

Preface

This class is for users who intend to use WordPress 2.7 as a site-building tool. The recommendations for this class are: a hosting account ready to accept WordPress, familiarity with FTP, a good grasp of HTML and CSS and a methodical approach to processes. Basic PHP skills helpful but *not* required.

Contents

The following subjects will be covered in this three-hour class:

Hour 1

- WordPress.org – finding the core code, plugins and the Codex
- Server requirements – minimums and recommendations
- First steps – create database and user, set up configuration file
- Lighting the LAMP – getting the code package loaded and running
- Preferences – what to set first and why
- Permalinks – .htaccess – what it is and why you should know
- Root vs subfolder – pros and cons

Hour 2

- Plugins – finding, installing, activating – some of my favorites
- Themes – finding, installing, activating, basics of customizing
- Themes 2 – inside the theme files, what they do and basic tweaks
- Category templates – making sections look different
- Page templates – making your homepage look different

Hour 3

- Navigation – using the stock widgets as well as plugins
- Disaster control – backing up, exporting, moving a site
- Troubleshooting – tracking down problems
- Q & A

WordPress.org – finding the core code, plugins and the Codex

The core code can always be found here:

<http://wordpress.org/download/>

Older versions can be found here:

http://codex.wordpress.org/WordPress_Versions

Plugin repository is here:

<http://wordpress.org/extend/plugins/>

Theme Directory is here:

<http://wordpress.org/extend/themes/>

Server requirements – minimums and recommendations

Hosting partners:

<http://wordpress.org/hosting/>

My recommendations:

Small sites, individuals - [Laughing Squid.net](http://LaughingSquid.net)

Large sites, companies – BlueHost.com

Local – Indra's Net – fantastic customer service, real people, *really* good. Relatively-new to WordPress hosting, but got it 99.9% right right out of the gate – the last 0.1% they worked with me quickly and politely to solve issues. A little pricey, but worth it if you value the personal touch.

Minimum requirements:

- PHP version 4.3 or greater
- MySQL version 4.0 or greater
- (Optional) Apache mod_rewrite module
(for clean URIs known as Permalinks)
Personally, I don't consider this optional. Permalinks don't just *look* better, they're easier for print publishers to deal with and users to *remember*.

First steps – create database and user, set up configuration file

The two recommended hosts have good solid control panels that make creating a database as simple as 1-2-3. The key things to remember: get the *final actual name* of the database and make *sure* the user has *all* rights. BlueHost appends the username to the database file and database user, don't miss that or you'll spin your wheels until you figure it out. If you decide to go with Indra's Net, they'll create the database, database user and password for you. Many non-control-panel hosts handle this step this way.

The config.php file

Here's the source code that matters:

```
// ** MySQL settings - You can get this info from your web host **
//
/** The name of the database for WordPress */
define('DB_NAME', 'putyourdbnamehere');

/** MySQL database username */
define('DB_USER', 'usernamehere');

/** MySQL database password */
define('DB_PASSWORD', 'yourpasswordhere');

/** MySQL hostname */
define('DB_HOST', 'localhost'); ← usually doesn't need changed

/**#@+
 * Authentication Unique Keys.
 * Change these to different unique phrases!
 * You can generate these using the {@link
https://api.wordpress.org/secret-key/1.1/ WordPress.org secret-key
service}
 *
 * @since 2.6.0
 */
define('AUTH_KEY', 'put your unique phrase here');
define('SECURE_AUTH_KEY', 'put your unique phrase here');
define('LOGGED_IN_KEY', 'put your unique phrase here');
define('NONCE_KEY', 'put your unique phrase here');
/**#@-*/

/**
 * WordPress Database Table prefix.
 *
 * You can have multiple installations in one database if you give
each a unique
 * prefix. Only numbers, letters, and underscores please!
 */
$table_prefix = 'wp_';

/** WordPress absolute path to the Wordpress directory. */
if ( !defined('ABSPATH') )
    define('ABSPATH', dirname(__FILE__) . '/');
```

Lighting the LAMP – getting the code package loaded and running

Note: all these steps are covered in even-greater detail at:

http://codex.wordpress.org/Installing_WordPress#Famous_5-Minute_Install

Once you've got WordPress loaded, the database and database user account created and the config.php file edited, you're ready to go. Just surf to your domain and you'll see this:



Enter your Blog Title and Your E-mail account and click 'Install WordPress'

You should see:



The installation process is now complete and your WordPress site is up!

