

9 Maak zelf een WordPress theme

Dit hoofdstuk wordt als extra download gratis aangeboden aan lezers van het boek *Kickstart WordPress* door Robbert Ravensbergen, in 2009 verschenen bij Pearson Education, ISBN 9789043019538. Het is mede gebaseerd op de Engelse tutorials van Themetation.com.

Je kunt het theme en de bronbestanden die horen bij dit hoofdstuk downloaden op www.robberoverwordpress.nl en op www.pearsoneducation.nl/downloads.asp

In dit hoofdstuk zal ik de basisbeginselen van WordPress-themeontwerp uitleggen. Het voert hier te ver om ook alle benodigde kennis van Photoshop, (X)HTML, CSS en PHP te bespreken. Enige basiskennis over deze onderwerpen is absoluut een voordeel wanneer je zelf gaat ontwerpen voor WordPress. Ik kan de volgende boeken, verschenen bij Pearson Education, aanbevelen wanneer je hier meer over wilt weten:

- *HTML en CSS, 2e editie – de Basis* door Andree Hollander
- *Het beste van CSS* door Ward van der Put
- *Snel op weg HTML, XHTML en CSS* door Elisabeth Castro.
- *PHP5 en MySQL – de Basis* door Ghica van Emde Boas

Zelfs wanneer ik de basiskennis van de hierboven genoemde onderwerpen oversla is er nog genoeg te leren en zal ik hier niet alles kunnen behandelen. Over het maken van WordPress themes alleen al zijn (Engelstalige) boeken verschenen. Ik zal me hier daarom tot de kern moeten beperken. Je zult echter ook merken dat wanneer je al wat ervaring hebt met het

maken van websites en/of het programmeren in PHP dat een basisthema voor WordPress niet heel erg ingewikkeld hoeft te zijn. En nu snel aan de slag!

Tools

Ik herhaal hier nog even wat je nodig hebt om serieus aan de slag te gaan met themeontwerp of websiteontwikkeling in het algemeen.

- Een Ftp-programma
- Een goede editor zoals UltraEdit of Dreamweaver
- Photoshop en/of Illustrator of vergelijkbare beeldbewerkingssoftware. Je zult echter zien dat negen van de tien tutorials die je op internet tegenkomt gebaseerd zijn op het gebruik van Photoshop.
- Een lokale testomgeving. Gebruik bijvoorbeeld Xampp.
- Gebruik FireFox en installeer de Webdeveloper toolbar-plugin, eventueel aangevuld met Firebug.
- Een ontwerp gemaakt in bijvoorbeeld Photoshop
- Doorzettingsvermogen. Het maken van themes is een tijdrovend karwei, zeker wanneer je nog maar weinig ervaring hebt met het zelf ontwerpen en coderen van websites.

Om te beginnen

Een WordPress-thema kent een vrij eenvoudige structuur. Het bevat minimaal de volgende bestanden:

- `index.php`
- `style.css`
- `header.php`
- `footer.php`
- `sidebar.php`

`Index.php` is in feite het hoofdbestand. `Style.css` bevat alle CSS-elementen waarmee bepaald wordt hoe je website er exact uit komt te zien. De meeste WordPress-themes bestaan echter uit meer dan deze vier bestanden. Dat komt doordat het voor de ontwikkelaar eenvoudiger is een structuur aan te brengen en bijvoorbeeld de WordPress-archieffunctie aan te roepen in een bestand dat je `archives.php` noemt. Je kunt in feite zo veel bestanden gebruiken als je maar wilt. Voor de werking van het thema is dat echter niet noodzakelijk. Het aantal bestanden zegt dan ook niets over de kwaliteit van het thema.

Voordat we enthousiast aan de slag gaan met het intikken van PHP- en CSS-code: hoe moet het theme er eigenlijk uit gaan zien? Inderdaad, we hebben een ontwerp nodig. Maar dat had je zelf vast ook al bedacht. Ik ga me in dit hoofdstuk niet helemaal richten op het maken van een websiteontwerp met Photoshop. Daar zul je zelf mee aan de slag moeten! Ik heb wel enkele (Engelstalige) websites voor je, waarin stap voor stap wordt uitgelegd hoe je dit aanpakt:

- Six Revisions: <http://sixrevisions.com/tutorials/>
- Grafpedia: www.grafpedia.com/tutorials/create-a-wordpress-interface-in-photoshop
- HV Designs: <http://hv-designs.co.uk/2009/07/02/wordpress-layout-5/>



AFBEELDING 9.1 Het ontwerp uit de tutorial van Six Revisions.

Kies bij het maken van het ontwerp van je theme voor één van de volgende basis lay-outs.



