



*Från Statisk HTML till
Dynamisk Wordpress Tema*

Ϩλυσμικισκ λλοισβισ22 ισμα

Vi Börjar Från HTML...

Idag ska vi omvandla en statisk HTML dokument till en dynamisk Wordpress tema!

För att kunna följa med är det viktigt att du har ett par grejer, nämligen: lokal server (t.ex. MAMP, XAMP eller liknande), en installation av Wordpress, en kodredigerare som kan koppla till en server (inte nödvändigt men bra att ha), och till sist en statisk XHTML dokument, d.v.s. webbsidan som du ska omvandla.

HTML dokumentet som jag kommer att jobba med kan du se här nedanför:

```
<!doctype html>
<html lang="sv">
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1" />
  <title>The Basic Theme</title>
  <meta name="viewport" content="width=device-width, initial-
scale=1.0" />
  <link type="text/css" rel="stylesheet" href="style.css" />
  <style type="text/css"></style>
</head>
<body class="wrapper">
  <header>
    <!-- LOGO -->
    <h1>Blogg Titel</h1>
    <h2>En liten truditidutt om bloggen</h2>
    <nav class="header-menu">
      <ul>
        <li><a href="#hem">Hem</a></li>
        <li><a href="#galleri">Galleri</a></li>
        <li><a href="#portfolio">Portfolio</a></li>
        <li><a href="#kontakt">Kontakt</a></li>
      </ul>
    </nav><!-- /nav -->
  </header>
  <div class="post-pagination clear">
    <ul>
      <li><a href="#1">Nästa</a></li>
      <li><a href="#2">Föregående</a></li>
    </ul>
  </div>
  <aside class="sidebar">
    <nav class="sidebar-menu">
      <ul>
        <li><a href="#0">Kategori0</a></li>
        <li><a href="#1">Kategori1</a></li>
        <li><a href="#2">Kategori2</a></li>
        <li><a href="#3">Kategori3</a></li>
      </ul>
    </nav>
  </aside>
  <div class="main">
```

```

<div id="content">
  <article class="entry">
    <section class="entry-meta">
      <h2><!-- Titel --></h2>
      <span>>Datum & amp; författare</span>
    </section><!-- /entry-meta -->
    <section class="entry-content">
      <p>Inläggs-innehåll</p>
    </section><!-- /entry-content -->
    <section class="entry-meta">
      <span>Kategori & amp; Kommentarer</span>
    </section><!-- /entry-meta -->
  </article><!-- /entry -->
</div><!-- /content-->
</div><!-- /main -->
<footer class="main-footer">
  <div>
    <span>&copy; 2014 by Daniel Karjanlahti</span>
    <nav class="footer-menu">
      <ul>
        <li><a href="facebook.com">Facebook</a></li>
        <li><a href="googleplus.com">Google+</a></li>
        <li><a href="twitter.com">Twitter</a></li>
      </ul>
    </nav>
    <aside class="credits">
      <span>Temat är skapad av
      <a href="mailto:karjanlahti76@gmail.com">Daniel
Karjanlahti</a>
      &nbsp;&copy; 2014 <a href="http://
cr8gr8designs.com">cr8gr8designs.com</a></span>
    </aside>
  </div><!-- /fat_footer -->
</footer><!-- /footer -->
</body><!-- /wrapper -->
</html>

```



Som du kan se har jag med det mesta, och resultatet ser du nedanför:

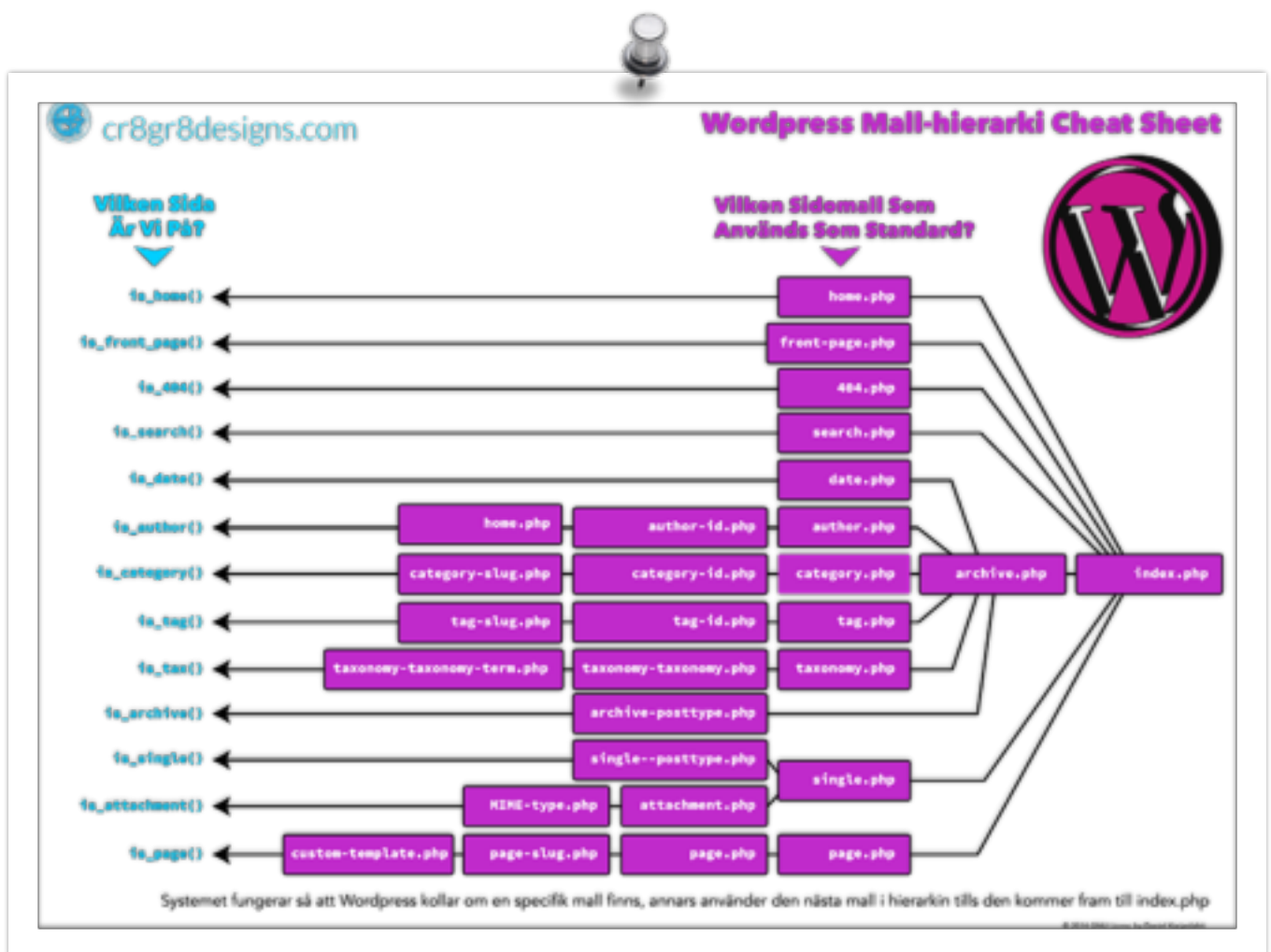
Tre Kärnbegrepp

För att verkligen förstå vad Wordpress teman gör, måste du förstå tre primära kärnbegrepp:

1. Mall-hierarki
- 2 "Loopen"
3. Hur krokar (hooks) fungerar (vi ska dock inte fördjupa oss i dom här)

Man kan se på Wordpress som en fabrik som omvandlar inlägg och sidor (som lagras i databasen) till webbsidor.

Standardfiler för ett tema



page.php	- visar en enkel sida.
single.php	- visar ett enkelt inlägg.
front-page.php	- en splash sida, visas istället för index.php och home.php (om den finns).
home.php	- används som hemsida (om den finns).
404.php	- error, kunde inte hitta sida.

Wordpress Codex - [wp_enqueue_script](#) och [wp_enqueue_style](#).

style.css	- (måste finnas med) är huvud-stilmallen för temat, och innehåller en kommentar som anger information som tema- namn, författar-namn, tema-URL, författar-URL, Taggar, version, och licens.
index.php	- standardmallen för alla sidor (väljs om inget annat har angetts)
functions.php	- temafunktioner, härifrån styr vi sidan med hjälp av funktioner, hooks och genom att enqueue stilar och script.
header.php	- innehåller head och header delen, med meny, titel, slogan, logo mm.
footer.php	- sidfot har innehåll som t.ex. licens, årtal, författare, social meny, eller widgetar.
sidebar.php	- sidospalt, ofta där widgetar placeras (vi ska ha en plats för en dynamisk meny där).

Wordpress fabriken tar hand om olika typer av innehåll och den väljer mall efter en mall-hierarki, och därför måste det alltid finnas en mall för hur den ska visa ett inlägg, sida, sökresultat eller kategori, och den grundmallen heter `index.php`.

Wordpress mallar är som pusselbitarn, där vissa mallar, som sidhuvud och sidfot används på alla sidor, medan andra används enbart under särskilda omständigheter.

Våra mall-delar

Vi behöver inte använda oss av alla, men några stycken blir det. Till att börja med ska HTML dokumentet delas upp i ett antal delar:

header.php	- innehåller allt från doctype, head och till efter header delen.
loop.php	- skriver ut inlägg, kategorier och sidor
sidebar.php	- ska innehålla sidospalten med widget område.
footer.php	- footer delen där vi stänger body & html taggarna, plus andra vi möjligen öppnat och placerar en meny-widget plats.
index.php	- ska anropa delarna
functions.php	- innehåller funktionerna som ska styra temat (där ska vi också <i>enqueue</i> eventuella script & stilar).

Style.css & tema-mappen

För att Wordpress ska känna igen ett tema måste den innehålla en `style.css` med header information, och eftersom vi ska ha flera filer ska dom in i en mapp.

Vi börjar med skapa en mapp i `/wordpress/wp-content/themes/` och jag ska kalla mitt tema för The Basic Theme, och då får mappen heta `the-basic-theme`. I den mappen ska vi skapa `style.css` som ska innehålla följande header information:

```
/*
Theme Name: The basic Theme
Theme URI: http://cr8gr8designs.com
Description: En Basic Wordpress Start Tema av Daniel Karjanlahti på <a href="http://:cr8gr8designs.com">cr8gr8designs.com</a>. Det här temat är en startmall för en Wordpress tema.
Version: 1.0
Author: Daniel Karjanlahti
Author URI: http://cr8gr8designs.com
Tags: responsive, mobile-ready, flexible, CSS3, HTML5
Text Domain: the-basic-theme
*/
```

Om du går till admin panelen och väljer » Utseende » Teman, så kan du där se och aktivera det nyskapade temat:



Header.php

Nu ska vi flytta över vår HTML sida till PHP filer. Vi börjar uppifrån och skapar header .php delen, och du ska välj allt från den inledande DOCTYPE deklaration till öppnande div elementet med klassen "main" och flytta det till din header .php fil.