Preferences – what to set first and why

Once you log in to your site you'll be at the *Dashboard* on the *General Settings* page.

General Settings

Tagline

You'll first want to set the 'Tagline' to something other than 'Just another WordPress blog' – think of this as your 'motto.' In the latter half of this class we'll talk about about hiding this text and using an image-only header file.

Membership

In general, it's more effort to maintain a site where 'Anyone can register' – spammers love this and will harrass you endlessly. I encourage leaving it unchecked

and registering users manually, assuming you're going to have anything other than subscribers.

New User Default Role

Never never never EVER set this to 'Administrator' unless access to your site is strictly limited. You *might* want to do this if your entire site is behind a password or on a private server. User roles can be changed at any time but user **names** cannot.

Date and Time Format

You can pick a display format for all dates and times WordPress displays. You can change this after launch and changes are comprehensive.

A discussion of the date and time format and use is at:
[http://codex.wordpress.org/Formatting Date and Time](http://codex.wordpress.org/Formatting_Date_and_Time)

Writing Settings

Default Post Category

If your site has a commonly-used posting category, you may want to set the *Default Post Category* to that category to prevent users (and yourself!) from creating posts in the default non-removable '*uncategorized*' category.

Remote Publishing

If you're a hardcore road warrior and into 'moblogging' you might want to enable this option. It lets you simply send e-mail to your site and it auto-publishes into the specified post category. Note: if the recipient address gets popularized, you may find someone else running your site. I'd be very very cautious using this feature and keep the address a carefully-guarded secret as well as never *never never* EVER cc folks. If (when!) they '*reply all,*' guess what. Right. All that said, a great way to report live from events with nothing more elaborate than a web-enabled cell phone.

Update Services

If you're looking for traffic, update services are your friend. If you're using Feedburner to propagate your RSS feeds, you'll definitely want to look into these.

Reading Settings

This settings panel is where you specify how the blog presents information.

'Front page displays' lets you set your site up as a 'blog' or as a page-based website.

Note: you can change this setting at any time so your homepage can be changed easily to support special events.

'Blog pages show at most' controls how many postings show up in the category index pages. By careful use of the *'more'* tag to split posts or using *excerpts* you can make your blog categories much more user-friendly.

'For each article in a feed, show' controls what goes into your RSS feed – note, full-text feeds can become a performance issue if your site gets heavy traffic. My advice is to get a feed reader and check what you're sending out.

Discussion Settings

'Default article settings' the first two check boxes (notifications) can cause performance issues if your site gets a lot of traffic. The last one *'Allow people to post comments on the article'* controls whether or not posts and pages have the *'allow comments'* checkbox in the editor to be checked by default. Generally I leave this unchecked on most user sites to avoid pages erroneously getting a comment field.

'Comment author must fill out name and e-mail' to me, is a no-brainer. Folks can put bogus values in, but all comments report their IP address – more on that later.

'Users must be registered and logged in to comment' – this lets you control who can comment. If you control registration and require this, only people you've granted access to can post comments. Assuming you trust these folks, you could conceivably turn off moderation.

Automatically closing comments is a good idea if you get lots of comments – same is true for threaded comments and breaking comments into pages.

'E-mail me whenever' lets you know when people have left comments. If you're getting too much mail, you might want to turn these two checkboxes off and manage comments only through the site.

'Before a comment appears' lets you specify that an admin must always approve comments. I strongly encourage checking this option.

'Comment moderation' lets you control exactly which comments even make it to moderation. You can specify that comments with links (almost always spam) or containing certain words (viagra comes mind) always get moderated. The Akismet plugin is also invaluable for keeping this under control.

'Comment Blacklist' lets you send comments with naughty words or comments from undesirable IP addresses to limbo, no moderation needed. Caution – this means you

can never *ever* use those words in a comment – the comments won't show up. Maybe if an admin posts it, but that could be very confusing to users. Use with care.

Avatars. Almost everyone has a image avatar these days. This lets you control *if* they show up and *how* they show up. You can also filter out ones with offensive images.

Permalinks – .htaccess – what it is and why you should know

If you're going to use permalinks (and I encourage you to), when you set up your code files, you need to make sure you have an .htaccess file.

Mac users take especial note – naming a file on your local machine with a leading period makes it invisible! Also, some FTP tools need to be specifically adjusted to display 'dot files'.

