TASK

WordPress - Themes

Visit our website

Introduction

THEMING IN WORDPRESS

In the last task, you saw how to pick and apply a premade theme to your WordPress site. Although these are often very customisable, you may have very specific needs to customise it even further, for instance, if you're building a site for a client that has a specific branding aesthetic. In this task, we'll cover how to style a WordPress site to your *exact* needs.

LOCAL WORDPRESS DEVELOPMENT

Unfortunately, WordPress.com (WP) places some inconvenient restrictions on customisation abilities for free accounts. In order to access the full array of capabilities, you'll be installing and running your own copy of WordPress on your machine and hosting a website yourself.

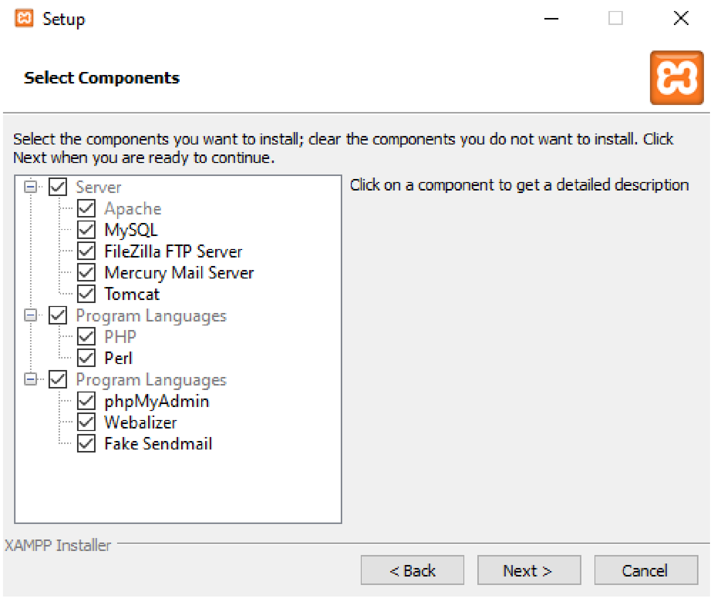
Another misfortune of WP is that it does not officially support SQL Server (the DB platform you've been learning on so far). For this reason, you will need to install XAMPP to host your WP locally on your machine.

XAMPP INSTALLATION

XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends. It consists mainly of Apache HTTP, MariaDB database, and interpreters for scripts written in PHP and Perl. This will allow you to build WP sites offline, on a local web server on your computer.

Please follow the following steps to install XAMPP:

1. Go to the Apache Friends website, **http://apachefriends.otg/downloads.html**, and download the file for the platform you are using. After downloading it, launch the installer and the setup process should begin.
2. Depending on your operating system, you may receive a security warning. Do not worry, XAMPP is safe to install. When the setup wizard pops up, click "Next", and in the next window, you will be asked to select components you want to install. Options like Apache and PHP are required and are installed automatically. Since our goal is to host WordPress locally, you must check MySQL and phpMyAdmin boxes. You can uncheck the rest if you want to.



Setup

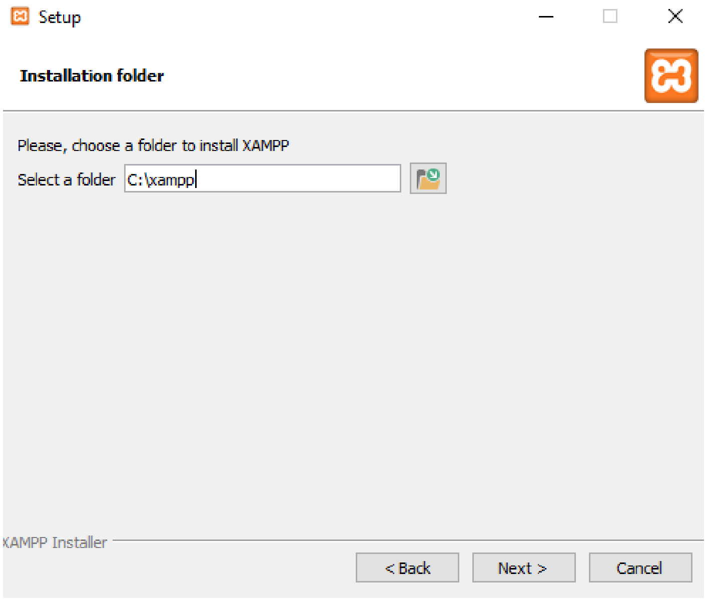
Select Components

Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

- ☑ Server
  - ☑ Apache
  - ☑ MySQL
  - ☑ FileZilla FTP Server
  - ☑ Mercury Mail Server
  - ☑ Tomcat
- ☑ Program Languages
  - ☑ PHP
  - ☑ Perl
- ☑ Program Languages
  - ☑ phpMyAdmin
  - ☑ Webalizer
  - ☑ Fake Sendmail

Click on a component to get a detailed description

Next, select the folder where you would like to install XAMPP on your computer. It's up to you if you want to create a new folder or use the default path, i.e. **C:\xampp**.



