Software Developer's JOURNAL
new ideas & solutions for professional programmers

Vol.2 No.13 Monthly Issue 13/2013 (19) ISSN 1734-3933

Building Your Own WordPress Sites from Scratch

Most Exciting Tips and Instructions Written by Best Experts

Dear Readers,

We would like to present you with our first WordPress issue! As promised, you will find tutorials varying on different levels of complexity and a wide range of topics.

With a strong following worldwide, WordPress has become one of the most popular content managements systems (CMS) used for web development. Its popularity continues to grow as the flexibility and dynamics of the platform are being pushed further everyday. Without further adieu, here it is a brand new SDJ issue!

We've started a partnership with some of the best developers in the industry who are willing to share their knowledge and practical skills with you. Some of them will contribute to our magazine on a regular basis, so we would be very thankful for any feedback.

What's inside? Jeremy Holcombe and Lynda Gilmore explain the very basics of building a WordPress site, while Jim Gallaher shares secrets of Self Hosted WordPress and Gary Glasscock tells how to create a Simple Responsive Website.

You'll also find Jeffrey Zinn's great networking concepts, clearly expressed in the article 'How To Speed Up Your WordPress Website', Tish Briseno's tricks in 'How to Enhance your Website with Plugins & Widgets', and Brett Widmann's article covering the basics of keeping WordPress secure from getting hacked.

Don't forget to look through our unique +EXTRA articles. Dr. Yildirim Kocdag presents his 'Beyond Mobile Gestures' and Kevin Donnigan shares some news about 'Mobile Navigation: User Friendly Techniques For Small Devices'. Futhermore, Matthew Ruppert's new input about 'Mistakes' and a special gift – additional ebook devoted to author's WordPress instructions.

Enjoy your reading!

Anna Łakomy

& SDJ Team

SDJ Team:

Editor in Chief: Anna Lakomy
anna.lakomy@sdjournal.org

Editorial Advisory Board: Kishore PV

Special thanks to our Beta testers and Proofreaders who helped us with this issue. Our magazine would not exist without your assistance and expertise.

Publisher: Paweł Marciniak
Managing Director: Ewa Dudzic
Production Director: Andrzej Kuca
andrzej.kuca@sdjournal.org

Art. Director:
Jan Duda, Marta Kamieńska

DTP: Jan Duda, Marta Kamieńska
Marketing Director: Ewa Dudzic

Publisher: Hakin9 Media SK
02-676 Warsaw, Poland Postepu 17D
Phone: 1 917 338 3631
http://en.sdjournal.org/

Whilst every effort has been made to ensure the highest quality of the magazine, the editors make no warranty, expressed or implied, concerning the results of the content's usage. All trademarks presented in the magazine were used for informative purposes only.

DISCLAIMER! The techniques described in our magazine may be used in private, local networks only. The editors hold no responsibility for the misuse of the techniques presented or any data loss.

Table of Contents

First steps:

Getting involved:

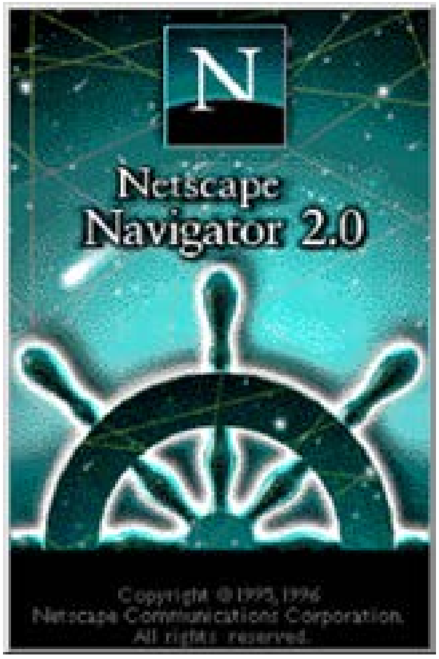
EXTRA +

EXTRA ++

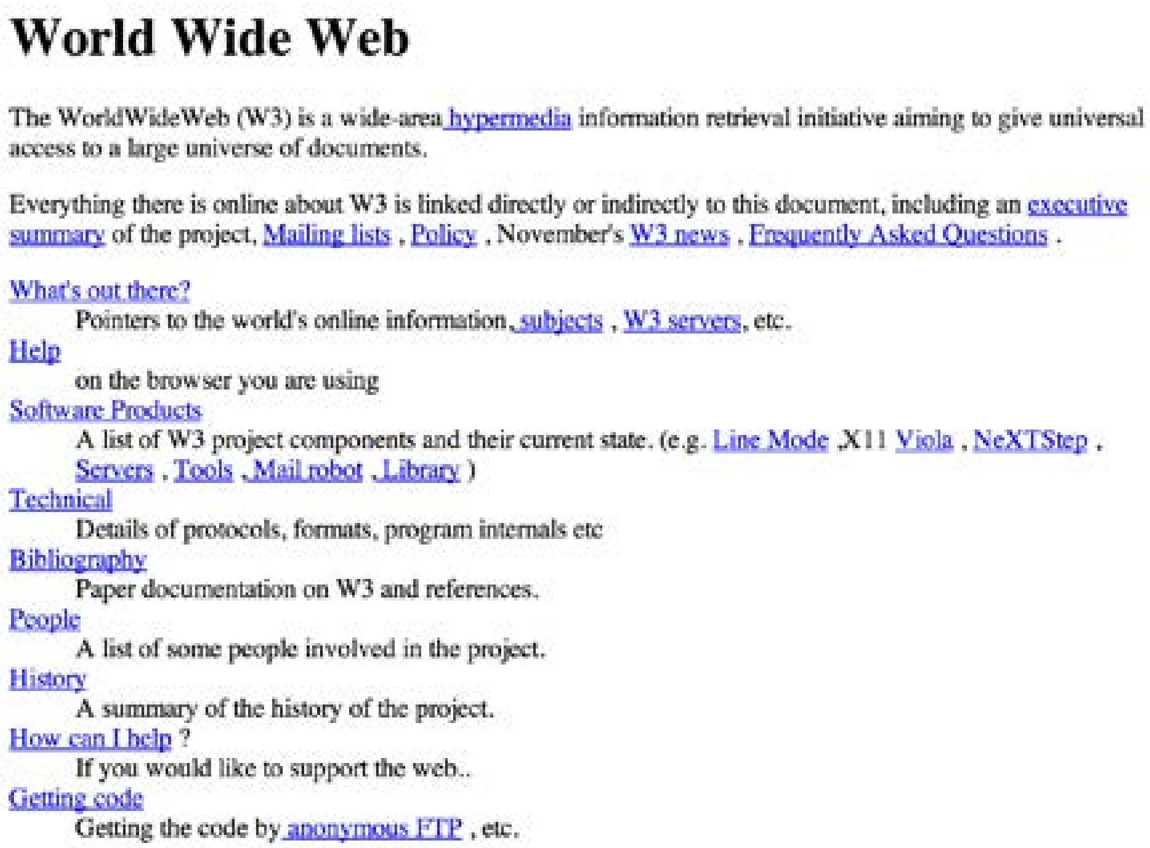
A Brief History of The Web (Very Brief)

Not long ago I cam across an article about "The World's First Website." Without going into any debate about what exactly that means, or whether or not British physicist Tim Berners-Lee can be fully credited for inventing the World Wide Web at CERN in 1989, this much is true: The early days of the World Wide Web were, well, how should I put this? A bit static.

By *static* I mean a couple of things. There was a palpable excitement about it. The first time I ever viewed a website it was on a text-only browser called *Lynx*. The content was dynamic only insofar as it had been created at some point and uploaded to a web sever. I didn't realize that what I was viewing was being fed to my **VAX/VMS** terminal window was being served by something called web server, or that the web server was very likely **CERN httpd**. I didn't mind that the pages didn't change from day-to-day. Nor did I mind that the websites weren't pretty. For the time being, I was sufficiently fascinated by the endless links from topic to topic.



Copyright © 1995, 1996
Netscape Communications Corporation.
All rights reserved.



World Wide Web

The WorldWideWeb (W3) is a wide-area hypermedia information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an executive summary of the project, Mailing lists , Policy , November's W3 news , Frequently Asked Questions .

What's out there?
    Pointers to the world's online information, subjects , W3 servers, etc.
Help
    on the browser you are using
Software Products
    A list of W3 project components and their current state. (e.g. Line Mode ,X11 Viola , NeXTStep , Servers , Tools , Mail robot , Library)
Technical
    Details of protocols, formats, program internals etc
Bibliography
    Paper documentation on W3 and references.
People
    A list of some people involved in the project.
History
    A summary of the history of the project.
How can I help ?
    If you would like to support the web..
Getting code
    Getting the code by anonymous FTP , etc.

The First Web Page at CERN

In retrospect, those early days of the World Wide Web moved by pretty quick. Soon we had things like Slipknot, Mozilla, and Netscape (or should I say *Netscape Navigator*?), to display web pages in increasingly flashy ways. Soon it seemed everyone had a website—and most of those websites were, well, awful. Using free hosing sites like Tripod, Angelfire, Earthlink, and The Globe. With anyone and everyone creating websites, we became accustomed to an overload of wasted web space. Remember the centered flashing red text? Remember the slow-loading GIFs? Remember the obligatory under construction images? YUCK!

Blogging became a trend around 2002, and it wasn't long before multitudes of what amounted to online journals full of amateur political commentary or teen angst exploded. With the advent of Twitter and Facebook, those who desired little more than public announcement of personal opinions found more 'appropriate' outlets. Blogging survived, still survives (thrives, even), but only the best blogs receive significant traffic. WordPress and Blogger were two of the early blogging tools. Blogger, purchased by Google, has floundered. Sites created by Blogger stand out like a broken URL. WordPress, on the other hand, has grown and adapted. These days it is unfair to call WordPress a blogging tool. It can be just that, but its much more. It's a full functioning content management system. It's an expandable, customizable, scalable foundation upon which a vast range of websites are built.

Here's the bad news: The bad sites are still out there, and more of them are being created every day. The good news? The WWW audience has become more shrewd, with higher expectations of appearance and content. The even better news? WordPress makes it easy to create a foundation upon which those with something to share can do so in a way that contributes and does not detract.

WordPress From Scratch

About a week ago I got a call from a friend of mine who is looking to create a website for his small business. This friend is a small business owner—a contractor who worked on upgrades and additions to homes. "Archie," I said, "the best way to go about this is to let me set up a WordPress site for you. Once it's set up, I think you'll find it is extremely easy for you to maintain yourself."

"Isn't WordPress for blogging?" he asked.

It used to be it. It certainly still is. However, WordPress has evolved tremendously from its initial release in 2003. With thousands of templates, plugins, and easy customizability, WordPress has become one of the single best ways for businesses, small to large, to host their web presence. Don't think of WordPress as a blogging tool. Think of it, instead, as a powerful website creation tool, with powerful CMS, administration, feedback, sharing, and expansion capabilities.

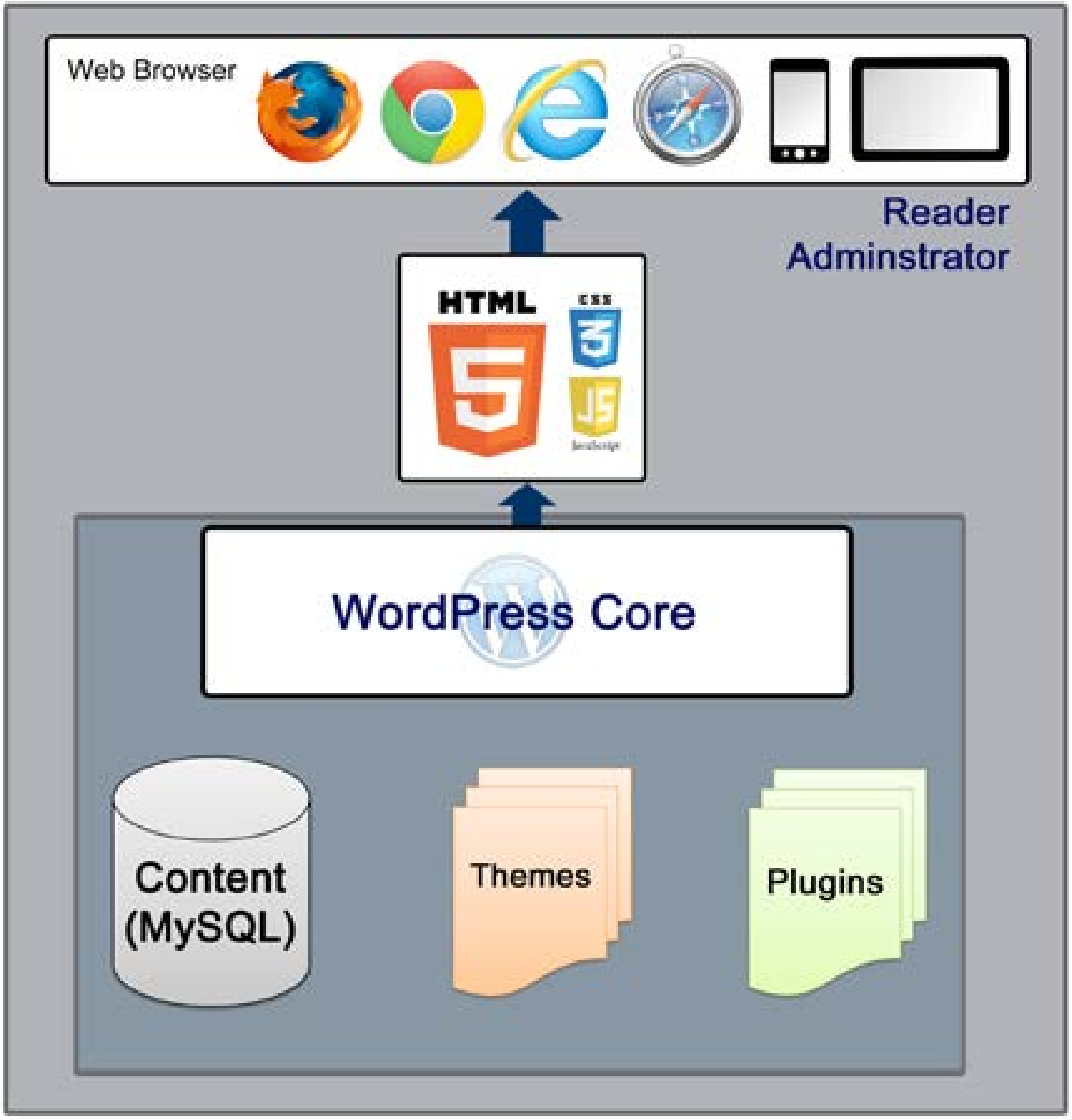
So while WordPress has become a robust, powerful solution for websites of all kinds (see the WordPress showcase at http://wordpress.org/showcase/ to see a surprising and eclectic range of sites using WordPress), it remains a powerful blogging tool as well. Sure, some out there may suggest "blogging is soooo 2006," but as for me, I remain an avid blogger and blog reader. The change in blogging over the years, with the growth of Facebook, Twitter and Google+, has been a good one. Those who created a blog simply as a way of keeping in touch with family or talking about themselves have found a voice in a more appropriate forum. Many of the remaining, most popular blogs, are those with original and interesting content. If I may editorialize for a moment, I suggest that before you start blogging, and even if you are continuing a history of blogging, strive to create content. Anyone can post links. A great blogger contributes!

What I personally love about Wordpress is this: It passes both the mom test and the uber-geek test. By *mom test*, I mean this: Could my mom use it? Yes (sorry mom!). By *uber-geek test*, I mean that it is open and configurable enough that even the most demanding power user will find satisfaction in the level of control granted by WordPress.

Due to its popularity, a key advantage is that the plugin architecture continues to grow. And with templates that are highly customizable, most of them free, it is indeed very easy for non-techies to administer their own sites. That said, for a self-hosted WordPress site, there is a bit of technical expertise required to get things up and running. With many providers, perhaps most, offering WordPress hosting, setup of a custom installation can be as easy as a few clicks. And for those running their own servers, given the knowledge required to do such a thing, installation of WordPress will seem quite simple.

It is necessary to clarify some terms here. WordPress is software. That is, WordPress, used generically, refers to all of the server-side PHP implementation that creates the functionality for management of all the blog/site settings, from posts to pages to plugins to settings. WordPress itself is not an ISP or hosting solution of any kind. WordPress is all the "stuff" that organizes and present content.

