Episode 1

Hello World!, Hello Server!

Introduction

Welcome to **PHP Fundamentals**. My name is **Julio Quevedo** and I will be your instructor.
The first question to ask is why PHP. PHP was created about 25 years ago, so why do I need to learn it and not the flavor of the month? Well, the answer is that PHP powers close to 80% of websites out there. That is a considerable number, making PHP an essential language to learn for developers. Some of the most popular sites like Facebook, Yahoo, WordPress, Flickr, and Wikipedia all use PHP.
In this course you will learn the basic syntax of PHP, starting with types, variables, expressions and then we'll move on to control structures, functions, and arrays. Finally, we will cover files and forms. You will also learn how to set up a server to process PHP and MySQL which we will touch briefly. By the end of this course, you will be able to write exciting and fun PHP programs confidently.
Each section of this course comes with a guide for the content and activities that have to be completed before moving on to the next section.

PHP Specialist Certification

Completing this course will make you eligible to earn the **PHP Specialist** Designation from Framework Television. This certification is designed to represent a fundamental understanding of the PHP programming language and its applicability to web development. There are three steps to earn the certification:

1. Complete all the sections in the PHP Fundamentals course. You'll complete a section by reading the section guide (You're doing that now!), watching the associated videos, and completing the code exercises.
2. Submit a correct solution to the Lab Exercise at the end of each chapter. This is critical as learning to code is not a spectator sport!
3. Ask an instructor to validate that you are ready for certification. If you have completed all the sections of the course program, you will be certified.

Upon earning certification, your certificate will be displayed and validated by Credential.net. You'll be provided with a URL that you can use as your proof of certification. Credential.net will also allow you to link the certification to your LinkedIn account— And it is highly recommended that you do so.

Note: As of the 2019 edition of our certification program, we no longer require an online exam to earn your certification. We felt that completing the lab exercises was better proof of your programming ability than a multiple choice exam.

Goals

In this section of the course your goals are:

- □ To write your first PHP script.
- □ To download and start the Apache server and PHP interpreter.
- □ To launch the script and see the result in the browser.
- □ To find small errors in your PHP code and correct them (known as debugging).

Writing your first PHP script

Let's dive right in. PHP is different from other web languages like JavaScript or Python in that PHP needs a server to run. I already have an Apache server installed on my computer so the first thing I will do is write a simple PHP script and launch it in the browser and then I will show you how to set up a server on your machine.

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>Hello World from PHP!</title>
</head>
<body>
    <?php
    echo "<h1>Hello World!</h1>";
    echo "<h2>My name is Robert</h2>";
    echo "<p>I am a master saxophonist.</p>";
    echo "I can solve a Rubik's cube in 25 seconds";
    ?>
</body>
</html>
```

It can't get any easier than that. Now this looks like an HMTL file, but it is in fact a file with a .php extension. Without this extension the interpreter won't know that is it a PHP file and the browser won't render it correctly. This is one way to embed PHP within HTML. As our projects grow bigger we will start to keep our PHP files separate just like we do with CSS or JavaScript.

Setting Up a Server

There are several ways to set up a server to run PHP on your computer. In this course, we will use a package called MAMP.
MAMP stands for:

- macOS, the operating system;
- Apache, the web server;
- MySQL, the database management system; and
- PHP, Perl, or Python, the programming languages used for web development.

Windows users don't worry it runs on Windows as well. We are going to head to the MAMP website to download what we need. There you will find two choices MAMP and MAMP PRO. We only need the free option MAMP. Click on the download button and install it on your computer.

Once you have installed MAMP, use Finder or Launchpad on a Mac, or use the search button or Windows Explorer on a Windows machine to find it. Open the application by double-clicking on it, and you should see the MAMP panel. Click on Start Servers, and you're done. Now you can write PHP programs on your computer.

Before we do that, let's be a little curious and click on the Open WebStart Page on the MAMP panel if it hasn't opened already. Sometimes it does this automatically when you start the servers. On this page, you will find the phpinfo link on the top navigation bar, click on it. Here you will find two crucial pieces of information that you need:

1. SERVER_NAME and SERVER_PORT which together they are localhost:8888
2. DOCUMENT_ROOT which on my computer is /Applications/MAMP/htdocs

It is within the htdocs folder that you must save your PHP files for the server to find it. This is very important. This is one of the most common mistakes beginner developers make. They save their PHP files in a different folder and guess what? They don't work, and then you waste precious time trying to figure out what the problem is. Now you won't have that problem. On the URL field on your browser (this where you type a web address like google.com) if you type **localhost:8888/myfile.php** you will open the file called "myfile.php" located inside the htdocs folder.

Now we are going to do one more thing. We are going back to the phpinfo page to look for two variables: the first one is called **display_errors** and the second one is called **display_startup_errors**. As you can see they are both set to Off. We need to change their value to On. How do we do that? We look for the **Loaded Configuration File**. Let's go to our code editor to open this file. On my computer, the path is **/Applications/MAMP/bin/php/php7.2.10/conf/php.ini**. It should be something similar on yours depending on where you downloaded the MAMP application. In our code editor let's look for display_errors and display_startup_errors, set them to On and that's it. That was the hardest part of this course I promise. What this will do is output to the browser any errors made in our coding which as you might imagine will be very helpful. This is recommended only in the development phase and not during production in which it could be dangerous since sensitive data may be exposed.

Code Review

Now that we have our server up and running let's go to our code editor and browser and let's examine the code.

As a first step let’s look at the code without any PHP:

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>Hello World from PHP!</title>
</head>
<body>
</body>
</html>
```