Nu ska vi ändra en del av det så att den blir dynamisk:

```
<!--[if IE]><![endif]-->
<!DOCTYPE html>
<!--[if lt IE 7]><html class="no-js ie6"><![endif]-->
<!--[if IE 7]><html class="no-js ie7"><![endif]-->
<!--[if IE 8]><html class="no-js ie8"><![endif]-->
<!--[if gt IE 8]><!-->
<html class="no-js" <?php language_attributes(); ?><!-- bara den här
syns -->
<!--<![endif]-->
<head>
    <!--==== META TAGGAR ====>
<!-- Följande linje säkerställer att webbläsare inte använder quirks mode
-->
    <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
<!-- Denna linje talar om charset till webbläsaren, & undviker okända
tecken! -->
    <meta charset="<?php bloginfo('charset'); ?>" />

    <meta name="description" content="Keywords">
    <meta name="author" content="<?php echo AUTHOR; ?>">
<!-- Nästa tagg tar bort/återställer standardzoom på mobil enhet (iPad
och iPhone) -->
    <meta name="viewport" content="width=device-width, initial-
scale=1, maximum-scale=1">
    <!--==== LÄNK TAGGAR ====>
<!-- En länk till RSS-flödet på din webbplats. -->
    <link rel="alternate" type="application/rss+xml" title="<?php
bloginfo('name'); ?> RSS2 Feed" href="<?php bloginfo('rss2_url'); ?>" />
<!-- En länk för Pingback URL för din webbplats -->
    <link rel="pingback" href="<?php bloginfo('pingback_url'); ?>" />

    <!--==== TITEL ====>
<!-- skapar en dynamisk titel, beroende på sida -->
    <title><?php
        if (!is_home()) { wp_title(); } else { echo 'Hem'; } /* page
title */ ?> | <?php bloginfo('name'); /* blog name */ ?>
    </title>

<!--
Ha ALLTID med action hooken wp_head() strax före den avslutande </ head>
i ditt tema, för
```

annars kommer många plugins att inte fungera, för dom använder hooken för att lägga till stilar, skript och metataggar.

```
-->
```

```
<?php wp_head(); ?>
```

```
</head>
```

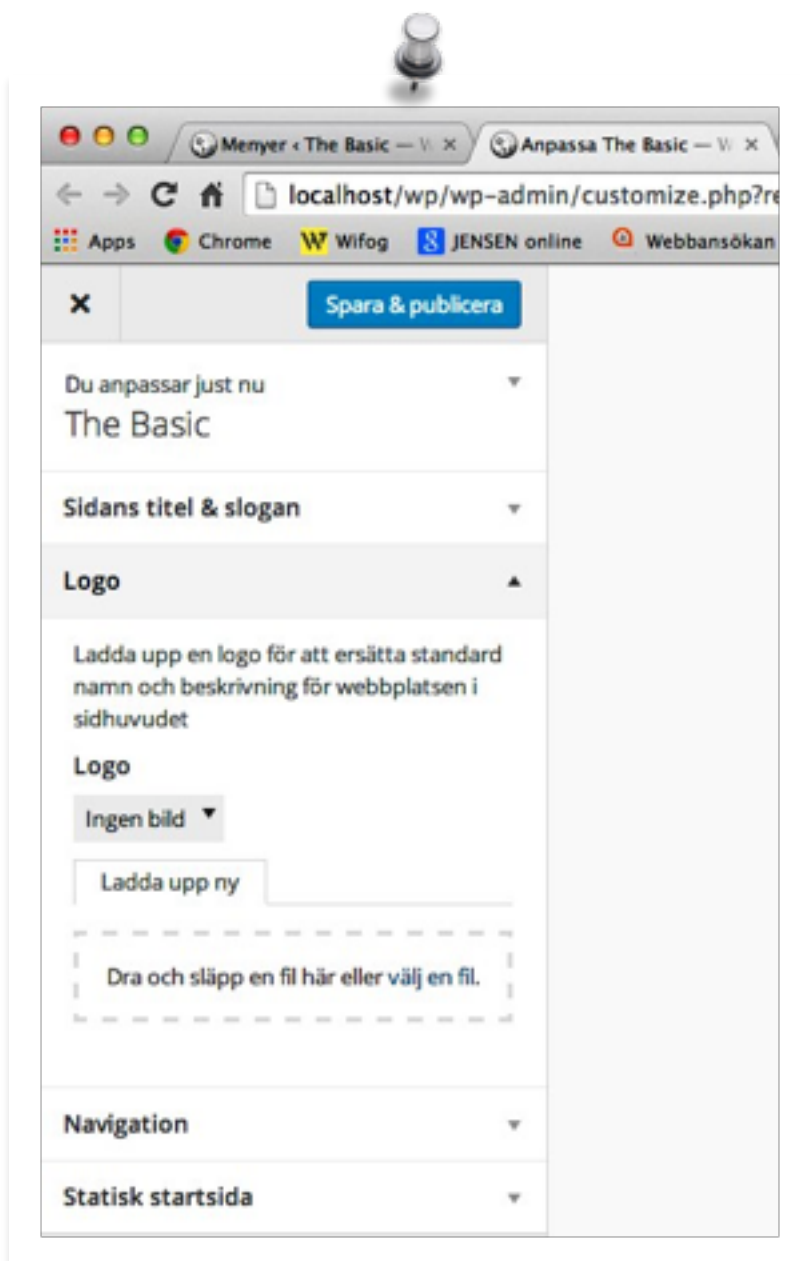
Och det var bara <head> delen, nu har vi kvar <header> delen med huvudtitel, slogan, meny mm. Vi fortsätter från body taggen i header.php:

```
<!-- body taggen har en hook som hjälper tema/plugin att styla med CSS -->
```

```
<body <?php body_class($class); ?>>  
    <header class="main_header">
```