WordPress is developed by Automattic, Inc., a web development company that provides a number of other services, most of which tie-in nicely with it's flagship product. It is released under the GNU General Public License, meaning that all the code behind it is open source, and can be contributed to by anyone (anyone with the necessary skills, that is). One of the most popular blogging/CMS/website creation tools around, it is capable of handling very large amounts of data. It scales well enough for a small blog to a large news site.



WordPress.com vs. WordPress.org

When it comes to starting a WordPress site, there are two major options to consider: self-hosting or *wordpress.com* hosting. Each has its benefits and drawbacks, but self-hosting (either on your own server or through a hosting provider) provides the greatest amount of freedom. How to approach hosting is a question only you can answer. It's all a matter of cost, freedom of control, and technical ability/confidence.

Wordpress.com hosting may be a good option for many, but it is with limitations. In the case of a *wordpress.com* hosted site, available themes, plugins, and customizations are limited. A user cannot implement or install custom plugins, and tailoring is limited to that which is provided by *WordPress.com*. Still, it can be a great option for a number of reasons. First and foremost: It's free! This means that some users will see ads on your page (*free* isn't always ***free***).

It's simple and easy, and one need not pay for additional hosting. If you're willing to spend a few bucks, there are custom themes, fonts, colors, and domain name customization. *WordPress.com* hosting is easy, fast, stable, and secure. You'll never have to concern yourself with backups or scramble if your site gets hacked. All such security and damage control is handled by *WordPress.com*.

As a true business front or for hosting a personal site wherein one desires great customization, the limitations are likely more than such a customer desires. For this article, I am focusing on the setup of a self-hosted *wordpress.com* site.

Before continuing, note that there have been many updates to WordPress since it's initial release in 2003. For this article, WordPress 3.5+ is discussed.

Design

Web pages and style preferences are similar in many ways to fashion trends—What is considered sleek looking now will be considered dull (or downright ugly) in a year or two.

If you're setting up a website for a customer, this is easily overcome by simple switching of templates. New templates are always being created, and people spend many hours tailoring and testing them. Any web designer who has struggled with CSS and worked long hours to make sure a site renders correctly across multiple platforms and browsers will understand just how tedious this can be.

Because WordPress is so widely used, there is no shortage of new, trendy looking templates to choose from. Sure, you can spend hours upon hours creating your own (and some people, myself included, consider this *fun*), but starting with a great theme and customizing it a little here and a little there is a much safer, more timesaving approach.

Even for those who prefer to work start from scratch, there are a number of barebones templates available, created with the intent of laying out the basics (including specific WordPress hooks and tags). For the brave, those who prefer to go it alone and create something, I most definitely recommend starting from a barebones template.

Back to the shortcomings of a *WordPress.com* hosted site: With limited configurability and a limited number of template to choose from, your ability to create something entirely unique is very much limited. Some of the templates (many of the best ones) cost (anywhere from $50-120 per year), as does CSS, font, and color configurability. So if you do go the *WordPress.com* hosted route, be aware that viewers will likely stumble upon other *WordPress.com* sites that appear similar to your own.

That said, since it's free, it may not be a bad idea to get started and to understand WordPress is to set up a test blog for free at WordPress.com (assuming you have no prior experience with WordPress). If you do this, please be a good *web citizen*: Creation of a WordPress.com site ties up a name (e.g. flimflam.wordpress.com) that someone else may actually want to use. So be sure to remove the site and the blog so that others are not blocked from using it. It is frustrating when attempting to use a domain name or a blog name that is tied up by an unused or temporary site.

To sum up what I've stated above, there are really three options:

- WordPress.com hosting
- A paid hosting provider
- Self-hosting (Your own IP Address, server, domain name, installation, and configuration of both WordPress and the server.)

Hosting Providers

WordPress.org presents a few options to users in need of hosting. (For more details see http://wordpress.org/hosting/). For further details on the differences between WordPress.com and WordPress.org, see http://en.support.wordpress.com/com-vs-org/.

The remainder of this article is about setting up a self-hosted WordPress site. After setting up a WordPress self-hosted site, the most obvious immediate difference will be the appearance of the left-menu. Menu options for Upgrades, Ratings, Polls, and certain plugins will not be immediately available in the WordPress.org (self-hosted) dashboard.

Before Starting

If installing WordPress on a Windows server, there will be a few additional requirements, including installation and configuration of Apache, MySQL, and PHP. With so many differing server environments, from Mac to Windows to all of the various Linux distributions, environment needs are configured. The most simple installations will require some knowledge of hosting environments (assuming you intend on fully hosting yourself), including configuration of PHP, MySQL, and Apache. In a self-hosted setup, you will want to be well aware of security considerations. The overall configuration of Apache Web Server is beyond the scope of this article. Although I will touch on a few points, if self-hosting (not using a subscription hosting provider), take note that experience in web configuration and security is necessary.

Although any web server that supports MySQL and PHP will work, WordPress recommends either Apache (as do I) or Nginx.

Minimal WordPress installation requirements are:

- HP 5.2.4 or greater
- MySQL 5.0 or greater
- The mod_rewrite Apache module

In this article I will assume Apache HTTP Server is used (as is commonly the case).

PHP

WordPress and it's plugins are written in PHP. If you are a software developer, it is likely that you have at least some knowledge what PHP is. In case you do not, PHP is simply an interpreted scripting language used in many LAMP systems. LAMP is an acronym for common web applications that use Linux, Apache HTTP Server, MySQL database (although PostgreSQL is also commonly used), and PHP, Perl, or Python for backend serviced and rendering HTML. By this definition, WordPress is a LAMP application.

> ***Definition***
>
> *LAMP is a combination of free, open source software. The acronym LAMP refers to the first letters of Linux (operating system), Apache HTTP Server, MySQL (database software), and PHP, Perl or Python, principal components to build a viable general purpose web server.*
>
> **Wikipedia.org, https://en.wikipedia.org/wiki/LAMP_(software_bundle)**

While it is not necessary to know PHP, you will likely see some PHP code here and there. This is particularly true if you venture into editing the appearance of your theme beyond simple CSS modifications, modify, or write a plugin. Syntactically, PHP may seem foreign to those without prior experience, but it not a difficult language for a programmer to understand.

JavaScript

Modification of self-hosted *WordPress.org* sites will not require an expert level of JavaScript understanding. However, as with PHP, a number of bells and whistles can be added with JavaScript. (*WordPress.com* hosted blogs prevent inclusion of JavaScript enhancements.) This is another major benefit of using a hosted *WordPress.org* site (along with the ability to include jQuery or other JavaScript libraries).

Graphics

The default WordPress themes generally come with a canned set of graphics for use as headers, backgrounds, buttons, and separators. While it may be acceptable at first to use the default content, with so many other sites using similar themes, I encourage you to create, minimally, a custom header. Most themes will include a description of recommended header dimensions. I generally use Photoshop for my needs, but GIMP ("GNU Image Manipulation Program") is a powerful free alternative (*www.gimp.org*). Even if you are building a site for a customer, you will find it necessary to modify provided images to the necessary dimensions. As a sidebar, I don't recommend using an image that is too small for a particular area. For example, if a recommended header size is 800x600 pixels, a 200x100 image stretched to fit the area will look awful (not to mention unprofessional).

For buttons and headers, when customized, I prefer using the PNG format, which provides the compression and clarity of JPG, but allows transparent backgrounds.

Definition

Portable Network Graphics (PNG) *is a raster graphics file format that supports lossless data compression. PNG was created as an improved, non-patented replacement for Graphics Interchange Format (GIF), and is the most used lossless image compression format on the World Wide Web.*

Web Browsers

While modifying your WordPress site, always be sure to verify appearances in all of the most popular browsers, not simply your personal favorite. As I watch the hit statistics on my blog (*matthewrupert.net*), I've noticed a sharp increase in the number of users coming from mobile clients and Google Chrome. Perhaps because my blog is technical in nature, most readers are using either Google Chrome or Firefox. There are surprisingly few using Internet Explorer.

I always test my web pages on Chrome, Firefox, Internet Explorer, and Safari. If you do not happen to have a Mac available for testing, take comfort in knowing that if your site looks the way you want it to in Chrome and Firefox, it will likely render appropriately on Safari.

Newer WordPress themes have settings for mobile browsers, which may be enabled or disabled. I recommend enabling mobile rendering whenever possible. Think about your personal website reading. I'm willing to bet that you often use Android or iOS on a phone or tablet.

Installation

Guided vs. Non-Guided Installation

Assuming your hosting provider does not provide an interface for installation (e.g. cPanel), you'll need to know your way around the command line at least a bit. If your hosting provider does provide a cPanel interface, or something similar, chances are there is already a very easy WordPress installation path. In either case, after WordPress is installed, you'll want to configure some things. Installation through a hosted interface certainly simplifies the first few steps, and if this is your situation, skip ahead. Unless you are installing WordPress on your own server, changes are Apache is already installed.

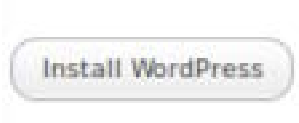
As long as your server has PHP and MySQL installed, the platform matters little (although the vast majority of WordPress installations are hosted on Linux). Because there are so many possible configurations and hosting provider setups, there is no straightforward answers to some of the questions that may arise. Throughout this guide I will attempt to answer those questions as much as possible. Be aware that things like PHP installation, MySQL installation, database setup, and web server configuration are highly dependent on many variables.

That said, even if it is necessary to install everything from Apache to PHP, a knowledgeable software/administrator will find installation to be straightforward (although a bit longer than the "famous 5 minutes" that WordPress advertises—particularly if you are setting everything up from scratch). Even the database setup is easy. All you need to do as the installer is create an empty database with permissions for your WordPress account to access it. For anyone with limited operating system and web administration experience, I don't recommend attempting to install "from scratch" without first learning a few things about web hosting.

If installing our your own server, there's no reason you can't install WordPress locally on a laptop if you wish, just to figure things out before attempting a *production* install. All the tools needed, MySQL, PHP, and Apache, are all capable of running on nearly any OS. This isn't a bad idea, as you may find it necessary to tinker for a while before attempting to real thing.

Installing Using cPanel/Hosting Provider Services

My hunch is that those with limited technical knowledge are willing to pay a hosting provider, and hosting providers generally include web interfaces (such as cPanel) that make all of this simple. These web-hosting control panels, along with *autoinstallers* such as **Softaculous**, **Fantastico**, and **Installatron**, remove the difficult parts. Some hosts provide multiple autoinstallers, and these programs often include all the database creation steps and domain mapping.



The cPanel software package a control panel commonly used by hosting providers. This software, and other packages like it, gives website owners easy ways to manage their websites and services. Installing WordPress, in this case, is very simple, especially is your MySQL user database is already configured (and it probably is). In this case, your host will provide an installation program such as **Installatron** or **Softaculous**.

In general, a **cPanel** installation of WordPress includes the following steps:

1. Log in to your cPanel. (this is typically something like https://<my-host-provider>/cpanel)
2. Use **MySQL Database Wizard** (These database steps may not be necessary. Your hosting provider may have already created your database. All you need to do is provide a WordPress table prefix and database access credentials.)
3. **Create a Database:** enter the database name and click Next Step.
4. **Create Database Users:** enter the database username and the password. Note that this is the database username and password—not the WordPress administrator username and password, which will be set up later.
5. **Add User to Database**: Select All Privileges checkbox before moving on to the next step.
6. **Complete the Database Setup:** Be sure to take note of the database name and user. There is nothing as frustrating as a forgotten hostname, username, databasename, or password. For this database, hostname will likely be localhost (127.0.0.1), as it is accessed by the server locally, not via some external path.

To set up the database, the installation process will request an administrator username, password and WordPress table prefixes. These table prefixes are used to distinguish WordPress tables from other tables (assuming the user has only one database schema per account).

Other hosting providers may allow for multiple MySQL databases, in which case it will be necessary to choose a database as well as WordPress table prefixes. Since there is no single approach among the many providers

Installation to Using Existing Database on Hosting Provider

If your hosting provider already provides a MySQL database created for each user (as many do), and you are not using a automatic installer (such as Softaculus), it is a good idea to check the *wp-config.php* file to gather this information about the WordPress installastion. This step is necessary for the MySQL configuration settings required later. There will be a default administatory username and password in the *wp-confing.php* file, so take note of this information. When you have WordPress up and running, one of the first things you will want to do is change the admin password.

MySQL Installation (Self-Hosting or Hosting Provider without MySQL)

First things first, MySQL must be installed. MySQL, a popular choice of database for use in web applications and LAMP applications (such as WordPress, provides a powerful, free (it is open source) database. WAMP and MAMP are not unheard of acronyms (Windows, Linux, Apache, PHP/Perl and Mac, Linux, Apache, PHP/Perl), and as long as the configuration is correct, WordPress can be served from Windows and Mac OSX without issue.

In the unlikely scenario that are not self-hosting and your hosting provider does not provide MySQL, it will be necessary to find a new hosting provider or work with the provider to have MySQL set up for your account. If you are self-hosting, and in complete control of your server environment, fear not: Installing and configuring MySQL is straightforward.

Installing MySQL, assuming your system does not already have it available, can be just a bit confusing, as there are several editions to choose from. MySQL Enterprise Edition and MySQL Cluster CGE are commercial packages. Although MySQL is open source, these particular editions are meant for businesses. They cost, and provide much more functionality and support than what most WordPress users require. We want the MySQL Community Edition (licensed under GPL). This can be downloaded from *http://dev.mysql.com/downloads*. Here you'll see a number of other MySQL tools that are also available for free under the GPL, but for now, all that is really necessary is MySQL Community Server.

MySQL can be extracted from a tar file (on all operating systems) or as a native installer (on some operating systems). When possible, a native installer is the simplest approach.

Note: With MySQL now being "owned" and hosted by Oracle, it is necessary to create an account to download even the free edition. If downloading directly from MySQL/Oracle, the downloaded file is a zip or a tar.gz file. This means that installation will include expanding the contents in a desired location. For those on non-Linux systems without package management utilities such as yum and apt-get, installers and/or tar bundles are available at http://dev.mysql.com/downloads/mysql.

Only once instance of MySQL server is required, as multiple WordPress sites can utilize schemas on the same MySQL database. The only requirement here is that each WordPress site has unique table prefixes to distinguish the MySQL database table names of each installation of WordPress. This is very easy to do—In fact, WordPress installation prompts for a database table extension during setup.

Most hosting providers provide a single database instance for their customers, and this is not a problem whatsoever for hosting multiple WordPress sites. In fact, a number of different applications may use that database. Not to worry! WordPress will know which tables belong to it.

RedHat/Fedora

With RedLinux (RedHat Enterprise, Fedora, CentOS), installation is as easy as using the yum installer:

```
sudo yum install mysql-server mysql
```

If you are a RedHat user, chances are you're already quite familiar with yum. Yum is the RedHat software package manager, used for installation and updates (as well as removal) of packages on all RPM-based systems. The nice thing about installing via yum is that all dependences can easily be included in a single command. The entirety of WordPress cannot be installed via yum, but php, MySql, and Apache HTTP can be installed with the single line:

```
yum install mysql-server php httpd
```

Ubuntu

Ready for this? You're gonna like this! Everything you need to do to install WordPress on Ubuntu is as follows:

```
   sudo apt-get install wordpress
   sudo ln -s /usr/share/wordpress /var/www/wordpress
   sudo apt-get install mysql-server
   sudo bash /usr/share/doc/wordpress/examples/setup-mysql -n
wordpress localhost
   sudo /etc/init.d/apache2 restart
```

The second line above simply creates a symbolic link from /var/www/wordpress (the place Apache serves files from) to the WordPress installed path. In the third step, while installing MySQL, you'll be prompted for an administrator/root password. Enter one, and don't forget it!

The "setup-mysql" step creates the necessary MySQL WordPress tables (and similar steps can be taken on other platforms).

Ready for this? Open up a browser and navigate to sudo bash /usr/share/doc/wordpress/examples/setup-mysql -n wordpress localhost

Voila! Boom goes the dynamite! There you have it! There on that page you see a form with some things that need to be entered before your WordPress site is actually a *real site*, but for all intents and purposes, the heavy lifting is done. To be clear, this isn't what one might consider a complete installation. In this example WordPress is running as root, and therefore not servable to the outside world. There are a number of Apache owner and group configuration steps required here that, while not overly complex, run beyond the scope of this article.

