



TASK

WordPress - Overview

[Visit our website](#)

Introduction

TECHNOLOGICAL ADVANCEMENT AT ITS FINEST

Before PHP came along, people used a scatter of languages to do backend programming — from the old Java Server Pages to the 1994 edition of Perl. PHP transformed the backend web scene because of its built-in support for common web app operations. Then, Laravel was introduced to make development on PHP even more powerful and easy. Just when people thought it couldn't get any easier, WordPress entered the arena and made creating a website so simple, literally anyone could do it.

In this task, we'll investigate how WordPress does this, and why it gained so much popularity among developers and laymen alike.

WHAT WORDPRESS IS ALL ABOUT

WordPress isn't like any web development framework out there. WordPress isn't really a framework. It is a highly-configurable website rendering program. So unlike Laravel, where you write code using it for support to build your own website, WordPress offers an *already* fully-capable website, just waiting for you to customise it to your needs.

And the best part is it comes packaged with an easy-to-use GUI. So by using a familiar interface with buttons, text boxes and pictures, you can create a website without writing a single line of code. This is what makes WordPress so popular — a business person who has a great idea, but no programming experience, can create a website in a matter of minutes completely free. Today, **over a third of all websites** are powered by WordPress.

You may be wondering then why we bother teaching web development at all if it's so easy to do for free without any expertise. The reason web developers are still very in-demand is because WordPress (and its alternatives) can't do everything. There are some complex features, especially when integrating with other services like e-commerce, which WordPress isn't built to do.

For scenarios like this, you could create a site from scratch, but luckily the developers over at WordPress had the foresight to allow plugins. What typically happens for a great number of websites is the heavy lifting is handled by WordPress, while the developers are free to write their own plugins (usually in PHP)

to give the website more complex functionality. In this sense, WordPress is sort of like a framework, but it's really a small part of what it can do right out of the box.

In this task, we'll look at setting up your own publicly available WordPress site using nothing but a web-based GUI. In a future task, we'll cover how to write your own custom PHP plugin for it.

HOW WORDPRESS TYPICALLY RUNS

WordPress can be observed in the wild as coming in two varieties: hosted and self-hosted. Self-hosted is when you download your own copy of WordPress, install it on your own server and expose it to the internet. This is common for small and medium enterprise websites and is usually combined with custom plugins.

Hosted is when a service provider manages WordPress installations for a bunch of different customers. This has the advantage of the simplicity of setup and maintenance (automatically handled for you by the service provider) but can come with certain technical restrictions or financial costs. There are many different WordPress service providers (since it's open-source), and many of them have free tiers that allow you to get the basics going.

In this task, you'll be creating and publishing your own WordPress site from the original WordPress service provider — WordPress.com.

BOOTING YOUR SITE

Head over to WordPress.com and click "Start your website". You'll first be asked to create an account, and then to choose a domain. Custom domains that end in a TLD (Top-Level Domain, like .com, .net, .org, etc.) come at an annual cost. The WordPress service provider allows you to create your website as a subdomain. That is, **yourwebsite.wordpress.com**. The only real disadvantage here is that your site's URL doesn't have its own identity. But, you can't beat the price (free), so go ahead and pick any name. You'll be creating your own developer profile, so the name should ideally include your name.

hyperiondev.wordpress.com	Free	Select
hyperiondev.org	Registration fee: \$18 Free Renews at \$18 / year	Select
hyperiondev.software	Registration fee: \$25 Free Renews at \$25 / year	Select

On the next screen, click “Start with a free site” and, after a few seconds, you should see the admin console of your freshly created website.


BASIC IDENTITY SETUP

On the left bar, click on the Manage dropdown and then on Settings. Here you should fill in your site title (website name), and a catchy tagline. Here’s our example:

Site profile

Save settings

Site icon



Change

Site title

Harold Hyperion

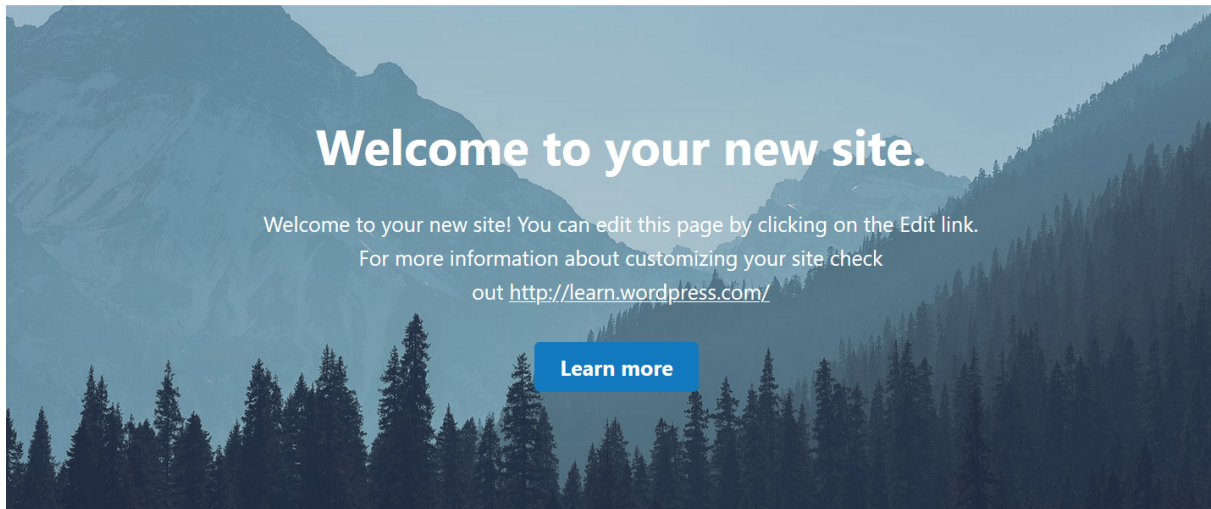
Site tagline

Hi! I'm Harold and I'm a passionate PHP developer.