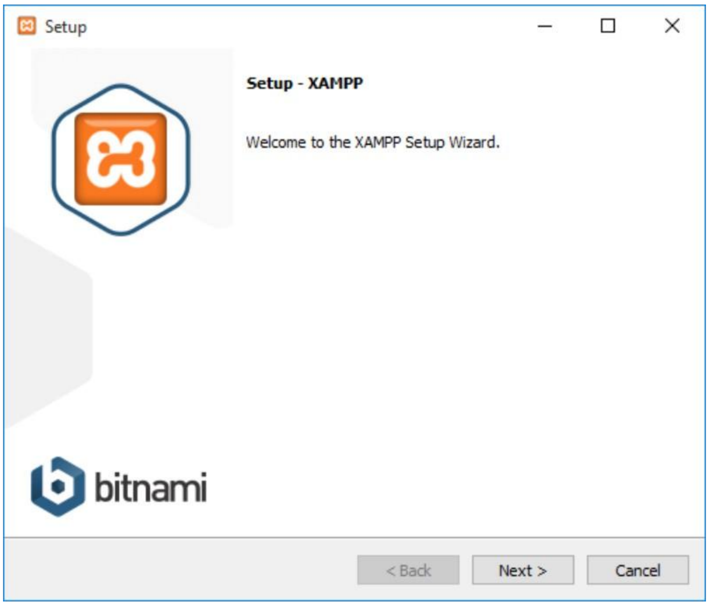
Setup

Installation folder

Please, choose a folder to install XAMPP

Select a folder C:\xampp

After choosing the installation directory, you will be asked to install Bitnami. This extension makes installing third-party CMS programs like WordPress, Joomla! and Drupal hassle-free. We will manually install WordPress, so uncheck "Learn more about Bitnami for XAMPP" and click "Next".

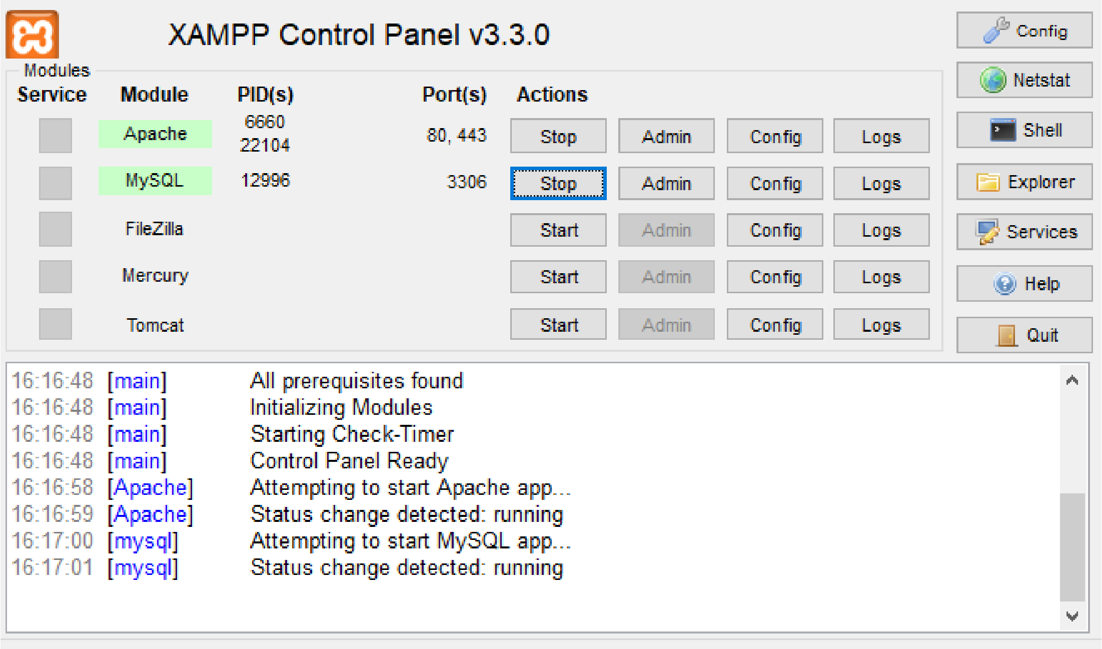
3. 

Welcome to the XAMPP Setup Wizard.

XAMPP should be ready to install but you may encounter a security/firewall warning. Make sure you check "Private networks, such as my home or work network", and click "Allow Access".
After the installation, you will be asked if you would like to start the XAMPP Control Panel. Leave the box checked and click "Finish". The Control Panel will now open.
4. If you accidentally unchecked the box, you can go to your XAMPP folder on your computer and manually run the XAMPP Control Panel.

Before using XAMPP, please check that you do not have any conflicting applications running in the background which are using the same ports. XAMPP will not be able to function properly, if it has to *share* ports with other applications.

If no errors occurred during the installation process, the control panel would show black and blue text below the list of the panel’s components (Apache, MySQL etc.). All of the elements have their separate Start/Stop, Admin, Config, and Logs buttons.



Service	Module	PID(s)	Port(s)	Actions			
□	Apache	6660 22104	80, 443	Stop	Admin	Config	Logs
□	MySQL	12996	3306	Stop	Admin	Config	Logs
□	FileZilla			Start	Admin	Config	Logs
□	Mercury			Start	Admin	Config	Logs
□	Tomcat			Start	Admin	Config	Logs

```
16:16:48  [main]      All prerequisites found
16:16:48  [main]      Initializing Modules
16:16:48  [main]      Starting Check-Timer
16:16:48  [main]      Control Panel Ready
16:16:58  [Apache]    Attempting to start Apache app...
16:16:59  [Apache]    Status change detected: running
16:17:00  [mysql]     Attempting to start MySQL app...
16:17:01  [mysql]     Status change detected: running
```

