**Day 3**

**PHP (Personal Home Page)**

PHP is an HTML-embedded, server-side scripting language designed for web development. It is also used as a general-purpose programming language. It was created by Rasmus Lerdorf in 1994 and appeared in the market in 1995. Much of its syntax is borrowed from C, C++, and Java.

PHP codes are simply mixed with HTML codes and can be used in combination with various web frameworks. Its scripts are executed on the server. PHP code is processed by a PHP interpreter. The main goal of PHP is to allow web developer to create dynamically generated pages quickly.

A PHP file consists of texts, HTML tags and scripts with a file extension of .php, .php3, or .phtml. You can create a login page, design a form, create forums, dynamic and static websites and many more with PHP.

PHP is a popular server-side language that is particularly good for web applications. Some of the largest companies and organizations from around the world utilize PHP for their operations. A large amount of web sites and applications are powered by PHP; therefore, an understanding of the PHP language is mandatory to fully understand and accept the power behind popular frameworks (such as Laravel, CodeIgniter or Symfony), and how popular websites and applications may be handling user's data.

**Key features of PHP**

* PHP stands for Hypertext Preprocessor.
* PHP is a server-side scripting language like ASP.
* PHP supports various databases like MySQL, Oracle, Sybase, Solid, PostgreSQL, Informix etc.
* PHP is an open source software and it is free to download and use.

**Advantages of PHP**

* **Free of Cost:** PHP is open source and all its components are free to use and distribute.
* **Platform independent:** PHP is platform independent and can be run on all major operating systems.
* **Compatible with almost all servers:** PHP is compatible with almost all servers used today.
* **Secure:** PHP has multiple layers of security to prevent threats and other malicious attacks.
* **Easy to learn:** PHP has a very easy and understandable syntax. Its codes are based on C, C++ and embedded with HTML so it is very easy to learn for a programmer.
* **Performance**: Script written in PHP executes much faster then those scripts written in other languages such as JSP & ASP.
* **Open Source Software**: PHP source code is free available on the web, you can develop all the version of PHP according to your requirement without paying any cost.
* **Platform Independent**: PHP are available for WINDOWS, MAC, LINUX & UNIX operating system. A PHP application developed in one OS can be easily executed in another OS also.
* **Compatibility**: PHP is compatible with almost all local servers used today like Apache, IIS etc.
* **Embedded**: PHP code can be easily embedded within HTML tags and script.

## The Language

PHP is an interpreted language. This means that you will write code statements (lines of code) and when a page is requested, the PHP ***interpreter*** will load your PHP code, parse it and then execute it. This differs from other languages, such as Java or C#, where the source code is compiled and then executed. This is useful for web development in the fact that you do not have to re-compile your source code for trivial code changes and the changes have immediate effect on all subsequent requests.

PHP is written as standard text files with the **.php** extension. PHP files are often saved within a folder in a web server's public directory (or a ***web root* directory**). On most systems this will either be named public or public\_html. For example, if a file was saved as index.php in a web root directory, a user could access it by typing .

**http://www.example.org or http://www.example.org/index.php.**

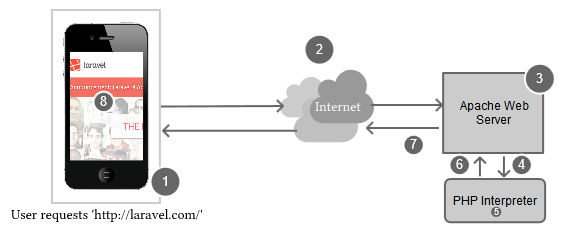
## The Request Life-cycle

What exactly is happening when a user types in the URL <http://example.org> ?

When a user types in **http://example.org** in a Web client (a browser, for instance), the client issues a GET request to the server. When Apache gets this request, it looks for a file named index.php (or index.html, remember the directory indexes from earlier?). If a file named index.php is found, Apache essentially says "Hey, this is a PHP file because it has the .php extension. I am going to give this to the PHP interpreter". After Apache decides that is is a PHP file, it gives it to the PHP interpreter. When PHP receives the file it reads through it and executes any PHP code it can find. After it is done with the file, the PHP interpreter gives the output of the code, if any, back to Apache. When Apache gets the output back from PHP, it sends that output back to a browser which renders it to the screen.

**The main goal of PHP is to generate some HTML document that a browser can render.**

However, modern applications built with client-side MVC frameworks often see the role of PHP change to just interacting with server-side data storage. Let's take another look at this process with a diagram. In this diagram, we will assume the user is going to the Laravel website at http://laravel.com/. The following figure has circled numbers that will highlight the various stages of the request. A step-by-step explanation of each step follows the figure.



**Steps**

1. The user enters **http://laravel.com** into their browser and taps/hits **enter**.
2. After the user has tapped/hit **enter**, the browser sends the page request over the Internet to the web server.
3. The web server gets the request and analyzes the request information. Apache realizes that we didn't specify a file, so it looks for a directory index and finds **index.php**.
4. Since Apache knows to send files that end with the **.php** file extension to the PHP interpreter, it asks PHP to execute the file.
5. In this step, PHP is executing the code contained in the **index.php** file from the request. During this step, PHP may interact with databases, the file system or make external API calls, amongst other things.
6. After PHP has finished executing the `index.php` file, it sends the output back to Apache.
7. Apache receives the output from PHP and sends it back over the Internet to a user's web browser. This is called the `web response`.
8. The user's web browser receives the response from the server, and renders the web page on a computer or device.

**<!DOCTYPE>**

**<html>**

**<body>**

**<?php**

**echo "<h2>Hello World PHP</h2>";**

**?>**

**</body>**

**</html>**

**PHP installation**

To install PHP use **AMP (Apache, MySQL, PHP)** software stack. It is available for all operating systems. There are many AMP options available in the market that are given below:

* **WAMP** for Windows
* **LAMP** for Linux
* **MAMP** for Mac
* **SAMP** for Solaris
* **FAMP** for FreeBSD
* **XAMPP** (Cross, Apache, MySQL, PHP, Perl) for Cross Platform: It includes some other components too such as FileZilla, OpenSSL, Webalizer, Mercury Mail etc.

**Variable declaration in PHP**

**<?php**

**$title="Gerbera";**

**$price=15;**

**$quantity=2000;**

**echo "Flower  is: $title <br/>";**

**echo "Unit Price is: $price<br/>";**

**echo "Quantitis available are : $quantity <br/>";**

**?>**

**PHP function**

**<?php**

**function sayHello($name,$age){**

**echo "Hello  $name,  you are $age years old<br/>";**

**}**

**function sqaure($n){**

**return $n\*$n;**

**}**

**sayHello("Sameer",29);**

**sayHello("Vijay",27);**

**sayHello("Rani",24);**

**echo "Square of 4 is: ". sqaure (4);**

**?>**