



WordPress

Build a WordPress theme from HTML5 template

- ◆ Nikolay Chochev
- ◆ Fmi Plovdiv



WEB CMS SYSTEMS

- ◆ Web Content Management Systems (CMS)
 - ◆ Software system providing Web site authoring, collaboration and administration tools
 - ◆ Users with little or no knowledge of HTML can create and manage Web sites content
- ◆ CMS typical functions:
 - ◆ Create / maintain Web sites
 - ◆ Add / edit / delete pages, documents, news, images, files, forms, polls, events, blogs, ...
 - ◆ Manage navigation, users, permissions, SEO, ...

BENEFITS OF CMS SYSTEMS

- ◆ WYSIWYG editor
 - ◆ No need for advanced HTML or CSS skills
 - ◆ No need for Dreamweaver / FrontPage / etc. or Wiki syntax to learn
 - ◆ Can paste content from a word processor
- ◆ Web-based access
 - ◆ Edit content from any computer
 - ◆ Staff can change content immediately
 - ◆ No waiting for Web admin to upload the content
 - ◆ Support for many types of documents and files

EXAMPLES OF CMS SYSTEMS

- ◆ Open Source CMS Systems

- ◆ Joomla

- ◆ Drupal

- ◆ WordPress

- ◆ DotNetNuke

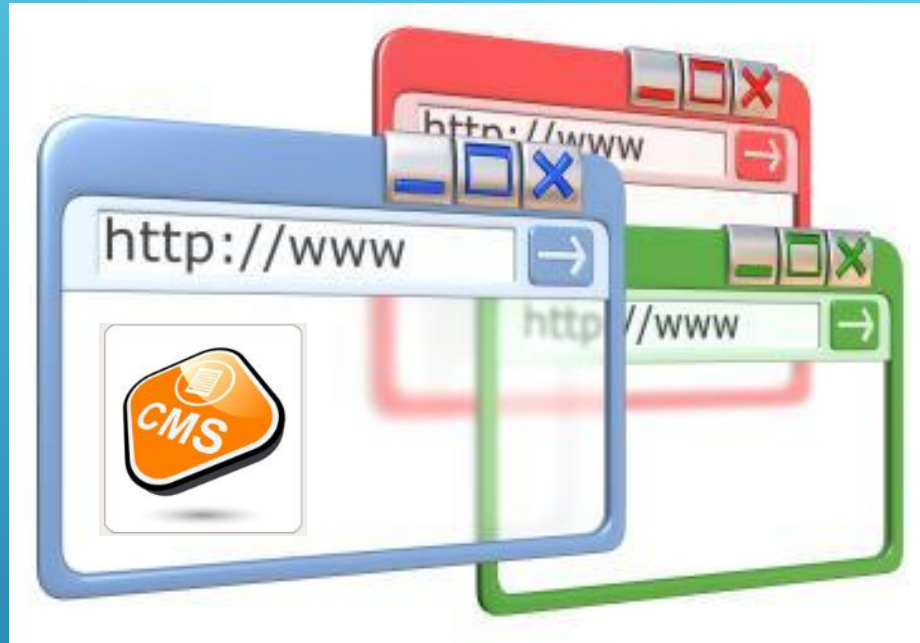
- ◆ Commercial CMS Systems

- ◆ SharePoint (MOSS)

- ◆ Telerik Sitefinity



Statistics: <http://trends.builtwith.com/cms>



WEB CONTENT MANAGEMENT SYSTEMS (CMS SYSTEMS)

Live Demo

WORDPRESS

Working with WordPress



WHAT IS WORDPRESS?

- ◆ WordPress is one of the world's leading blog and Web CMS systems
 - ◆ Open-source (free) software
 - ◆ Written with PHP and MySQL database
 - ◆ Stores all Web site contents in the database and images / files in the file system
 - ◆ Very powerful, easy-to-use, SEO friendly
 - ◆ Very large community – themes, plugins, ...
 - ◆ Official Web site: www.wordpress.org

INSTALLING AND RUNNING WORDPRESS

◆ Installing WordPress

- ◆ In the cloud (use WordPress as service)
 - ◆ Just create an account at www.wordpress.com
- ◆ At your Web hosting provider
 - ◆ Download it (e.g. **wordpress-3.8.zip**)
 - ◆ Unzip it in some public directory at the server
 - ◆ Create the database and run the install script
- ◆ Self-hosted
 - ◆ Install XAMP + WordPress at your local machine
 - ◆ XAMP = Windows + Apache + MySQL + PHP