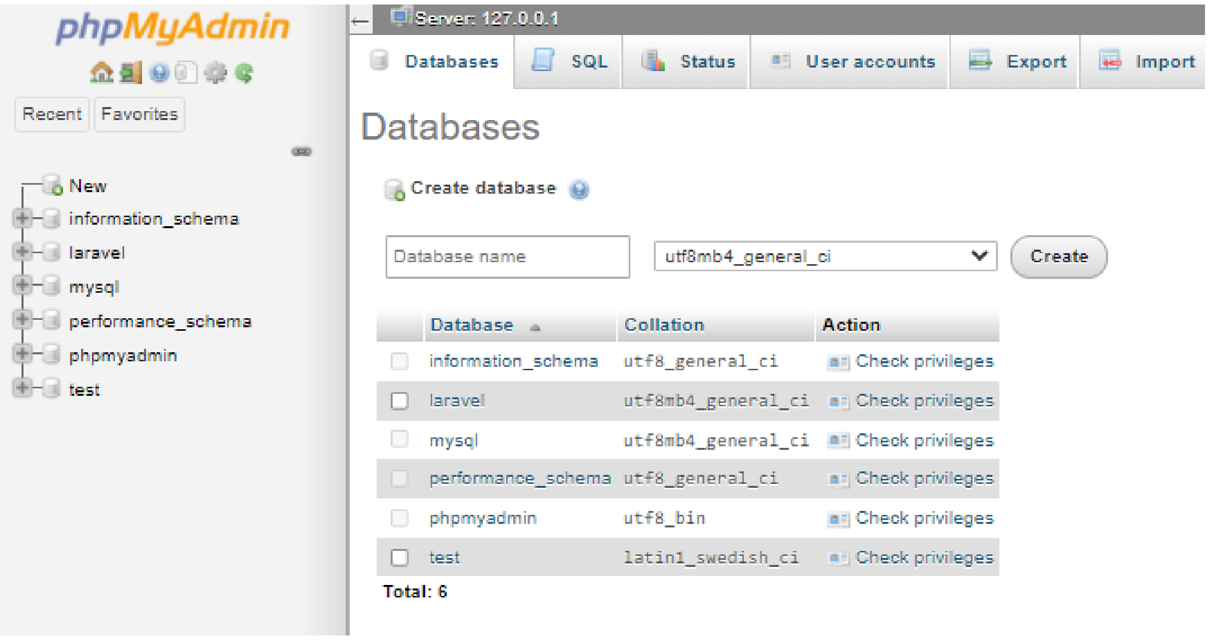
Run Apache and MySQL by clicking their respective Start buttons. If the components started without problems, you would see a green background behind their names.

Congratulations! You have successfully installed the XAMPP server on your computer.

SETTING UP WORDPRESS DATABASE USING XAMPP

Once you have your Apache and MySQL server running, you need to create a database and add database credentials.

Let's start with creating the database that we will be using for our WP site. Click on the MySQL "Admin" button on the XAMPP Control Panel. This should load the phpMyAdmin page within a tab in your browser.



Databases

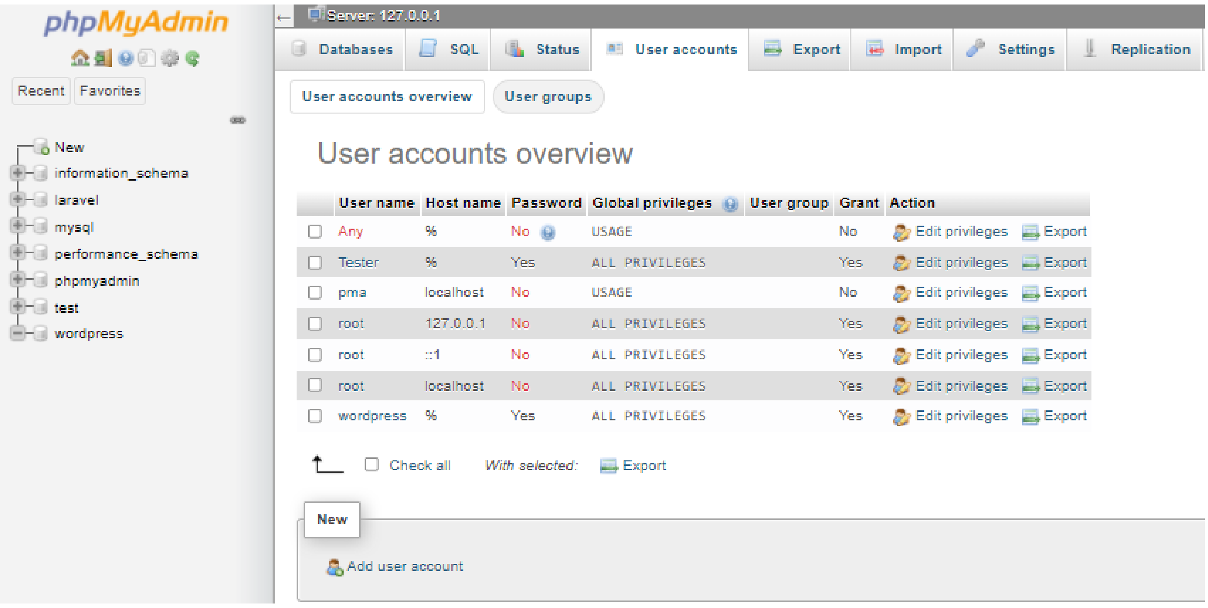
Create database

Database name | utf8mb4_general_ci | Create

	Database	Collation	Action
□	information_schema	utf8_general_ci	Check privileges
□	laravel	utf8mb4_general_ci	Check privileges
□	mysql	utf8mb4_general_ci	Check privileges
□	performance_schema	utf8_general_ci	Check privileges
□	phpmyadmin	utf8_bin	Check privileges
□	test	latin1_swedish_ci	Check privileges

Total: 6

To create a database, click on “New” in the sidebar or click on the “Databases” tab as shown above. Next, we will be creating new database credentials.