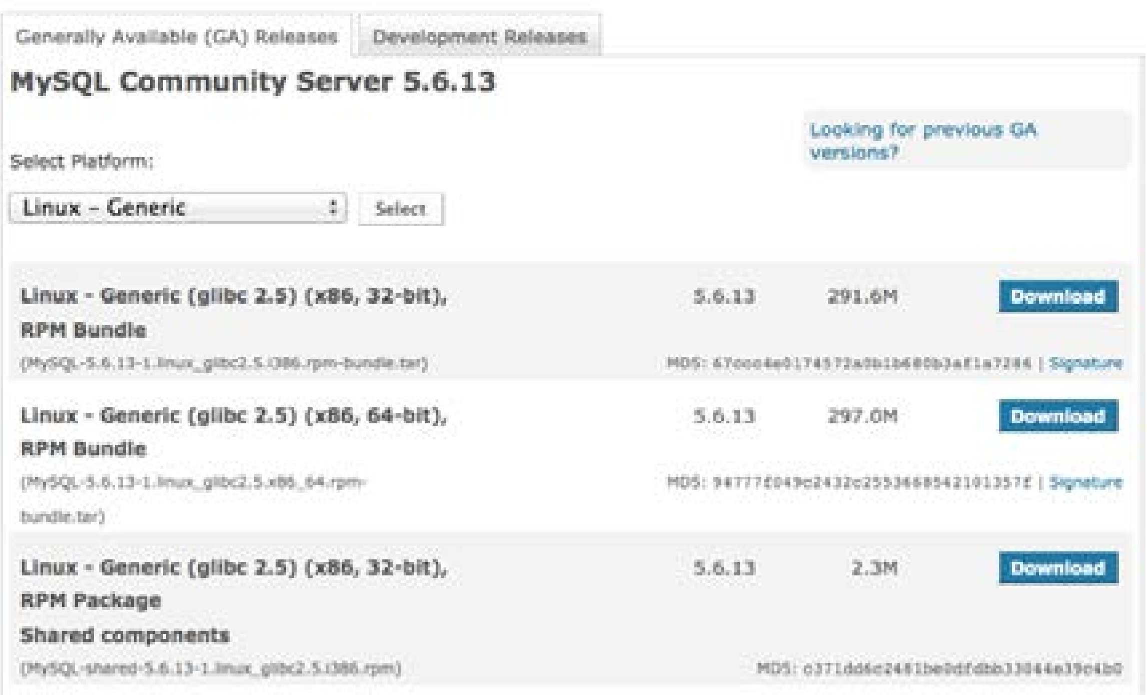
Microsoft Windows

A MySQL Installer for Windows (MSI) can be found at *http://dev.mysql.com/downloads/installer*.

Mac OS

MySQL provides a native dmg installer for Mac OS. See the MySQL downloads page.

Other Operating Systems



MySQL Community Server 5.6.13

Looking for previous GA versions?

Select Platform:

Linux – Generic

Linux - Generic (glibc 2.5) (x86, 32-bit), RPM Bundle (MySQL-5.6.13-1.linux_glibc2.5.i386.rpm-bundle.tar)	5.6.13	291.6M	Download MD5: 67ccc4e0174572a0b1b680b3af1a7286 | Signature
Linux - Generic (glibc 2.5) (x86, 64-bit), RPM Bundle (MySQL-5.6.13-1.linux_glibc2.5.x86_64.rpm-bundle.tar)	5.6.13	297.0M	Download MD5: 94777f049c2432c25536685421013 57f | Signature
Linux - Generic (glibc 2.5) (x86, 32-bit), RPM Package Shared components (MySQL-shared-5.6.13-1.linux_glibc2.5.i386.rpm)	5.6.13	2.3M	Download MD5: c371dd6c2481be0dfdbb33044e39c4b0

If using an OS for which there is no native MySQL installer available, it will be necessary to download and extract the MySQL application from a *tar.gz* file.

Now that MySQL has been installed, we won't worry about it for a bit (ore on this later). The next step is putting the WordPress core files in place.

SSH/Command Line

While the installation tools provide the capability of installation without ever touching the command line, those installing on their own servers, or using a server with SSH available, will find that knowledge of using the command line is necessary.

FTP Client

WordPress settings can usually be modified using the existing administrator pages. Even custom themes can be uploaded directly through the user interface. If you prefer to modify the files directly (rather than making changes locally, uploading, and testing), a good FTP client might speed the process. See *http://codex.wordpress.org/FTP_Clients*.

Downloading Wordpress Manually

The current version of WordPress (3.5.2 at the time of this article) can be downloaded from *http://wordpress.org/download/* as a *.tar.gz* or .zip file. WordPress touts a an easy "5-minute installation," and, assuming other requirements are already set up (hosting with MySQL available), this really is the case. There are ways to run WordPress with other databases, but MySQL is the standard. If the previous steps (using apt-get, yum, or another platform specific installer are not satisfactory), manual download and extraction is easy enough.

1. Download and Extract WordPress
   The first step is to download WordPress. Assuming you're working in a Linux environment (or on a Mac), grab the **.***tar.gz* file. A .zip file is also available. Obviously you can grab it through your favorite web browser, but since we're going to be working at the command line anyway, it's just as easy to use curl or **wget** to retrieve the latest WordPress build. Fortunately, the folks have

   WordPress have made this very simple. Prior to downloading, go ahead and change to the directory in which you wish to install WordPress.

   Depending on your Apache configuration (see DocumentRoot in *httpd.conf*), this location will vary. In most Linux distributions it is /var/www (but don't assume this).

   Depending on your particular installation, and assuming this is a server that you have some degree of control over, the path to install depends a lot on personal preference. Do you plan on hosting *only* this WordPress site? If so, you may want to install directly in the Apache DocumentRoot.

   Chances are, however, that you want to install WordPress in a subdirectory. I recommend using a subdirectory, as it is much easier to configure Apache to point to another path than it is to move WordPress after it has been installed. This is certainly true if you have existing web content at the root level.

   Should you attempt to put WordPress's *index.php* and the same directory level of an existing *index.php*, chances are the results won't be your desired goal. If what you have at that level is old content that you wish to get rid of, remove it first. If you instead want WordPress to be available at the same DocumentRoot (both the existing *index.html* and WordPress's *index.php*), you'll need to make a decision now about what you want available and how you want it to be accessed. It could be that Apache first reads PHP files when searching for an index. In this case, when a client hits http://<my-url> , your WordPress will be displayed. This would require clients wishing to hit the old *index.html* to explicitly enter *http://my-url/index.html* in the browser. This is not a recommend approach. Instead consider installing WordPress in a subdirectory (e.g., /var/www/html/wp). By installing in this manner, users (as well as site links from whatever your existing content is) can access your WordPress site in a consistent, expected way. This also enables better domain and subdomain mapping within your Apache configuration.

So here are my initial steps:

```
cd /var/www/html/
sudo mkdir wp
sudo wget http://wordpress.org/latest.tar.gz
or
sudo curl -O http://wordpress.org/latest.tar.gz
```

(For the sake of this article, I am skipping over explanation of sudo and other Unix commands. If you are uncomfortable with the above, and attempting to install WordPress on your own server, it is highly recommended that you first understand the basics.)

2. Set permissions

   The typical Apache installation is set up with User and Group Apache. Other systems may use www or _www. If unsure, just check the *httpd.conf* file and search for the User and Group settings:

   ```
   User Apache
   Group Apache
   ```

   In the previous step, while extracting the .*tar.gz* file, the WordPress directory, and all the files within were extract with the current user's Group and current user as the the owner. This needs to be corrected for Apache. The Group and Owner. To change this to something that Apache has the correct read/write permissions, we simply need to change the Group and Owner. To do so recursively:

   ```
   cd /var/www/html
   chown –R apache:apache wp
   ```

The above command changes the owner and group of each of the files and subdirectories in the wp directory to apache:apache. It may be possible to run WordPress without setting these permissions, but it is proper and most secure to have the correct Apache owner and group settings so that we need not worry about access permissions.

In the above steps, I assumed WordPress would be installed in the webroot in a /wp subdirectory. This may or may not be what you wish to do. WordPress can be placed in a subdirectory of wwwroot, in which case default user access is:

http://<url>/wp/

DNS mapping, virtual hosts, or Apache redirects can always be used to redirect from http://<url>/wp to http://<url>. This is my preferred approach, as it offers greater flexibility as a web administrator later. You may wish to configure a subdomain by way of Apache so that your WordPress site is accessed like so:

http://<my-site-name>.<my-domain-name>

Plugins will be addressed later, but it is worth noting here that there is a great plugin available for mapping various categories, post types, and pages to different subdomains: WP Subdomains: *http://wordpress.org/plugins/wp-subdomains-revisited/*

WordPress Installation

Now that WordPress has been extracted into your Apache served directory, you may be able to perform a quick test to see that the server is working by viewing *http://localhost/wp* in a browser. Here you'll see a list of the installed WordPress files (assuming Apache is configured to allow viewing of a directory index through mod_autoindex). The missing link here is the URL redirect and/or the document root for Apache. To be clear, this won't be the case if you did the more simple cPanel or easy installation steps including the configuration of MySQL using /usr/share/doc/wordpress/examples/setup-mysql -n wordpress localhost.

If, however, MySQL and/or PHP have yet to be configured, the following steps are necessary.

DocumentRoot, the Apache *httpd.conf* configuration setting is set by default in many Apache installations to be the root of your user web part (for example, /var/www or /user/var/www, or something similar). With full control of Apache, there are several approaches that can be taken to site redirection. If you intend only to serve your WordPress site, DocumentRoot can point to /var/www (for example). A more flexible solution is to utilize DNS redirects or VirtualHosts so that you are not limited to only hosting a single site. This is all a matter of Apache configuration, the details of which go well beyond the scope of this article.

At the root level of the served WordPress path (the DocumentRoot or the path that is redirected to when your domain is accessed), an *index.php* file appears (this already exists in the /wp directory that was created during WordPress extraction). This is the php file that redirects to the WordPress site. It is necessary that your web server is properly configured to automatically read *index.php* files upon access, and that no other index files compete with it (i.e., there should be only one index file at the DocumentRoot).

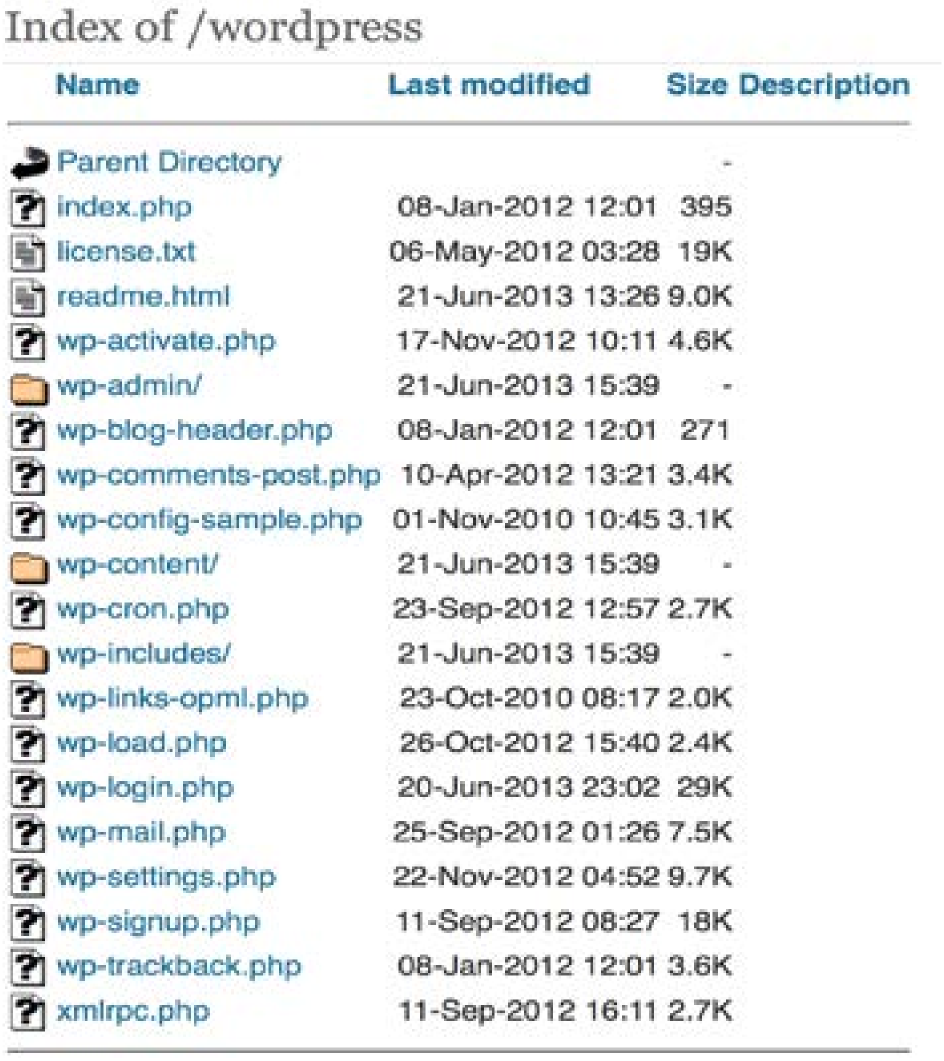
The contents of this file are as follows:

```
<?php
/**
 * Front to the WordPress application. This file
doesn't do anything, but loads
 * wp-blog-header.php which does and tells
WordPress to load the theme.
 *
 * @package WordPress
 */

/**
 * Tells WordPress to load the WordPress theme
and output it.
 *
 * @var bool
 */
define('WP_USE_THEMES', true);

/** Loads the WordPress Environment and
Template */
require('./wp-blog-header.php');

Of course, MySQL has has been installed, so
there may still be a few configuaton steps. so
you'll only see a listing of files (depending on
Apache configuration, you may get a Forbidden
error).
```



Index of /wordpress

Name	Last modified	Size	Description
Parent Directory		-	
index.php	08-Jan-2012 12:01	395	
license.txt	06-May-2012 03:28	19K	
readme.html	21-Jun-2013 13:26	9.0K	
wp-activate.php	17-Nov-2012 10:11	4.6K	
wp-admin/	21-Jun-2013 15:39	-	
wp-blog-header.php	08-Jan-2012 12:01	271	
wp-comments-post.php	10-Apr-2012 13:21	3.4K	
wp-config-sample.php	01-Nov-2010 10:45	3.1K	
wp-content/	21-Jun-2013 15:39	-	
wp-cron.php	23-Sep-2012 12:57	2.7K	
wp-includes/	21-Jun-2013 15:39	-	
wp-links-opml.php	23-Oct-2010 08:17	2.0K	
wp-load.php	26-Oct-2012 15:40	2.4K	
wp-login.php	20-Jun-2013 23:02	29K	
wp-mail.php	25-Sep-2012 01:26	7.5K	
wp-settings.php	22-Nov-2012 04:52	9.7K	
wp-signup.php	11-Sep-2012 08:27	18K	
wp-trackback.php	08-Jan-2012 12:01	3.6K	
xmlrpc.php	11-Sep-2012 16:11	2.7K	

Notice that there are three main directories in the WordPress extract: wp-admin, wp-content, and wp-includes. The files within wp-admin and wp-includes are, effectively, the core php files. The wp-content directory is where custom files, themes, plugins, and other media are placed. Expect the content of this directory to grow as your WordPress site is built and modified. Presentation code, that which is presented through the web browser (posts, pages, categories, links, comments, etc.), is generated dynamically by PHP code, so don't expect to find any html files in these paths. It is highly unlikely that you will need to directly manipulate any of the code in these paths. WordPress version updates and all customization is best done through the provided WordPress administration screens.

Apache/Web Server Configuration

Since WordPress is a PHP application, make sure Apache is set up for PHP (that is, the Apache PHP module is included and configured). This should be as simple as going to your Apache configuration file (*httpd.conf*) and uncommenting the following line:

```
LoadModule php5_module libexec/apache2/libphp5.
so
```

In a typical Linux environment. Apache configuration files are located in /var/conf/httpd of /var/www/conf. This all depends on where it was installed.

For this article, I will assume that you already have a basic installation of Apache Web Server complete. Configuration of Apache can be very simple or very complex, and it is a topic too large to cover here. In your hosting environment is under your control, some Apache configuration knowledge is required. There is no requirement that Apache be used as the web server, but it is the most commonly used web server, both for WordPress and most PHP applications.