WORDPRESS DASHBOARD

- ◆ The WordPress Dashboard is:
 - ◆ The administrative interface of WordPress
 - ◆ Create / edit / delete pages, posts, categories, tags, comments, documents, images, files, ...
 - ◆ Manage users, themes, widgets, plugins, settings, ...
- ◆ User-friendly UI
- ◆ WYSIWYG editor
- ◆ Media library



WORDPRESS THEMES

Using WordPress Themes

WORDPRESS THEMES

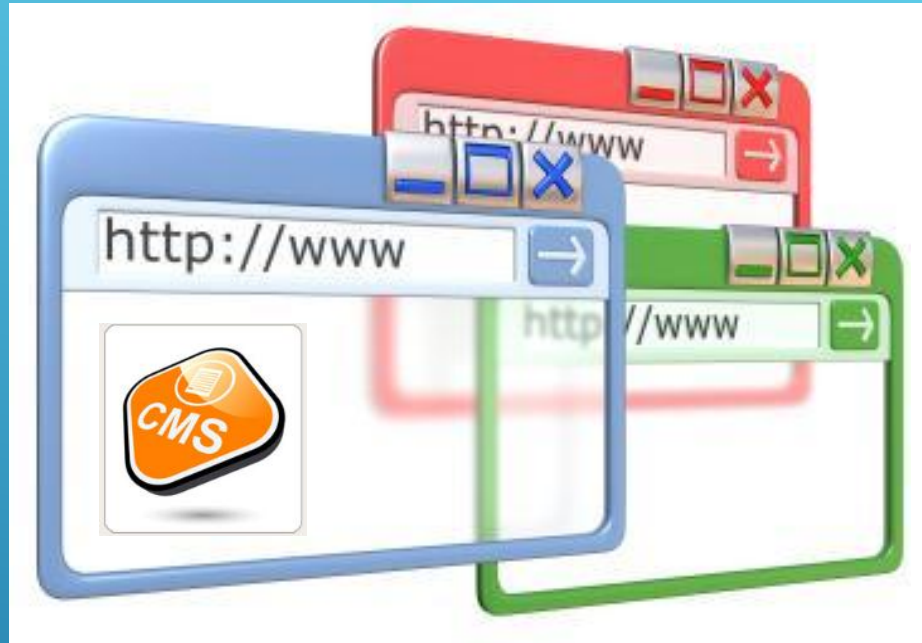
- ◆ WordPress supports themes
 - ◆ Themes define the structure, layout (the UI)
 - ◆ Can support or not widgets and menus
 - ◆ Can be single column / multi-column
 - ◆ Can be free or paid
 - ◆ The best themes are usually free
- ◆ Users can switch between the installed themes
- ◆ Official WordPress themes directory:
 - ◆ <http://wordpress.org/extend/themes/>

CUSTOM WORDPRESS THEMES

- ◆ Typical WordPress themes consist of
 - ◆ Main index template – **index.php**
 - ◆ Page template – **page.php**
 - ◆ Header – **header.php**
 - ◆ Footer – **footer.php**
 - ◆ Sidebar – **sidebar.php**
 - ◆ Comments template – **comments.php**
 - ◆ Styles – **styles.css**
 - ◆ Theme functions – **functions.php**

WORDPRESS

- ◆ Super mega powerful!
 - ◆ (surprise, surprise) 😊
- ◆ Over 90 000 000 websites online
- ◆ Tens of thousands of themes and plugins (both free and premium)
- ◆ Ready for eCommerce, Groupon, Q&A, forum, social network, ticketing system and more



INTEGRATING HTML5 TEMPLATE INTO A WP THEME

- ◆ We want to integrate our HTML sliced design into WordPress
- ◆ The end results are:
 - ◆ Same look and feel (design)
 - ◆ Dynamic functionality
 - ◆ Working menus, areas, blocks, comments and other CMS-related elements

SUPER IMPORTANT

- ◆ A PSD is not:
 - ◆ A brochure
 - ◆ A calendar
 - ◆ A wallpaper / magazine cover / whatever
- ◆ Site design should work with:
 - ◆ Different content (thousands of pages)
 - ◆ Different resolutions/browsers/operating systems/
platforms