User accounts overview

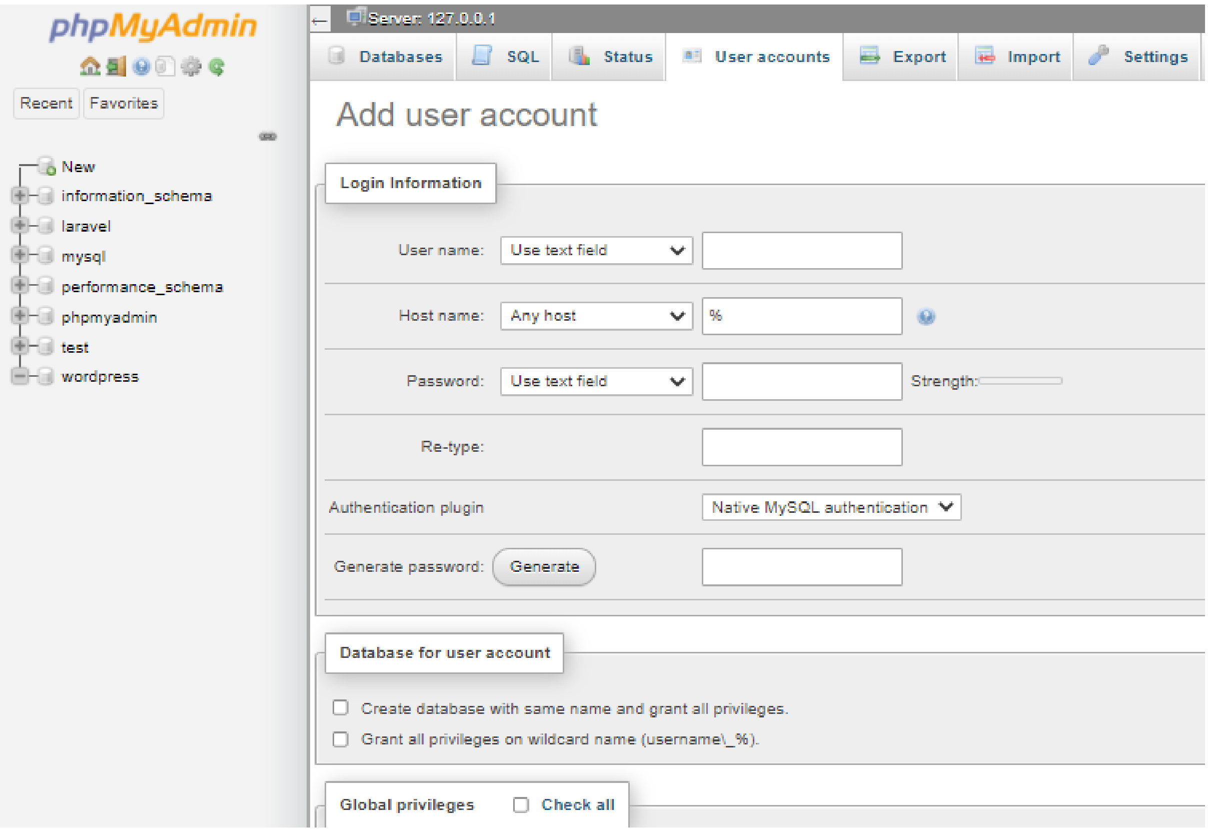
	User name	Host name	Password	Global privileges	User group	Grant	Action	
□	Any	%	No	USAGE		No	Edit privileges	Export
□	Tester	%	Yes	ALL PRIVILEGES		Yes	Edit privileges	Export
□	pma	localhost	No	USAGE		No	Edit privileges	Export
□	root	127.0.0.1	No	ALL PRIVILEGES		Yes	Edit privileges	Export
□	root	::1	No	ALL PRIVILEGES		Yes	Edit privileges	Export
□	root	localhost	No	ALL PRIVILEGES		Yes	Edit privileges	Export
□	wordpress	%	Yes	ALL PRIVILEGES		Yes	Edit privileges	Export

□ Check all *With selected:* Export

New

Add user account

Click on the “User accounts” and you will be taken to the view as shown in the above image. Once you have done that, click on “Add user account”.



Add user account

Login Information

User name: Use text field

Host name: Any host %

Password: Use text field Strength:

Re-type:

Authentication plugin Native MySQL authentication

Generate password:

Database for user account

□ Create database with same name and grant all privileges.

□ Grant all privileges on wildcard name (username_%).

Global privileges □ Check all

Add the following details in the form shown above:

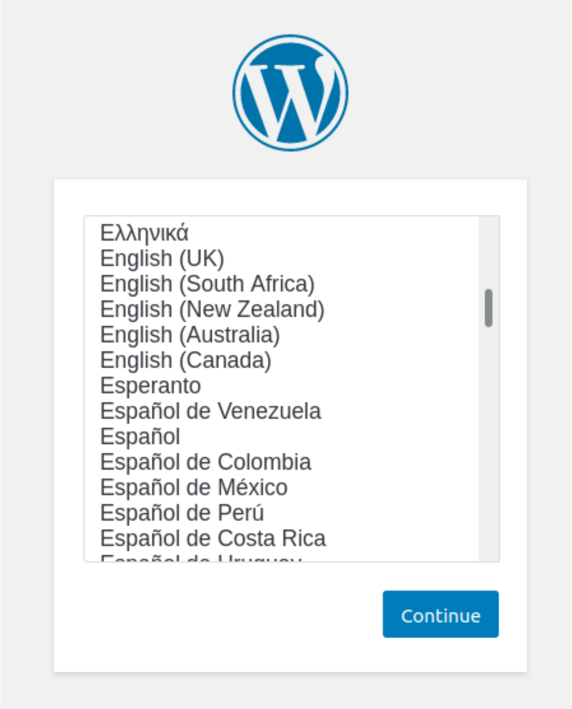
- User name: wordpress
- Password: WordPress (repeat it in the “Re-type” field)
- Check “Global privileges”

To add the new user account, click on the “Go” button.

