

## Unit-2:

### Introduction to PHP:

- PHP started out as a small open source project that evolved as more and more people found out how useful it was. Rasmus Lerdorf unleashed the first version of PHP way back in 1994. PHP is a recursive acronym for **"PHP: Hypertext Preprocessor"**.
- PHP is a server side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites.
- It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server.
- PHP is pleasingly zippy in its execution, especially when compiled as an Apache module on the Unix side. The MySQL server, once started, executes even very complex queries with huge result sets in record-setting time.
- PHP supports a large number of major protocols such as POP3, IMAP, and LDAP. PHP4 added support for Java and distributed object architectures (COM and CORBA), making n-tier development a possibility for the first time.
- PHP is forgiving: PHP language tries to be as forgiving as possible.
- PHP Syntax is C-Like.

### Common uses of PHP:

- PHP performs system functions, i.e. from files on a system it can create, open, read, write, and close them.
- PHP can handle forms, i.e. gather data from files, save data to a file, thru email you can send data, return data to the user.
- You add, delete, and modify elements within your database thru PHP.
- Access cookies variables and set cookies.
- Using PHP, you can restrict users to access some pages of your website.
- It can encrypt data.

### Characteristics of PHP

Five important characteristics make PHP's practical nature possible:

- Simplicity
- Efficiency
- Security
- Flexibility
- Familiarity

### ***"Hello World" Script in PHP:***

To get a feel for PHP, first start with simple PHP scripts. Since "Hello, World!" is an essential example, first we will create a friendly little "Hello, World!" script.

As mentioned earlier, PHP is embedded in HTML. That means that in amongst your normal HTML (or XHTML if you're cutting-edge) you'll have PHP statements like this:

```
<html>
<head>
<title>Hello World</title>
<body>
    <?php echo "Hello, World!";?>
</body>
</html>
```

It will produce following result:

```
Hello, World!
```

If you examine the HTML output of the above example, you'll notice that the PHP code is not present in the file sent from the server to your Web browser. All of the PHP present in the Web page is processed and stripped from the page; the only thing returned to the client from the Web server is pure HTML output.

All PHP code must be included inside one of the three special markup tags that are recognised by the PHP Parser.

```
<?php PHP code goes here ?>

<?    PHP code goes here ?>

<script language="php"> PHP code goes here </script>
```

Most common tag is the `<?php...?>` and we will also use same tag in our tutorial.

From the next chapter we will start with PHP Environment Setup on your machine and then we will dig out almost all concepts related to PHP to make you comfortable with the PHP language.

In order to develop and run PHP Web pages three vital components need to be installed on your computer system.

- **Web Server** - PHP will work with virtually all Web Server software, including Microsoft's Internet Information Server (IIS) but then most often used is freely available Apache Server. Download Apache for free here: <http://httpd.apache.org/download.cgi>
- **Database** - PHP will work with virtually all database software, including Oracle and Sybase but most commonly used is freely available MySQL database. Download MySQL for free here: <http://www.mysql.com/downloads/index.html>
- **PHP Parser** - In order to process PHP script instructions a parser must be installed to generate HTML output that can be sent to the Web Browser. This tutorial will guide you how to install PHP parser on your computer.

## The problem with other Technologies

- ✓ **PHP** is an open-source page scripting/templating system that is very similar to JSP and ASP.
- ✓ It defines its own scripting language, which looks and feels a lot like Perl.
- ✓ JSP uses Java as its scripting language (although some implementations support JavaScript, such as Caucho). ASP uses VBScript.

- ✓ PHP is very popular -- it is used on over a million web sites -- but its main advantage (IMHO) seems to be that the language, being more "scripty" and Perl-like, is less intimidating to the great unwashed mass of HTML monkeys and hackers. In the long run, JSP and Java provide a more powerful system.

