Contents

1 Getting started

Welcome to the exciting world of the interactive web with PHP. This chapter demonstrates how to create a dynamic development environment with a web server and the PHP engine.

Introducing PHP

The most appealing modern websites provide a customized user experience by dynamically responding to some current condition – user name, time of day, latest blog, shopping cart contents, etc. Many of these dynamic websites are created using PHP.

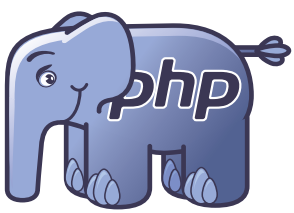
What is PHP?

PHP is a widely-used general purpose scripting language that is especially suited for web development and can be embedded into HTML. It was created by programmer Rasmus Lerdorf, as a set of scripts to maintain his website that he released as "Personal Home Page Tools (PHP Tools) version 1.0" on June 8, 1995.

The tools were extended in the version 2 release of 1997, and the name changed to become a recursive acronym "PHP: Hypertext Preprocessor" in version 3 the following year. Performance, reliability and extensibility was improved in 2000 with the release of PHP4, which was powered by the Zend engine virtual machine.

Subsequently, PHP5 was released in 2004 powered by the new Zend II engine and produced as free software by the PHP group. A planned experimental version PHP6, that intended to introduce native Unicode support throughout PHP was abandoned. The current version PHP7 was released in 2015, and is powered by the latest Zend 3 engine that offers improved performance. Today PHP is installed on over 20 million websites and 1 million web servers.

This is the official logo of the PHP project – the official online home of PHP can be found at **php.net**



This is the "elePHPant" – the mascot of the PHP project, designed by Vincent Pontier.

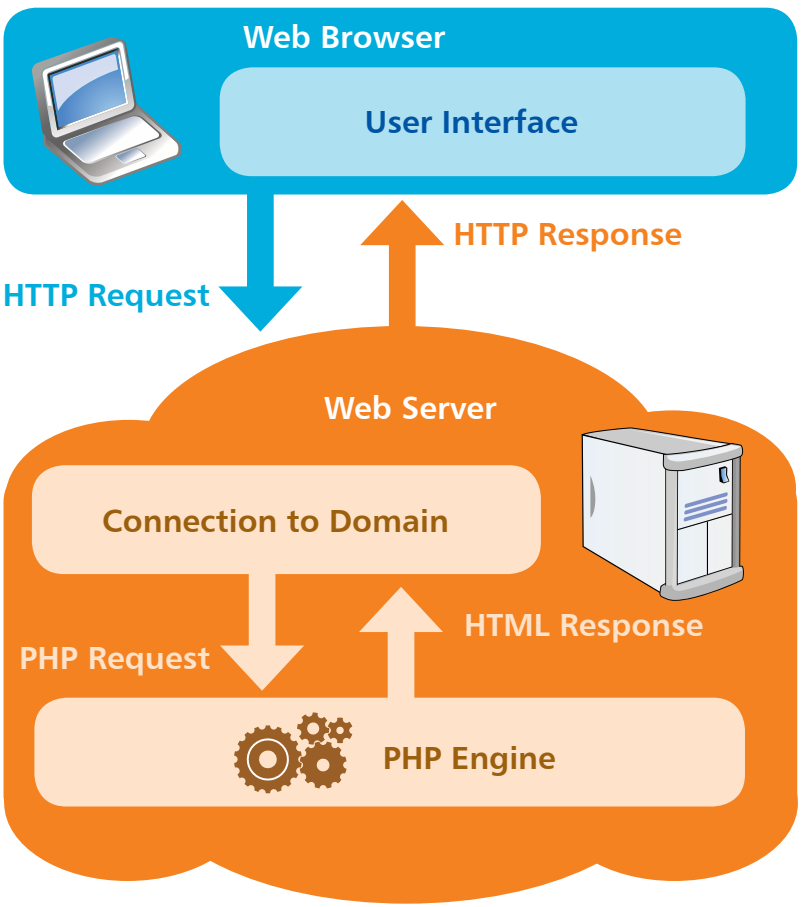
Why is PHP popular?

- PHP is extremely simple for a newcomer, but offers many advanced features for a professional programmer.
- PHP code is enclosed in special start and end processing tags that allow you to jump into and out of "PHP mode", to implement instructions within an HTML document.
- PHP code is executed on the server ("server-side"), unlike JavaScript code that is executed in the browser ("client-side"). The client receives the results of running the script without knowing what the underlying code was. Recently, server-side has become to be known as "The Cloud".