Stöd för uppladdning av logo

Nästa steg är att lägga till stöd för uppladdning av en logo, genom att skriva in stöd för en Tema » Anpassa meny.



Så här kommer uppladdningskontrollen för logo att se ut

I functions.php

I functions.php skapar vi funktionen `add_section` från klassen `WP_Customize_Manager`:

```
function the_basic_theme_customizer($wp_customize) {
    /* skapa en ny sektion för en logo uppladdning */
    $wp_customize->add_section('the_basic_theme_logo_section',
array(
    'title' => __('Logo', 'the_basic_theme'), // titel
    'priority' => 30, // prioritet
    'description' => 'Ladda upp en logo för att ersätta standard
namn och beskrivning för webbplatsen i sidhuvudet', //
beskrivning
    ));

    /* lägg till vår nya inställning */
    $wp_customize->add_setting('the_basic_theme_logo');

    /* lägg till en kontroll med add_control funktionen för en
bilduppladdare så att man kan välja logo. Funktionen
WP_Customize_Image_Control finns i codex: */
    $wp_customize->add_control(new
WP_Customize_Image_Control($wp_customize, 'the_basic_theme_logo',
array(
    'label' => __('Logo', 'the_basic_theme'),
    'section' => 'the_basic_theme_logo_section',
    'settings' => 'the_basic_theme_logo'
    )));
} /* till sist registrera funktionen */
add_action('customize_register', 'the_basic_theme_customizer');
```

Tillbaka Till header.php

Nästa steg blir att använda oss av funktionen som placerar en Anpassnings-menyn, med möjligheten att ladda upp en bild för logo, så i header.php skriver du in följande:

```
<!-- LOGO -->
<?php if (get_theme_mod('the_basic_theme_logo')) : ?>
    <div class='site-logo'>
        <a href='<?php echo esc_url(home_url('/')); ?>'
title='<?php echo esc_attr(get_bloginfo('name', 'display')); ?>'
rel='home'>
            <img class="logo" src='<?php echo
esc_url(get_theme_mod('the_basic_theme_logo')); ?>' alt='<?php echo
esc_attr(get_bloginfo('name', 'display')); ?>'>
        </a>
    </div>
<?php else : ?>
```

Om Ingen Bild Finns?

Om ingen bild finns satt för logo, ska standard blogg-titel och blogg-slogan visas istället:

```
<!-- om ingen bild för logo har satts visa titel & slogan -->
    <div class="logo-replacement">
        <h1 class='site-title'><a href='<?php echo
esc_url(home_url('/')); ?>' title='<?php echo
esc_attr(get_bloginfo('name', 'display')); ?>' rel='home'><?php
bloginfo('name'); ?></a></h1>
        <h2 class='site-description'><?php
bloginfo('description'); ?></h2>
    </div>
<?php endif; ?><!-- /logo -->
```

Huvudtitel & Slogan

Och så vidare i `header.php`, där vi fortsätter med att lägga till en dynamisk titel och slogan, vilka man ställer in när man installerar Wordpress, men det går ju enkelt att ändra på från » Utseende » Anpassa » Sidans Titel & Slogan:

(Jag vet att vi gjorde det precis, men dom försvinner ju om en bild laddas upp för logo)

```
    <!-- Titel & Slogan -->
    <div class="site-name half left">
<!-- titel & slogan ska in dynamiskt, med metoderna get_bloginfo() &
bloginfo()
och sanera URL med esc_url (mer om esc_url längre ner) -->
        <h1 class="the-title"><a href='<?php echo
esc_url(home_url('/')); ?>'><?php binfo('name'); ?></a></h1>
        <h2 class="the-description"><?php
bloginfo('description'); ?></h2>
    </div>
</header>
<div class="main"><!-- avslutas i footer.php -->
```