### ***Here is a list of reasons why JSP is better than PHP:***

- Anything you can do with PHP, you can do with JSP; the reverse is not true
- JSP is much more powerful, since it has access to all the Java libraries. PHP only has access to PHP libraries
- JSP is Object-Oriented, so leads to cleaner code that's easier to debug, maintain, and improve. (PHP also allows objects, but the object model is more primitive, and most scripted pages ignore PHP objects and just use normal variables.)
- The equivalent syntax in JSP is just as simple to learn, so you can get up and running just as quickly -- that is, there's no **extra** startup cost in using Java, at least not a significant one
- Java programmers (as opposed to 15-year-old hackers or HTML monkeys) appreciate the importance of a clean language with complex OO data structures and strong typing
- With JSP, if the code inside a page gets too big, or if you want to use it elsewhere, you can cut it out, make it into a Java class, and invoke it from anywhere in your application (even not from a page). With PHP, you're stuck inside the HTML box.
- JSP's concept of state management and persistence is more explicit and powerful than PHP's. With JSP, you can specify whether a variable persists for the page, the request, the session, or the application (or if it's just local to the function). The JSP engine automatically does the right thing with cookies so you have access to the variable on later requests. With PHP, you just have "global" and "not global", you don't have automatic session management, and have to do your state thing manually with cookies or hidden variables.

#### **PHP:**

- ✓ fast to develop (only small webs)
- ✓ no security.
- ✓ Nodebugging.

#### **JSP:**

- ✓ security
- ✓ reuse: You can make frontend in swing too.
- ✓ speed: Once compiled, then much faster.
- ✓ persistence
- ✓ thread-save
- ✓ load: It can serve more users.
- ✓ debug: classes could be debuged well.

#### **JSP:**

JSP is easy to learn and allow developers to quickly produce web sites and applications in an open and standard way. JSP is based in Java, an object oriented language. JSP offers a robust platform for web

development.

Main reasons to use JSP are:

- Multi platform
- component reuse by using Java Beans
- It also incorporates EJB
- it has every advantages that java provides

### **Servlet:**

A servlet is a java component that can be plugged into Java enabled web server to provide custom services. Servlets are designed to work with in request response processing model. In a Request/Response model, a client sends a request to a server and the server responds by sending back a reply message can come in the form of an

- HTTP URL
- FTP URL
- a custom protocol

### **JSP VERSUS SERVLETS**

- ✓ Before the advent of JSP, the most used java technology that could generate dynamic web page content was Java Servlets. Because JSPs eventually are compiled into Java servlets, you can do as much with JSPs as you can do with Java servlets.
- ✓ However Coding JSPs is easier than coding Java servlets. With Jsp, you place static text by coding HTML tags as opposes to java servlets ,in which you place static text by coding a many println statements. With JSPs,
- ✓ you can change static text by changing HTML;and with servlets you can change static text by modifying a java servlet( this also adds to the compile/ debug cycle)
- ✓ Combining Java Code and java cells into an HTML page is more easy than using straight java code in a Servlet. Jsp syntax gives you a shortcut for coding dynamic Web pages, typically requiring much less code than java Servlet syntax..

## **Downloading, Installing and Configuring PHP:**

### **Downloading**

- ✓ Although PHP comes bundled with most Linux distributions nowadays, you should download the latest stable version from the PHP Web site.
- ✓ To decrease download time, choose from the approximately 100 mirrors residing in more than 50 countries,a list of which is available here: <http://www.php.net/mirrors.php>.
- ✓

## Configuring PHP:

### Configuring PHP at Build Time on Linux:

- ✓ Building PHP as described earlier in the chapter is sufficient for getting started; however, you should keep in mind many other build-time options are at your disposal.
- ✓ You can view a complete list of configuration flags (there are more than 200) by executing the following:

```
%>./configure --help
```

- ✓ To make adjustments to the build process, you just need to add one or more of these arguments to PHP's configure command, including a value assignment if necessary.
- ✓ **For example**, suppose you want to enable PHP's FTP functionality, a feature not enabled by default. Just modify the configuration step of the PHP build process like so:  

```
%>./configure --with-apxs2=/usr/local/apache2/bin/apxs --enable-ftp
```
- ✓ As another example, suppose you want to enable PHP's Java extension. Just reconfigure PHP like so:  

```
%>./configure --with-apxs2=/usr/local/apache2/bin/apxs \>--enable-java=[JDK-INSTALL-DIR]
```
- ✓ One common point of confusion among beginners is to assume that simply including additional flags will automatically make this functionality available via PHP.
- ✓ This is not necessarily the case. Keep in mind that you also need to install the software that is ultimately responsible for enabling the extension support. In the case of the Java example.