GETTING STARTED WITH WORDPRESS USING XAMPP

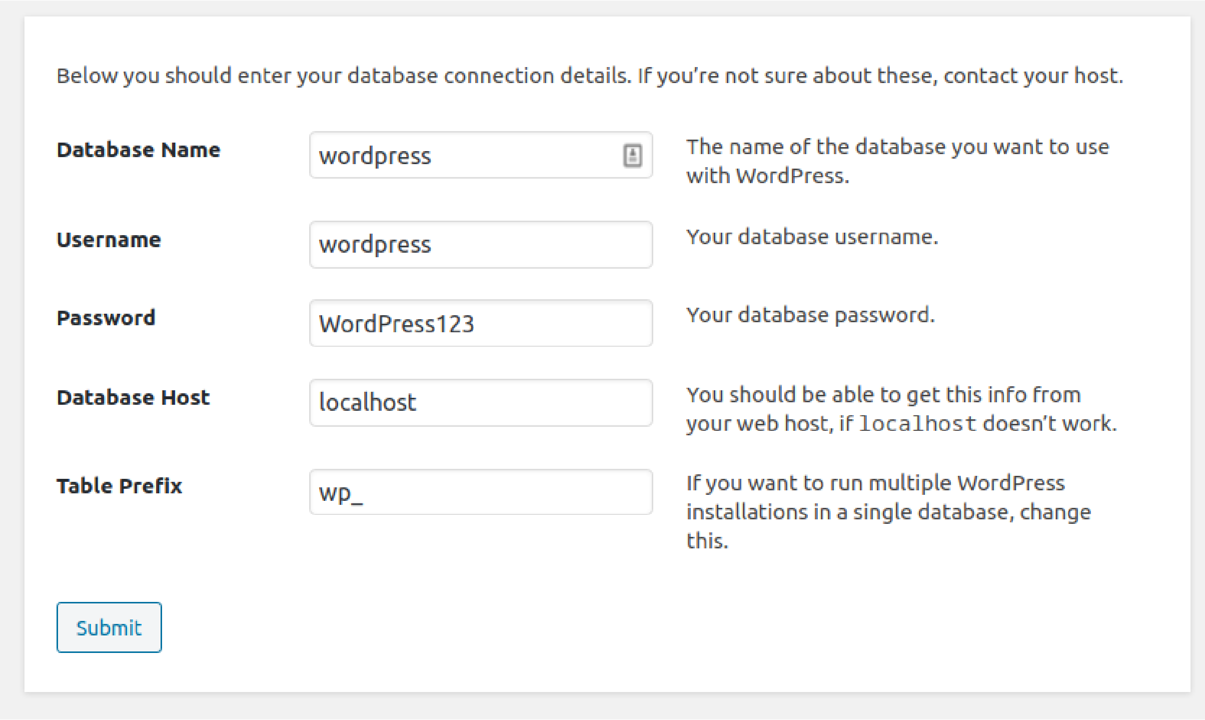
The following steps will guide you through downloading WordPress to your computer and show you where you should store your local WordPress site.

1. Go to **https://wordpress.org/download/** and download WP. You should receive a zip file containing the folder with the WP site.
2. Open the folder and select the “wordpress” folder and move it to the **C:\xampp\htdocs** directory.
3. Click the “Admin” button, for the Apache server, on the XAMPP Control Panel. A new tab in your browser will open with a welcome screen. Change the URL of the current web page to **http://localhost/wordpress** and then the following will appear.



Ελληνικά
English (UK)
English (South Africa)
English (New Zealand)
English (Australia)
English (Canada)
Esperanto
Español de Venezuela
Español
Español de Colombia
Español de México
Español de Perú
Español de Costa Rica
Español de Uruguay

Pick your favourite flavour of English, and hit "Continue". The next screen explains that we'll be setting up the WP database. Hit Continue, and fill in the appropriate details in the HTML form:



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 use with WordPress.
Username	wordpress	Your database username.
Password	WordPress123	Your database password.
Database Host	localhost	You should be able to get this info from your web host, if `localhost` doesn't work.
Table Prefix	wp_	If you want to run multiple WordPress installations in a single database, change this.

Clicking the "Submit" button will test your connection and inform you if there are any errors. After connection success, click "Run the installation". You will be asked for some basic information about you and your site. You can use the same details as your WP site from the previous task. All information entered here is only valid for your local copy of WordPress. You'll be creating an account that works on your machine only, but it's still important to remember the username and password as it's what you'll be using to edit your site locally.

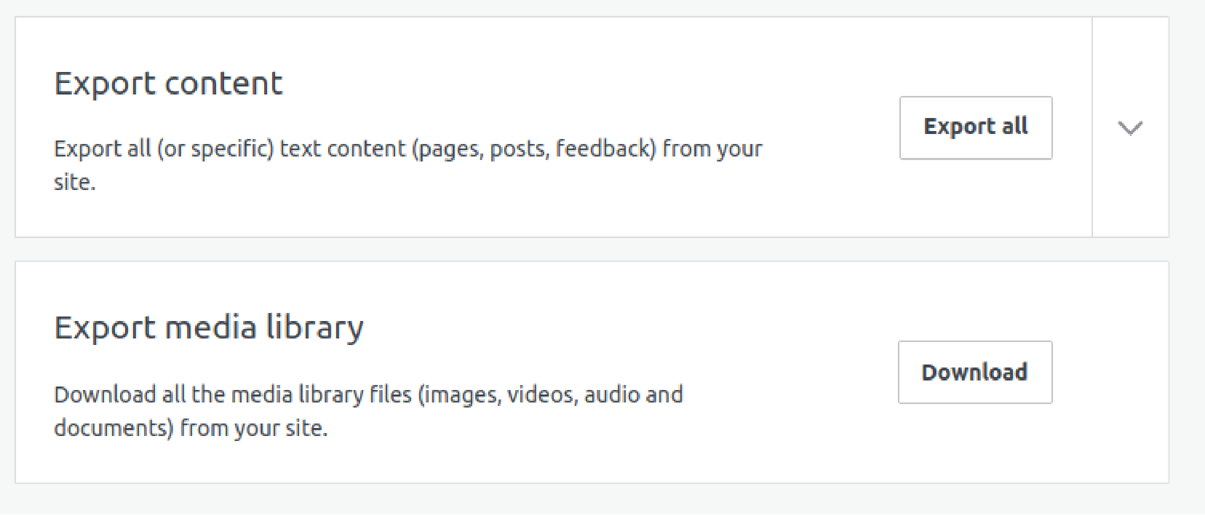
Click "Install WordPress" to install WP locally. This will set up the database with appropriate tables and content to get you started with a basic site.

When complete, click "Login" and then log in to get to your local admin console. You'll notice the console looks a bit different to the one on WordPress.com, but rest assured it has all the same features (and actually some more too).

