

How to build a simple PHP website

Want to learn how to build a basic website? Previously, you could start with HTML, but for now, the best solution is PHP.

Want to learn how to build a basic website? Previously, you could start with HTML, but for now, the best solution is PHP. Although you will need some HTML knowledge to get started, PHP has become the optimal choice for building websites, both static and dynamic.

To learn PHP, the best way to get started is with a simple PHP website.

Instructions for building a PHP website from A to Z

1. Why choose PHP for website development?
2. What do you need to build a PHP website?
3. Get started with PHP: Syntax
4. Building a website: Structure
5. Start your website: PHP header
6. Put the content in the body of the PHP website
7. Use PHP footer in web page
8. Combine all the pieces together
9. Is PHP the best choice for encrypted websites?

Why choose PHP for website development?

Many options are available for web development over the years. Start with simple HTML, followed by HTML with embedded CSS or CSS file references. When dynamic web pages appear, there are two main options: ASP (later ASP.NET) and PHP.

According to statistics (as surveyed by W3Techs), PHP is much more popular, with nearly 82% of websites using it as a server-side programming language. Compare this to just under 16% using ASP.

ASP was supported until 2022. PHP (which stands for PHP Hypertext Preprocessor) has proven itself to be more successful, mainly thanks to easier integration with Linux.

Because this open source operating system runs on most web servers, this is not surprising.

What do you need to build a PHP website?

Before you begin, make sure that you have installed a plain text editor or a PHP ready development environment. You can start coding PHP with a simple tool like Windows Notepad. The examples found in this tutorial were written in Notepad ++.

You should also have a PHP web server to upload your files. This could be a remote server or a local computer with an LAMP environment (Linux, Apache, MySQL, PHP) or WAMP (Windows, Apache, MySQL, PHP) installed. If you use Windows, follow this WAMP installation guide to get started.

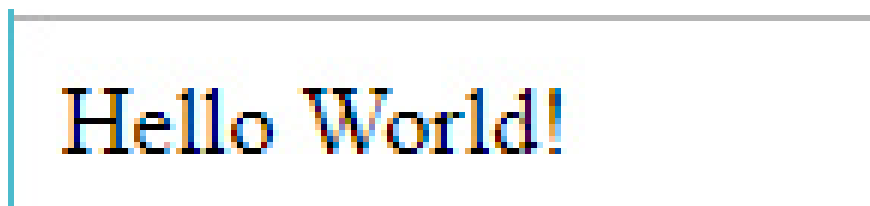
Finally, you will need an FTP program to upload your files to the web server.

Get started with PHP: Syntax

The basic syntax for PHP uses a set of braces, with each function ending with a semicolon, as follows:

For websites, almost every use case of PHP depends on the **echo** command . This command instructs the browser to export text and content in quotes. For example:

Note that HTML is also included in quotes. The output for this command will usually appear as:



Building a website: Structure

Whatever code you write for your website, you'll need to know the structure of the site before proceeding. This tutorial will show you how to create a unique page from reusable PHP files. They can be used to create additional pages or you can choose a different approach.

No matter which direction you see the site developing, take the time to jot down a quick plan on paper. You can then refer to this, to check the intended content or see which page to link it to.

The example in the article will be a basic homepage, including biographical information and some images.

For this simple PHP site, you will create a unique PHP page filled with content from 3 HTML pages. The **index.php** file you create can then be edited by modifying words and images from the original HTML file.

The code examples shown below are screenshots. You will find the original code in the GitHub repository, free for everyone to download at:

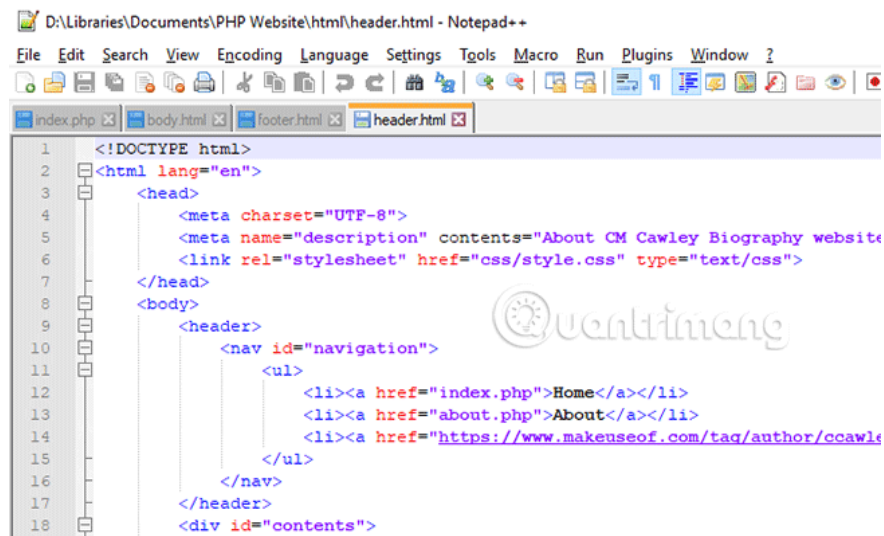
<https://github.com/atomickarma/muobasicphp>

Start your website: PHP header

To get started with the website, you will need to build 3 websites. They are based on a basic structure of header, body and footer.