AFBEELDING 9.2

Natuurlijk kun je het ingewikkelder maken en voor vier kolommen kiezen, voor een tweede header of footer, enzovoort. In feite kan vrijwel elk ontwerp omgezet worden in een WordPress-theme. Maar je maakt het jezelf makkelijker als je begint met een dergelijk ontwerp. Een ontwerp dat binnen deze gebruikelijke methode valt zal ook eenvoudiger om kunnen gaan met widgets en plugins. Omdat hier de ruimte ontbreekt om alle mogelijkheden die er zijn te bespreken, kies ik voor een eenvoudige lay-out met twee kolommen. Ook de Cleanr-template die gebruikt wordt op www.robbertoverwordpress.nl gebruikt dezelfde structuur. Mijn Photoshop-ontwerp ziet er als volgt uit:



AFBEELDING 9.3

Merk op dat dit ontwerp bestaat uit een header, een hoofdkolom (main), een rechter zijkolom (sidebar) en een footer. Maar er is nog een extraatje: we hebben een ruimte gemaakt om een bepaald artikel meer aandacht te geven. De Engelse term hiervoor is *featured post*.



AFBEELDING 9.4

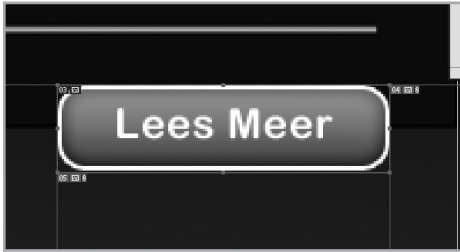
Voordat je begint met het ontwerpen van je eigen theme in Photoshop, volgen hier eerst nog een aantal tips. Daarbij geldt zeker: oefening baart kunst. Kijk hoe andere ontwerpen in elkaar zitten en houd rekening met de volgende punten.

- Zorg voor voldoende ‘witruimte’. Probeer niet te veel informatie op één pagina te proppen.
- Wees zuinig met het gebruik van kleuren en fonts. Gebruik liever één of twee kleuren in verschillende tinten, dat zorgt voor rust. Kijk bijvoorbeeld naar de kleuren in je belangrijkste foto of je logo en borduur daar op voort. Hetzelfde geldt voor het gebruik van lettertypes. Gebruik je te veel verschillende types en/of stijlen, dan wordt het al snel een bonte kermis.
- Maak gebruik van een subtiel verloop van kleuren (ook wel gradients genoemd) en gebruik schaduwen om een diepte-effect te creëren.
- Gebruik een basisgrid zoals je bijvoorbeeld kunt vinden op <http://960.gs>. Hier vind je voorbeelden van een indeling van je website in kolommen en kun je direct de bijbehorende voorbeeldbestanden voor Photoshop, inclusief de benodigde CSS, downloaden. Het gebruiken van een grid is geen must, maar het kan je wel helpen om een evenwichtig ontwerp voor je site te maken. Zeker in het begin is zo'n hulpmiddel meer dan welkom.

Een andere mogelijkheid voor het maken van een eigen theme is het gebruiken van een zogenaamd ‘theme framework’. Een voorbeeld is het Thematic framework van ThemeShaper (<http://themeshaper.com/thematic/>). Hiermee heb je eigenlijk al een basis theme met alle functionaliteit in handen. Het enige dat je dan zelf moet gaan doen is het verzorgen van de opmaak. Het schrijven van de CSS code dus, op basis van het ontwerp dat je gemaakt hebt.

Ontwerp opknippen

Je uiteindelijke ontwerp ga je nu opknippen in stukjes. De Engelse term hiervoor is *slice and dice*. Je doet dat door goed naar je ontwerp te kijken en alle losse elementen die je nodig hebt te kopiëren naar nieuwe bestanden en vervolgens op te slaan voor het web in gif-, png- of jpg-formaat. Voor sommige elementen, bijvoorbeeld buttons, ligt erg voor de hand hoe je dat moet doen. Maar voor bijvoorbeeld de achtergrond geldt dat je goed moet kijken welke stukjes je nodig hebt. Laten we eenvoudig beginnen met een button:



AFBEELDING 9.4

Er zijn wat verschillende methoden om het ontwerp met behulp van Photoshop op te knippen. Zelf kies ik er echter altijd voor om een deel te selecteren en (verenigd) te kopiëren naar een nieuw document.

Merk op dat je steeds ver moet inzoomen om dit goed te kunnen doen. Ik kopieer de button naar een nieuw document, kies dan **Opslaan voor het web** (Save for the web) en sla het bestand op in het formaat png-24.

Voor het vastleggen van de achtergrondafbeeldingen moet ik goed kijken naar de opbouw van mijn Photoshop-ontwerp en de juiste selectie maken. Je slaat vervolgens een hele smalle afbeelding (bij voorkeur 1 pixel breed) op, die we straks met behulp van CSS zullen herhalen.