EXPORTING AND IMPORTING

If you go to **http://localhost/wordpress** in your browser, you'll see a default website. For convenience's sake, WP allows exporting and importing of site content. This is useful for making backups, or in our case, moving from one WP server to another.

Log in to **wordpress.com** and click on "My Site" (top-left) to get to your website's admin console. Under the tools dropdown, hit "Export". Click on "Export all" to get a zip of all your written content (pages and posts).



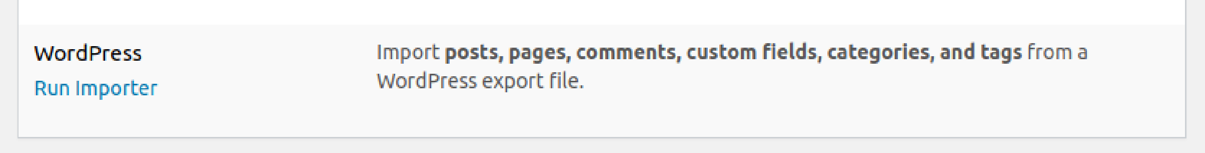
Export content

Export all (or specific) text content (pages, posts, feedback) from your site.

Export media library

Download all the media library files (images, videos, audio and documents) from your site.

Now head to your local WP admin console, click on "Tools" and then "Import". It'll list the most popular import plugins (there are many places from which you can import). Click on "Install Now" under WordPress. It should take a mere few seconds. That is how easy installing a WP plugin usually is.



WordPress
Run Importer

Import **posts, pages, comments, custom fields, categories, and tags** from a WordPress export file.

Now hit "Run Importer". The file to import is a WXR file, which is WordPress's custom flavour of XML used to markup WP content. It will be inside the zip you exported earlier. Unzip it, select it in the importer, and hit "Upload". It will now ask you if you want to use the same author for the content, or if you want to create a new one. If you're happy with your online WP username as the article author, leave it as is.

Be sure to check "Download and import file attachments" — this will import relevant pictures and other media you used to create your online WP site. It will automatically be downloaded, and so might take a while. Hit "Submit" when ready.

After importing finishes, head back over to **http://localhost/wordpress** to see your imported posts and pages from your online WP site. Of course, the styling will be different, and we'll talk about that soon. For now, take a screenshot of your site with the imported content — you'll be submitting it in a compulsory task.

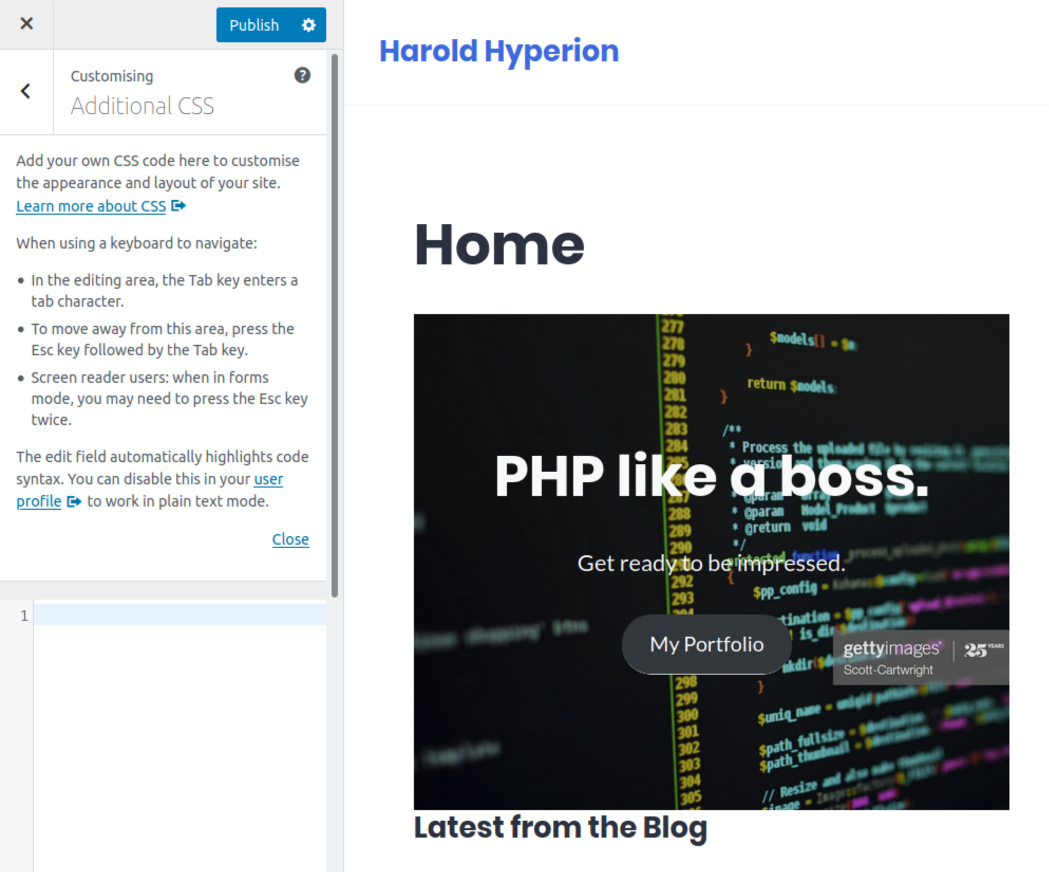
At this point it gives us no pleasure to inform you about one of the irkings of capitalism. In order to export your WP theme, you need to install the appropriate plugin, which would normally have been fine, but at WordPress.com (and most alternatives), installing plugins requires a paid account. So there is no way to transfer your site theme customisation for free.

What you can do is find the same theme you used and customise it yourself again from the start, but we do not require you to do this. Today's lesson is about custom themes, so we'll be making our own alterations to the theme anyway.

