripts/jquery-1.7.js"></script>
```

Body-gedeelte

Na het head-gedeelte van de webpagina komt het body-deel. Dit is omsloten met het element **body**:

```
<body>
[content]
</body>
```

In het body-deel staat al je content: teksten, afbeeldingen, video's en muziek, tabellen.

Ik maak redelijk veel gebruik van **Tabel**. Daarmee breng ik structuur aan in mijn pagina. Ik gebruik dan de cellen van de tabellen om de tekst, foto, video, enz., een plek te geven. Hieronder nog enkele tabellen met gekleurde achtergrond en ruimte tussen de inhoud van de vakken:

Hiernaast staat een tabel met een gekleurde achtergrond. Dit gebeurt hier niet in de TABLE-tag, maar in de beide TR-tags. Er is tussen de vak-inhoud en de rand een tussenruimte van 15 pixels. De ruimte tussen de vakken is op 10 pixels gesteld. De code daarvoor luidt:

```
<table cellspacing="10" cellpadding="15" border="0">
<tr bgcolor="#9999FF">
  <td>eerste vak<br>(in de eerste rij)</td>
  <td>tweede vak<br>(in de eerste rij)</td>
</tr>
<tr bgcolor="#9999FF">
  <td>derde vak<br>(in de tweede rij)</td>
  <td>vierde vak<br>(in de tweede rij)</td>
</tr>
```

In plaats van die tekst in de cellen van de tabel, kun je daar de code voor een plaatje of bijvoorbeeld een video plaatsen. Ook is het mogelijk om in een cel weer een code voor een nieuwe tabel te plaatsen. Zo kun je heel precies plaatsen waar alles te zien zal zijn op de website.

Opmerking: in plaats van `<tr bgcolor="#9999FF">` kan er ook `class="bgcolor"` staan om de kleur in CSS vast te leggen.

Het is handig om 'inspringen' te gebruiken om structuur aan te brengen. Je kunt dan de verschillende elementen beter herkennen.

Er zijn heel veel codes, maar dat is hier nu niet te doen.

Links

https://nl.wikipedia.org/wiki/Lijst_van_HTML-elementen

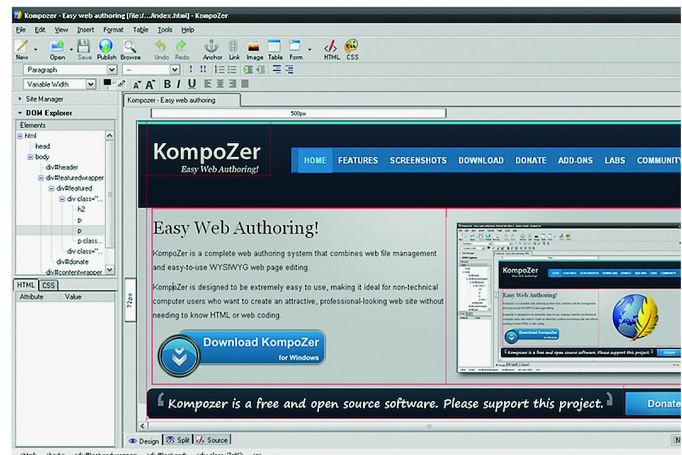
Een andere website met voorbeelden:

<https://www.w3schools.com>

In bovenstaande links vind je alle codes die je nodig hebt.

Nu is het alleen nog nodig om een editorprogramma te hebben om al die codes te kunnen typen.

Er zijn er vele; je kunt werken in *Kladblok* of *Word* en elke tekstverwerker is bruikbaar.



Wel is het handig om een editor te hebben die volgens het beginsel WYSIWYG (What You See Is What You Get) werkt. Bij voorkeur een editor die zowel de bron als het ontwerp laat zien. KompoZer is zo'n editor. *Daarover een volgende keer.*

eerste vak (in de eerste rij)	tweede vak (in de eerste rij)
derde vak (in de tweede rij)	vierde vak (in de tweede rij)