Back at the command line, Apache needs to be restarted for this configuration change (inclusion of the PHP module) to take effect.:

```
sudo apachectl restart
```

Next we need to set up MySQL. Most Linux environments already have it installed, and again, if you're dealing with a hosted environment, much of this can be skipped. If this is your situation, you simply need to create a database to be used for WordPress.

MySQL Database

There are certainly ways that WordPress can be configured with databases other than MySQL, but such configurations are not officially supported, and may lead to a number of technical gotchas. That said, there is really no good reason to attempt a database other than MySQL. On the contrary, to do so will cause more problems than it is worth.

Because many hosting providers include WordPress, it configuring the MySQL database may not be necessary. Most hosting providers include a pre-configured MySQL database for each user, or installation scripts that make the process as simple as clicking and choosing schema names. If this is your situation, see your hosting provider's help and support pages (generally hosting providers provide some sort of control panel page, which may include handy services like Softaculous or Installatron).

While not necessary, myPHPAdmin offers a simple web-based interface for administering a MySQL database. If you prefer to avoid the command line for future administration, this may be worth looking into. Chances are, if your hosting provider uses guided installation programs like those already mentioned, your MySQL administration can already by done using myPHPAdmin or a similar tool.

If you have only one database and it is already in use, you can install WordPress in it - just make sure to have a distinctive prefix for your tables to avoid over-writing any existing database table.

The MySQL installation includes documents, manuals, and detailed installation instructions. Start by viewing the README and INSTALL-BINARY files in the MySQL path (e.g., /var/local/mysql).

The following steps are necessary to create a specific mysql user and group.

```
shell> groupadd mysql
shell> useradd -r -g mysql mysql
shell> cd /usr/local
shell> tar zxvf /path/to/mysql-VERSION-OS.tar.
gz
shell> ln -s full-path-to-mysql-VERSION-OS
mysql
shell> cd mysql
shell> chown -R mysql .
shell> chgrp -R mysql .
shell> scripts/mysql_install_db --user=mysql
shell> chown -R root .
shell> chown -R mysql data
shell> bin/mysqld_safe --user=mysql &
# Next command is optional
shell> cp support-files/mysql.server /etc/
init.d/mysql.server
```

Note: The above example assumes Linux commands. In a Mac or Windows environment, it is best to use the System Preferences/Control Panel interfaces to great users and groups.

As with Apache HTTP Server, MySQL administration is a very large subject. There are many factors to consider when hosting a database with MySQL. For WordPress hosting, all of the heavy lifting of database schema creation is handled. All we need to do is create the database, user, and permissions. To create all that, we're going to create a separate schema.

To do this, we need to go back into MySQL at the command line (MySql must be running, obviously. This setup of this can be done via the UI or as is done in the above shell commands, with the line "cp support-files/*mysql.server* /etc/*init.d/mysql.server*." In this case, the mysql server must first be started. If it is not running, go ahead and do /etc/*init.d/mysqld* start from the command line. You will probably want to set MySQL up to run as a default service.

In the following steps we'll create a schema specifically for WordPress. We'll create a database called "wordpress" and a user for that database called "wp." Remember the admin/mysql user password that you created previously. Alternatively, "root" can be substituted as the username. After this setup, however, there won't be much of a need to use root for WordPress tasks.

```
shell>  mysql -u mysql –p
Enter password: Welcome to the MySQL monitor.  Commands end with ; or
\g. Your MySQL connection id is 5340 to server version: 3.23.54   Type
'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>CREATE DATABASE wordpress;

Query OK, 1 row affected (0.00 sec)

mysql> GRANT ALL PRIVILEGES ON wordpress.* "wordpressusername"@"hostname"
  -->IDENTIFIED BY "password";

Query OK, 0 rows affected (0.00 sec)

mysql> FLUSH PRIVILEGES;

Query OK, 0 rows affected (0.01 sec)

mysql> EXIT

Bye $
```

The above steps created the initial MySQL user. These steps did not create specific WordPress schema. This task is done by the WordPress setup. The beauty of this is that the rest of our setup can be done via the browser.

Installing WordPress using cPanel

If your hosting provider provides the *cPanel* hosting control panel (and it is very likely that it does), congratulations! The steps to installing WordPress are dramatically simplified.

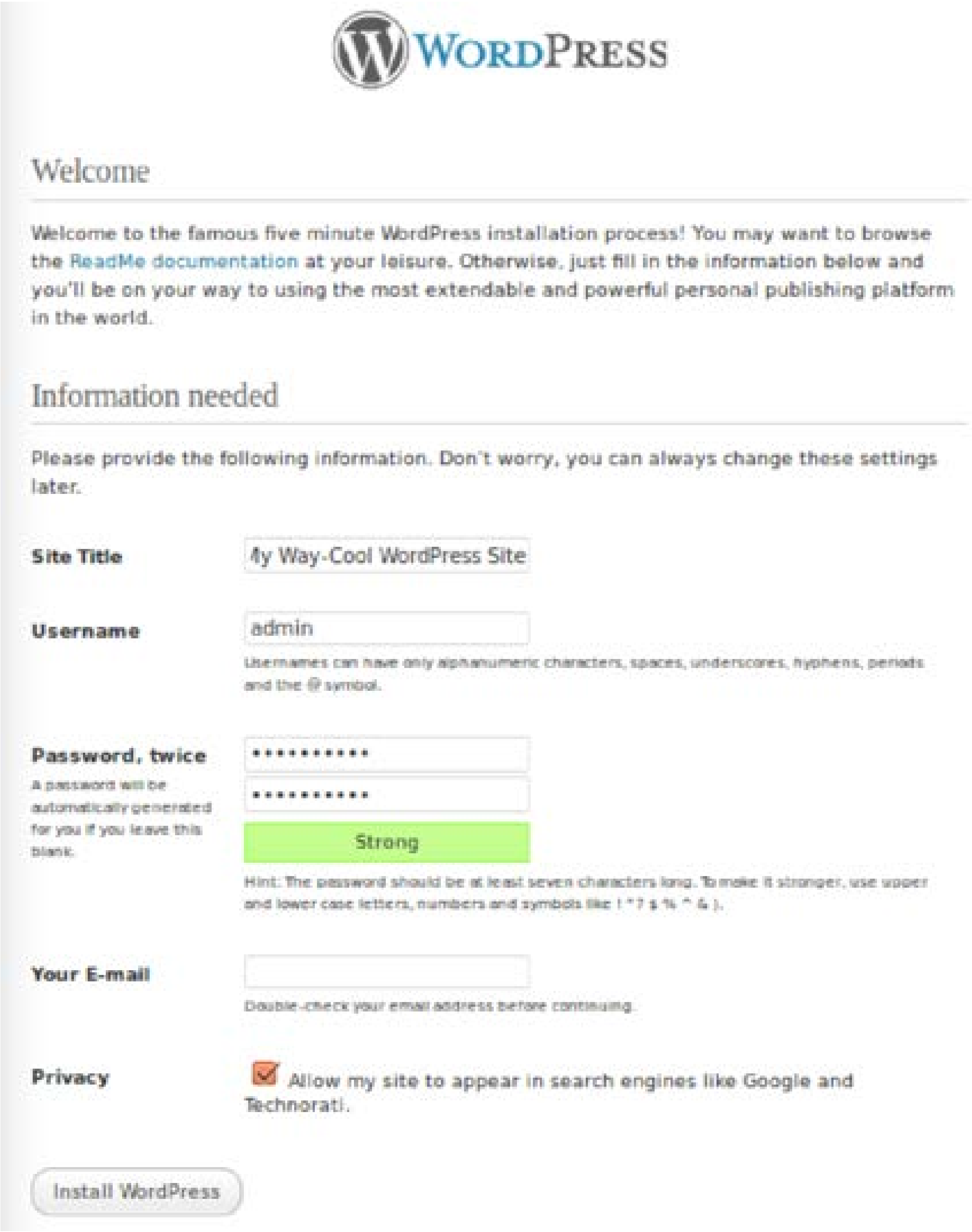
Often, this database has already been created. Not to worry—WordPress can use that single database. You will find during the cPanel WordPress installation that you are prompted to enter a WordPress table name prefix, the default being **wp_**. WordPress will create all the tables it needs within the single database that your hosting service has provided. If this is your situation, the following steps can be avoided. If you are unsure, go ahead and attempt the database creation steps. At some point it will become clear whether or not you can create a new database and if one already exists.

1. Log in to your *cPanel* user interface.
2. In the Databases section **MySQL Database Wizard**. Note: Some hosting providers allow only one database per user.
   1. **Create a Database**. You'll need to choose a database name. "WP" works, if you're not feeling creative. Your hosting provider may allow multiple databases. If this is the case, others can be created later. For now we are simply creating a database for use by WordPress.
   2. **Create Database Users.**
3. **Add User to Database.** Select the **All Privileges** checkbox.
4. **Complete. Take** note the database name and user—you do not want to forget these! At this point, it is wise to record the *hostname*, *username*, *databasename*, and the password you chose. In general, the hostname is localhost, as this is the host

that MySQL connects to, and it is very likely that the database that was created is on the same machine. This is not necessarily true, so take note.

Permissions correct, database created, admininstrator username and password set, we're ready to get into WordPress and configure it.

Now that WordPress, MySQL, PHP, and Apache are installed and configured, try hitting *http://localhost/wordpress* again.



Welcome

Welcome to the famous five minute WordPress installation process! You may want to browse the ReadMe documentation at your leisure. Otherwise, just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Don't worry, you can always change these settings later.

Site Title: 1y Way-Cool WordPress Site

Username: admin

Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods and the @ symbol.

Password, twice: •••••••••• / ••••••••••

A password will be automatically generated for you if you leave this blank.

Strong

Hint: The password should be at least seven characters long. To make it stronger, use upper and lower case letters, numbers and symbols like ! " ? $ % ^ &).

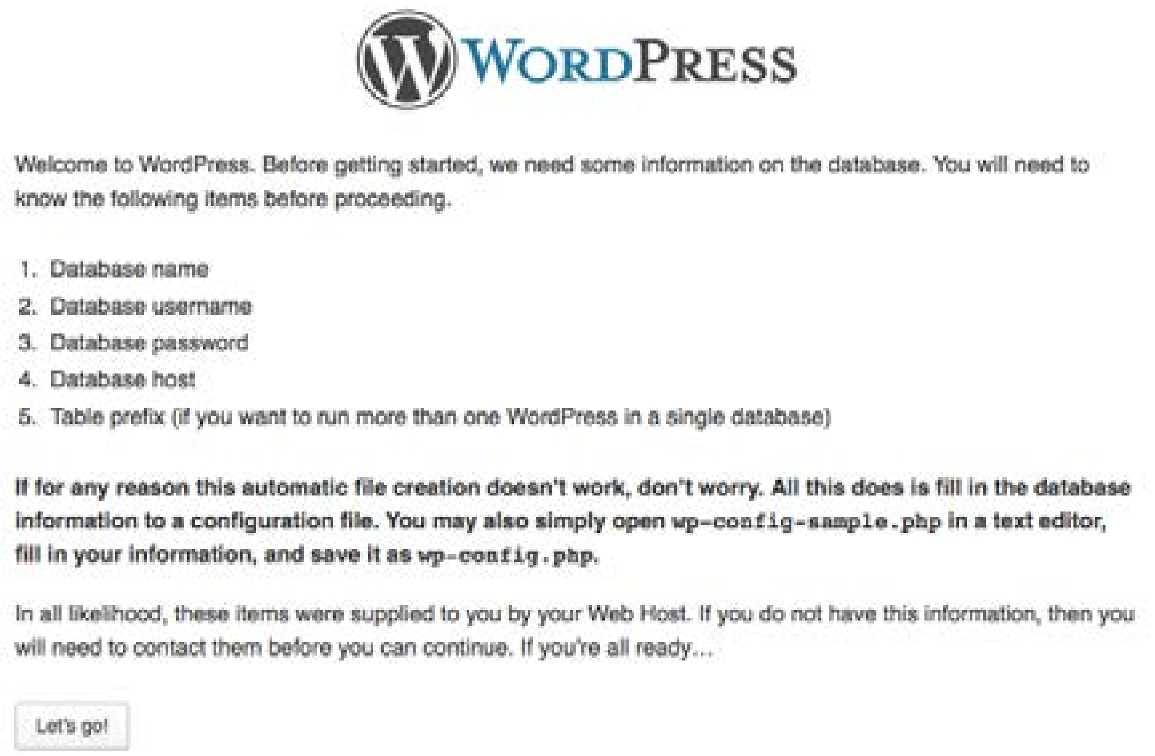
Your E-mail:

Double-check your email address before continuing.

Privacy: ☑ Allow my site to appear in search engines like Google and Technorati.

Initial WordPress Configuration

The first time WordPress is loaded it will recognize that it has not yet been configured. In your favorite web browser, go to *http://localhost/wp* (or http://localhost/, depending on where you installed). You should see a page like this:



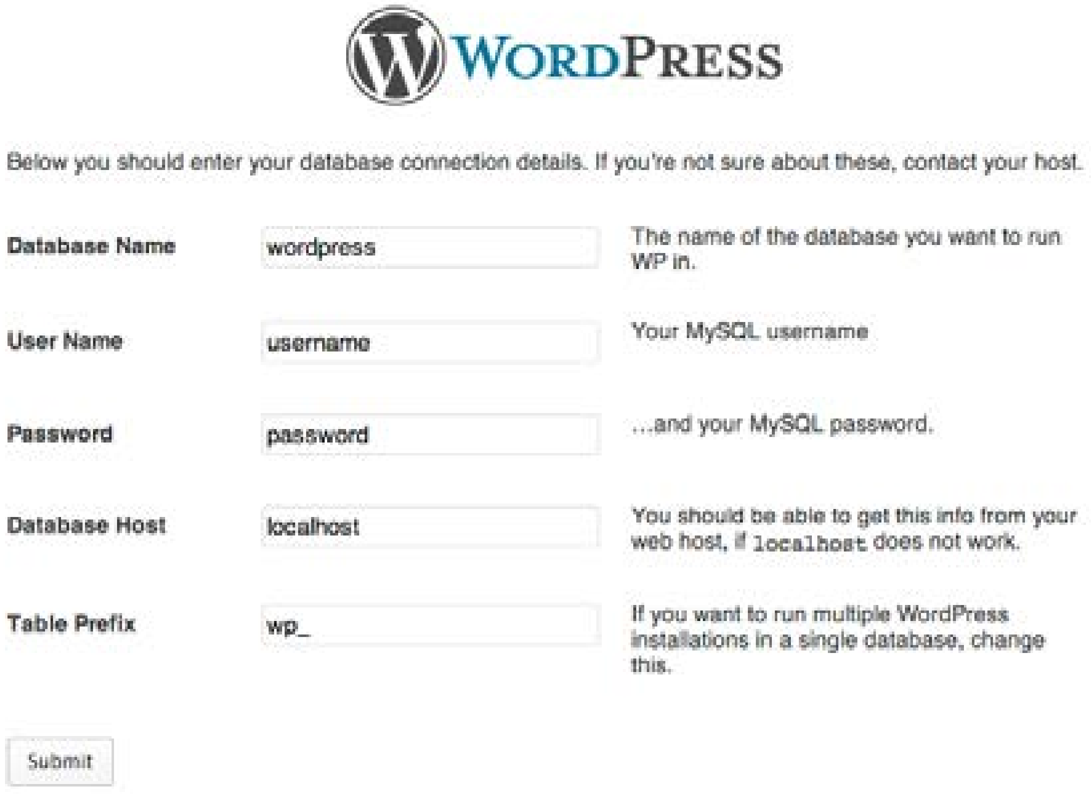
Welcome to WordPress. Before getting started, we need some information on the database. You will need to know the following items before proceeding.

1. Database name
2. Database username
3. Database password
4. Database host
5. Table prefix (if you want to run more than one WordPress in a single database)

If for any reason this automatic file creation doesn't work, don't worry. All this does is fill in the database information to a configuration file. You may also simply open `wp-config-sample.php` in a text editor, fill in your information, and save it as `wp-config.php`.