AFBEELDING 9.5

Ik sla het bestand op voor het web en geef als naam header-background.jpg. Dit doe je ook voor de achtergrond van de hoofdkolom en de footer. Verder hebben we de afbeeldingen voor de RSS-feed en het Twitter-icoon nog nodig.

Ook moeten we natuurlijk het logo niet vergeten. Doordat we gebruikmaken van een niet-standaard font is het noodzakelijk het logo als afbeelding op te slaan. Aan het eind van dit hoofdstuk leg ik nog uit hoe je je logo wijzigt, zodat je je eigen websitenaam hier kunt plaatsen.



AFBEELDING 9.6

Als laatste slaan we de achtergrond voor het menu op. Hier is het even opletten! Doordat het menu een tab-structuur heeft met links en rechts afgeronde hoeken, heb ik in totaal drie afbeeldingen nodig. Één voor de linkerzijde, één voor de rechterzijde en een afbeelding van 1 pixel breed voor het middelste deel van de tab.



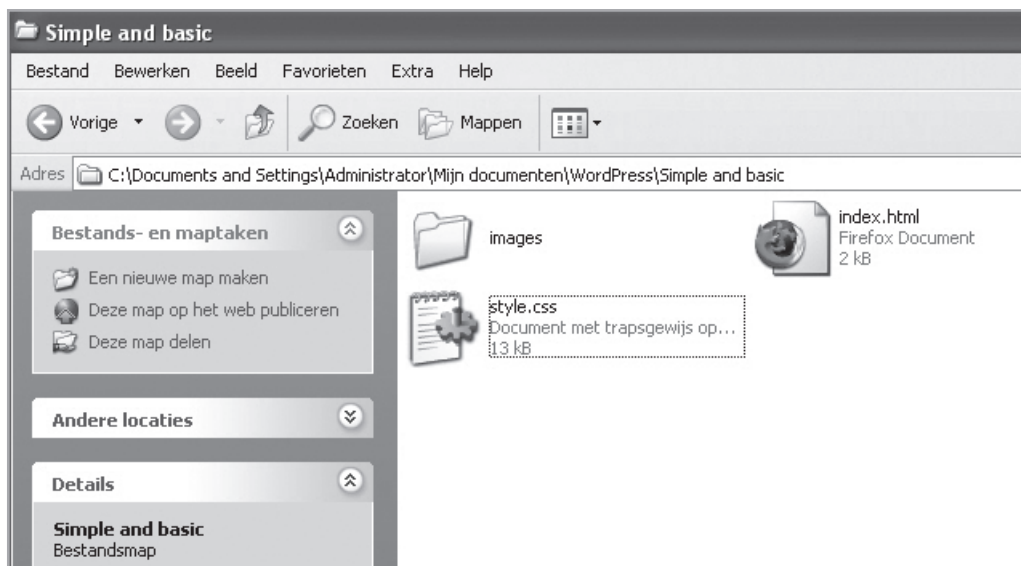
AFBEELDING 9.7

Ontwerp naar code omzetten

Nu je de eerste stappen genomen hebt kun je het ontwerp gaan omzetten naar HTML- en CSS-code.

Als je geïnteresseerd bent in een meer uitgebreide beschrijving van het maken van een WordPress theme, kijk dan eens naar het boek *WordPress Theme Design* door Tessa Blakeley Silver (uitgegeven door Packt Publishing en alleen in het Engels beschikbaar). Daarin wordt ook dieper ingegaan op het omzetten van je ontwerp met 'slicing and dicing'.

Maak ergens in je map met documenten en data een submap aan waarin je alle bestanden voor het theme kunt plaatsen. Geef de map de naam van je theme. Ik noem mijn theme 'SimpleAndBasic'. Binnen die map begin je met het aanmaken van een map 'images'. Ook maak je met je editor twee lege bestanden aan: `index.html` en `style.css`. De structuur ziet er dan als volgt uit:



AFBEELDING 9.8

Let op: je hoeft niet alle code letterlijk over te typen. Dat zou alleen maar erg foutgevoelig zijn. Vanaf de pdf kun je kopiëren en plakken. Daarnaast is het natuurlijk ook mogelijk om de code vanuit het kant-en-klare theme over te nemen in je eigen bestanden. Of gebruik het bestaande theme as basis voor de structuur en wijzig alleen de opmaak door de styling aan te passen door middel van het CSS-bestand!

Ik begin met wat basisinformatie in het bestand style.css op te nemen:

```
/*  
Theme Name: SimpleAndBasic  
Theme URI: http://www.robberoverwordpress.nl  
Description: Ontwerp gemaakt voor KickStart WordPress  
Author: Robbert Ravensbergen  
Version: 0.1  
Tags: simple, two columns, dark  
*/
```

Dit stukje tekst doet in feite niets anders dan beschrijven waar het bestand voor is. Het staat tussen de commentaartekens (`/*`), waardoor het verder geen functie heeft. Natuurlijk neem jij hier je eigen gegevens op. Sla het bestand even op voor je verder gaat!