Like I mentioned before this looks just like regular HTML, but if we are going to add PHP to it, then it must be saved with a .php extension for the interpreter to recognize it.

Let’s now take a look at the PHP portion of our code:

```
<?php
    echo "<h1>Hello World!</h1>";
    echo "<h2>My name is Robert</h2>";
    echo "<p>I am a master saxophonist.</p>";
    echo "I can solve a Rubik's cube in 25 seconds";
?>
```

To include PHP in a file we enclose our code between the
opening **<?php** tag,
and the closing **?>** tag.
Everything in between those tags will be parsed by PHP, everything outside the tags will be ignored by the interpreter.

Let’s now open the developer tools in our browser. Enter CTRL-COMMAND-I in Chrome, or F12 in IE or Edge. If we now examine the html in the browser, we’ll see that the PHP tags are gone, as are the echo commands. What happened is that the PHP interpreter parsed the code and sent plain HTML to the browser.

Taking a closer look, we can see that we used the echo statement in two different ways. The first three echo lines include HTML tags enclosed in double quotes; when this is sent to the browser, it is interpreted as HTML. The last line contains a string with no HTML.

Lab Exercises

The following exercises are meant to help you practice your coding skills. They need to be submitted for you to get a certification. As stated before coding is not a spectator sport, the more you practice (and make mistakes) the more you will learn.

1 Editing Code

Most of the time a developer spends is editing code already created by other developers and not writing new code from scratch. As your first exercise open the file hello_world.php and make a few changes:

1. Change the name given to your name.
2. Change the facts given to two facts about yourself.
3. Add a

   tag outside the PHP tags and write what your favorite food is.

4. Save the file as hello_world_from_yourname.php. Use your first name in place of “your name”.

2 Using print

The print statement in PHP is another way to output content to the screen. Echo and print are similar to each other, but there are some differences. In the file, hello_world_from_yourname.php change every echo statement to print. Do you see the same output as before?

3 Debugging Code

Copy and paste the following code: hello_world_with_errors.php on your code editor. This is what is looks like:

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>Hello World from PHP!</title>
</head>
<body>
    <php>
    echo "<h1>Hello World!<h1>";
    echo "<h2>My name is Robert</h2>"
    echo "<p>I am a master saxophonist.</p>";
    echo I can solve a Rubik's cube in 25 seconds";
    ?>
</body>
</html>
```

Carefully compare it to the hello_world.php file and find the 5 errors. Most of the time debugging will involve more than just looking at two files side by side, but the point in this first chapter is to make you aware that even a small error will break your application.

In the next episode, we will take a closer look at the syntax of PHP and at the most common way to integrate it with HTML. See you then.