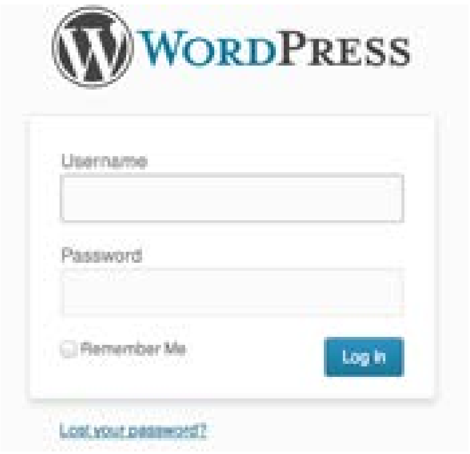
In all likelihood, these items were supplied to you by your Web Host. If you do not have this information, then you will need to contact them before you can continue. If you're all ready…

Click on that "Let's Go" button to get to the following page. This is where WordPress prompts for database location.



Below you should enter your database connection details. If you're not sure about these, contact your host.

Database Name	wordpress	The name of the database you want to run WP in.
User Name	username	Your MySQL username
Password	password	...and your MySQL password.
Database Host	localhost	You should be able to get this info from your web host, if `localhost` does not work.
Table Prefix	wp_	If you want to run multiple WordPress installations in a single database, change this.



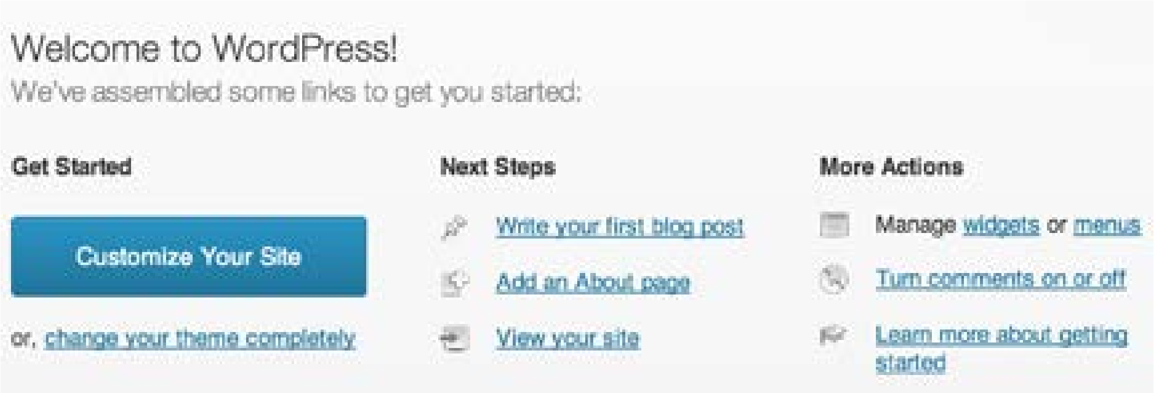
We previously created a database named **wordpress**, and granted all privileges to the user **wp**. The database host, as far as WordPress is concerned, is localhost. It is possible that MySQL is hosted elsewhere, but in our case we are hosting our web content, WordPress, and database all on the save server. (If **localhost** does not work for you, try **127.0.0.1**.)

After submitting these initial details it's time to login to your administrative pages (http://<my-site>/<sub-directory>/wp-admin).

WordPress Codex

WordPress's central documentation system, the **WordPress Codex** can be found at *http://codex.wordpress.org*. If searching for help on a WordPress subject, it's likely that you'll end up here. The documentation and help system found here provides details on nearly all of the features, plugins, and customization. The Codex is where advanced usage details, for those who wish to dive in and take things to another level, are found.

Further WordPress Configuration



Welcome to WordPress!

We've assembled some links to get you started:

Get Started	Next Steps	More Actions
Customize Your Site	Write your first blog post	Manage widgets or menus
	Add an About page	Turn comments on or off
or, change your theme completely	View your site	Learn more about getting started

We've installed WordPress. Piece of cake! But for users accustomed to WordPress hosted blogs, it is necessary to highlight some of the differences. The clean WordPress installation we have just performed doesn't have all the plugins that are immediately available on a hosted WordPress site. When using a hosted WordPress site, features such as spam block, social network sharing, statistics, and subscriptions are already present. For those of us installing WordPress from scratch, these plugins need to be installed. Additionally, hosted WordPress sites present the user with a set up templates to choose from. This may seem handy, but plugins and template availability are limited on *WordPress.com* hosted sites.

Per your web server setup, you blog is available at one of the two following addresses:

http(s)://my-blog/
http(s)://my-blog/<sub-path>

And the administration user interface is available at one of the two addresses (again, depending on your web server setup):

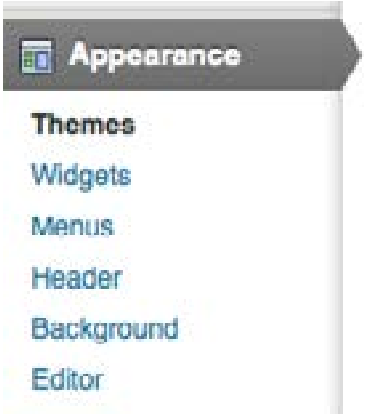
http(s)://my-blog/wp-admin
http(s)://my-blog/<sub-path>/wp-admin

Themes

Appearance ->Themes

Thousands of themes exist, and new ones are being created daily. The pimary source of themes (as well as plugins) is the WordPress Codex (this is where WordPress locates themes that are displayed while browsing within the admin pages).

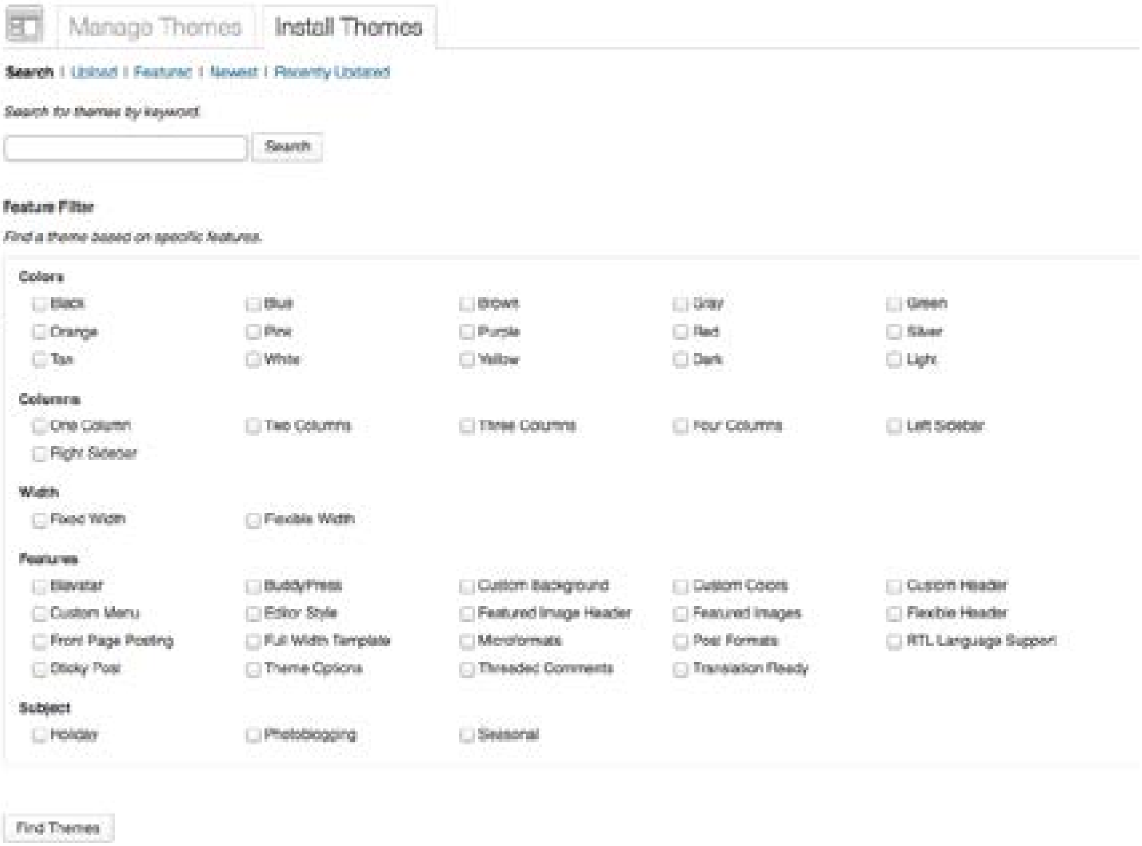
Themes are installed and managed in the administration interface at **Appearance-Themes**. There are two tabs available at the top of the Themes admin screen: **Manage Themes** and **Install Themes**. Manage Themes is where you'll go to view all the themes you've installed. From this tab you can preview and set themes as active. But we're not ready for this—not yet.



The second tab, **Install Themes**, is the first step. WordPress has shipped with a default theme, but let's go ahead and select something different.

Selection of a theme for installation is pretty simple. On the Install Themes page, you will notice that there are a number of selection criteria options to help whittle down the search results when browsing a theme. Which criteria you select is largely dependent on the objectives and personal preferences you have in mind for your site. While theme colors can be selected from here, my suggestion is that you don't filter by color. Remember, colors can very easily be changed. Some of the other features, such as RTL support, number of columns, custom menus, and sticky posts are not as easy to modify. So choose a theme that, at a high level, meets your basic needs.

You can also search by keyword here, but you may find that this limits the results more than you may wish. On the other hand, it might be very useful. For example, if this site is intended as a showcase of your photography, go ahead and search using the keyword "photo" to see a number of themes that are tailored for just such a thing.



Manage Themes | Install Themes

Search | Upload | Featured | Newest | Recently Updated

Search for themes by keyword.

Search

Feature Filter

Find a theme based on specific features.

Colors

□ Black □ Blue □ Brown □ Gray □ Green
□ Orange □ Pink □ Purple □ Red □ Silver
□ Tan □ White □ Yellow □ Dark □ Light

Columns

□ One Column □ Two Columns □ Three Columns □ Four Columns □ Left Sidebar
□ Right Sidebar

Width

□ Fixed Width □ Flexible Width

Features

□ Blavatar □ BuddyPress □ Custom Background □ Custom Colors □ Custom Header
□ Custom Menu □ Editor Style □ Featured Image Header □ Featured Images □ Flexible Header
□ Front Page Posting □ Full Width Template □ Microformats □ Post Formats □ RTL Language Support
□ Sticky Post □ Theme Options □ Threaded Comments □ Translation Ready

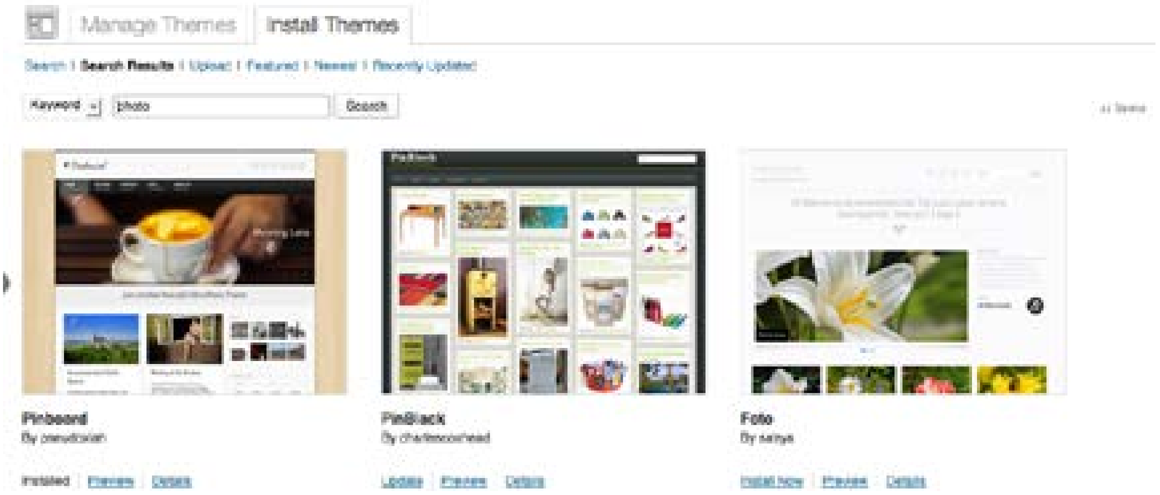
Subject

□ Holiday □ Photoblogging □ Seasonal

Find Themes

Below each theme thumbnail there are a few links:

- **Details:** See a description of the theme.
- **Preview:** See an example of what the installed theme might look like. This is not an Active Theme. Rather, this link displays a sample page (as screen shots) using the theme.
- **Install Now:** This link is displayed if the link has not yet been downloaded ("installed"). Download the theme to your local themes directory. This makes the theme available for activation.
- **Update:** The update link is available only if the theme has already been installed and a new version of the theme is available. If you update a theme that is currently the active theme of your site, the changes will take place immediately. Some of these changes may be significant. See the theme details or the theme creator's site for details.
- **Live Preview:** This link is available only if the theme has been installed. Clicking opens a preview window showing what your site will look like if the theme is activated (extremely useful).
- **Activate:** This link is displayed only after the theme has been Installed. This link applies the theme to your pages, posts, and overall appearance.


Manage Themes
Install Themes
Search | Search Results | Upload | Featured | Newest | Recently Updated
Keyword
photo
Search
Pinboard
By pseudoxian
Installed | Preview | Details
PinBlack
By charlescourhead
Update | Preview | Details
Foto
By satya
Install Now | Preview | Details

While there are thousands of themes available for download from various locations (some cost, many do not) WordPress has made it very easy to find almost any them you would ever want by hosting many of them. Themes available from WordPress can easily be downloaded through the administrator interface. Given the volume to choose from make use of the theme search filters that are available:

- **Search:** Search by keyword and feature.
- **Popular:** The link pretty much says it all: The most popular themes used. Be aware, sometimes it's nice to stand out in the crowd.
- **Trending:** Search for themes that are increasing in popularity. This is useful because popular themes may be popular only because of wide use, and not because they are currently among the best available.
- **Featured:** Browse a collection of WordPress selected, quality themes.
- **New and Recently Updated**: Browse the most recently created and updated themes. Many themes have been around for a while, but because of continuous improvements (such as the inclusion of responsive design), such themes can still appear appealing to a current web audience.

Should you decided on an initial theme that is not hosted by WordPress, it will need to be installed by uploading a .zip or .*tar.gz* theme file. This is done in the admin UI at Appearance->Themes->Install Themes->Upload. To start, let's just choose a theme that is available through WordPress. For those wanting to use a theme found elsewhere, this can be done after the blog/site is all set up.

Themes come in many varieties. Some are very basic, others extremely complex. Some themes provide a very general look and feel, while others are heavily tailored to a specific type of website. If you're a photo blogger, small storeowner, contractor, and just a simple blogger, start by looking for themes that best fit your desired use.

Before choosing a theme, it might be best to create a few dummy posts. This will help as you preview the appearance. While it can be fun to browse themes and see what your blog/site will look like, it can also be a distraction from completely the setup. Since this site is going to be self-hosted, remember that your theme will be fully customizable. So choose a theme that is close to what you like for now.

Definition

Installed Theme: *A theme that has been downloaded and unpacked in the WordPress wp-content/themes directory. Most themes include around 20 CSS and PHP files.*

(*http://wordpress.org/themes/*). Installing and setting a theme is a two-step process: Installing the theme(s) and activating a theme.

Definition

Active Theme: *A theme that has been downloaded and unpacked in the WordPress wp-content/themes directory and is selected as the current active theme.*

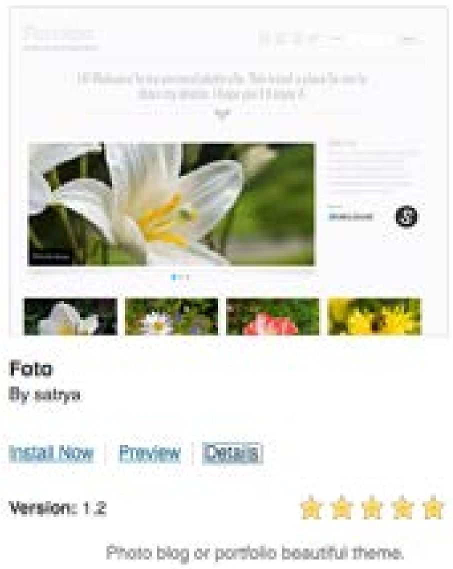


Foto
By satrya

Version: 1.2

Photo blog or portfolio beautiful theme.