As you might have guessed, the header includes header information. However, information for the browser is also included, such as the standard HTML used, along with CSS references.

Start by creating a file called **header.html** and then add the necessary header information.

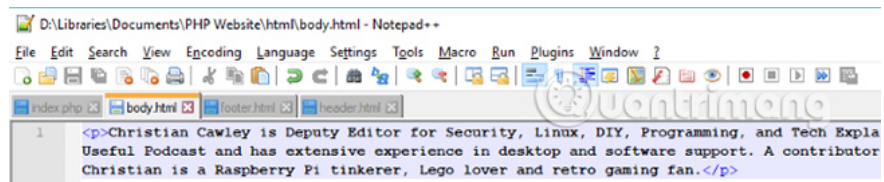


```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <meta name="description" contents="About CM Cawley Biography website">
6     <link rel="stylesheet" href="css/style.css" type="text/css">
7   </head>
8   <body>
9     <header>
10      <nav id="navigation">
11        <ul>
12          <li><a href="index.php">Home</a></li>
13          <li><a href="about.php">About</a></li>
14          <li><a href="https://www.makeuseof.com/tag/author/ccawley">
15            About the author</a></li>
16        </ul>
17      </nav>
18    </header>
19    <div id="contents">
```

In this example, the article used a basic CSS file that you will see referenced in its /css/ directory itself. This file will be called when the page loads in your browser, applying the necessary fonts and layouts.

Put the content in the body of the PHP website

Every website has a piece of content known as 'body'. This is the part of the page you are reading, what you are currently reading is the body of this page.



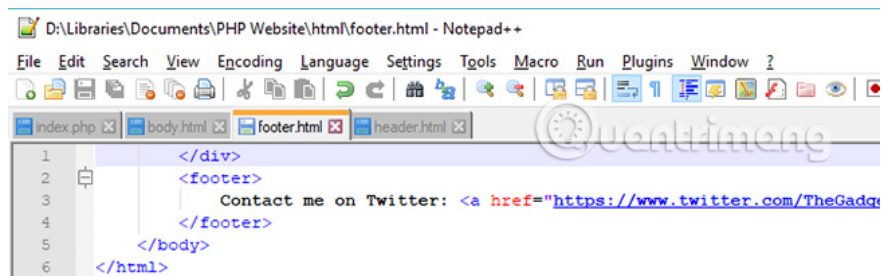
```
1 <p>Christian Cawley is Deputy Editor for Security, Linux, DIY, Programming, and Tech Explains. He is a contributor to the Useful Podcast and has extensive experience in desktop and software support. A contributor to the site, Christian is a Raspberry Pi tinkerer, Lego lover and retro gaming fan.</p>
```

Create a file named **body.html** and add the information you want included in the page. You can add anything you like.

Use PHP footer in web page

Next is the footer section of the web page. Create this section as **footer.html** and add some content. This may be copyright information or perhaps some useful links for anyone who visits your site.

Footer could be something such as:



```
D:\Libraries\Documents\PHP Website\html\footer.html - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
index.php body.html footer.html header.html
1      </div>
2      <footer>
3          Contact me on Twitter: <a href="https://www.twitter.com/TheGadg
4      </footer>
5  </body>
6 </html>
```

With the code added, save the file.

Combine all the pieces together

With 3 separate HTML files in / **html** /, you can use PHP echo to combine them into a single page.

Create a new PHP file named `index.php` with the following three lines:

Save, upload to server, then browse to **index.php**. You should see the complete web page in your browser. Remember that the actual PHP file you have open in your browser is only three lines long.

Finally, you can add a few highlights to PHP with the last line, including a copyright notice:

Copyright © CM Cawley

This message will appear in the **index.php** file under the footer. Notice how the **echo date ('Y')** displays the current year in 4 digits. You can change the display by referring to this list of W3Schools options. For example, a lowercase 'y' will display the year in 2-digit format, instead of 4.

Use CSS to position and style it, as you would any other element. Find the CSS for this project in GitHub, along with other files.

Congratulations, you have created your first simple PHP website.

Is PHP the best choice for encrypted websites?



As you know, PHP is not the only way to develop website. Many frameworks already exist for dynamic database-based web experiences, including JavaScript, related technologies and software such as Adobe Dreamweaver.

However, if you are looking to get started with web development, then be smart in getting an appreciation for the basics. If you have a good understanding of building blocks for HTML, CSS and PHP websites, you will get the job done.

Refer to the article: [9 reasons you should equip a little knowledge of HTML and CSS](#) for more details.

Hope you are succesful.

You finished reading the article "**How to build a simple PHP website**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.