Dynamisk Meny

Nu kommer menyn i `header.php`, men vi ska ju ha en dynamisk meny vilket betyder att vi behöver lägga till stöd för en sådan i `functions.php`. Menyn kan sedan ställas in i » Admin Panelen » Utseende » Menyer:

```
// functions.php
// lägg till stöd för menyer
```

```

add_theme_support('menus');

/* det är alltid bra att kolla att funktionen inte finns definierad någon
annanstans */
if(!function_exists('register_the_basic_theme_menus')) {
    function register_the_basic_theme_menus() {
        register_nav_menus(
            array(
                'header-menu' => __('Header Meny'),
/* vi lägger till två extra menyer som vi lägger in i footer.php och
sidebar.php */
                'sidebar-menu' => __('Sidebar Meny'),
                'footer-menu' => __('Footer Meny')
            )
        );
    }
}

/* till slut lägger vi till en action hook */
add_action('init', 'register_the_basic_theme_menus');

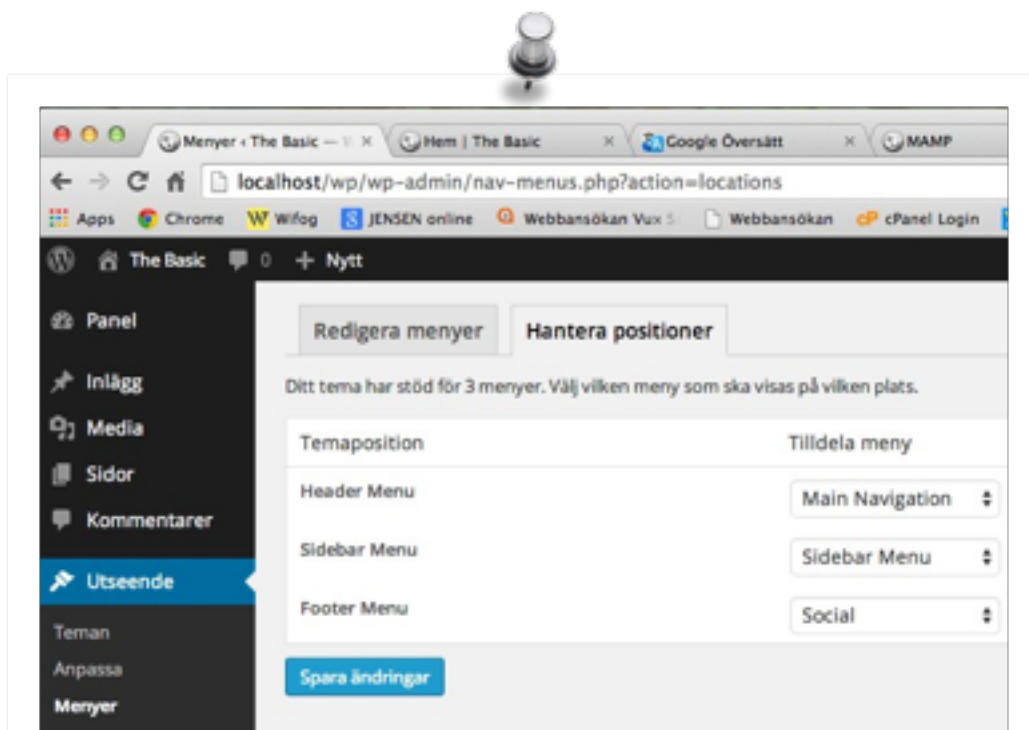
```

Och så skriver vi in själva menyn i header.php där vi också anger det omslutande elementet och en klass till den:

```

<!-- header.php -->
    <!-- användardefierad meny "Header Meny" -->
    <?php
    wp_nav_menu(array(
        'theme_location' => 'header-menu', /* placerings
inställning */
        'container' => 'nav', /* omslutande element */
        'container_class' => 'header-menu'));/* klass för nav */
    ?><!-- /nav class="main-navigation" -->
</header>
<div class="main"><!-- stängs i footer.php

```



Sidebar.php

Och så har turen kommit till sidospalten, där vi också skulle lägga in en meny.

```
<aside class="sidebar">
<?php
/* sidebar-meny */
wp_nav_menu(array(
    'theme_location' => 'sidebar-menu',
    'container' => 'nav',
    'container_class' => 'sidebar-menu'));
?>
</aside>
```

Loop.php

Wordpress Loopen är själva grundstommen i så gott som varje sida på bloggen, eftersom det är de block av kod som har ansvaret för att hitta och visa dina inlägg.

Loopen är där du kan specificera exakt vad som ska hända för varje inlägg som visas på en viss sida, och den täcker allt från vilka tjänster och hur många inlägg du vill visa, till vilken information som borde gå in i sidan, länkar i titeln, taggarna, även *kroppen* i inlägget själv.

Grundloopen ser ut så här:

```
<?php
if ( have_posts() ) {
    while ( have_posts() ) { the_post();
        /* inläggs innehåll här */
    } /* slut while */
} /* slut if */
?>
```

Vi ska skapa en egen fil för vår loop, för den kommer att innehålla en hel del (den kan också lägga direkt i `index.php`) och vi ska använda en förenklad `if` och `while` sats:

```
<!-- loop.php -->
<div id="content">
    <!-- SIDNUMRERING kommer senare -->
    <!-- loopen börjar med att fråga - om inlägg finns...? -->
    <?php if (have_posts()) : ?>
```

```
<!-- medan inlägg finns : skriv ut inlägg -->
<?php while (have_posts()) : the_post(); ?>
```

Utvalda Bilden

Här ska vi lägga till inläggets "Utvalda Bilden" (om det finns en), och det gör man med funktionen `the_post_thumbnail()` som kan ha ett antal parametrar för upplösning:

```
the_post_thumbnail(); // Utan parametrar -> 'post-thumbnail'
the_post_thumbnail( 'thumbnail' ); // Miniatur (default 150px x 150px max)
the_post_thumbnail( 'medium' ); // Medium upplösning (default 300px x
300px max)
the_post_thumbnail( 'large' ); // Stor upplösning (default 640px x
640px max)
the_post_thumbnail( 'full' ); // Full upplösning (original size
uploaded)
the_post_thumbnail(array(100, 100)); // Annan upplösning
```

Sedan finns ett antal standardklasser som Wordpress skapar:

```
img.wp-post-image
img.attachment-thumbnail
img.attachment-medium
img.attachment-large
img.attachment-full
```

Du kan också ge Inlägg miniaturbilder sin egen klass.