CMS INTEGRATION

- ◆ Every CMS has it's own specifics and expectations in terms of markup
- ◆ It's best to know the system in depth in order to draw the appropriate design and slice it properly

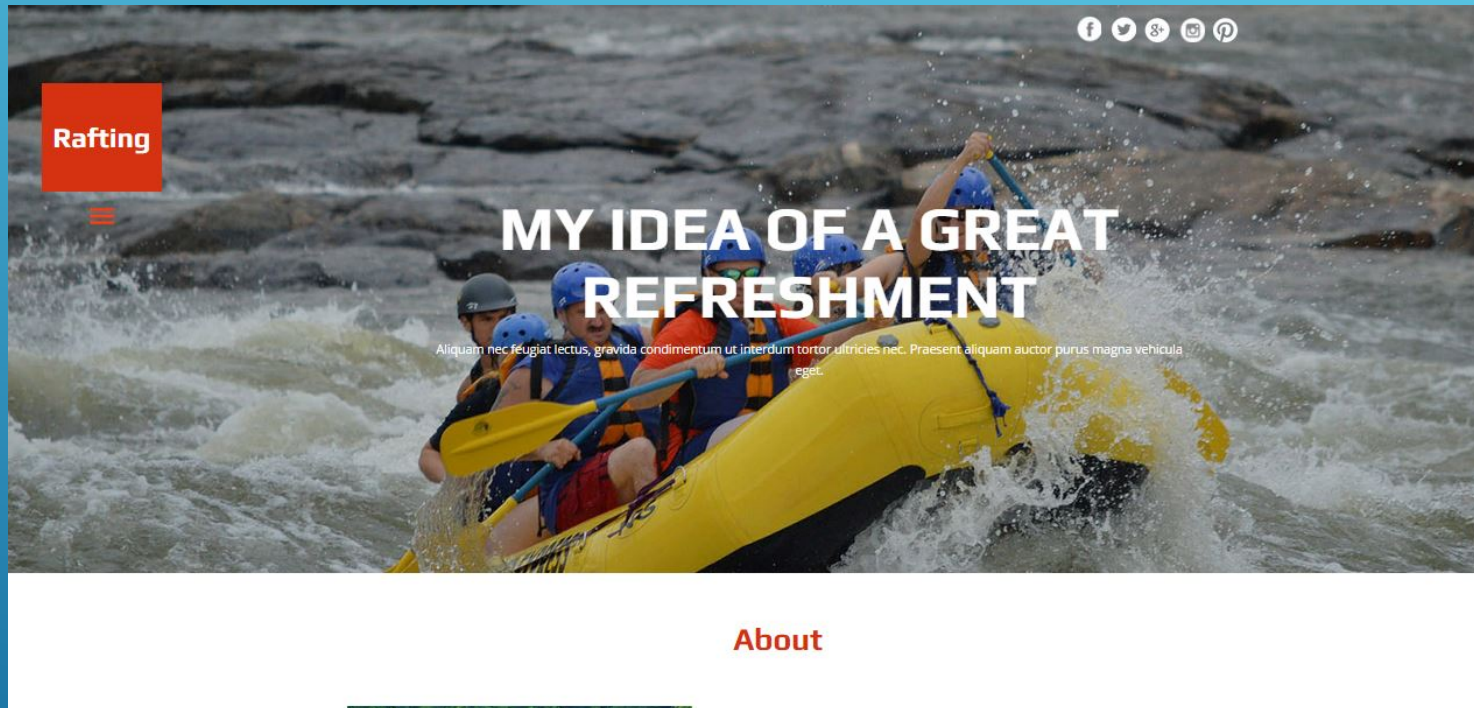
WORDPRESS

Let's integrate!