Open nu het bestand index.html en voeg daar de code in:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
<HTML xmlns="http://www.w3.org/1999/xhtml">  
<head>  
<title>WordPress theme</title>  
<link rel="stylesheet" href="style.css" type="text/CSS" media="screen" />  
</head>  
<body>  
  
</body>  
</HTML>
```

De code die we nodig hebben om straks de juiste onderdelen op de juiste plaats te tonen komt in het `<body>` gedeelte terecht.

Plaats direct onder de <body>-regel de volgende code:

```
<!--wrapper-->
<div id="wrapper">
  <!--header-->
  <div id="header">
    <div id="logo"><h1><!--Hier komt het logo--></h1></div>
    <div id="menu">
      <ul>
        <li class="page_item"><a href="#">Products</a></li>
        <li class="page_item"><a href="#">Services</a></li>
        <li class="page_item"><a href="#">About</a></li>
      </ul>
    </div><!--menu-end-->
  </div><!--header-end-->

  <!--content-->
  <div id="content">
    <!--main-col--><div id="main-col"></div><!--main-col-end-->
    <!--sidebar--><div id="sidebar"></div><!--sidebar-end-->
  </div><!--content-end-->

</div><!--wrapper-end-->

<div id="footer"></div>
```

Dit is de basisstructuur voor je WordPress-theme! Wat gebeurt hier? Met behulp van de div-codes bouwen we een structuur van onderdelen. Merk bijvoorbeeld op dat de delen ('de divs') voor het logo en het menu onderdeel zijn van de div 'header'. Daarnaast bevat de div 'content' de onderdelen 'main-col' (de hoofdkolom links) en de 'sidebar' (die we rechts gaan plaatsen). Bovenstaande code bepaalt alleen de structuur, het zegt nog helemaal niets over welke informatie er in die structuur getoond wordt en hoe dat eruit komt te zien.

Overigens, de structuur is nog niet helemaal compleet zo, er ontbreken nog wat elementen. We hebben nu al wel een hoofdkolom gemaakt, maar voor de inhoud ervan moeten we ook nog ruimte reserveren.

Vervang de code in het gedeelte voor de hoofdkolom nu als volgt:

```

<!--main-col-->
  <div id="main-col">

    <!--featured-post--><div id="featured-post"><!--featured post hier --></
div><!--featured-post-end-->
    <!--post--><div class="post"><!--blog posts komen hier--></div><!--post-end-->

</div><!--main-col-end-->

```

Hiermee plaatsen we in de hoofdkolom de ‘featured post’, gevolgd door een div waar straks de reguliere WordPress-berichten getoond moeten worden. Daarna richten we ons op de rechterkolom, de sidebar. Vervang de code van het sidebargedeelte door het volgende:

```

<div id="sidebar">
  <!--feed--> <div id="feed"></div><!--feed-end-->
  <!--search--><div id="searchform"></div><!--search-end-->
  <!--sidebar kolommen -->
    <div id="sidebar-columns">
      <!--sidebar1--><div id="sidebar1"></div><!--sidebar1-end-->
      <!--sidebar2--><div id="sidebar2"></div><!--sidebar2-end-->
    </div><!--sidebar-columns-end-->
</div><!--sidebar-end-->

```

In de rechterkolom beginnen we met de rss- en Twitter-buttons, die plaatsen we in de ‘feed’ div. Daarna komt het zoekformulier. Vervolgens wordt de kolom in twee delen gesplitst, die we gaan gebruiken voor het plaatsen van WordPress-widgets.

CSS code

De basisstructuur is hiermee vastgelegd. Nu moeten we gaan beschrijven hoe alle elementen eruit komen te zien. Welke kleur hebben ze, hoe breed zijn ze, enzovoort. Dat doe je in het bestand style.css. Open dat bestand en voeg de volgende code toe.

```

#wrapper {
  width: 960px;
  margin: 0 auto;
  color: #ffffff;
}

```

```

#header {
    width: 960px;
    height: 114px;
    float: left;
    background: url(images/header-background.jpg) repeat-x top left;
}

#content {
    width: 960px;
    float: left;
    background: #707070 url(images/background.jpg) repeat-x top left;
}

#content #main-col {
    width: 600px;
    float: left;
    position: relative;
    padding-right: 30px;
}

#content #sidebar {
    width: 330px;
    margin-top: 20px;
    float: left;
    position: relative;
}

#content #sidebar #sidebar-columns {
    width: 330px;
    float: left;
    position: relative;
}

#footer {
    width: 100%;
    float: left;
    background: url(images/footer-background.jpg) repeat-x top left;
    height: 84px;
}

```