In a few words, explain what this site is about.

Later, you can add your own site icon to personalise it even more.

Now in the left side panel, click on Design, and then on Customize. This will take you to the website customiser, which is a GUI interface used to alter the site with real-time changes. To start off with, your site should be quite generic:



Latest from the Blog

There are some options you can tweak in the side panel, but it's best not to get too carried away here, as most of the options are tied to the current theme. One of the wonderful features of WordPress is that you can easily apply themes to overhaul your whole site in just a few clicks. Head out of this screen by clicking the X at the top left.

Now under the Design submenu in the left side panel, click on Themes. Here you'll see a mountain of different themes to choose from. Some are professional curations, some are proprietary, and some are user-contributed. All of them are customisable to your exact liking. Pick one you think will fit your personal website well. On a theme's page, click "Activate this design" and then on Customize.

You'll now be back at the site customiser but may notice a lot has changed. There will probably be some more menu options on the side panel indicating more customisation ability. Feel free to tweak these in your own time.

YOUR OWN PAGES

Head back out to the admin console, click on the Site dropdown, and then on Pages. This is where your website's static pages will be. These are usually unchanging informational pages about you or the website. By default, WordPress creates four pages, named earlier. Click on the About page to go to its page editor. The default page will look different per theme, but you should edit this space to

describe yourself. Add in a photo and some content about yourself. Here's what we did:

About



Credit: Getty Images

Harold Hyperion – your tech whizz in PHP.

I have a lot of experience writing back-end web applications in PHP. People often criticise PHP for being a monolithic language, but really with PHP anything is possible. In fact, I wrote this website in PHP. (Just kidding, this is definitely WordPress).

Click Update (top-right corner), and then head back by clicking on the “W” at the top-left corner.

Now open your home page and do the same — edit it to your liking.

When you get back to your pages overview, go ahead and add a new page (button to the right of the page). It'll ask you to pick a layout for the page. You'll now be making a page to list a summary of the project's you've done so far. The example layouts are grouped by category — you might find something you like under the Portfolio heading. Select a layout you think appropriate and click “use layout” (top-right). You'll now be taken to a page editor.

Edit the page to include all the Capstone Projects you've done so far (and any other personal tech projects you've been involved with). We recommend using images, even if only technology logos (like Java, JS, PHP, Docker, etc.) or application screenshots. It immediately gives the viewer a good impression. Add short descriptions and links to the appropriate GitHub repos, if they exist. Finally, click Publish (top-right).

YOUR OWN BLOG

If you want to, WordPress lets you use your website as a blog too. You can write about whatever you want — be it your learning experiences while coding, or your personal philosophy about life. This is a fantastic feature and greatly helps people that stumble upon your site to gauge your character.

Head back to the admin console and, under the Site dropdown, click on Posts. This is your post overview page. Posts are simply blogged articles that each follow exactly the same format (as set by you or your theme). You should see some example posts. Ignore them and create your own right now (button to the right).

Make the title “Hello World” and write two or three paragraphs about what brought you to create your own website (please don’t give us the credit, we’re sure you have your own motivations to have a personal website).

When done, click on Publish (top-right).

LET THE WORLD MARVEL

It’s time to let the world get to know you. Head to the admin console, and click on your site name at the top left. A render of your site-to-be should appear on the main screen. Click on “Visit site” (top-right). This will show you a complete render of what your site will look like once you make it public. To do so, click on “Launch site” (top-right). WordPress may try to persuade you to buy your own domain, but ignore them and click on “Skip purchase” if harassed. After a bit of loading, you should be in business.



Compulsory Task 1

Ensure you have done the following (from the guide above):

- Chosen a custom theme for your site.
- Edited the About page to describe you.
- Edited the Home page to suit you.
- Created a Portfolio page listing projects you have done.
- Written your first blog post.
- Published your site.

Compulsory Task 2

Make the following improvements to your website:

- Add your own site icon. It needn't be very professional. If you're out of ideas, pick [one of these icons](#) (licensed for use under creative commons).
- Customise the main menu to include a link to your portfolio page.
- Add a link to your portfolio both from your Home page and from your About page.
- Sprinkle in some images for aesthetics. Hint: you can use most [Getty Images](#) pictures under a Creative Commons licence, provided you give attribution. That's what we did in our example website's design.
- Submit a link to your published website in a file called **my-website.txt**.



Rate us

Share your thoughts

HyperionDev strives to provide internationally-excellent course content that helps you achieve your learning outcomes.

Think that the content of this task, or this course as a whole, can be improved or think we've done a good job?

[Click here](#) to share your thoughts anonymously.

