Content Management Systems and WordPress

Next Steps in Web Development

At this point, you probably have had some experience creating websites by writing HTML pages and CSS stylesheets in a code editor.

However, you may have noticed that HTML and CSS alone do not give you the tools to make a large, complex website.

Some things a large website may need that can't be done with HTML and CSS:

- Allow users to sign up for accounts
- Allow users and admins to post and modify content, that is stored persistently
- Allow users to retrieve content, get updates, etc.
- Allow admins to update the site easily and frequently without writing any code:
  - Add new pages or areas to the site
  - Update the design

You can probably think of even more things!

Server-Side Scripting

As an example of a large complex website, think of **Brightspace.** (Used by tens of thousands of students and teachers at this college and around the world.)

When an instructor posts a new assignment, obviously they don't have to open the code files for the site in an editor and change the HTML!

Similarly, when you do a search on the site for a particular course, there obviously isn't anyone who quickly writes the HTML code for the results page and sends it to you - the results page is automatically generated someone how on the server.

*This demonstrates the needs for **server-side scripting** and **a database.** The website needs some kind of program running on the server that can accept inputs from users, store necessary data for later, and generate HTML pages, with CSS, on the fly.*

Server-Side Scripting

In the example of Brightspace, a **program running on the server** accepts files, discussion posts, assignment solutions, etc. that have been submitted by users, and **stores them in a database**. Similarly, when you try to view one of these things, a program retrieves the necessary thing from the database and automatically builds the HTML page, which is sent to your browser.

Content Management Systems

As web developers, we could learn how to write these programs (in a language like JavaScript, Java, PHP, Ruby, etc.) and design databases (using a system like MySQL, MongoDB, etc.) We could essentially code a site like this from scratch, even though it would be time consuming and expensive.

However, it turns out we don't have to reinvent the wheel; instead, we could use a ready-made solution that allows us to do all the things we need with only a minimal amount of programming, or even no programming at all!

This is where **Content Management Systems** come in.

Content Management Systems

Content Management Systems, or **CMS**s, allow site administrators to quickly, easily, and cheaply build complex sites without the need to write any code or know much about how databases work. These sites can be customized to meet their needs and updated with minimal effort.

New features, functionalities, and design templates can be added to a CMS site without the need to do any custom coding or design.

A content management system allows anyone, including those without expertise in software or web development, to create and maintain complex websites.

Content Management Systems

Some popular CMSs, along with their market share, are the following (from Kinsta):

- WordPress (64.7%)
- Shopify (5.4%)
- Joomla (3.3%)
- Squarespace (2.5%)
- Wix (2.4%)
- Drupal (2.3%)
- ... and many others!

Content Management Systems

Some popular CMSs, along with their market share, are the following (from Kinsta):

- **WordPress (64.7%)**
- Shopify (5.4%)
- Joomla (3.3%)
- Squarespace (2.5%)
- Wix (2.4%)
- Drupal (2.3%)
- ... and many others!

According to W3Techs, **more than 40% of sites on the web use WordPress**. It's worth noting that estimates vary, and I've seen numbers from other sources as low as 30%. Whichever source you consult, it seems safe to say that roughly 1/3 of all websites on the web are powered by WordPress, meaning it's an influential technology worth learning.

Therefore, WordPress is the CMS we will focus on in this course.

WordPress

Here are some things that can be easily done with WordPress:

- Posting and updating new content
- Creating navigation menus
- Providing search functionality, as well as categorization and tagging content
- Using templates to quickly build complex layouts and maintain consistent structure on each page
- Managing users, allowing new users to register, and providing different levels of privileges
- Using a WYSIWYG editor to easily edit pages, and even the structure of the site itself
- Installing **themes** to completely change the design of the site
- Installing **plugins** to add advanced functionality, such as shopping carts, payment systems, etc.
- ***Customizing the site by creating one's own themes and plugins***

WordPress

WordPress started 20 years ago as a blogging platform, but nowadays, it can be used as a basis to create **any kind of website,** with complete customizability. WordPress can be used two ways:

- WordPress**.com**
  - Quickly sign up and create a site using a pre-existing template;
  - Hosted for free on WordPress's servers;
  - A paid tier allows you more customization and to choose a better domain name.
- WordPress**.org**
  - Complete freedom to customize;
  - Must host it yourself and install the software from scratch;
  - Totally free, community-driven, and open source.