### Customizing the Windows Build

- ✓ A total of 45 extensions are bundled with PHP 5.1 and 5.2, a number that was pared to
- ✓ 35 extensions with the current alpha version of PHP 6. However, to actually use any of these extensions, you need to uncomment the appropriate line within the php.ini file.  
 For example, if you'd like to enable PHP's XML-RPC extension, you need to make a few minor adjustments to your php.ini file:
  1. Open the php.ini file and locate the extension\_dir directive and assign it  
**C:\php\ext\**. If you installed PHP in another directory, modify this path accordingly.
  2. Locate the line; extension=**php\_xmlrpc.dll**. Uncomment this line by removing the preceding Semicolon. Save and close the file.
  3. Restart the Web server and the extension is ready for use from within PHP. Keep in mind that some Extensions have additional configuration directives that may be found later in the php.ini file.

### Run-Time Configuration

- ✓ It's possible to change PHP's behavior at run time on both Windows and Linux through the **php.ini** file.
- ✓ This file contains a myriad of configuration directives that collectively control the behavior of each product.
- ✓ The remainder of this chapter focuses on PHP's most commonly used configuration directives, introducing the purpose, scope, and default value of each.

## Managing PHP's Configuration Directives:

- ✓ **The php.ini File:** The PHP distribution comes with two configuration templates, php.ini-dist and php.ini-recommended. You'll want to rename one of these files to php.ini and place it in the location specified by the **PHPIniDir** directive found in Apache's **httpd.conf** file.

## The Apache httpd.conf and .htaccess Files:

- ✓ When PHP is running as an Apache module, you can modify many of the directives through either the httpd.conf file or the .htaccess file. This is accomplished by prefixing directive/value assignment with one of the following keywords:
  - **php\_value:** Sets the value of the specified directive.
  - **php\_flag:** Sets the value of the specified Boolean directive.
  - **php\_admin\_value:** Sets the value of the specified directive. This differs from **php\_value** in that it cannot be used within an .htaccess file and cannot be overridden within virtual hosts or .htaccess.
- ✓ **php\_admin\_flag:** Sets the value of the specified directive. This differs from **php\_admin\_value** in that it cannot be used within an .htaccess file and cannot be overridden within virtual hosts or htaccess.

## Programming in a Web environment:

### Choosing a Code Editor:

#### 1. Adobe Dreamweaver CS3:

- ✓ Formerly known as Macromedia Dreamweaver MX, Adobe's Dreamweaver CS3 is considered by many to be the ultimate Web designer's toolkit.
- ✓ Intended to be a onestop application, Dreamweaver CS3 supports all of the key technologies, such as Ajax, CSS, HTML, JavaScript, PHP, and XML, which together drive cutting-edge Web sites.

#### 2. Notepad++:

- ✓ Notepad++ is a mature open source code editor and avowed Notepad replacement available for the Windows platform. Translated into 41 languages,
- ✓ Notepad++ offers a wide array of convenient features one would expect of any capable IDE, including the ability to bookmark specific lines of a document for easy reference; syntax, brace, and indentation highlighting; powerful search facilities;

#### 3. PDT (PHP Development Tools):

- ✓ The PDT project (<http://www.eclipse.org/pdt/>) is currently seeing quite a bit of momentum

#### 4. Zend Studio:

- ✓ Zend Studio is far and away the most powerful PHP IDE of all commercial and open source offerings available today
- ✓ Facilities integrating code with popular databases such as MySQL, Oracle, PostgreSQL, and SQLite are also offered, in addition to the ability to execute SQL queries and view and manage database schemas and data.



- ✓ Zend Studio ([http://www.zend.com/products/zend\\_studio/](http://www.zend.com/products/zend_studio/)) is available for the Windows, Linux, and Mac OS X platforms in two editions: standard and professional.