```
padding-top: 40px;
}
```

Deze code bepaalt voornamelijk de breedte van de verschillende onderdelen. Merk op dat het belangrijk is om bij de header, de footer en de content de juiste afbeeldingsnamen te gebruiken. Pas die dus aan als je zelf met andere afbeeldingen werkt. Dat geldt ook voor eventuele kleurcodes, zoals de #707070 in ons content-deel. Gebruik je andere kleuren dan moet je hier de juiste kleurcodes gebruiken.

Hierna ga ik verder met de styling voor het menu. Gebruik de volgende code, waarbij je ook nu weer je eigen bestandsnamen en kleurcodes gebruikt. Het menu wordt aan de rechterzijde van het scherm uitgelijnd.

```
/*--menu--*/
#header #menu {
    float: right;
    width: 960px;
    margin-top: 5px;
}

#header #menu ul {
    list-style: none;
    float: right;
    text-align: right;
}

#header #menu ul li { /* page_item is wordpress default class for page list */
    float: right;
    height: 33px;
}

#header #menu ul li a {
    padding: 4px 8px;
    display: block;
    margin: 0 7px 0 0;
    float: left;
    background: url(images/menu-middle.png) repeat-x top left;
    text-decoration: none;
    font: 900 14px Arial, Helvetica, sans-serif;
}
```

```

}

#header #menu ul li a:hover {
    color: #FFFFFF;
    background: url(images/menu-middle-hover.png) repeat-x top left;
}

#header #menu ul li.current_page_item a, #header #menu ul li.current-cat a {
/* current_page_item is wordpress default class for active page list */
    color: #89A23C;
}

```

Dit wordt al wat ingewikkelder. Merk op dat een menu bestaat uit een ongenummerde lijst of ul (unnumbered list) en elk menu-item daarin als een li wordt weergegeven. De codes current_page_item en current-cat worden gebruikt om het actieve menu-item te accentueren.

Ten opzichte van het oorspronkelijke ontwerp heb ik het menu wat vereenvoudigd. De tabs die ik daar gebruikte hadden afgeronde hoeken. Nu is het niet heel erg ingewikkeld om dat te bouwen, maar elke extra levert meer code op. Omwille van de overzichtelijkheid heb ik het hier daarom weggelaten.

We coderen ook het zoekformulier hier nog even:

```

/*--search--*/
#searchform {
    width: 330px;
    float: left;
    background: url(images/search.jpg) no-repeat top left;
    height: 48px;
    margin-bottom: 20px;
}

#searchform form {
    float: left;
    width: 310px;
    margin: 10px;
    display: inline;
}

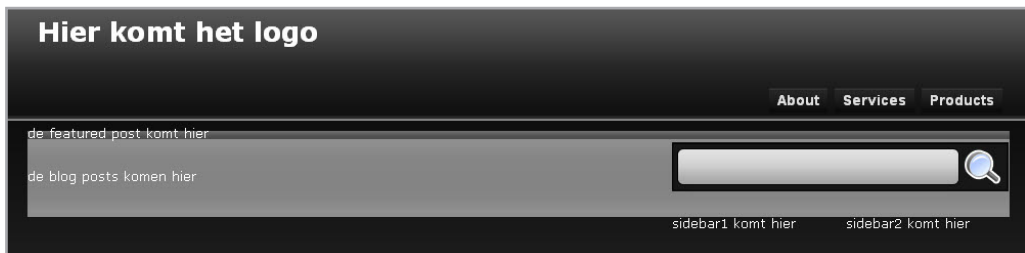
```

```
#searchform form input#s {
    width: 270px;
    background: url(images/form.png) no-repeat top left;
    height: 20px;
    border: 0;
    float: left;
    margin-right: 10px;
    padding: 3px 0 0 7px;
    color: #666;
}