You can take any file, delete it's contents and rename it .htaccess to create this file. Since WordPress will need to edit this file, you'll want to set its permissions to 777 so WordPress can update it, then go to the '*Permalinks*' settings page and choose the scheme you'd like. I generally use the 'Day and name' scheme. When you *Save Changes* WordPress will tell you that it successfully updated the .htaccess file OR it will advise you to do so yourself if the permissions aren't set up so that it can do it.

If you choose to use a caching plugin, be warned that it –will- edit your .htaccess file. Tinkering with this file can and *will* render your site inoperable. Proceed with care and *always make backups*.

If you need to have specific urls for special reasons, you can do 'redirects' that allow users to go to a 'custom' url and be automatically redirected to the final destination. This is also helpful when content is moved or urls change. Note – caching schemes and redirects are compatible but care should be taken to avoid 'circular' redirects which will definitely upset the host server and disable your site.

Root vs subfolder – pros and cons

Putting your WordPress install into a subfolder creates a number of issues when you're ready to move it to the 'root' of the site. Since all the database references as well as the pointer to your theme are stored in the database, just moving your code files to the root isn't enough, you have to change the General setting to the root as well. There's a long and detailed discussion at:

[http://codex.wordpress.org/Giving WordPress Its Own Directory](http://codex.wordpress.org/Giving_WordPress_Its_Own_Directory)

that explains this issue in depth. I encourage you to read it FIRST if you're planning to put WordPress in a subdirectory.

Generally speaking, I try to *always* put WordPress in the root of the site and leave it there. It's fine to create a 'dev' site (using the wp_ table prefix in the config.php file) and have a work area, but don't attempt building WordPress in a subdirectory and then moving it to the root when you're all done unless you're *confident* you can execute *all* the steps at the url listed above. Otherwise you'll end up with a broken site, an unhappy client and a real headache.

End of hour 1

Hour 2

Plugins – finding, installing, activating – some of my favorites

Plugins are where WordPress gets its superhero tights and cape – however, just like Superman, the wrong plugin or conflicts can be just like Kryptonite. Treat new plugins with care and don't just toss a new one into the mix and expect it to work and play well with others. If you start seeing problems in your WordPress site, new plugins are suspect #1. Most plugin authors write them for free, if you visit their support forums, be polite and keep in mind their support to you is *optional*.

WordPress.org maintains a large library of plugins at:

<http://wordpress.org/extend/plugins/>

Since there are currently over 4,000 plugins, here's a list of some of my favorites:

- [Akismet](#) – controls comment spam
- [antileech](#) – stops content-scrapers, bane of bloggers
- [bluetrait-event-viewer](#) – logs events, spots hackers
- [calendar](#) – (very) simple scheduling tool
- [cforms](#) – Best-of-class form creation data-capture
- [custom post templates](#) – mimics the page template feature
- [dd-import-users](#) – lets you bring in large sets of users/subscribers
- [dd-multi-col-links](#) – lets you convert your link list into multi-columns
- [event-calendar](#) – more elaborate calendar tool
- [firestats](#) – very elaborate stats tool
- [Jeromes keywords & manager](#) – old but reliable
- [kimili-flash-embed](#) – allows simple placement of Flash elements
- [limit-login-attempts](#) – stop brute-force password cracks
- [mtr-podcast-recorder](#) – capture podcasts directly
- [multi-level-navigation](#) – excellent drop-down menus
- [mypageorder](#) – drag/drop menu arrangement
- [nextgen-gallery](#) – excellent image gallery manager
- [page link manager](#) – simple page exclusion from menus

- [reveal-ids-for-wp-admin](#) – for advanced users
- [search pages](#) – very simple hack to force page searches
- [simpleflickr](#) – embed image galleries from FLICKR
- [sitemap-generator](#) – creates self-maintaining XML site map for Google
- [stats](#) – simple and effective statistics viewer
- [subscribe2](#) – lets users subscribe to postings
- [surveygizmo](#) – use the survey gizmo service inside wordpress
- [tinymce-advanced](#) – unleash the visual editor
- [vipers-video-quicktags](#) – embed all kinds of video easily
- [wordpress-navigation-list-plugin-navt](#) – amazing menu manager
- [wp-db-backup](#) – simple way to capture your database
- [wp-lightboxJS](#) – popup image effect
- [wp-simpleviewer](#) – another simple image display tool
- [wp-super-cache](#) – handle heavy traffic

Themes – finding, installing, activating, basics of customizing

Themes are essentially 'skins' for your site.