The New icon pictured above appears in this book to indicate new features introduced in PHP version 7.

Understanding The Cloud

Whenever a user asks to view a web page in their browser, it requests the page from the web server and receives the page in response, via the HTTP protocol. Where a web page contains PHP script, the web server will first call upon the PHP engine to process the code before sending the response to the web browser:



The ensuing pages describe how to create a development environment for interactive websites by installing the following server-side technologies on your own computer:

- **Web Server** – Abyss Web Server X1 Free Personal Edition
- **PHP Engine** – PHP 7.0.4

HTTP (HyperText Transfer Protocol) is the common communication standard that allows any computer connected to any web server to access files across the web.

The examples in this book are created and tested with these software versions but may require modification for other versions.

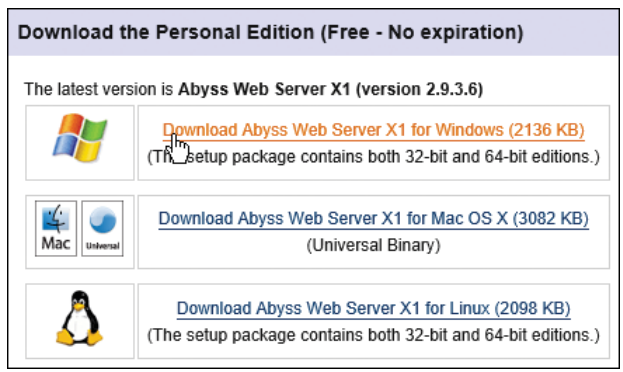
Installing the Abyss server

Abyss X1 is a free compact web server available for Windows, Mac OS X, and Linux operating systems available for download at **aprelium.com**
Despite its small footprint, Abyss supports many powerful features including dynamic content generation with server-side scripts – so is an ideal companion for PHP.

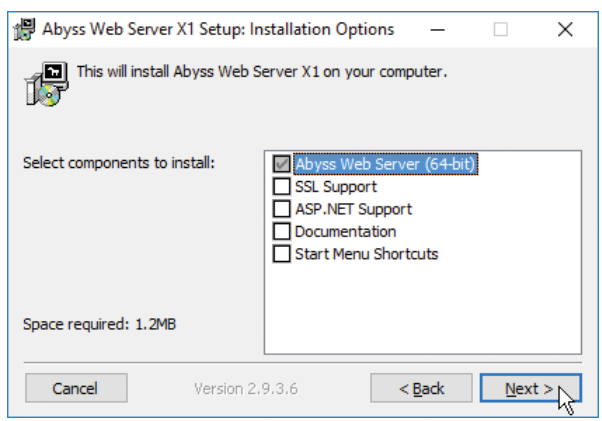
The Abyss Web Server can be installed on your own computer to provide an environment for interactive PHP website development:

Further guidance on installation of the Abyss Web Server is available at aprelium.com/abyssws/start.html

1 Download the Abyss X1 Web Server setup package for your system from **aprelium.com/abyssws/download.php**

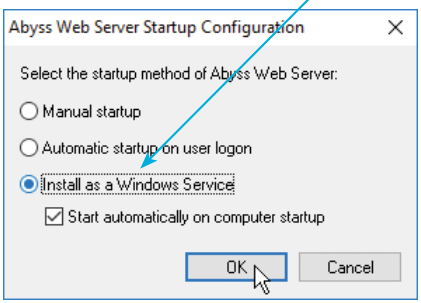


2 Run the setup installer and **Agree** the License terms, then select the Abyss Web Server component and click **Next**



The Abyss setup package for Windows is an executable file named **abwsx1.exe** that you run to install the web server.

3 Accept the suggested location of **C:\Abyss Web Server** then choose to **Install as a Windows Service**



After the installation process completes, your system's default web browser will open, displaying the Abyss Web Server Console.

4 Select your preferred language, then enter a name and password for future access to the Abyss Console

5 Now, log in using your chosen name and password to see the Abyss Console confirm the server Status as "Running"

6 Type **http://localhost** into your browser address field then hit **Enter** – to see the default Abyss "Welcome" page

If you choose the "Manual startup" option, the Abyss logo will not appear in your system tray for easy start/stop control and access to the server console. Instead, the console can be found with your browser at **http://localhost:9999** or numerically at **http://127.0.0.1:9999** ("localhost" is an alias for the IP address 127.0.0.1).