#searchform form input#searchsubmit {
    width: 20px;
    background: url(images/search.png) no-repeat center left;
    height: 24px;
    border: 0;
}
```

Maar natuurlijk zijn we er hier nog niet mee. In feite hebben we nu alleen de CSS-code voor de lay-out, het menu en het zoekformulier geschreven. Er is nog veel meer nodig. Denk bijvoorbeeld aan de styling van de rest van de sidebar, het plaatsen van het logo, de sectie voor reacties, enzovoort. Het voert hier te ver om alle CSS-code letterlijk af te drukken en te behandelen. Ik raad je echter aan om het CSS-bestand dat bij het theme zit dat je kunt downloaden uitgebreid te bestuderen. In combinatie met het gebruik van de webdeveloper-toolbar moet je in staat zijn om te bepalen welke code waarvoor gebruikt wordt. Aan het einde van dit hoofdstuk laat ik nog een voorbeeld zien tijdens het vervangen van het logo.

We hebben nu een index.html en style.css die aan elkaar gekoppeld zijn. Open je index.html in je browser dan zou je ongeveer zoiets moeten zien:



AFBEELDING 9.9

Dat lijkt er nog niet echt op! Dat komt deels doordat nog niet alle CSS-code klaar is, maar ook doordat we vrijwel geen inhoud tonen. Er staan enkele tijdelijke teksten die ik in de HTML-code heb opgenomen, maar verder mis je vooral 'hoogte' om een goed beeld te krijgen. Maar ook met het deel dat we tot nu toe gaan hebben zie je dat we op de goede weg zijn! Maak het CSS-bestand verder af naar je eigen ontwerp en gebruik de style.css van het theme uit dit voorbeeld eventueel als referentie.

Van HTML naar WordPress PHP-code

De volgende stap is het omzetten van je HTML/CSS-code naar een WordPress-theme. Eindelijk! Daarbij hoef je aan de CSS-code niet veel te doen. Maar de index.html wordt volledig vervangen door een veelvoud aan PHP-bestanden. We hebben de volgende lijst met bestanden nodig voor ons theme:

- index.php Het hoofdbestand waarmee het theme gestart wordt.
- header.php Bevat alle code voor de header.
- featured-post.php De code voor het artikel in de spotlight.
- sidebar.php De code voor de sidebar aan de rechterzijde.
- footer.php De code voor de voettekst.

Deze vormen de basis, maar we hebben nog meer bestanden nodig. Overigens is dat verder opsplitsen niet in alle gevallen noodzakelijk, maar het houdt het theme wel overzichtelijk:

- sidebar-left.php De code voor de linker sidebar.
- sidebar-right.php De code voor de rechter sidebar.
- searchform.php Bevat de code om te kunnen zoeken.
- search.php Bevat de code voor de resultaten van de zoekactie.
- page.php Code voor het tonen van de WordPress-pagina's.
- single.php Toont één artikel volledig.
- comments.php Hoe wordt het commentaar getoond bij je bericht?
- functions.php Bevat enkele algemene PHP-functies voor het theme.

Daarmee zijn we er nog steeds niet, maar hiermee heb je een groot gedeelte van je theme toch wel in kaart gebracht. Je krijgt met behulp van de afbeelding een idee voor welk gedeelte van je website de verschillende bestanden gebruikt worden.



AFBEELDING 9.10

Laten we gaan kijken hoe de inhoud van deze bestanden eruit komt te zien. We beginnen met de `index.php`.

```
<?PHP get_header(); ?>

<!--content-->
<div id="content">

    <!--main-col-->
    <div id="main-col">

        <?PHP include('featured-post.php'); ?>
        <!--post - categorie 3 nemen we niet mee -->
        <?PHP query_posts($query_string.'&cat=-3&&order=DESC');

        while (have_posts()) : the_post(); ?>

            <div class="post" id="post-<?PHP the_ID(); ?>">
                <div class="post-info">
                    <?PHP echo get_avatar( get_the_author_id(), '85', '' ); ?>
```

```

        <em><?PHP the_time('F jS, Y') ?></em>
        <span class="post-tag"><?PHP the_tags(", ' . ', ''); ?></span>
    </div>

    <div class="entry">
    <div class="cat">Categorieën: <?PHP the_category(', '); ?></div>
        <h2><a href="<?PHP the_permalink() ?>" rel="bookmark" title="Permanent
Link to <?PHP the_title_attribute(); ?>"><?PHP the_title(); ?></a></h2>
        <?PHP the_content(""); ?>

    </div>

    <p class="metadata"><?PHP comments_popup_link('Geen reacties', '1
Reactie', '% Reacties'); ?> . <a href="<?PHP the_permalink() ?>#more-<?PHP the_
ID(); ?>">lees meer</a></p>

    </div><!--post-end-->
    <?PHP endwhile; ?>
    <div class="navigation">
        <div class="alignleft"><?PHP posts_nav_link("",'', '&laquo; Previous En-
tries') ?></div>
        <div class="alignright"><?PHP posts_nav_link('', 'Next Entries &raquo;', '')
?></div>
    </div>