CUSTOM THEMING

After you're happy with a basic theme (that does not need to match your online WP site), we're ready to get into custom theming. WP works like any other normal website, in that at the end of the day it's just HTML, CSS and JS working together to render the web content you see. Most elements on your site (titles, paragraphs, buttons, etc) have CSS classes and IDs that are used to give them style.

You as a developer with a background in CSS can take advantage of this through the use of WP's CSS customisation functionality. In your local admin console, click on "Appearance", and then "Customise". You'll be presented with the familiar theme customiser. At the very bottom of the side panel menu, click on "Additional CSS". This feature lets you add your own custom CSS code to modify your site any way you please.



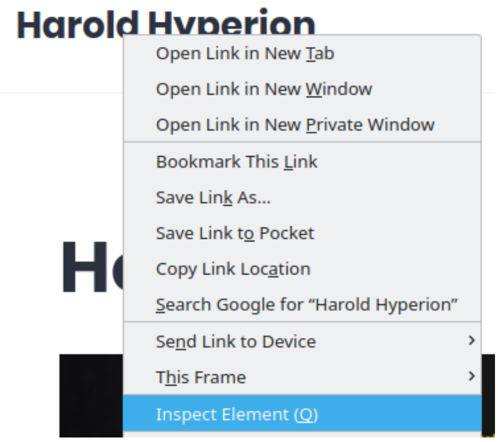
Add your own CSS code here to customise the appearance and layout of your site.
Learn more about CSS

When using a keyboard to navigate:

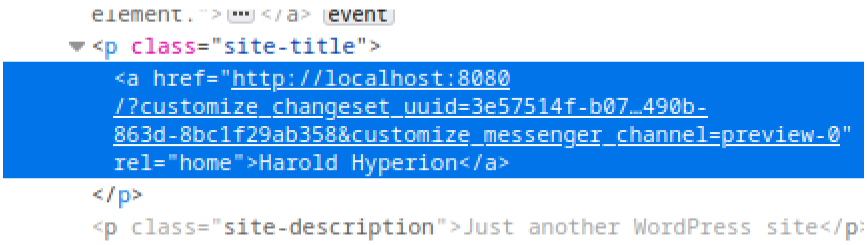
- In the editing area, the Tab key enters a tab character.
- To move away from this area, press the Esc key followed by the Tab key.
- Screen reader users: when in forms mode, you may need to press the Esc key twice.

The edit field automatically highlights code syntax. You can disable this in your user profile to work in plain text mode.

You can use normal CSS selectors like HTML tags, but the easiest way to find how to select specific elements is by inspecting it with your developer tools in your browser. For instance, to modify the site's title (Harold Hyperion in our case), right-click on it, and select "Inspect Element".



This will open your browser’s developer tools, which is a feature you may have had exposure to when doing other web development. It is a set of tools used to examine the inner workings of a web page. The element inspector, which is the specific tool opened, allows you to see the full HTML definition of all elements. By right-clicking on an element, the inspector will automatically jump to the clicked-on element.



```
element."> … </a> event
▼<p class="site-title">
   <a href="http://localhost:8080
   /?customize_changeset_uuid=3e57514f-b07…490b-
   863d-8bc1f29ab358&customize_messenger_channel=preview-0"
   rel="home">Harold Hyperion</a>
 </p>
 <p class="site-description">Just another WordPress site</p
```

You should see something like this: a `<p>` tag containing an `<a>` tag. You’ll notice that it has the class **`site-title`**. In your CSS editor (on the left panel), write some CSS to increase the size of the site’s title:

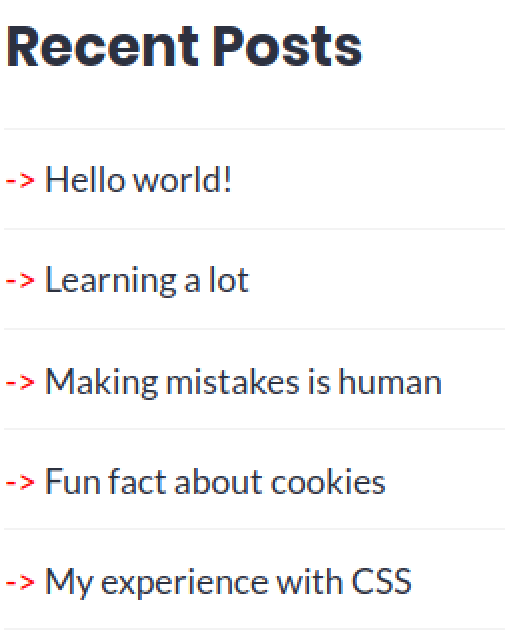
```
.site-title {
      font-size-adjust: 1.2;
}
```

The change should render immediately.

Custom CSS can be used to change quite literally anything in your site, including multiple items at once. For instance, Harold’s site has a Recent Posts section and, to alter it, we may use the following selector (after having discovered they are listed as `<li>`s):

```
li::before {
      content: '-> ';
      color: red;
}
```

Result:



Recent Posts

-> Hello world!

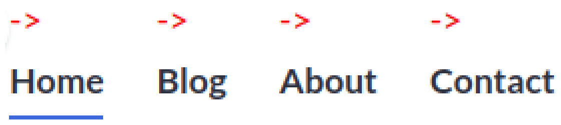
-> Learning a lot

-> Making mistakes is human

-> Fun fact about cookies

-> My experience with CSS

This can be a double-edged sword, however, since it could cause unintended changes in other places of your site. For instance, one may discover that Recent Posts isn't the only place that uses `<li>`s.