The first step is installing a theme. Note that an Installed Theme is not necessarily an active theme. The process of installing a theme in WordPress sets that theme as one that is available to your website. Only after a theme has been installed can it be viewed (Active Preview) or set as the active theme. While browsing themes to install, you can view the current version, community rating, and details. Some themes have great details highlighting theme features and purpose. Other themes are a little light on the details, so it may be necessary to install them theme and take a look.

Installed themes appear in the wp-content/themes. After installing a few, take a look there to see what has changed. You may see something like this:

```
twentythirteen
twentytwelve
my-custom-theme
/var/www/wp/wp-content/themes $
```

Looking more closely (cd into a theme directory), see the files that are installed for the theme:

```
/css                          functions.php
/languages                    footer.php
/js                           editor-style.css
/inc                          editor-style-rtl.css
tag.php                       content.php
style.css                     content-status.php
single.php                    content-quote.php
sidebar.php                   content-page.php
sidebar-front.php             content-none.php
search.php                    content-link.php
screenshot.png                content-image.php
rtl.css                       content-aside.php
page.php                      comments.php
page-templates                category.php
index.php                     author.php
image.php                     archive.php
header.php                    404.php
```

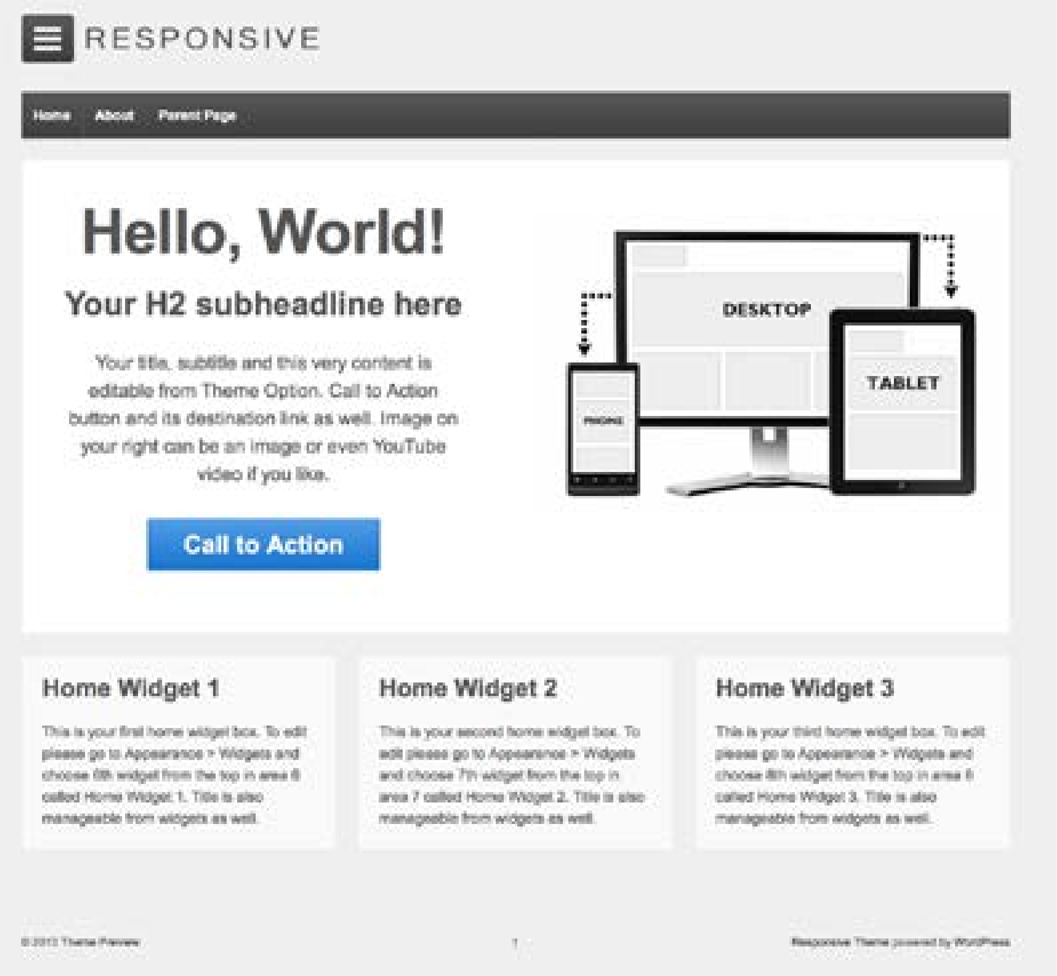
An experienced web developer may surmise (correctly) that the above files contain Multilanguage support, JavaScript, stylesheets and PHP specific to both end user/viewer experience as well as administrative pages. A theme may include certain bells and whistles that extend the administrator's ability to modify certain elements.

While browsing themes to install, you will notice a Preview link for each one. This preview will not show you what your site will look like with the theme (not exactly). This is simply a link to a sample of the theme. Before viewing an **Active Preview**, the theme must be installed.

Definition

Active Preview: *A preview window which is shows the blog/site with using an installed theme, pulling all posts, settings, and active plugins so as to display the appearance should the user activate the theme.*

Responsive Themes



Hello, World!

Your H2 subheadline here

Your title, subtitle and this very content is editable from Theme Option. Call to Action button and its destination link as well. Image on your right can be an image or even YouTube video if you like.

Home Widget 1

This is your first home widget box. To edit please go to Appearance > Widgets and choose 6th widget from the top in area 6 called Home Widget 1. Title is also manageable from widgets as well.

Home Widget 2

This is your second home widget box. To edit please go to Appearance > Widgets and choose 7th widget from the top in area 7 called Home Widget 2. Title is also manageable from widgets as well.

Home Widget 3

This is your third home widget box. To edit please go to Appearance > Widgets and choose 8th widget from the top in area 8 called Home Widget 3. Title is also manageable from widgets as well.

© 2013 Theme Preview 1 Responsive Theme powered by WordPress

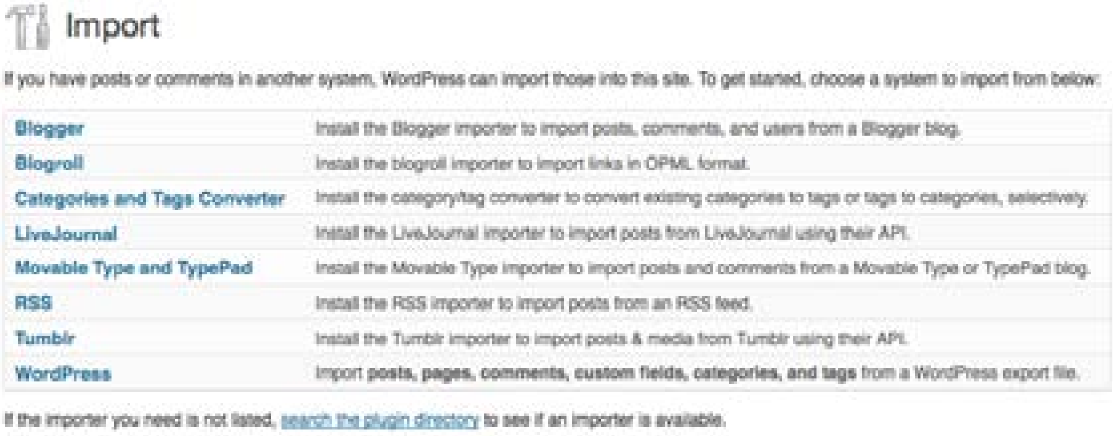
You will likely notice a number of themes that describe themselves as @media "responsive." Essentially, this just means that the theme uses CSS3 media queries to adjust the layout, size, and/or orientation of page for best display on a particular device (including hardcopy print, tablets, smartphones and, naturally, plain old computer screens). Many responsive themes include hideaway sidebars and/or horizontally scrollable content.

For some interesting details on the history behind responsive designs, see (*http://www.alistapart.com/articles/responsive-web-design*).

Importing an Existing Blog

Tools->Import

Import functionality is available as a plugin, but I mention it here because it may be an important early step of your configuration.



Import

If you have posts or comments in another system, WordPress can import those into this site. To get started, choose a system to import from below:

Blogger	Install the Blogger importer to import posts, comments, and users from a Blogger blog.
Blogroll	Install the blogroll importer to import links in OPML format.
Categories and Tags Converter	Install the category/tag converter to convert existing categories to tags or tags to categories, selectively.
LiveJournal	Install the LiveJournal importer to import posts from LiveJournal using their API.
Movable Type and TypePad	Install the Movable Type importer to import posts and comments from a Movable Type or TypePad blog.
RSS	Install the RSS importer to import posts from an RSS feed.
Tumblr	Install the Tumblr importer to import posts & media from Tumblr using their API.
WordPress	Import **posts, pages, comments, custom fields, categories, and tags** from a WordPress export file.

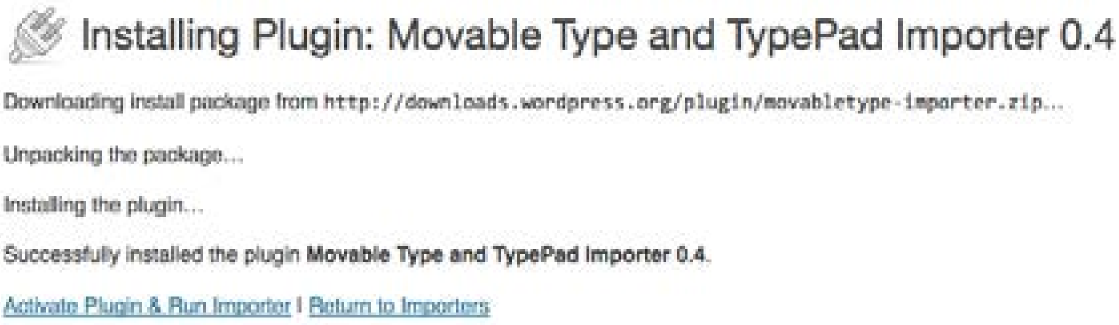
If the importer you need is not listed, search the plugin directory to see if an importer is available.

In the likely case that your new self-hosted WordPress site is intended to replace an existing blog, WordPress offers an import feature that works very well. For details on all of the various import possibilities, see *http://codex.wordpress.org/Importing_Content*.

As might be expected, importing from an existing WordPress site to a self-hosted site is the most straightforward process. This is often done when users import from a *WordPress.com* hosted site to a self-hosted WordPress site. In most scenarios, the import process is as follows:

1. Log into your old blog as administrator.
2. Export the blog. If your previous blog allows export as XML, you will want to do so.
3. Log into your new hosted WordPress site.
4. Navigate to Manage->Import in admin panels.
5. Choose the appropriate Import type the list. From here, you will be prompted to install the importer plugin for the type of site you are importing from. Install the plugin (click the Install Now button).
6. Upon successful installation of the plugin, you will see two links, ***Activate Plugin & Run Importer***, ***and Return to Importers***. Assuming you are now ready to import, click the **Activate Plugin & Run Importer** link.
7. In the final step, decisions will have to be made regarding the type of import. If importing from an existing WordPress site, the XML file created in step 2 must be selected and imported. For other blogs the import process may use the blog's RSS feed to auto generate posts on your new site.
8. In some import types, you may be prompted to map authors and categories.

If importing from a non-WordPress site, there is likely to be some cleanup work required following the process. Depending on the type of site that is being imported from, the WordPress import plugin may not recognize authors, categories, tags, dates, or comments. Be aware that, while the WordPress import scripts often do import comments, in some cases it is not possible. Comments on old posts may not be imported to the new site during this process—it all depends on the import type.



Installing Plugin: Movable Type and TypePad Importer 0.4

Downloading install package from `http://downloads.wordpress.org/plugin/movabletype-importer.zip`…

Unpacking the package…

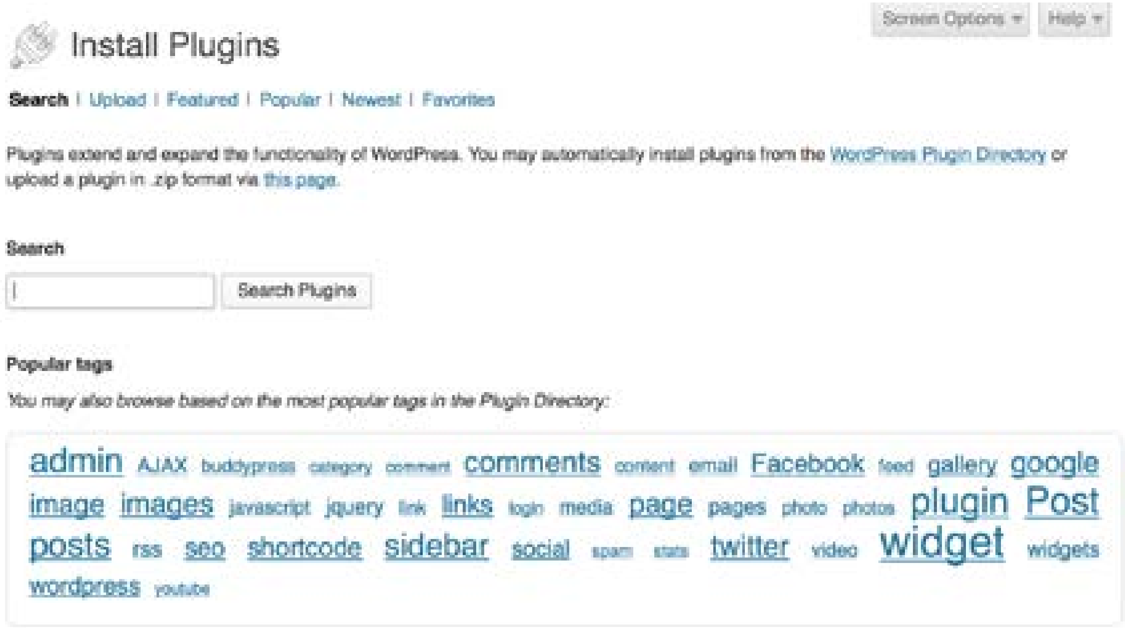
Installing the plugin…

Successfully installed the plugin **Movable Type and TypePad Importer 0.4**.

Activate Plugin & Run Importer | Return to Importers

Plugins

Plugins->**Installed Plugins**
Plugins->**Add New**
Plugins->**Edit**



Install Plugins

Plugins extend and expand the functionality of WordPress. You may automatically install plugins from the WordPress Plugin Directory or upload a plugin in .zip format via this page.

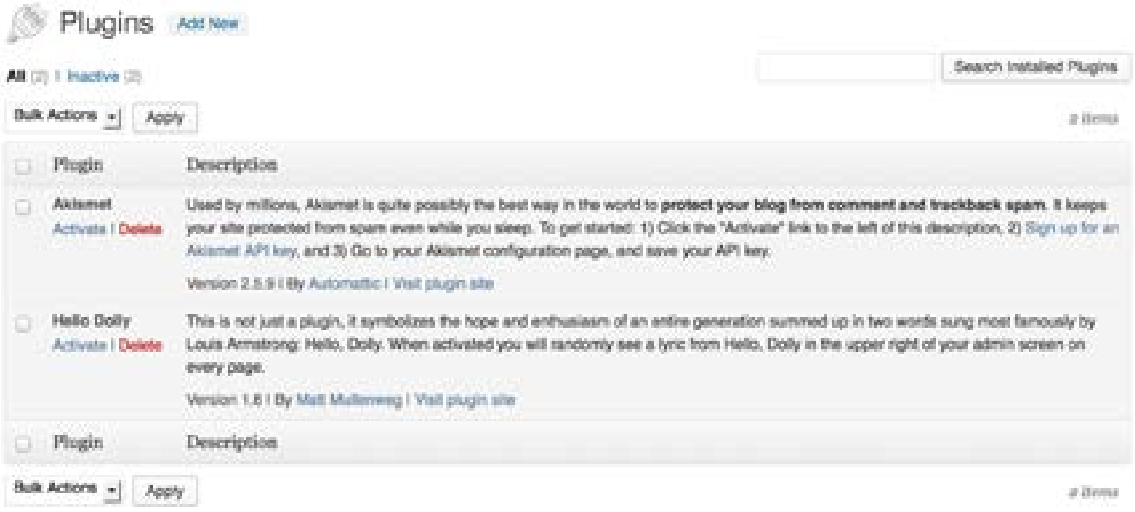
Search

Popular tags

You may also browse based on the most popular tags in the Plugin Directory:

admin AJAX buddypress category comment comments content email Facebook feed gallery google image images javascript jquery link links login media page pages photo photos plugin Post posts rss seo shortcode sidebar social spam stats twitter video widget widgets wordpress youtube

As with themes, plugins are available directly through WordPress or via third party downloads. Always beware the source of a download, as, albeit uncommon, a plugin from an unknown provider could introduce a virus. Many plugins can be installed, but no plugin does anything until activated on the activation and configuration is completed within the Plugins administration page. There are certain plugins that you may find absolutely critical to have. These include plugins for post editing, SEO statistics, social network sharing, and creating short links to posts. Others may add a certain amount of whiz-bang flare to your site.