</div><!--main-col-end-->

<?PHP get_sidebar(); ?>
</div><!--content-end-->

</div><!--wrapper-end-->
<?PHP get_footer(); ?>

```

Wat gebeurt hier nu precies? Allereerst wordt met de code 'get_header()' de header.php aangeroepen. Daarna volgt de div-structuur voor de hoofdkolom en starten we de featured-post.php. Daarna wordt met de functie 'query_posts()' de berichten opgehaald. Hierbij nemen we berichten in categorie drie niet mee. Merk op dat dit afhankelijk is van de wijze waarop

je WordPress hebt ingericht. Een artikel dat je wilt tonen als ‘featured post’ plaatsen we dus in een categorie met id nummer drie. Maar dat nummer kan natuurlijk bij elke WordPress installatie verschillen.

In het vervolg van de code gaan we met een ‘while-loop’ één voor één de berichten door en plaatsen we de gewenste informatie voor elk bericht.

Na de code voor de hoofdkolom volgt nog het aanroepen van de sidebar.php en de footer.php. En daarmee is de structuur voor je theme klaar. Maar ook hier hebben we nog veel meer code nodig om alles in elkaar te passen. Als voorbeeld toon ik de inhoud van de header.php

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<HTML xmlns="http://www.w3.org/1999/xhtml" <?PHP language_attributes(); ?>>

<head profile="http://gmpg.org/xfn/11">
<meta http-equiv="Content-Type" content="<?PHP bloginfo('HTML_type'); ?>; charset=<?PHP bloginfo('charset'); ?>" />

<title><?PHP wp_title('&laquo;', true, 'right'); ?> <?PHP bloginfo('name'); ?></title>

<link rel="stylesheet" href="<?PHP bloginfo('stylesheet_url'); ?>" type="text/css" media="screen" />

<link rel="alternate" type="application/rss+xml" title="<?PHP bloginfo('name'); ?> RSS Feed" href="<?PHP bloginfo('rss2_url'); ?>" />
<link rel="alternate" type="application/atom+xml" title="<?PHP bloginfo('name'); ?> Atom Feed" href="<?PHP bloginfo('atom_url'); ?>" />
<link rel="pingback" href="<?PHP bloginfo('pingback_url'); ?>" />

<?PHP if ( is_singular() ) wp_enqueue_script( 'comment-reply' ); ?>
<?PHP wp_head(); ?>
</head>
<body>
<!--wrapper-->
<div id="wrapper">
<!--header-->
    <div id="header">
```

```

        <!--logo-->
        <div id="logo"><h1><a href="<?PHP echo get_option('home'); ?>"
title="<?PHP bloginfo('description'); ?>"><?PHP bloginfo('name'); ?></a></h1></
div><!--logo-end-->

        <!--page-navigation-->
        <div id="menu">
        <!-- <em><?PHP bloginfo('description'); ?></em> -->
                <ul>
                        <?PHP wp_list_pages('title_li=&sort_column=menu_
order&depth=2'); ?>
                                <?PHP wp_list_categories('sort_column=name&sort_order=a
sc&style=list&children=true&hierarchical=true&title_li=0'); ?>
                                </ul>
                </div><!--page-navigation-->

</div><!--header-end-->

```

Doordat de header.php als eerste wordt aangeroepen vanuit de index.php beginnen we hier met de basis HTML-informatie en het laden van de stylesheet. Daarna volgen de divs voor de wrapper, de header, het logo en het menu. Het menu wordt gemaakt door een combinatie van WordPress pagina's en categorieën. Hiervoor gebruik je de functies `wp_list_pages` en `wp_list_categories`. De CSS code voor het menu is overigens nog niet ver genoeg uitgewerkt, waardoor er momenteel maar 1 niveau ondersteund wordt.

Wanneer we kijken naar de code voor de sidebar (inderdaad, in sidebar.php) dan ziet dat er als volgt uit:

```

<!--sidebar-->
<div id="sidebar">

        <!--feed-->
        <div id="feed">
                <a href="http://twitter.com/joomblocks" target="_blank"
class="twitter">RSS</a>
                <a href="<?PHP bloginfo('rss2_url'); ?>" class="rss">RSS</a>
        </div><!--feed-end-->

        <!--search-->

```

```
<?PHP include (TEMPLATEPATH . '/searchform.php'); ?>
```

```
<!--sidebar-->  
<div id="sidebar-columns"><h3>Sidebar</h3>  
    <?PHP include('sidebar-left.php'); ?>  
    <?PHP include('sidebar-right.php'); ?>  
</div><!--sidebar-columns-end-->
```

```
</div><!--sidebar-->
```

Ik begin met de div 'feed' en plaats daarin twee links naar mijn Twitter-account en de RSS feed. Natuurlijk zou je nog meer links kunnen plaatsen, naar bijvoorbeeld LinkedIn, Hyves, Facebook etc. De opmaak daarvan moet je natuurlijk dan nog wel even regelen met behulp van extra CSS-code.