In the Abyss console, click the "Configure" button then the General icon to see the default HTTP Port is 80 and the default Documents Path (where your web pages will reside) is **/htdocs**

Installing the PHP engine

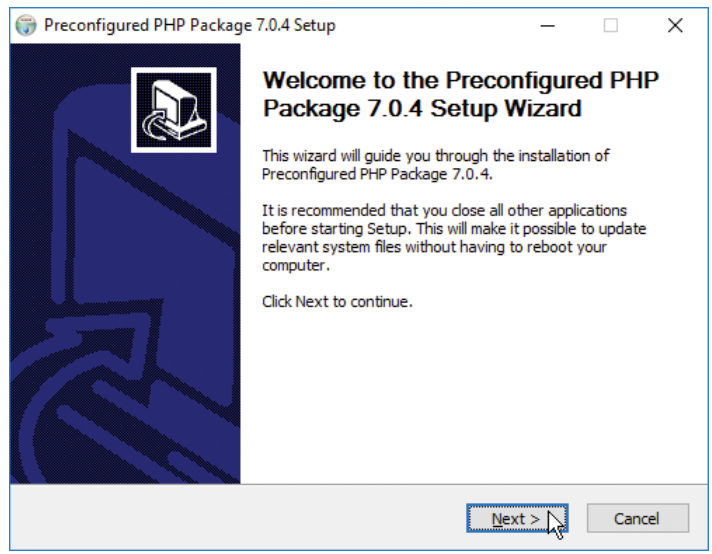
The PHP interpreter "engine", which implements PHP scripts within web pages, is available for Windows, Mac OS X and Linux operating systems as a free download at **php.net**

Additionally, a pre-configured package for the Abyss Web Server on Windows is available from **aprelium.com** and is recommended for a simple fast installation:

Further guidance on installation of PHP is available at php.net/manual/en/install.php

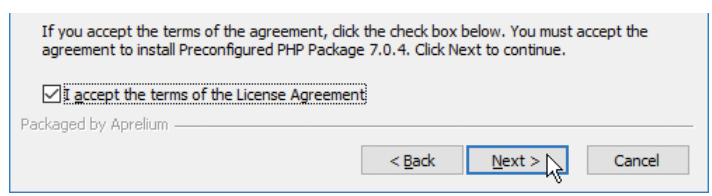
1. Download the PHP setup package for your system from **aprelium.com/downloads**

2. Run the downloaded executable file to launch the Setup Wizard then click on the **Next** button to begin

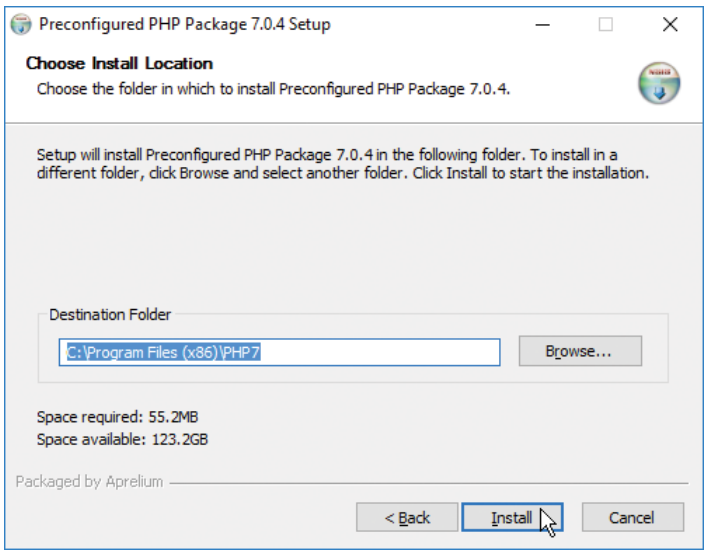


If you are installing PHP for Abyss on Windows from php.net, be sure to choose one of the VC14 Thread Safe versions – as they require fewer Windows dependencies.