OUR TARGET

- ◆ Some random free HTML5 template



- ◆ Make sure that:
 - ◆ Code is valid and understandable
 - ◆ Code is reusable
 - ◆ Template markup seems WP-compatible
- ◆ Note: some super basic PHP knowledge is required

STEP 1

- ◆ Download the HTML template
- ◆ Browse all required resources:
 - ◆ HTML
 - ◆ CSS
 - ◆ JS
 - ◆ Fonts

STEP 2

- ◆ Setup development environment – XAMPP, LAMP/WAMP/MAMP
- ◆ Create a database for the new install
- ◆ Install WordPress Create a new folder in wp-content/themes

STEP 3

- ◆ Add the required files there:
 - ◆ style.css
 - ◆ index.php
- ◆ These are the two mandatory files for a WordPress theme

STEP 4

- ◆ Add an appropriate heading to style.css:

```
1  /*
2      Theme Name: Sample CodeWeek Theme
3      Theme URI: http://chochev.eu
4      Description: Theme for CodeWeek Demonstration
5      Author: nikych
6      Version: 1.0
7  */
```

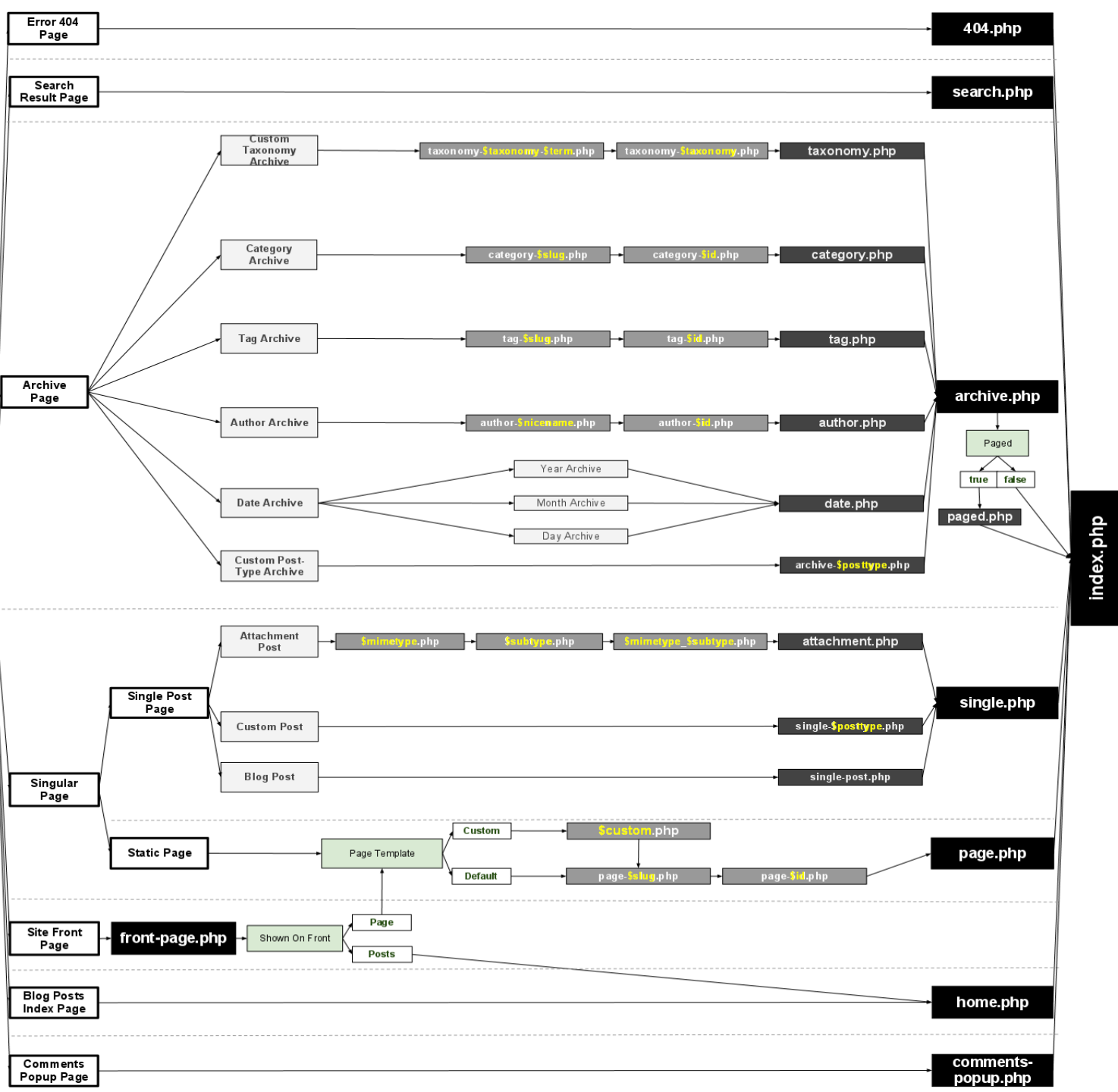
STEP 5

- ◆ Migrate your template CSS to style.css
- ◆ If there are more than 1 stylesheet files, there are 2 options:
 - ◆ Include them in header with normal `<style>` tag
 - ◆ Use `wp_enqueue_style` in functions file