Visa inlägg Miniatur med en klass "align-right":

```
<?php the_post_thumbnail('thumbnail', array( 'class' => 'align-
right' )); ?>
```

Vi Skriver in den Utvalda Bilden med stöd i `functions.php`

Man måste också skriva in stöd för *Utvalda Bilden* i `functions.php`, men vi börjar med att sätta in koden i `loop.php`, där vi först kollar om en bild finns till inlägget (på det sättet kan du lägga till flera funktioner senare):

```
<!-- loop.php -->
<?php if (has_post_thumbnail()) : ?>
    <!-- vi lägger till permalänk till
    <a href="<?php the_permalink(); ?>" title="<?php
the_title_attribute(); ?>">
    <?php the_post_thumbnail(); ?>
    </a>
<?php endif; ?>
```

Och i `functions.php` blir det bara en rad:

```
// functions.php
add_theme_support('post-thumbnails');
```

Vidare i loopen

Nu fortsätter vi med loopen igen, och där har turen kommit till själva inlägget:

```
<!-- loop.php -->
<article class="entry post" id="post-<?php the_ID(); ?
">
    <section class="entry-meta">
        <!-- titel med länk till inlägget -->
        <h2><a href="<?php the_permalink() ?>"
rel="bookmark" title="Permanent Länk till <?php the_title(); ?>"><?php
the_title(); ?></a></h2>
        <!-- datum & författare -->
        <span><?php the_time('F jS, Y') ?> <!--
skriven av <?php the_author() ?> --></span>
    </section>

    <section class="entry-content">
        <!-- om man lägger till "more" i en HTML kommentar i ett
inlägg
        kommer detta att stå / annars bara inlägget -->
        <?php the_content('Läs resten av det här inlägget
» '); ?>
    </section>

    <section class="entry-meta">Postad i Kategorin » <?php
the_category(', ') ?> <strong>|</strong>
        <!-- en redigera länk för administratörer -->
        <?php edit_post_link('Redigera', '', '<strong>|</
strong>'); ?>
        <!-- länk till möjlig/a kommentar/kommentarer -->
        <?php comments_popup_link('Inga kommentarer »', '1
Kommentar »', '% Kommentarer »'); ?>
    </section>
</article><!-- /entry post -->
<?php endwhile; ?><!-- /while -->
<!-- SIDNUMRERING -->

<!-- om inga inlägg hittas -->
<?php else : ?>
    <h2 class="center">Kunde inte hitta</h2>
    <p class="center">Tyvärr söker du efter något som inte finns där.</
p>
    <?php include (THEME_DIR . "/search.php"); ?>
<?php endif; ?><!-- /if -->
</div><!-- /content -->
```

Okej nu har vi nästan kodat hela temat, men det finns ett par saker kvar att ta hand om.

Pagination (förse en sida med ett sidnummer)

Du kanske märkte att jag skrev SIDNUMRERING i kommentarer, det är en funktion vi ska lägga till så att man ska kunna navigera mellan äldre och nyare inlägg, vilket vi gör med hjälp av en numrerad meny. Den ska läggas innan eller efter loopen, för annars kommer den att dyka upp så många gånger som det finns inlägg.

Okej vi börjar med funktionen, som ska in i, ja just det, `functions.php`:

```
// functions.php
// Numrerad inläggs Pagination
if (!function_exists('the_basic_theme_pagination')) {
    function the_basic_theme_pagination() {

        $prev_arrow = is_rtl() ? '&rarr;' : '&larr;';
        $next_arrow = is_rtl() ? '&larr;' : '&rarr;';

        global $wp_query;
        $total = $wp_query->max_num_pages;
        $big = 999999999; /* en osannolikt hög integer */
        if ($total > 1) {
            if (!$current_page = get_query_var('paged'))
                $current_page = 1;
            if (get_option('permalink_structure')) {
                $format = 'page/%#%/';
            } else {
                $format = '&paged=%#%';
            }
            echo paginate_links(array(
                'base' => str_replace($big, '%#%',
esc_url(get_pagenum_link($big))),
                'format' => $format,
                'current' => max(1, get_query_var('paged')),
                'total' => $total,
                'mid_size' => 3,
                'type' => 'list',
                'prev_text' => $prev_arrow,
                'next_text' => $next_arrow,
            ));
        }
    }
}
```

Och så koden som ska in i `loop.php`:

```
<!-- loop.php -->
<div class="post-navigation">
    <?php the_basic_theme_pagination(); ?>
</div>
```