3. Next, accept the License terms then click on the **Next** button to proceed with the installation

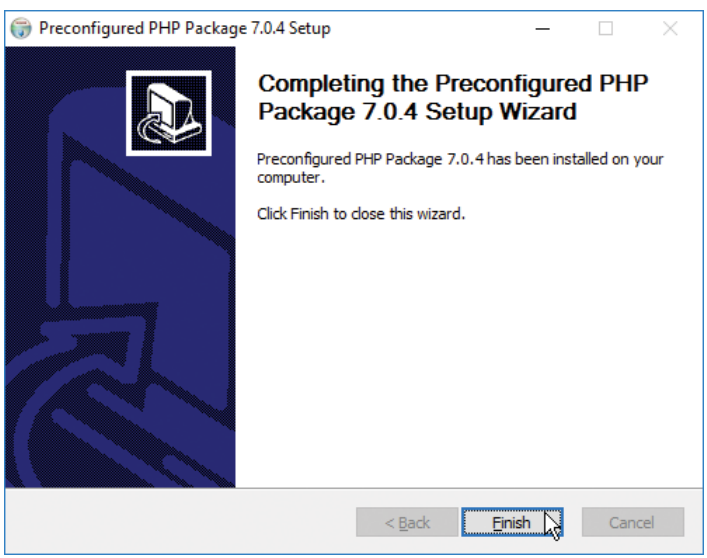


4 Accept the suggested Destination Folder located at **C:\Program Files (X86)\PHP7** then click the **Install** button



The PHP installation location will be required when configuring the Abyss Web Server to integrate with PHP – make a note of the Destination Folder.

5 Finally, after the installation completes, click on the **Finish** button to close the Setup Wizard



Following installation of PHP, the web server cannot yet execute PHP scripts until it is configured to recognize them and to find the PHP interpreter engine – all as described on the next page.

Integrating Abyss and PHP

The Abyss Web Server must be configured to recognize PHP scripts and employ the PHP interpreter when it encounters them. This is achieved in the Abyss Console by associating the file extension ".php" as being PHP scripts, and by specifying the location of the PHP engine on your system to interpret them:

1. Enter **http://localhost:9999** into your browser address field to launch the Abyss Web Server Console, then click the **Configure** button – to open the Configuration page

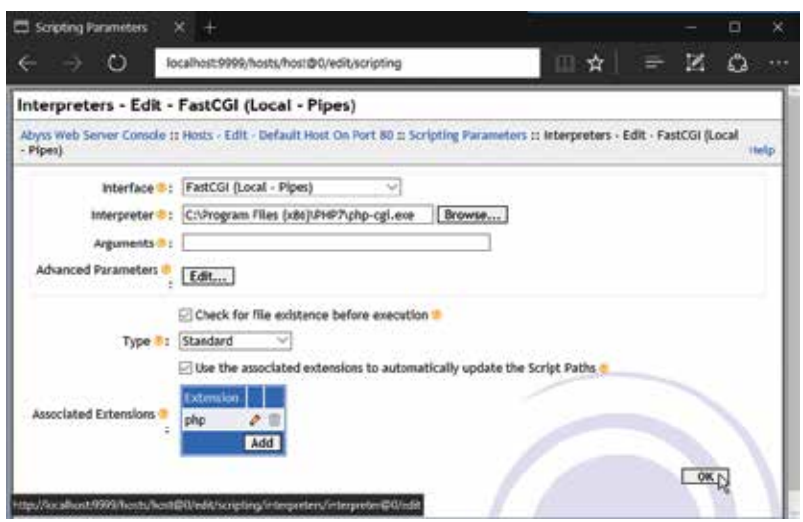
2. Click on the  **Scripting Parameters** icon – to open the Scripting Parameters page for editing

3. Ensure that the **Enable Scripts Execution** box is checked, then click the **Add** button in the Interpreters table – to open the Interpreters-Add page

4. Set the **Interface** parameter to "FastCGI (Local - Pipes)"

5. Set the **Interpreter** parameter to the PHP path location on your system of the **php-cgi.exe** file

6. Set the **Associated Extensions** parameter to "php", so your configuration should look like that shown below:



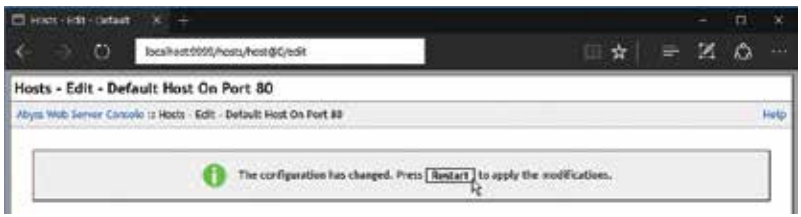
Further guidance on configuration of the Abyss Web Server is available online at **aprelium.com/abyssws/start.html**

To clarify the code examples in this book components of the PHP language are colored **blue**, programmer-specified names are **red**, numeric and string data is **black**, and comments are **green**.