STEP 6

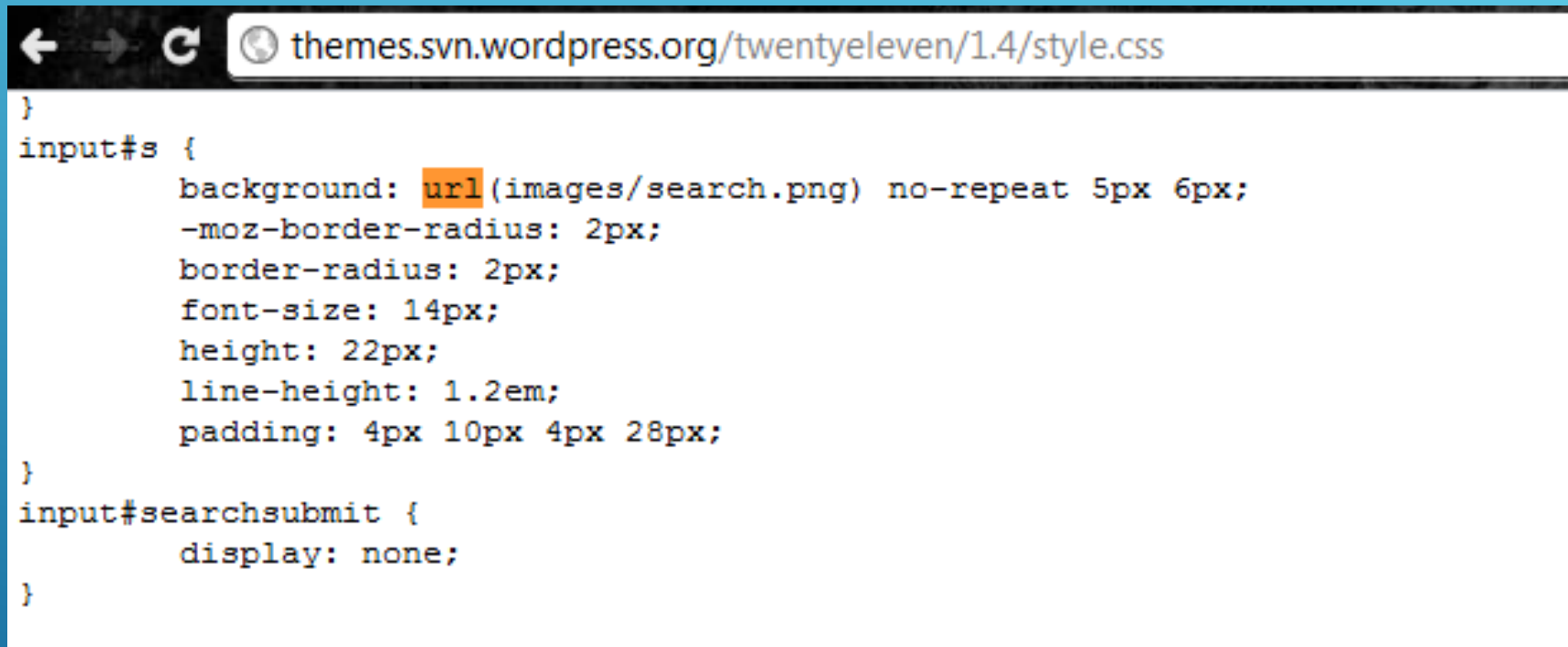
- ◆ Add header.php, footer.php and sidebar.php
- ◆ Separate the common parts of the site and move them to header, footer or sidebar pages
- ◆ There are other template files as well

What Page?



