CH-6: PHP

TOPIC 1: Introduction to Server Side Scripting

Scripts in Service Now fall into two categories:

- Client-side
- Server-side

This module is about server-side scripting. Server-side scripts execute on the ServiceNow server or database. Scripts in Service Now can do many, many things. Examples of things server-side scripts can do include:

- Update record fields when a database query runs
- Set field values on related records when a record is saved
- Manage failed log in attempts
- Determine if a user has a specific role
- Send email
- Generate and respond to events
- Compare two dates to determine which comes first chronologically
- Determine if today is a weekend or weekday
- Calculate the date when the next quarter starts
- Log messages
- Initiate integration and API calls to other systems
- Send REST messages and retrieve results

TOPIC 2: Introduction and Basic Syntax of PHP

What is PHP?

- HP is an acronym for "PHP: Hypertext Preprocessor"
- PHP is a widely-used, open source scripting language
- PHP scripts are executed on the server
- PHP is free to download and use
- **PHP** is a general-purpose server-side scripting language originally designed for web development to produce dynamic websites.
- PHP scripts execute on web server and serve WebPages to user on request.
- PHP was originally created by RasmusLerdorf in 1994. Programmer RasmusLerdorf initially created a set of C scripts he called "Personal Home Page Tools" to maintain his personal homepage. The scripts performed tasks such as displaying his résumé and recording his web-page traffic.
- These were released and extended to include a package called the Form Interpreter (PHP/FI). While PHP originally stood for "Personal Home Page", it is now said to stand for "PHP: Hypertext Preprocessor", a recursive acronym.
- PHP code is embedded into the HTML source document and interpreted by a web server with a PHP processor module, which generates the web page document. It also has evolved to include a

- command-line interface capability and can be used in standalone graphical applications.PHP can be deployed on most web servers and as a standalone interpreter, on almost every operating system and platform free of charge.
- In 1997 ZeevSuraski and AndiGutmans along with Rasmus rewrite PHP and released PHP version 3.0 in June 1998.After this release PHP becomes so much popular.
- The PHP version 4.0 was launched in May 2000.This version includes session handling, output buffering, a richer cire language and support for wide variety of web server platforms.
- The PHP 5.0 version released in 2004 with object oriented programming concept.

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, through email you can send data, return data to the user.
- You add, delete, modify elements within your database through 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

Basic PHP Syntax

A PHP script can be placed anywhere in the document.
A PHP script starts with <?php and ends with ?>:

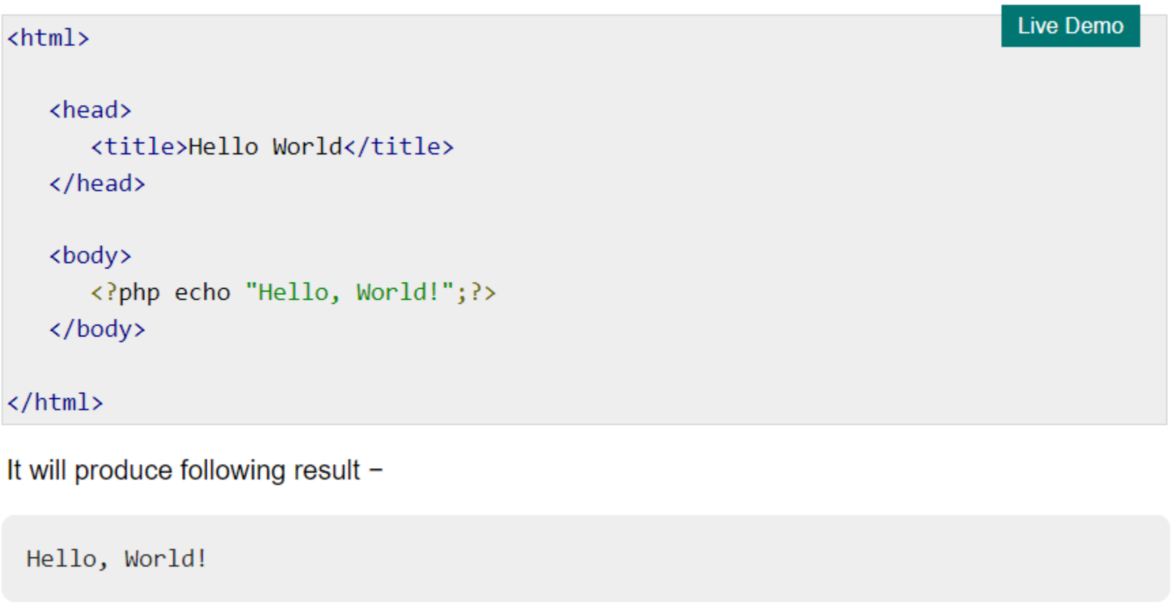
```
<?php
// PHP code goes here
?>
```

The default file extension for PHP files is ".php".
A PHP file normally contains HTML tags, and some PHP scripting code.

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>
   </head>

   <body>
      <?php echo "Hello, World!";?>
   </body>

</html>
```

It will produce following result −

```
Hello, World!
```

TOPIC 3: String Processing and Regular expression

PHP String Functions

strlen() - Return the Length of a String

The PHP `strlen()` function returns the length of a string.

```
<!DOCTYPE html>
<html>
<body>

<?php
echo strlen("Hello world!");
?>

</body>
</html>
```

str_word_count() - Count Words in a String

The PHP `str_word_count()` function counts the number of words in a string.

```
<!DOCTYPE html>
<html>
<body>

<?php
echo str_word_count("Hello world!");
?>

</body>
</html>
```

strrev() - Reverse a String

The PHP `strrev()` function reverses a string.

```
<!DOCTYPE html>
<html>
<body>

<?php
echo strrev("Hello world!");
?>

</body>
</html>
```

strpos() - Search For a Text Within a String

The PHP `strpos()` function searches for a specific text within a string. If a match is found, the function returns the character position of the first match. If no match is found, it will return FALSE.

```
<!DOCTYPE html>
<html>
<body>

<?php
echo strpos("Hello world!", "world");
?>

</body>
</html>
```

str_replace() - Replace Text Within a String

The PHP `str_replace()` function replaces some characters with some other characters in a string.