The **localhost** domain name is an alias for the domain IP address of **127.0.0.1** – so the Abyss Web Server Console can alternatively be addressed as **http://127.0.0.1:9999**

7. Click the **OK** button to validate your new configuration

8. Click the **Restart** button that next appears to apply the changes you have made to the Abyss configuration



The Abyss Web Server should now be running on your system, correctly configured to recognize that documents having the **.php** file extension should be interpreted by the PHP engine. Configuration can now be tested by creating a simple PHP script for service to your web browser by Abyss:

1. Open a plain text editor and exactly type the script below
```
<?php  phpinfo() ;  ?>
```

2. Save the script as **phpinfo.php** in the Abyss document path directory, typically at **C:\Abyss Web Server\htdocs**

3. Exactly enter the location **http://localhost/phpinfo.php** into your web browser's address field to see Abyss serve up a web page containing your PHP version information

Documents can only be interpreted by the PHP engine if served up by the web server using the HTTP protocol. You cannot simply open a PHP file in your browser directly. Always use the location **http://localhost/** to serve the examples in this book.

phpinfo.php

The source code of all examples in this book is available for free download at http://www.ineasysteps.com/resource-center/downloads.

PHP scripts are case-sensitive so you must copy the listed script using lowercase characters only.

Embedding PHP script code

PHP script may be embedded within HTML documents – meaning PHP and HTML code can both happily co-exist in the same file. All embedded PHP code must be contained within **<?php** and **?>** tags so it can be readily recognized by the PHP engine for interpretation. Typically the PHP code will write content into the body section of the HTML document, which is then sent to the web browser:

hello.php

1. Launch a plain text editor and create this valid barebones HTML5 document with an empty body section

```
<!DOCTYPE HTML>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Getting Started In PHP</title>
</head>
<body>

</body>
</html>
```

2. Insert tags into the body section to contain PHP code

```
<?php

?>
```

3. Now, insert between the PHP tags a descriptive comment and a line of code to write content into the body section

```
# Write the traditional greeting.
echo '<h1>Hello World!</h1>' ;
```

Notice that the descriptive comment is enclosed between the PHP tags for information purposes only.

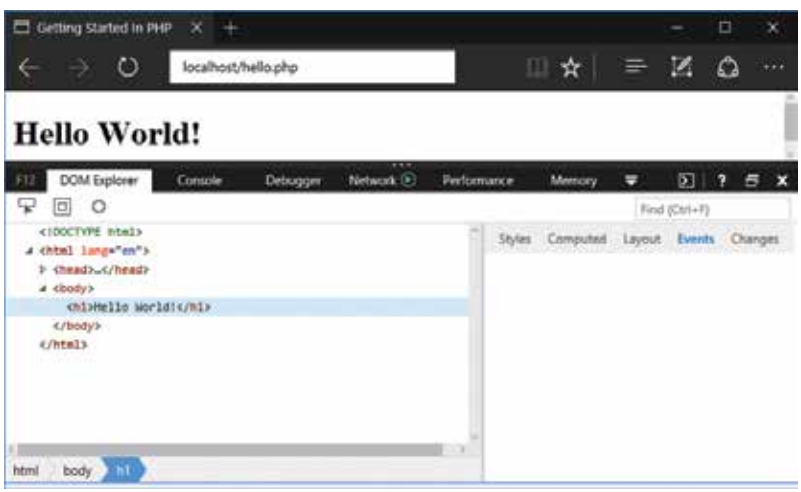
Code Writer

```
1  <!DOCTYPE HTML>
2  <html lang="en">
3  <head>
4  <meta charset="UTF-8">
5  <title>Getting Started In PHP</title>
6  </head>
7  <body>
8  <?php
9  #Write the traditional greeting.
10 echo '<h1>Hello World!</h1>' ;
11 ?>
12 </body>
13 </html>
```

4 Set the document encoding to UTF-8 format then save it as **hello.php** in the Abyss server's **/htdocs** folder

5 Next, enter the location **http://localhost/hello.php** into your web browser's address field to see Abyss serve up a web page containing content written by embedded PHP code

6 Now, use your web browser's **View Source** facility to see that PHP has written the content into the body section, including the HTML **<h1></h1>** heading tags



PHP script can be embedded in earlier versions of HTML in just the same way. Other examples in this book demonstrate embedded PHP script but do not repeatedly list the HTML code.

Windows' Notepad automatically adds a hidden "Byte Order Mark" (BOM) signature to the file, while other editors (such as Code Writer shown here) allow this to be omitted. Code Writer is a free text and code editor available from the Windows Store.