-> Home -> Blog -> About -> Contact

Best practice is to always be as specific as possible to avoid these unintended styling changes. For example, in our case, the element inspector reveals that Recent Posts list items are within a `<section>` with id `recent-posts-2`. We can therefore specify the CSS as such:

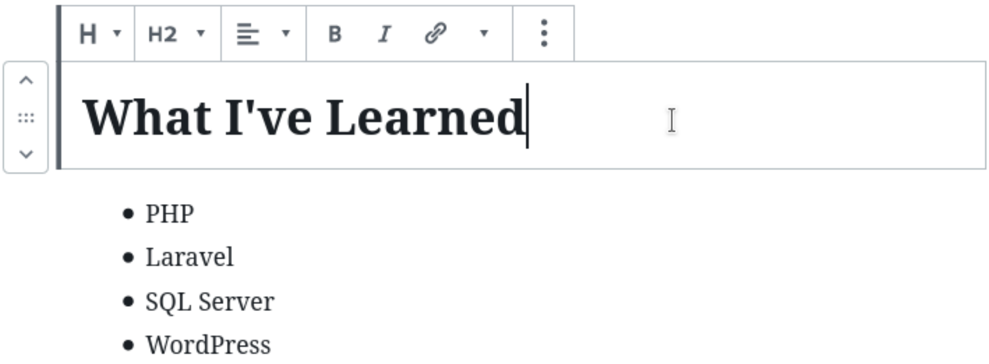
```
section#recent-posts-2 li::before {
      content: '-> ';
      color: red;
}
```

You can click around your site to see all your pages and posts and look for opportunities to restyle with custom CSS. Just don't get too carried away — there is often a lot you can customise by just interacting with the normal theme customiser interface.

When done, click on "Publish" to have all your changes go live on your website. Browse to it now in your browser to check that it worked.

CUSTOM CLASSES

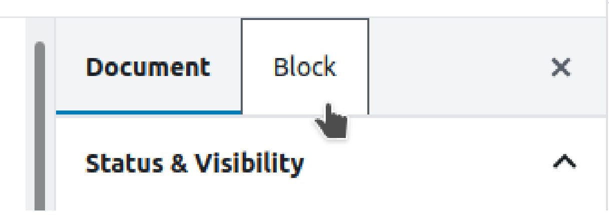
Sometimes you may want to theme certain parts of your site that don't have easily selectable elements. A good example of this is arbitrary content in your posts or pages. In a post or page, create a heading, or select an existing heading.



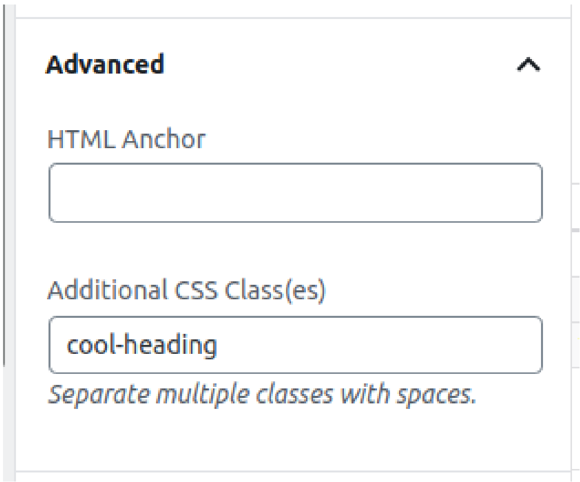
What I've Learned

- PHP
- Laravel
- SQL Server
- WordPress

On the right-side panel, click on "Block" (which refers to the specific content block you're currently editing).



Then, at the bottom, click on "Advanced". You'll see a box that allows you to define specific CSS classes for that block (in this case, heading). Add a class name there now.



HTML Anchor

Additional CSS Class(es): cool-heading

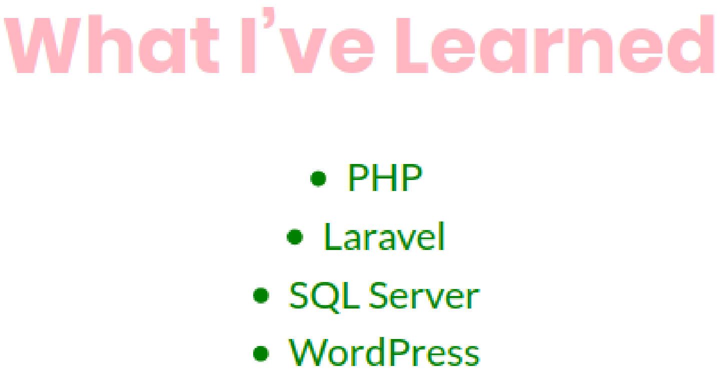
Separate multiple classes with spaces.

Do the same for any arbitrary paragraph. We're going to set our list block from above to class **`cool-list`**. Click "Update" when you're done.

Head back to the CSS editor in the theme customiser. In the site preview, navigate to your post or page you just added the classes for. Add any CSS you like to alter the style of your heading and paragraph. Here's what we did:

```css
.cool-heading {
      color: lightpink;
      text-align: center;
}

.cool-list li {
      color: green;
      text-align: center;
}
```



What I've Learned

- PHP
- Laravel
- SQL Server
- WordPress

Very importantly, this only affects items with those specific classes. You can now add the **`cool-heading`** class to any heading you want to format with this styling. Remember to hit "Publish" to save your changes.

Compulsory Task 1

Ensure you’ve done everything in the guide, including:

- Installed WP on your machine.
- Imported your posts and pages from your online WP site.
  - Submitted a screenshot of your local site at this stage.
- Made arbitrary CSS adjustments to your site (to confirm it works).

Compulsory Task 2

- Make at least three custom CSS changes to your website. Some ideas:
  - The site’s title.
  - Background colour/image
  - Recent posts/comments listing.
  - Footer
  - Link colours
  - Default cursor
- Make at least three custom CSS classes and apply it to at least one element each. Some ideas:
  - Headings
  - Lists
  - Paragraphs
  - Images
- Submit all your CSS code in a file called **custom.css** and provide screenshots showing how your CSS code styles your site. Two or three should be enough, depending on what precisely you opt to style.

Optional Bonus Task

- Update your online WP site with some custom CSS. You can make any styling changes your heart desires. Feel like a button is off by 3 pixels? No problemo.

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.