Plugins Add New

All (2) | Inactive (2)

Search Installed Plugins

Bulk Actions Apply 2 items

□	Plugin	Description
□	Akismet Activate | Delete	Used by millions, Akismet is quite possibly the best way in the world to **protect your blog from comment and trackback spam**. It keeps your site protected from spam even while you sleep. To get started: 1) Click the "Activate" link to the left of this description, 2) Sign up for an Akismet API key, and 3) Go to your Akismet configuration page, and save your API key. Version 2.5.9 | By Automattic | Visit plugin site
□	Hello Dolly Activate | Delete	This is not just a plugin, it symbolizes the hope and enthusiasm of an entire generation summed up in two words sung most famously by Louis Armstrong: Hello, Dolly. When activated you will randomly see a lyric from Hello, Dolly in the upper right of your admin screen on every page. Version 1.6 | By Matt Mullenweg | Visit plugin site
□	Plugin	Description

Bulk Actions Apply 2 items

Definition

Search engine optimization (SEO) *is the process of improving the visibility of a website or a web page in a search engine's "natural" or un-paid ("organic") search results. In general, the earlier (or higher ranked on the search results page), and more frequently a site appears in the search results list, the more visitors it will receive from the search engine's users. SEO may target different kinds of search, including image search, local search, video search, academic search,[1] news search and industry-specific vertical search engines.*

Wikipedia.org**,** https://en.wikipedia.org/wiki/Search_engine_optimization

Plugins are PHP extensions to existing WordPress features. From appearance improvements to SEO statistics to social networking, there are plugins for nearly everything imaginable. This is a very good thing… While you may be an experienced PHP developer, it is unlikely that you will find yourself needing to develop a custom plugin. There are plugins that are nice to have, and there are plugins that are necessary. With thousands available, I'll cover only a few of the most popular ones here. A plugin can be downloaded to your local computer from anywhere, and then uploaded to your WordPress site, but it is certainly much easier to

Step One: Locating a Plugin

WordPress provides a wealth of plugins, including many created by WordPress, and many more from 3rd party individuals and companies. Beware—A plugin may advertise itself as being free, but once installed, you may find that some of them push you to purchase additional features. I find that it is easier to remove that plugin and search for another option, with a few caveats (read on).

Behind the scenes, the plugin installation process is straightforward. Plugin files (PHP) are placed in the WordPress plugin directory (/wp-content/plugins). This directory is maintained by WordPress, so there is no need to maintain it outside of the admin pages. Plugins are packaged as .zip files. When a plugin is installed, the content is extracted into the /plugins directory.

If you are running on a server that does not have an FTP host installed, it will be necessary to download the plugin to a local path and then extract it to wp/wp-content/plugins. A much easier approach is to simply run an FTP server locally. This is because WordPress uses FTP or SFTP to transfer the plugin.

To start FTP on MacOS:

sudo -s launchctl load -w /System/Library/LaunchDaemons/*ftp.plist*

To start FTP on Linux:

```
    /sbin/service vsftpd start
```

Step 2a: Adding a Plugin Through WordPress's Provided List

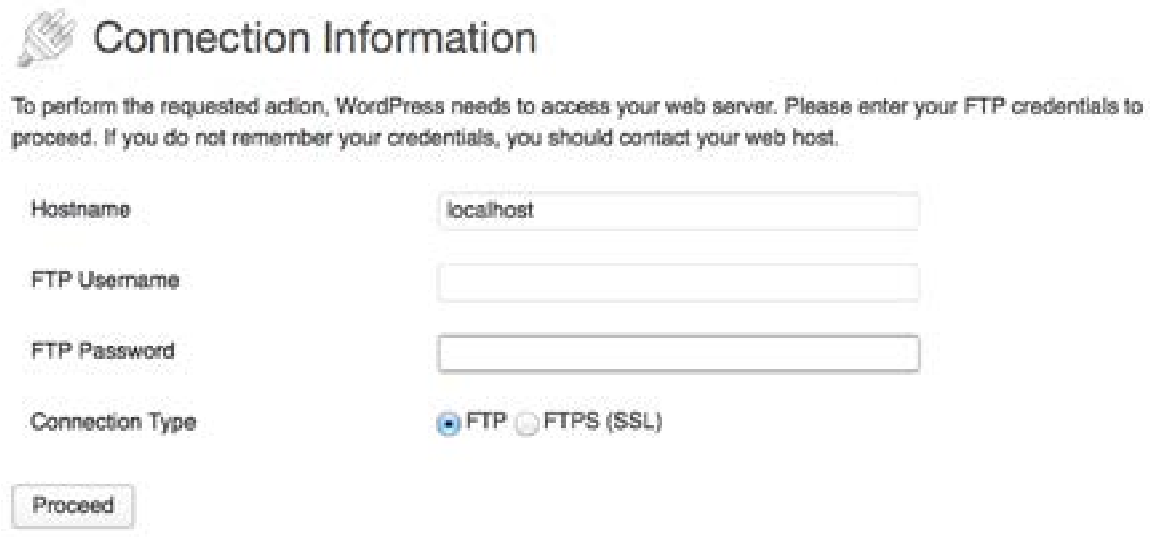
To install a plug-in, choose Plugins->Add New, which takes you to the Install Plugins page. You'll see five links at the top of the page that let you decide how to find the plug-in you want. Plugin's, unlike themes, do not provide active or inactive previews, so read all the details and understand what you are getting before you install. The plugin installation is pretty quick, but you will not notice any differences until the plugin is activated.

Step 2b: Adding a Third-Party or Custom Plugin by Upload

Once installed, a plugin still needs to be activated before it does anything, and plugins are easily uninstalled. In general, it is unlikely that you will damage your site by installing a plugin, but this is not always the case. Certain plugins permanently change database values For example, there are some that change URLs to posts, shortening them. Such plugins may be highly useful, but take heed: One such a plugin is installed it cannot be removed without breaking your site.

Step 3: Activating a Plugin

Activating a plugin is really a two-step process. Although many plugins require no customization, others do. Sometimes the configuration steps are clear. If they are not, try the plugin developer page for more details.



Connection Information

To perform the requested action, WordPress needs to access your web server. Please enter your FTP credentials to proceed. If you do not remember your credentials, you should contact your web host.

Hostname	localhost
FTP Username	
FTP Password	
Connection Type	● FTP ○ FTPS (SSL)

Installation of a plugin is similar to installation of a template in that it is a (simple) two-step process. The first step is uploading the template (which can be done through the WordPress administration interface). The second step is activating and configuring the template (some templates require no particular configuration, but many do). Some templates add entirely new links of the left side of the administration page. These are very convenient in many cases, know that your administrative experience may vary—it all depends on the plugin, and it might not be immediately clear where the changes took place. The best plugins provide details on such usage.

Plugins are not always consistent with regard to their setting modifications. Some templates, for example those that enhance post creation and editing features, are found in Posts section of the Administration pages. Others are seen in Appearance/Widgets.

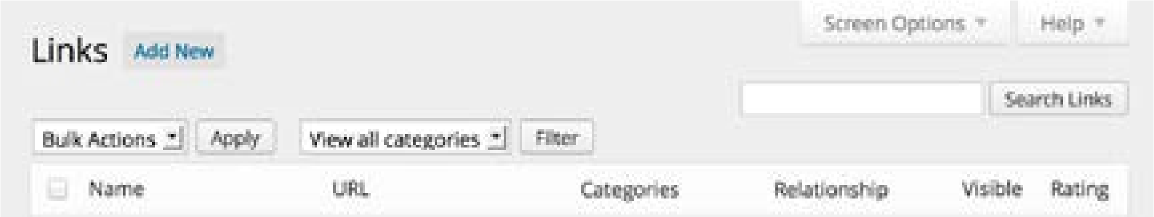
Important Plugins

Believe it or not, WordPress does not come with built-in search functionality (the same goes for custom links and permalinks). There exists a plethora of search plugins (many of them in the form of widgets) the allow you to place a search box on the side of posts, pages or both. Whatever the nature of your site, it is generally good to provide readers with the ability to search content. Find a plugin that allows you to limit search criteria. You may not want the search plugin to search comments or certain pages (or posts beyond a certain date).

Important Plugins: Links

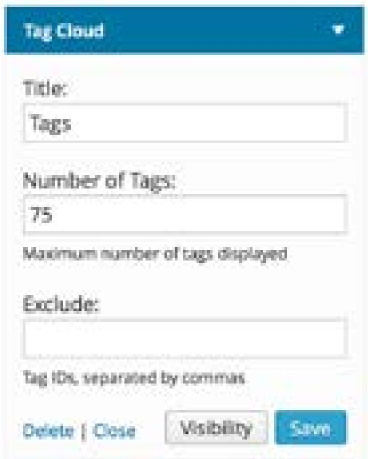
Links->**All Links**
Links->**Add New**
Links->**Link Categories**

The link features are not a part of the default *WordPress.org* installation. For those familiar with *WordPress.com* hosting, this is one of many differences in the basic setup. The official WordPress plugin for creating and managing links is Links Manager (see *http://wordpress.org/plugins/link-manager/*).



For a blogger, community site, or even a business site, it's always good to share links to friends, organizations of common interests, and partner. WordPress allows the creation of custom links, customizable by category. This isn't just a way to be a good web citizen—it may also help to drive traffic to your site.

Additionally, creative use of links can be used to link to content within your own site categories, and included as menus on a page or sidebar as a widget. Other link plugins include menu, category, tag and social media sharing plugins. Such plugins can make your site easy to navigate for a user and easily indexed by search engines.



Title: Tags

Number of Tags: 75

Maximum number of tags displayed

Exclude:

Tag IDs, separated by commas

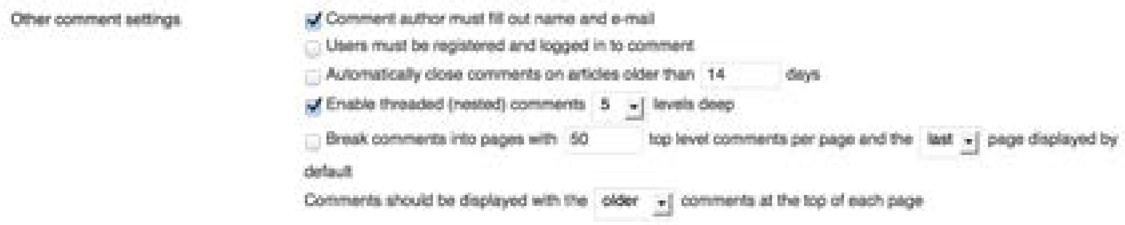
Important Plugins: Fighting Spam

It may seem odd that the subject of spam appears in the plugins section, but plugins are the best way to battle the problem.

It's inevitable: If you open your site up to comments, you will get spam. There are ways to battle this, the simplest of which is by configuring your site to limit comments only to logged in users (those with WordPress accounts). This is limiting if you wish to allow for anyone to comment, and it doesn't ensure that the logged in user isn't a spammer.

You can also lock comments down so that they do not appear until you have approved them. Comments are "held for moderation." This is certainly something worth considering if you run a business. The options here are self-explanatory.

Comment settings can be configured at **Settings**->**Discussion** settings.



Other comment settings

☑ Comment author must fill out name and e-mail

□ Users must be registered and logged in to comment

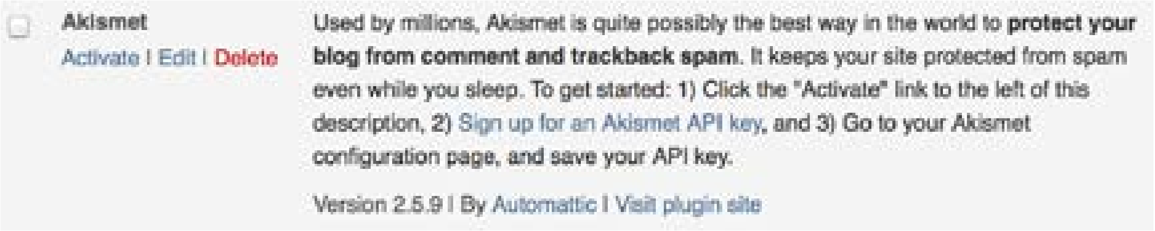
□ Automatically close comments on articles older than 14 days

☑ Enable threaded (nested) comments 5 levels deep

□ Break comments into pages with 50 top level comments per page and the last page displayed by default

Comments should be displayed with the older comments at the top of each page

To automate much of the pain associated with spam prevention, the most popular plugin is Aksimet:



□ Akismet

Activate | Edit | Delete

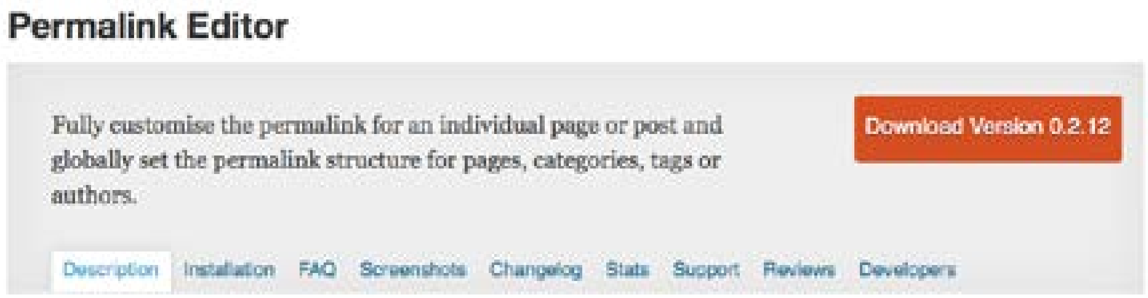
Used by millions, Akismet is quite possibly the best way in the world to **protect your blog from comment and trackback spam**. It keeps your site protected from spam even while you sleep. To get started: 1) Click the "Activate" link to the left of this description, 2) Sign up for an Akismet API key, and 3) Go to your Akismet configuration page, and save your API key.

Version 2.5.9 | By Automattic | Visit plugin site

While this plug requires a certain amount of configuration, including registration Aksimet (you will need to create an Aksimet key), its effectiveness against spam is highly effective. For personal use an Aksimet key is free, and for business use the cost is minimal, and well worth it if you wish to prevent potentially embarrassing or unprofessional spam from appearing on your site.