## Choosing a Web Hosting Provider:

- ✓ Generally speaking, hosting providers can be broken into three categories:

### 1. Dedicated server hosting:

- ✓ Dedicated server hosting involves leasing an entire Web server, allowing your Web site full reign over server CPU, disk space, and memory resources, as well as control over how the server is configured.

### 2. Shared server hosting:

- ✓ Shared server hosting is likely the ideal solution. Shared hosting providers capitalize on these factors by hosting numerous Web sites on a single server and using highly automated processes to manage system and network resources, data backups, and user support.

### 3. Virtual private server hosting:

- ✓ A virtual private server blurs the line between a dedicated and shared server, providing each user with a dedicated operating system and the ability to install applications and fully manage the server by way of *virtualization*.
- ✓ Virtualization provides a way to run multiple distinct operating systems on the same server.

## The anatomy of a PHP Page(or) Embedding PHP Code in Your Web Pages

- ✓ One of PHP's advantages is that you can embed PHP code directly alongside HTML. For the code to do anything, the page must be passed to the PHP engine for interpretation

**There are four delimitation variants:**

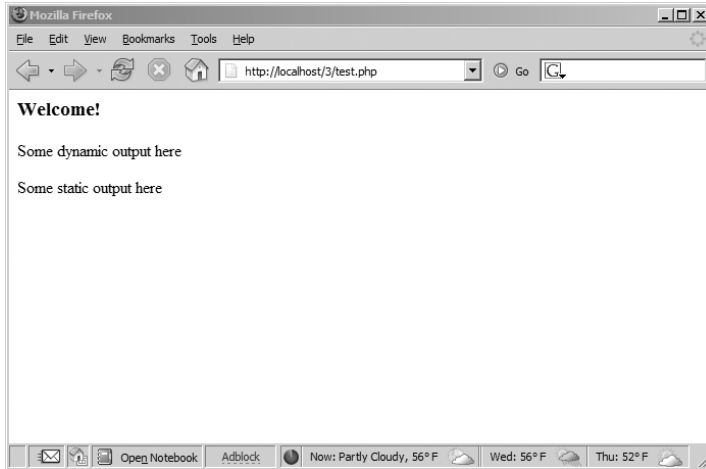
### 1. Default Syntax

The default delimiter syntax opens with **<?php** and concludes with **?>**, like this:

**Eg:**

```
<h3>Welcome!</h3>
<?php
echo "<p>Some dynamic output here</p>";
?>
<p>Some static output here</p>
```

**Output:**



## 2. Short-Tags: Syntax:

```
<?
print "This is another PHP example.";
?>
```

## 3. Script: Syntax:

```
<script language="php">
print "This is another PHP example.";
</script>
```

## 4. ASP Style:

- ✓ Microsoft ASP pages employ a similar strategy, delimiting static from dynamic syntax by using a predefined character pattern, opening dynamic syntax with **<%**, and concluding with **%>**.

### Syntax:

```
<%
print "This is another PHP example.";
%>
```

## ***Embedding Multiple Code Blocks:***

- ✓ You can escape to and from PHP as many times as required within a given page. For instance, the following example is perfectly acceptable:

```
<html>
<head>
<title><?php echo "Welcome to my Web site!";?></title>
</head>
<body>
<?php
$date = "July 26, 2007";
?>
<p>Today's date is <?=$date;?></p>
</body>
</html>
```



```
<?php
# Title: My PHP program
# Author: Jason
echo "This is a PHP program";
?>
```

## Commenting Your Code:

### Single-Line C++ Syntax

- ✓ Comments often require no more than a single line (*//*)

```
<?php
// Title: My first PHP script
// Author: Jason
echo "This is a PHP program";
?>
```

### Shell Syntax:

- ✓ PHP also supports an alternative to the C++-style single-line syntax, known as *shell syntax*, which is prefaced with a hash mark (#).

### Multiple-Line C Syntax ☹(/\* ..... \*/).

```
<?php
/*
Title: My PHP Program
Author: Jason
Date: July 26, 2007
*/
?>
```

-----*All the best*-----