STEP 7

- ◆ Fix all static image paths in the style.css file



```

}
input#s {
    background: url(images/search.png) no-repeat 5px 6px;
    -moz-border-radius: 2px;
    border-radius: 2px;
    font-size: 14px;
    height: 22px;
    line-height: 1.2em;
    padding: 4px 10px 4px 28px;
}
input#searchsubmit {
    display: none;
}

```

- ◆ Organize images in images/ folder

STEP 8

- ◆ Add the style.css file to the header of the site:

```
6 <link rel="stylesheet" type="text/css" media="all"  
7 href="<?php bloginfo( 'stylesheet_url' ); ?>" />
```

- ◆ Fill in the index.php code for all pages (everything left besides header/footer code)

STEP 9

- ◆ Clear out all static page content from your markup (the demo texts – title and static content)
- ◆ Use this snippet instead:

```
72  
73 <?php if(have_posts()):  
74     while(have_posts()):  
75         the_post(); ?>  
76
```

STEP 10

- ◆ Use predefined tags, such as:
 - ◆ `the_title()`
 - ◆ `the_content()`
 - ◆ `the_permalink()`
 - ◆ They work in a loop
 - ◆ A single post view is like an array with 1 element
 - ◆ Set working paths with `get_template_directory_uri`

STEP 11

- ◆ Move index.php to home.php or front.php if you want a landing page
- ◆ Replicate index.php, but:
 - ◆ Add sidebar section
 - ◆ Correct small dynamic details such as dates
 - ◆ Use date('Y'), date('M'), date('d') for y/m/d

◆ Define sidebar

codex.wordpress.org/Function_Reference/register_sidebar

Example

This will create a sidebar named "RightSideBar" with `<h1>` and `</h1>` before and after the title:

```
register_sidebar(array(
    'name' => __( 'Right Hand Sidebar' ),
    'id' => 'right-sidebar',
    'description' => __( 'Widgets in this area will be shown on the right-hand side.' ),
    'before_title' => '<h1>',
    'after_title' => '</h1>'
));
```

◆ Add sidebar to a page

```
<div class="sidebar">
<?php if(!dynamic_sidebar('right-sidebar')): ?>
    <div class="subnav">
        <h5>Navigation</h5>
        <ul>
            <li><a href="#">Link</a></li>
            <li><a href="#">Link</a>
                <ul>
                    <li><a href="#">Sub Link</a></li>
                    <li><a href="#">Sub Link</a></li>
                </ul>
            </li>
        </ul>
    </div>
<?php endif; ?>
</div>
```

STEP 14

- ◆ Define menu

```
001 | if ( function_exists('register_nav_menu')) {  
002 |     register_nav_menu(  
003 |         array(  
004 |             'top-menu' => __('brand new menu'), ) );  
005 | }
```

- ◆ Call it a few times or use register_nav_menus for several menus

STEP 15

- ◆ Add the menu to a block in the markup

```
001 wp_nav_menu(array(  
002 'theme_location' => 'top-menu',  
003 'menu' =>,  
004 'container' => 'div',  
005 'container_class' => 'menu-{menu slug}-container',  
006 'container_id' =>,  
007 'menu-class' => 'menu',  
008 'menu_id' =>,  
009 'echo' => true,  
010 'fallback_cb' => 'wp_page_menu',  
011 'before' =>,  
012 'after' =>,  
013 'link_before' =>,  
014 'link_after' =>,  
015 'items_wrap' => '<ul id="%1$s" class="%2$s">%3$s</ul>',  
016 'depth' => 0,  
017 'walker' =>  
018  
019 ) );
```

- ◆ theme_location is sufficient

STEP 16

- ◆ Call `comment_form()` or `comments_template(...)`:

```
84  
85     <?php comments_template('', true); ?>  
86  
87     <?php // comment_form(); ?>  
88
```

STEP 17

◆ Enjoy! 😊



WHAT'S NEXT?

- ◆ Once you're ready with the theme, you could:
 - ◆ Add some complex plugins for more functionality
 - ◆ Integrate some APIs to social networks or remote services
 - ◆ Work on the next theme of yours

RECOMMENDED PLUGINS

- ◆ Types and Views
- ◆ WPML
- ◆ WooCommerce
- ◆ BuddyPress
- ◆ S2Member
- ◆ Contact Form 7 / Gravity Forms
- ◆ Free and Paid

WANT TO BE A WP NINJA?

- ◆ Check out Theme Reviewers Team guides:
 - ◆ Theme Review
 - ◆ Theme Unit Test
 - ◆ WPTRT page
 - ◆ Theme-Check plugin
- ◆ Become a Pro and submit on ThemeForest and WPORG



Questions?