Footer.php

Den sista delen kommer ska in i filen footer.php:

```
<!-- i footer.php avslutar vi div class main, body & html -->
    </div><!-- /main -->
    <footer class="main-footer">
        <?php wp_nav_menu(array('theme_location' => 'footer-
menu', 'container' => 'nav', 'container_class' => 'footer-menu')); ?>
```

Credits / Erkännande

Nu tänkte jag lägga till dynamisk erkännande (credits) som baseras på en funktion som tar information från Wordpress databas om vem som skapat temat, och ett årtal som ändras dynamiskt. Vi skapar den i functions.php:

```
// functions.php
function the_basic_theme_credits() {
    $the_basic = wp_get_theme();
    echo "<span><span class='bold'>" . $the_basic->get('Name') . "</
span> skapad av <a href='" . $the_basic->get('AuthorURI') . "'><span
class='bold'>Daniel Karjanlahti</span></a></span><br />", "<span>&copy; "
. date('Y') . " GNU License <a href='http://'" . $the_basic-
>get('ThemeURI') . "'><span class='bold'>" . $the_basic-
>get('ThemeURI') . "</span></a></span>";
}
```

Och så kan vi anropa funktionen från footer.php:

```
<!-- footer.php -->
    <aside class="credits">
        <?php the_basic_theme_credits() ?>
    </aside>
</footer><!-- /footer -->
<!-- en hook för plugin och temat, för att lägga till script -->
<?php wp_footer() ?>
</body><!-- /wrapper -->
</html>
```

Index.php

Nu har vi alla delar för att temat ska fungera (hoppas jag). Vi kallar in dom olika delarna med `get_header()`, `get_sidabar()` och `get_footer()`, plus att vi kallar in loopen med funktionen `get_template_part()` som kan *anropa* vilken specialfil som helst:


```
<!-- index.php -->
<?php get_header(); // header.php ?>
    <?php get_sidebar(); // sidebar.php ?>
    <?php get_template_part('loop'); // loop.php ?>
<?php get_footer(); // footer.php ?>
```

Functions.php

Här är ett par väldigt användbara globala variabler: sökvägen till temats rotmapp och ta bort Wordpress genererad version info.

```
// functions.php
/** definiera sökväg till temat som en global variabel **/
define("THEME_DIR", get_template_directory_uri());

/** Ta bort Wordpress version info meta tagg **/
remove_action('wp_head', 'wp_generator');
```

wp_enqueue_style

Ett säkert sätt att lägga till (köa) en stilmall till en Wordpress genererad sida. Om det först registreras med `wp_register_style` kan det läggas till med hjälp av bara dess "handtag" (ett namn som man kan använda som referens). Funktionen ser ut så här:

```
<?php wp_enqueue_style( $handle, $src, $deps, $ver, $media ); ?>
```

`$handle` - (obligatoriskt) Namn, användas som ett *handtag* för stilmallen så man kan referera till den senare (konkatenera ditt temanamn i början så att den blir unik & inte skapar konflikter). Standard: ingen

`$src` - (valfritt) URL till stilmallen. t.ex. : "http://example.com/css/mystyle.css". Nödvändig bara om Wordpress inte känner till stilmallen. OBS! Skriv aldrig webbadresser direkt, utan använd i stället `plugins_url` (för insticksprogram) och `get_template_directory_uri` (för teman). Tillgångar utifrån kan anges med ett URL, dvs `'//stil.com/deras-stil.css'`. Standard: false

`$deps` - (array) (valfritt) En array med *handtag* (namn) som pekar mot varje formatmall som denna stilmall är beroende av, och det är mallar som måste laddas före detta stilmall. false om det inte finns. Standard: array()

`$ver` - (String) (valfritt) Version nummer, viktigt för webbläsare cachar ofta stilar och då missas redigeringar. Standard: array()

`$media` - CSS media typer som; screen, print mm. Standard: 'all'

```
function enqueue_styles() {
    /** REGISTRERA css/screen.css */
    wp_register_style('screen-style', THEME_DIR . '/css_path/
screen.css', array(), '1', 'all' );
    /** lägg stilen i kö */
    wp_enqueue_style('screen-style');
    /** ...eller bara */
    wp_enqueue_style('screen-style', THEME_DIR . '/css_path/
screen.css', array(), '1', 'all');
}
add_action('wp_enqueue_scripts', 'enqueue_styles');
```

wp_enqueue_script

Länkar en skript fil till en Wordpress genererad sida i rätt tid enligt skript beroenden, men bara om skriptet inte redan finns inkluderat och efter att alla *beroenden* har registrerats. Du kan antingen koppla ett skript med ett handtag som tidigare registrerats med hjälp av `wp_register_script()` funktionen, eller tillhandahålla denna funktion med alla parametrar som behövs för att koppla ett skript.