Note that the PHP **echo** instruction literally writes the entire content contained within the pair of **' '** single quote marks.

Scripting by the rules

Tag rules

When the PHP engine receives input from the web server it reads the input, from top to bottom, in a process called "parsing". During the parsing process, the PHP engine (a.k.a. parser) looks for the opening and closing **<?php** and **?>** tags and understands that the content between those tags is script code that it must interpret. Everything outside the **<?php** and **?>** tags is completely ignored, which allows PHP files to have mixed content and allows PHP code to be embedded within HTML like this:

```
<p>Ignored by PHP and displayed by the browser</p>

<?php echo 'Script code that will be parsed' ; ?>

<p>Also ignored by PHP and displayed by the browser</p>
```

Typically your embedded PHP code will always have opening and closing <?php and ?> tags around script.

In order for this to work properly it is therefore important that all your PHP script code must be enclosed between opening and closing **<?php** and **?>** tags, when embedded in an HTML file.

The only exception to this rule is when PHP script is written in a pure PHP file, which contains only code. In this case it is preferable to omit the closing **?>** tag like this:

```
<?php  echo 'Print this First' ;
       echo 'Print this Last' ;
```

Where your PHP script code intends only to insert a single string of text into an HTML document, you may optionally use the PHP short echo **<?=** and **?>** tags:

<?= 'Hello!' ; ?> is equivalent to **<?php echo 'Hello!' ; ?>**

Advanced PHP script code can also insert text only when a tested condition is met, like this:

```
<?php if ( $expression == true ) : ?>

        Insert this text only if the expression is true.

<?php else : ?>

        Otherwise insert this text.

<?php endif ; ?>
```

Conditional testing is fully explained and demonstrated later in this book, in Chapter 3.

Statement rules

Each statement within the PHP language must be terminated by a **;** semicolon character – just as each sentence in the English language must be terminated by a **.** period character. The semicolon is recognized by the parser as marking the end of an individual instruction that it must interpret. So a PHP code block containing two statements could look like this:

```
<?php echo 'First statement' ; echo 'Second statement' ; ?>
```

The closing **?>** tag of a block of PHP code automatically implies a semicolon, however, so you can optionally omit the semicolon terminating the last statement of a PHP block, like this:

```
<?php echo 'First statement' ; echo 'Second statement' ?>
```

It does no harm to terminate the last statement of a PHP block if you wish to do so.

Comment rules

It is often worthwhile adding comments to your PHP script code so it can be more easily understood by others, or by yourself when revisiting your code later. All whitespace and comments are completely ignored by the PHP parser so you can add as many comments as you like, without any adverse effect on performance.

Single-line comments may begin with a **#** hash character, or alternatively they may begin with a **//** double-slash sequence.

Multi-line (block) comments must be enclosed within **/*** and ***/** character sequences, as used in the C programming language:

```
<?php
        echo 'First statement' ; // A single-line comment.

        /* This is a multi-line comment
        containing two lines of comment. */

        echo 'Second statement' ;

    echo 'Final statement' ; # Another single-line comment.

?>
```

The // single-line comment style is also used in C++ programming and the # single-line comment style is also used in Unix/Linux BASH shell scripting.

Summary

- PHP is a scripting language that is especially suited for web development as it can be embedded in HTML
- PHP code is executed server-side on The Cloud, unlike JavaScript code that is executed client-side in the browser
- Where a web page contains PHP script, the web server will first call upon the PHP engine to process the code before sending the response to the web browser
- A local development environment can be created by installing a web server and the PHP engine on your own computer
- The **http://localhost** URL is an alias for the numerical IP address of **http://127.0.0.1**
- Access to the Abyss Web Server Console requires your user name and password, and is found at **http://localhost:9999**
- A pre-configured package for the Abyss Web Server is recommended for simple fast installation of the PHP engine
- The Web Server must be configured to recognize scripts so it will direct them to the PHP engine, by setting Interface, Interpreter, and Associated Extension parameters
- The PHP code instruction **phpinfo()** can be used to serve up a web page containing your PHP version information
- Documents containing PHP script can best be encoded using the popular UTF-8 character format
- The PHP **echo** instruction literally writes the text content contained within following quote marks
- All embedded PHP code must be enclosed within **<?php** and **?>** tags so it can be recognized by the PHP engine
- Each statement within the PHP language must be terminated by a **;** semicolon character
- Single-line comments may begin with a **#** hash character or with a **//** double-slash sequence
- Multi-line block comments must be enclosed within **/*** and ***/** character sequences