WordPress

WordPress started 20 years ago as a blogging platform, but nowadays, it can be used as a basis to create **any kind of website,** with complete customizability. WordPress can be used two ways:

- WordPress**.com**
  - Quickly sign up and create a site using a pre-existing template;
  - Hosted for free on WordPress's servers;
  - A paid tier allows you more customization and to choose a better domain name.
- WordPress**.org**
  - Complete freedom to customize;
  - Must host it yourself and install the software from scratch;
  - Totally free, community-driven, and open source.

WordPress: How it Works

WordPress sites consist of the following components:

- **Pages:** important areas of the site. Examples: contact, about us, etc.
- **Posts:** smaller units of frequently-updated content, often listed on the home page. Examples: blog posts, articles, news stories, comments, etc.
- **Categories & Tags:** allow one to place posts into categories and tag content with keywords, for organization and findability.
- **Plugins:** allow one to add special functionality to a site. Often maintained by third-party developers
- **Permalinks:** a system of generating URLs for resources dynamically; the site admin can control the format of URLs.
- **Blocks:** these are literally the building blocks for site content, which can easily be added, moved, and edited. Examples include paragraphs, headings, lists, etc. As you can see, blocks are kind of like HTML elements, but with more features.

WordPress: How it Works

WordPress sites consist of the following components: **(continued)**

- **Themes:** new themes can be downloaded and installed to change the look and feel of the site. Themes usually have some settings one can edit to change things like color, typography, etc. They're often developed by third parties.
- **Templates:** themes are made up of a collection of templates; they define how different parts of the site will look and behave. Examples: home page, post page, search results, 404 page, etc.

WordPress: How it Works

WordPress must be installed on a **web server** to work. (This makes sense, as its purpose is to create websites!) In this course, we'll start with a local development server for learning, building, and testing. Later, we'll deploy our sites to a real, live server, so that everyone can access it including your parents and friends.

*I've included instructions on **Brightspace** that walk you through the process of setting up a development server and creating your first WordPress site.*

WordPress: Technical Details

WordPress is programmed in the **PHP language** and uses a **MySQL database.** It runs on any server that supports these two technologies. (In our case, we'll use **Apache**, which is a popular server software.)

- **PHP:** stands for **PHP Hypertext Preprocessor.** It’s a server-side scripting language that was released in 1995, but is still very popular and powers much of the web. Beginner WordPress users don't need to know how to program in PHP, but we'll learn some of the basics so we can upgrade our skills to intermediate and advanced!
- **MySQL**: a very popular database management system, using the **SQL** database language, that allows one to easily add, modify, retrieve, and delete data. A database may seem unnecessary, as we're used to storing data in simple files, but MySQL is much faster, easier, efficient, and robust against errors. We'll use a program called **phpMyAdmin** to work with our WordPress databases.

WordPress: Technical Details

WordPress **blocks**, as you may recall, are the building blocks of content for themes, posts, and pages, similar to HTML elements, but with more features and customizability. These are built using the following technologies:

- **JavaScript:** you probably know about this one; it's a scripting language that is used nowadays for pretty much everything web related, but especially client-side scripting. WordPress uses this to program the logic of blocks.
- **React:** a popular framework for using JavaScript to create user interfaces. WordPress uses React under the hood for its block-based site editor.
- **JSON:** a format for storing, structuring, and transmitting data. WordPress uses this to define and store settings and styles for a theme.
- **HTML and CSS:** we're not completely done with these - they still form the basis of marking up and styling WordPress web pages!

WordPress: Final Thoughts

While WordPress has been around for 20 years, it has gone through a period of rapid development and change within just the last 18 months. WordPress development two years ago was very different than it is today. The changes—primarily introducing a block-based editor—have made it much easier to edit themes and content. Furthermore, even less programming is required than before, making theme development even more rapid and efficient.

However, there are a few pitfalls to being a WordPress developer right now: it's harder to find resources and documentation; many bugs in the new system have not been worked out yet; and new updates, with big changes, are somewhat frequent.

But overall, to me it feels like an exciting time to be a WordPress developer!