Detta är den rekommenderade metoden för att koppla JavaScript till en Wordpress genererade sida. Funktionen ser ut så här:

```
<?php wp_enqueue_script( $handle, $src, $deps, $ver, $in_footer ); ?>
```

\$handle - (obligatoriskt) Namn, användas som *handtag* för scripten (konkatenera ditt temanamn i början så att den blir unik & inte skapar konflikter). Standard: ingen

\$src - (valfritt) URL till stilmallen. t.ex. : "http://example.com/css/mystyle.css". Nödvändig bara om Wordpress inte känner till stilmallen. Skriv aldrig webbadresser till lokala stilar, använd i stället `plugins_url` (för insticksprogram) och `get_template_directory_uri` (för teman). Tillgångar utifrån kan anges med ett URL, dvs `'//stil.com/deras-stil.css'`. Standard:FALSE

\$deps - (array) (valfritt) En array av namn (handtag) på redan registrerade skript som det här skriptet är beroende av, och som måste laddas innan det här skriptet. Denna parameter är endast nödvändigt om skriptet med angivet `$handle` inte redan registrerats med hjälp `wp_register_script ()`. Man skriver handtag med små bokstäver. Standard: array()

\$ver - (String) (valfritt) Sträng som anger versionsnummer, om en sådan finns, den konkateneras i slutet av sökvägen som en query sträng (`script.js?1.0`). Om ingen version anges eller om man anger FALSE, då kommer Wordpress automatiskt att lägga till ett versionsnummer som är samma som den aktuella versionen av Wordpress. Om den är satt till NULL kommer ingen versionsnummer att sättas. Denna parameter används för att säkerställa att rätt version skickas till klientdatorn oavsett caching, och bör därför tas med om ett versionsnummer finns tillgänglig och är vettigt för skriptet.. Standard: FALSE

\$in_footer - (boolean) om scriptet ska laddas i footer delen precis innan avslutande body taggen. Standard: FALSE

```
function enqueue_scripts() {
    /** REGISTRERA först.. */
    wp_register_script('html5-shim', 'http://
html5shim.googlecode.com/svn/trunk/html5.js', array( 'jquery' ),
'1.0', FALSE);
    /** lägg i kö sen... */
    wp_enqueue_script( 'html5-shim' );

    /** ...eller bara lägga i kö */
    wp_enqueue_script('html5-shim', 'http://
html5shim.googlecode.com/svn/trunk/html5.js', array('jquery'),
'1.0', FALSE);
}

add_action( 'wp_enqueue_scripts', 'enqueue_scripts' );
```

wp_register_style & wp_register_script

Registrerar ett skript fil i Wordpress som länkas till en sida senare med hjälp av `wp_enqueue_script()`.

```
<?php wp_register_script( $handle, $src, $deps, $ver, $in_footer ); ?>
<?php wp_register_style( $handle, $src, $deps, $ver, $media ); ?>
```

(Förklaring för parametrarna kan du se ovanför i `wp_enqueue_script` och `wp_enqueue_style`)

404.php

Till sist är det bra att kunna hantera "Kunde Inte Hitta Sida" error, och till det skapar vi en ny mall som får heta just 404.php. Innehållet kommer inte att vara så mycket annorlunda förutom ett par saker:

```
<?php
/**
 * Mall för att visa 404 sidor (Not Found)
 * man kan lägga till get_header en parameter som t.ex. 404, då kommer
Wordpress
 * att leta efter en sida som heter header-404.php
 */
get_header() ?>
    <!-- _e() Visar värdet som översatt text från translate() -->
    <h1 class="error-title"><?php _e('Kunde Inte Hitta...', 'the-basic-
theme'); ?></h1>
    <p> Vi beklagar, men sidan du sökte har inte hittats! Det kan ha
flyttats eller tagits bort. </ P>
    <p> Jag klandrar dig inte, men har du kollat ditt adressfält? Det
kan finnas ett stavfel i webbadressen. </ P>
```

```
<p> Om det inte finns, kan du prova att söka på min hemsida för det innehåll du söker: </ p>
<?php get_search_form (); ?>
<p> Eller kanske du letar efter en av mina populära inlägg:</ p>
<?php get_template_part('loop'); /* loopen kommer bara att visa sökrutan */ ?>
<?php get_footer(); // footer.php ?>
```

Search.php

Man kan skapa en mall för söksidan, men jag tänkte visa ett exempel på enklare variant, som låter Wordpress bestämma var och när den ska visa sökrutan:

```
add_theme_support('html5', array('search-form'));
```

Och då kommer Wordpress att generera det här:

```
<form role="search" method="get" class="search-form" action="<?php echo home_url( '/' ); ?>">
  <label>
    <span class="screen-reader-text"><?php echo _x( 'Search for:', 'label' ) ?></span>
    <input type="search" class="search-field" placeholder="<?php echo esc_attr_x( 'Search ...', 'placeholder' ) ?>" value="<?php echo get_search_query() ?>" name="s" title="<?php echo esc_attr_x( 'Search for:', 'label' ) ?>" />
  </label>
  <input type="submit" class="search-submit" value="<?php echo esc_attr_x( 'Search', 'submit button' ) ?>" />
</form>
```

Nu är vi klara med basen för en Wordpress tema! Så nu kan du styla sidan och skapa specialmallar för kategorier, inlägg, sidor med mera, genom att följa Wordpress mallhierarkin:

<http://cr8gr8designs.com/fa-boken-html-till-wordpress-tema/>

http://codex.wordpress.org/Template_Hierarchy

Du kan alltid kolla funktionen `get_search_form` i Wordpress Codex, liksom alla andra funktioner.

TIPS! Du kan kolla upp vad för element och klasser Wordpress genererar, genom utvecklingsverktygen i t.ex. Chrome eller Firefox och » högerklicka på sidan och välj » "Inspect Element".

Så där, det var det! Hoppas att du lärde dig något, och till nästa gång, koda smart 😊 !