WordPress.org currently has over 600 themes at:

<http://wordpress.org/extend/themes/>

You can download a theme, unzip it and then upload it to your wp-content/themes folder, then from the Appearance / Themes activate it.

Changing themes will not alter or damage your site content.

Very extensive documentation on themes can be found at:

http://codex.wordpress.org/Using_Themes

The following are the files typically included within a Theme (though not *always*)

- 404 Template = 404.php
- Archive Template = archive.php
- Archive Index Page = archives.php
- Comments Template = comments.php

- Footer Template = footer.php
- Header Template = header.php
- Links = links.php
- Main Template = index.php
- Page Template = page.php
- Popup Comments Template = comments-popup.php
- Post Template = single.php
- Search Form = searchform.php
- Search Template = search.php
- Sidebar Template = sidebar.php
- Stylesheet = style.css

Themes 2 – inside the theme files, what they do and basic tweaks

Every theme has a CSS file. At the top you'll find something like this:

```
/*
Theme Name: Lunasea
Theme URI: http://www.macwebguru.com/
Description: Lunasea is a custom designed theme just for Luna Beach
Resort.
Version: 1.0
Author: Macguiguru
Author URI: http://www.macwebguru.com/
*/
```

If you start with a published theme and want to make it your own, first duplicate the entire theme folder, then customize this information any way you like, then activate the theme in the Appearance / Themes admin page.

The **header.php** file is where you'll insert navigation schemes or calls to navigation routines generated by plugins like NAVT or just hard-code your own. You can also choose which meta elements are included – you may not want folks to know your site is run with WordPress, so you might remove the Generator tag. This is also where you set the DTD your theme uses.

The **sidebar.php** file is where you can hard-code your own navigation, but I generally use the widgets functionality. If you happen to pick a theme that's not 'widget-ready' you can 'widgetize' it by adding a few lines of code to the sidebar.php file. Caution: mistakes in the PHP will cause your site to not render – keep backups so you can 'fall back and regroup' if needed.

The **index.php**, **page.php** and **single.php** control the appearance of category lists, pages and single posts respectively. By rearranging the elements in these theme files you can control the appearance of pages.

Note – there is a ‘loop’ structure in these files that you must –not- break or your site will stop functioning. You won’t lose any content, but things will just get wonky. Remember, backup backup backup. If you’re not skilled with PHP, move small pieces and use care to test each change. Even an inexperienced user can slowly iterate a design to the desired state.

Here’s a typical **index.php** file:

(refer to http://codex.wordpress.org/The_Loop as well as live file)

The **footer.php** file is where you can add specific customized information about you, the developer. Also, if the user doesn’t want their admin login page shown in the sidebar, you can turn that off and insert a link in the footer to `/wp-admin/` which will take the user to the dashboard, authenticating if they haven’t already.

The `searchform.php` file controls how your search appears, even if you’re using widgets. Here’s a typical `searchform.php`:

```
<form method="get" id="searchform" action="<?php bloginfo('url');
?>/">
<div><input type="text" value="<?php the_search_query(); ?>"
name="s" id="s" />
<input type="submit" id="searchsubmit" value="Search" />
</div>
</form>
```

You can add any special text you’d like or additional divs to allow special CSS formatting.

Category templates – making sections look different

More information about this section can be found here:

http://codex.wordpress.org/Category_Templates

By duplicating the either the **index.php** or **archive.php** file and renaming it: **category-#.php** you can specify a special appearance to any category’s list of posts. Note, the hierarchy for use of these theme files is as follows:

1. `category-6.php`
2. `category.php`
3. `archive.php`
4. `index.php`

Page templates – making your homepage look different

To create a page template, simply duplicate the page.php file, rename it and add this code to it's beginning:

```
<?php
/*
Template Name: Your Template Name Here
*/
?>
```

Once you've saved the file, the template will be available to apply to any page in the site. You can thus create a custom look for every page in your site. In the case of:

<http://www.roatandiving.net/>

The page templates allow the user to control which Flash presentation is applied. Presence and position of page elements can also be controlled this way.

(show example of changing page template)

Hour 3

This section is all demonstrations.

Navigation – using the stock widgets as well as plugins

Disaster control – backing up, exporting, moving a site

Troubleshooting – tracking down problems

Q & A