Permalinks

Settings->**Permalinks**



Permalink Editor

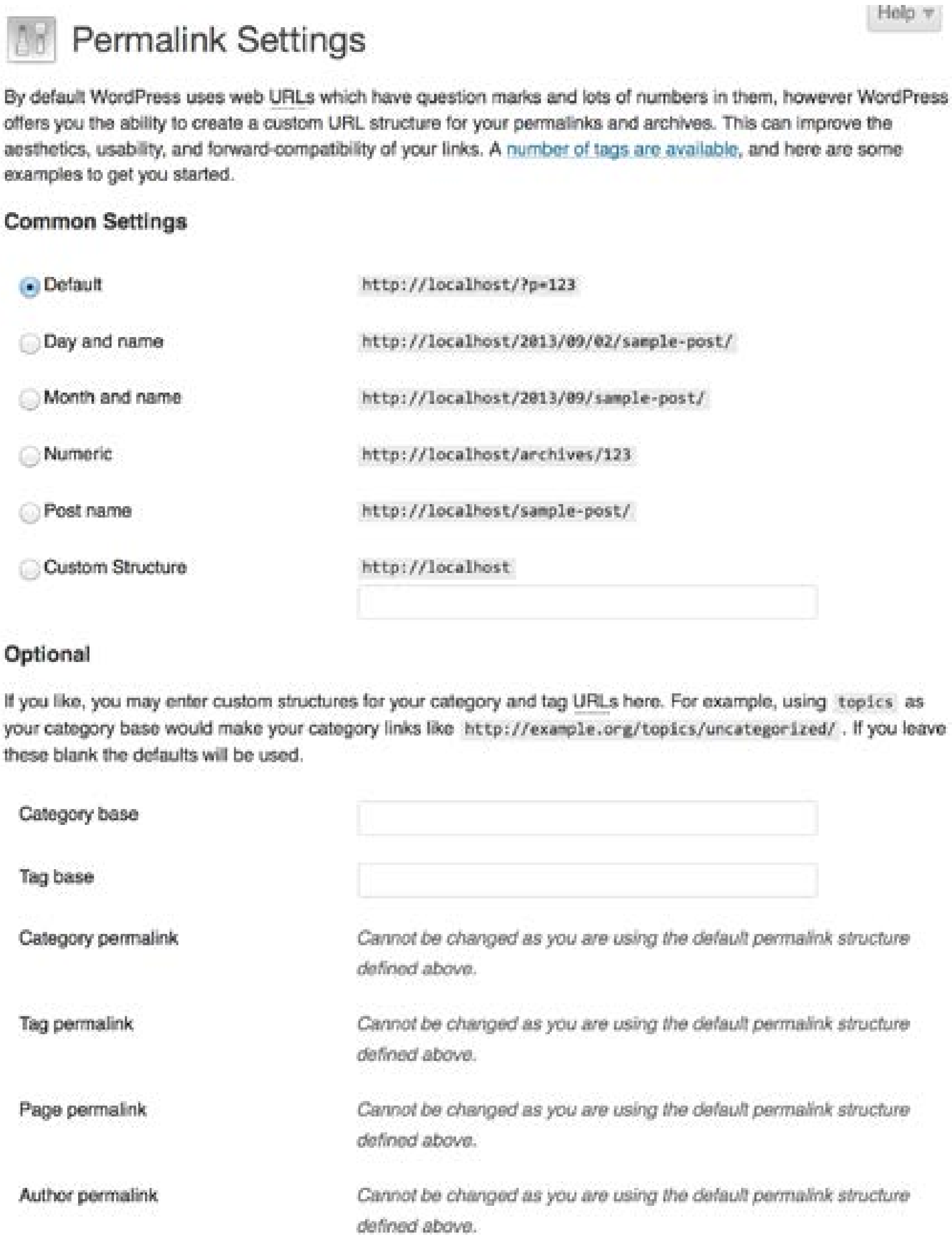
Fully customise the permalink for an individual page or post and globally set the permalink structure for pages, categories, tags or authors.

Without permalinks, you'll notice that page and post URLs contain GET data on the address line. For example, your very first post may look like this:

http://localhost/?p=1

There are other permalink plugin options, but the most common one (and the official WordPress plugin) is Permalink Editor (see *http://wordpress.org/plugins/permalink-editor/*).

After installing the permalink plugin, activated it at Plugins->Permalink Editor. The change won't be immediate obvious, as it must first be configured at Settings->Permalinks. This plugin is self-explanatory, with a number of permalink options being presented (see below).



Permalink Settings

By default WordPress uses web URLs which have question marks and lots of numbers in them, however WordPress offers you the ability to create a custom URL structure for your permalinks and archives. This can improve the aesthetics, usability, and forward-compatibility of your links. A number of tags are available, and here are some examples to get you started.

Common Settings

● Default	`http://localhost/?p=123`
○ Day and name	`http://localhost/2013/09/02/sample-post/`
○ Month and name	`http://localhost/2013/09/sample-post/`
○ Numeric	`http://localhost/archives/123`
○ Post name	`http://localhost/sample-post/`
○ Custom Structure	`http://localhost`

Optional

If you like, you may enter custom structures for your category and tag URLs here. For example, using `topics` as your category base would make your category links like `http://example.org/topics/uncategorized/` . If you leave these blank the defaults will be used.

Category base	
Tag base	
Category permalink	*Cannot be changed as you are using the default permalink structure defined above.*
Tag permalink	*Cannot be changed as you are using the default permalink structure defined above.*
Page permalink	*Cannot be changed as you are using the default permalink structure defined above.*
Author permalink	*Cannot be changed as you are using the default permalink structure defined above.*

My personal preference is to use the simple Post Name option. Combined with the option Category and Tag base names, each post and page can be assigned a unique pathname, generally one that is short enough to easily share. For a blog, you may wish to consider including months and or days in the post URLs.

Don't Overdo It With Plugins

As great as plugins can be, there are downsides to consider. The more plugins you have, the more risk you introduce for collision of features and need to maintenance and configuration, so consider the necessity of any plugin prior to installing it. Different plugins may work in slightly different ways, sometimes with overlapping features. For example, there are multiple plugins that provide social network sharing features. One plugin may provide features for sharing to Reddit, Facebook, and LinkedIn. Another may provide sharing to Reddit, Facebook, LinkedIn, Digg, Google+, and Email. If you install and activate both, the resulting user experience will be clumsy.

Test

Certain plugins can cause slight load delays, whether those plugins impact end-viewer experience, or your own administrative experience. As someone who has tinkered with a number of plugins, I have found more than a few that made with the admin pages, reader pages, or both, painfully slow. Sometimes this problem is made evident by user reviews. Other times, you'll have to discover this on your own. Always test response times, appearance, and function after installing a plugin. A plugin may result in some unexpected change. Therefore, while it may be tempting to install a bunch of great looking plugins all at once, it is good practice to install and configure one plugin at a time.

Impact on Load Times

I've personally been frustrated by plugins that advertise themselves as being free, only to find that, upon installation, they provide limited functionality and continuously prompt for purchase of the full version. Unless necessary, I tend to avoid such plugins.

Plugins can be written by anyone, so be cautious to install plugins from trusted sources. Read the plugin reviews and be careful to avoid any (or understand the risks) that may include security holes.

Finally, pay attention to the plugin activity, how widely it is used, whether or not it is maintained, and the age of the plugin. Some plugins can be rendered obsolete by newer versions of WordPress or newer plugins.

The Content: Posts and Pages

Setup is never really complete. Like the landscaping at a park, people expect new and ever-changing scenery.

This isn't to say that you're not ready to begin with content creation. Rather, a site that doesn't morph and change over time is, well… boring. Keep this in mind while as time passes by. New content—regular content—is a fair expectation of a reader. Even if your WordPress site is a semi-static business or informational page, pages that do not change over time begin to seem dull. Go ahead and try a new theme from time to time. Change your colors, logos, and arrangement. Dynamic content and new scenery are the things that keep visitors coming.

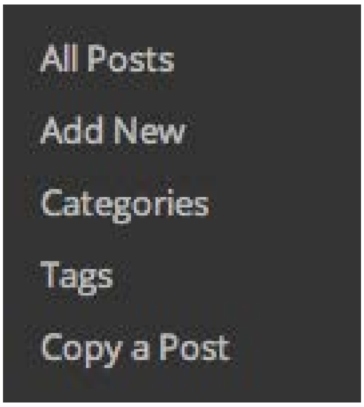
That said, dynamic content is why we're here in the first place. An unchanging set of web pages is created easily enough with HTML, CSS, and a few JavaScript goodies. With WordPress (as with any dynamic CMS, wiki, or blogging tool), the purpose is to promote and simplify ongoing dynamic content.

Even many businesses have embraced the idea of blogging—even if the regular posts are not specifically called *blogs*. Take a look at some of the biggies: *ibm.com*, *redhat.com*, *ge.com*, *exxonmobil.com*, *lilly.com*. Think of any company—large, successful company, and go to their website. What you'll find is not simply a website. It is more of an experience for the view--an opportunity to see what is new and interesting for the business in question. Typically, there on the first page you'll find information on careers, news, events, and conferences. It's what we've come to expect. See *https://wordpress.org/showcase* for an impressive list of WordPress powered sites.

With Posts and Pages, on the right side of the screen you will see a Publish section with a few options, including **Save Draft** and **Publish**. As a general rule, it's a good idea to Save Draft frequently while editing. This will save a draft of your post or page without publishing to the live website. As stable as WordPress is, we've all been frustrated by network connection losses, browser problems, and other applications when something unexpected happens and our work is lost.

Posts

Posts are the things that we think of when it comes to blogging. With WordPress, posts can be categorized, tagged, and formatted in different ways.



Post Format

Those already familiar with *WordPress.com* will know that various post formats are available:

- Standard
- Aside
- Image
- Video
- Quote
- Link
- Audio
- Gallery

These post formats allow various templates to modify the layout of a post depending on the format type. For example, and Aside Format post may appear with smaller texts, or off to one side. This all depends on the template that has been chosen. I typically select Standard for all of my posts, and rely on WordPress shortcodes and post editing, as I don't wish to have post appearances change profoundly when I choose a new template. This isn't to say that post formats should be avoided, but whether or not they meet your needs is subjective.

Pages

A key element separating WordPress from other traditional blogging tools is the ability to create pages that are not blog posts. These pages are still read from the database and read dynamically, but they can be displayed as separate entities—without a blog-like post date and time. It is perfectly reasonable, and common, to create an entire WordPress site without any blog posts.

Shortcodes

Shortcodes are WordPress tags that allow specific content to be embedded in a post. Like codes in a markup languages, **shortcodes** are replaced with dynamic content at presentation time. When using **WYSIWYG** editing mode, the fact that a shortcode is being used behind the scenes will not be obvious.

Shordcodes appear in square brackets ([ ]). There are many already included with WordPress and many plugins that add new ones or expand on existing.

A few **shortcodes** include:

- `[gallery]` – Displays an image gallery in a post or page
- `[slideshow]` – Embed a of photos within a post or page
- `[office]` Embed a document Microsoft Office Live document
- `[twitter-timeline]` - Embed a Twitter timeline

The list of available **shortcodes** is extensive and ever changing. Details can be found at *http://en.support.wordpress.com/shortcodes*/.

Tags

Admin Sidebar->**Tags**

Tags are different *things* than categories. Thing of Twitter *hashtags*—a set of short, preferably one word descriptions of a post –words that give a reader a quick view of the subject matter of a post. Tags can be especially handy for helping others to search for and find tops. A key difference between tags and categories is this: Tags are freeform words, created at post-time,, that label content in an informal, non-hierarchical manner. Along with a **Tag Cloud** plugin, regular use of tags can give a convenient visual of common site subject matter.

Categories

Posts->**Categories**

Post **Categories** (not to be confused with Link categories) may not always be necessary, but as a general rule, it is a good idea to create categories into which posts (and pages) can be organized. WordPress categories can be organized in a hierarchy, and this is a useful way to organize the appearance of menus (primary, secondary). WordPress is set up by default to categories posts as **uncategorized**. While categories are not critical, I will say that leaving all posts in this default category give the appearance of an unfinished website. By default, Pages are not categorized. However, there are plugins that allow this.

You do not necessarily need to determine all of your categories or the hierarchy of categories immediately. I've added and rearranged categories long after a site has been created. A great thing about WordPress is that the category organization isn't set in stone. For example, if you have two categories to begin with, say ***Operating Systems*** and ***Linux***, and each was initially created at the same level, if at some point you decided to move ***Linux*** underneath ***Operating Systems*** (making Operating Systems the parent category of ***Linux***), no links will break. If later still you decided to add a subcategory Windows to the ***Operating Systems*** parent, and ***RedHat***, ***Ubuntu***, and ***CentOS*** has subcategories of ***Linux***, it is easy to do. Keep in mind, however, that addition of new categories may prompt you go to back to old posts and re-categorize.

Creation of categories can also be done at post or page creation time. There have been a number of times when I've written a post, and, unsatisfied with my existing categories, I created one while editing the post. On the right side of the post editing window you will see your existing categories (there are two tabs in the **Categories** section: **All** and **Most Used**). At the bottom of the categories section is a link: **Add New Category**. Don't worry—clicking this link will not take you away from your post. It is handled right there on the same screen, without a page reload. Likewise, any **Category** can be removed at a later time. Doing so will not remove a post that is tied to the category—the category will simply be removed.

There are a number of considerations when creating categories, including hierarchy, volume of posts, posts per category, and so on. In previous example, if you believe that you may have just a few posts in the ***CentOS*** subcategory, perhaps it should instead by labeled ***Others***. And if you do use a hierarchy, think about what makes sense. It may not be clear to a user that some OS related posts is in a subcategory while others appear only in the parent category. Again, as with most of this, it is subjective and depends on your needs and the expectations of your readers. In any case, choose categories and category relationships that distribute your posts in a sensible way.

Reader View of Categories

Categories are visible to the user in a number of ways (all configurable). A sidebar widget and/or category display with each page or post can be presented. The Categories widget, when added to a sidebar or embedded in a page, does not automatically display categories in a hierarchy. While editing widgets, you can check or uncheck whether or not you wish for them to be displayed as a hierarchy (**Appearance**->**Widgets**).

Comments

There's a certain thrill when a new comment arrives on a blog post. It means that someone has been reading, and that someone felt compelled to follow up. Unfortunately, it could also mean that someone is trolling or spamming your site. Comments can be disabled, but in general, I recommend opening up posts for comments. It creates a much more interesting user experience. This may make it necessary to lock comments down to only registered users, or setting comments up so that they must be approved (by you) prior to display. A much easier approach is to use a plugin that guards against spam (such as Akismet, which is discussed later).

Rating

Post Ratings is another feature that users with previous *WordPress.com* experience may be surprised not to find by default on a custom *WordPress.org* site. There are a number of Post Rating plugins/widgets available. While it can be fun to see ratings on a post, sometimes there is little point to making such a feature available. Once again, this is a matter that depends on the purpose of your site. For a new site, or a blog with a number of subjects that invite user feedback, ratings cam be appropriate. For a business blog, ratings are likely something to avoid. (See *http://wordpress.org/plugins/rating-widget/* for an example of a Rating Widget.) The same could be said of other plugins, such as ***Polls*** and **Like on Facebook** enabling widgets.

That's That!

I've touched on a range of subjects here, from the basic details of installation, to configuration and plugins. WordPress has become a very mature solution over the course of ten years, and as such, there is much to learn about all the various features. It is possible to expand on each subject here, and the possibilities for customization are endless. For non-technical users, including small business owners who cannot afford to hire a web developer, WordPress makes creation of a professional website attainable. For the most talented web developers, WordPress expands the possibilities by performing all of the heavy lifting.

What the Internet needs more of, and what people want, is meaningful content. The framework laid out by WordPress allows us to move forward and create that content. So I'll end with this: There are plenty of outlets for link sharing and repeating the thoughts of others. Don't do this… Create and contribute!

Snoop Dogg: www.snoopdogg.com - Created with WordPress

Whether you are setting up a web presence for a landscaper, a contractor, a group of hobbyists, or for your personal interests, remember that there are many others. Countless others! Any website worth the effort required to create it is equally worth the effort to maintain it. It's the ongoing updating that keeps readers coming back. While quality is important, it's the content that keeps readers coming. Case in point: The Drudge Report. Love it or hate it, one cannot deny it's popularity. It's simplicity has become legendary. People don't visit Matt Drudge's simple page (sometimes dozens of times a day) for its beauty. They visit for the content.

(Of course, you and I are not Matt Drudge. We want our pages to have both current, great content, as well as an elegant, attractive appearance.)

Additional Recommended Reading (Just For Fun)

20 Years of a Free, Open Web: http://info.cern.ch/
Tim Berners-Lee: http://www.w3.org/People/Berners-Lee/
History of Blogging: https://en.wikipedia.org/wiki/History_of_blogging
Lynx Web Browser: https://en.wikipedia.org/wiki/Lynx_(web_browser)

Links

Akismet: http://akismet.com/
A List Apart: Responsive Web Design:
http://alistapart.com/article/responsive-web-design
Automattic: http://automattic.com/ (the makes of WordPress)
WordPress.org: http://wordpress.org/
WordPress.com: http://wordpress.com/
WordPress Showcase: http://wordpress.org/showcase/
cPanel: https://cpanel.net/
MySQL: https://www.mysql.com/
PHP: http://php.net/
Softaculous: http://www.softaculous.com/

Matthew Rupert
Matthew Rupert lives in Wake Forest, NC, and is a software architect with Ateb Inc., (Raleigh, NC). He has been designing and developing software for over 15 years. A graduate of Ball State University (Computer Science), he has worked in a range of software development and lead roles ranging from defense contractors to healthcare. He blogs regularly on software subjects at *http://matthewrupert.net*. Find him on Twitter: @matthewrupert