Na de links wordt de code in searchform.php aangeroepen. Hiermee voeg je een zoekfunctie toe aan je blog. Tenslotte splits ik de sidebar op in een linker- en een rechtergedeelte, waar ik weer aparte PHP bestanden voor gebruik. De code voor de linker- en rechter-sidebar zien er vrijwel hetzelfde uit. Links tonen we een lijst met categorieën en rechts de weblinks. Maar die inhoud wordt alleen getoond als de dynamische sidebar niet actief is. Met andere woorden: als de widget-plugin niet actief is dan ziet de sidebar er toch nog goed uit door de statische inhoud die we hier vastleggen. In de praktijk wordt de inhoud natuurlijk (dynamisch) bepaalde door de keuze in widgets die de gebruiker maakt.

```
<!--sidebar1-->  
<div id="sidebar1">  
    <ul>  
        <?PHP if ( !function_exists('dynamic_sidebar') || !dynamic_  
sidebar('sidebar1') ) : ?>  
            <?PHP wp_list_categories('title_li=<h2>Categorien</h2>'); ?>  
        <?PHP endif; // end of sidebar1 ?>  
    </ul>  
</div><!--sidebar1-end-->
```

Hoe nu verder?

De code die je tot nu toe gezien hebt moet je al op weg geholpen hebben. Bekijk verder de PHP code van het theme dat ik hier ontwikkeld heb. Maak je de code eigen zodat je weet wat waar precies gebeurt. En natuurlijk is het toegestaan om dit theme op te pakken en alleen het uiterlijk te wijzigen door het aanpassen van de CSS-code. Verder zijn er verschillende (Engelstalige) bronnen waar je terecht kunt voor meer informatie over het coderen van themes.

Begin bijvoorbeeld op: http://codex.wordpress.org/Theme_Development Ook belangrijk zijn de zogenaamde WordPress-template tags: http://codex.wordpress.org/Template_Tags Hier vind je referenties naar alle WordPress-tags die je in je theme kunt aanroepen. Daarnaast kun je WordPress-functies gebruiken, zoals we deden in de code voor de sidebar hierboven. Een referentie daarvan vind je op http://codex.wordpress.org/Function_Reference

Als laatste een link naar een eveneens Engelstalig artikel over WordPress-theme-hacks. Een verzameling tips en trucs om je theme net dat beetje extra te geven:

www.noupe.com/wordpress/mastering-your-wordpress-theme-hacks-and-techniques.html

Het eindresultaat wanneer ik het theme gebruik om mijn site weer te geven:

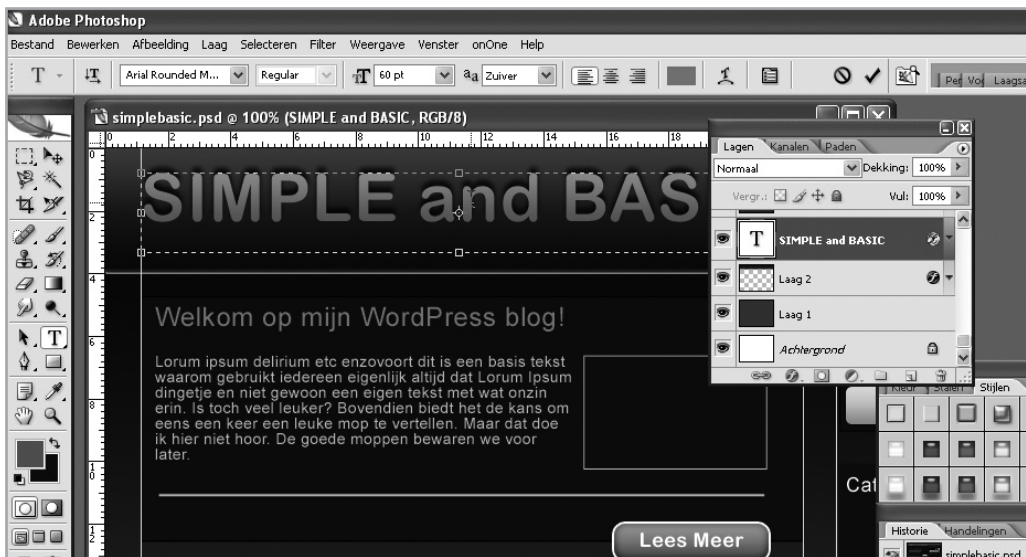


AFBEELDING 9.11

Het aanpassen van het logo

Wanneer je het theme dat ik hier gemaakt heb gewoon voor je eigen website wilt gebruiken dan mag dat natuurlijk. Maar in dat geval zul je enkele zaken moeten aanpassen naar je eigen situatie:

- De Twitter link verwijst nu naar <http://twitter.com/joomblocks> in de sidebar.php. Gebruik je eigen code of verwijder deze wanneer je geen Twitter gebruikt.
- De categorie-id voor de 'featured post'. Maak zowel een aanpassing in de featured-post.php als in de index.php.
- Het logo met de naam van je website. Gebruik Photoshop en open het bestand 'simplebasic.psd'. Selecteer de juiste laag en pas de tekst aan. Vervolgens exporteer je het logo als logo.png. Plaats deze in de map images van je theme en overschrijf het origineel.



AFBEELDING 9.12