```
<!DOCTYPE html>
<html>
<body>

<?php
echo str_replace("world", "Dolly", "Hello world!");
?>
```

```
</body>
</html>
```

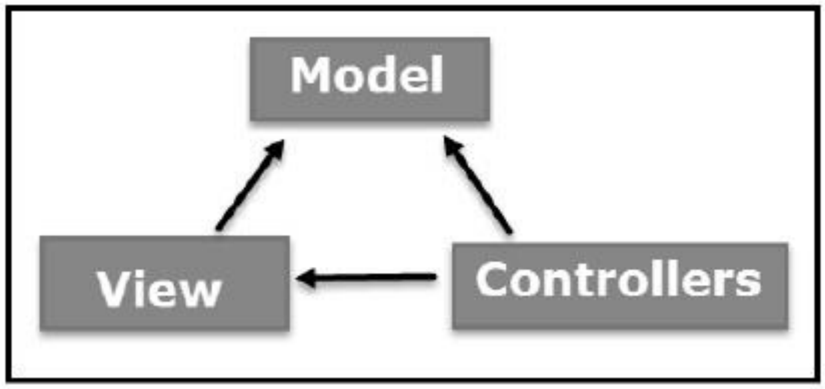
PHP - Regular Expressions

- Regular expressions are nothing more than a sequence or pattern of characters itself. They provide the foundation for pattern-matching functionality.
- Using regular expression you can search a particular string inside a another string, you can replace one string by another string and you can split a string into many chunks.
- PHP offers functions specific to two sets of regular expression functions, each corresponding to a certain type of regular expression. You can use any of them based on your comfort.
    - POSIX Regular Expressions
    - PERL Style Regular Expressions
- The structure of a POSIX regular expression is not dissimilar to that of a typical arithmetic expression: various elements (operators) are combined to form more complex expressions.
- The simplest regular expression is one that matches a single character, such as g, inside strings such as g, haggle, or bag.
- Lets give explanation for few concepts being used in POSIX regular expression. After that we will introduce you with regular expression related functions.
- **SLIDE NO. – 12 TO 23 IN PHP PPT**

TOPIC 4: Web Development Frameworks

MVC Components

Following are the components of MVC −



Model

The Model component corresponds to all the data-related logic that the user works with. This can represent either the data that is being transferred between the View and Controller components or any other business logic-related data. For example, a Customer object will retrieve the customer information from the database, manipulate it and update it data back to the database or use it to render data.

View

The View component is used for all the UI logic of the application. For example, the Customer view will include all the UI components such as text boxes, dropdowns, etc. that the final user interacts with.

Controller

Controllers act as an interface between Model and View components to process all the business logic and incoming requests, manipulate data using the Model component and interact with the Views to render the final output. For example, the Customer controller will handle all the interactions and inputs from the Customer View and update the database using the Customer Model. The same controller will be used to view the Customer data.

A PHP framework is a platform to create PHP web applications. PHP frameworks provide code libraries for commonly used functions, cutting down on the amount of original code you need to write.
There are many good reasons for using PHP frameworks as opposed to coding from scratch.

1. Faster Development

Because PHP frameworks have built-in libraries and tools, the time required for development is less.

For example, the CakePHP framework has the Bake command-line tool which can quickly create any skeleton code that you need in your application.

Several popular PHP frameworks have the PHP Unit library integrated for easy testing.

2. Less Code to Write

Using functions that are built-in to the framework means that you don’t need to write so much original code.

3. Libraries for Common Tasks

Many tasks that developers will need to do within web apps are common ones. Examples are form validation, data sanitization, and CRUD operations (Create, Read, Update, and Delete). Rather than having to write your own functions for these tasks, you can simply use the ones that are part of the framework.

4. Follow Good Coding Practices

PHP frameworks usually follow coding best practices. For example, they divide code neatly into a number of directories according to function.

They force you to organize code in a cleaner, neater, and more maintainable way.

Frameworks also have their own naming conventions for entities which you should follow.

5. More Secure Than Writing Your Own Apps

There are many PHP security threats including cross-site scripting, SQL injection attacks, and cross-site request forgery. Unless you take the right steps to secure your code, your PHP web apps will be vulnerable.

Using a PHP framework is not a substitute for writing secure code, but it minimizes the chance of hacker exploits. Good frameworks have data sanitization built-in and defences against the common threats mentioned above.

6. Better Teamwork

Projects with multiple developers can go wrong if there isn't clarity on:

- Documentation
- Design decisions
- Code standards

Using a framework sets clear ground rules for your project. Even if another developer isn't familiar with the framework, they should be able to quickly learn the ropes and work collaboratively.

7. Easier to Maintain

PHP Frameworks encourage refactoring of code and promote DRY development (Don't Repeat Yourself). The resulting leaner codebase needs less maintenance.

You also don't have to worry about maintaining the core framework, as that's done for you by the developers.

Framework differences

- Every framework is created differently. Symfony works on reusable components and provides the best modularity. Symfony also utilizes the model and controller for developing a web application, which may look rusty for many new developers, but it works. Also, Symfony is a good example of the modular framework. You can use the 30 components provided by Symfony in your project in a modular fashion.
- Yii uses an MVC framework. (Symfony does provide support for MVC, which is discussed in more detail in Is

Symfony2 a MVC framework on the blog.sznapka.pl site.)

- Symfony can be used for rapid development and complex projects. Even though there is a debate on which framework is better for complex projects, Symfony does showcase brilliant complexity handling compared to other frameworks. Yii also utilizes components, but is not as modular as Symfony. Laravel doesn't provide a modular approach as sharp as the other two frameworks.
- If you are looking for a modular framework, go for Symfony. Otherwise, Laravel and Yii are both great choices.

PHP Form Handling

We can create and use forms in PHP. To get form data, we need to use PHP superglobals $_GET and $_POST.

The form request may be get or post. To retrieve data from get request, we need to use $_GET, for post request $_POST.

PHP Get Form

Get request is the default form request. The data passed through get request is visible on the URL browser so it is not secured. You can send limited amount of data through get request.

Let's see a simple example to receive data from get request in PHP.

File: form1.html

```
<form action="welcome.php" method="get">
Name: <input type="text" name="name"/>
<input type="submit" value="visit"/>
</form>
```

File: welcome.php

```
<?php
$name=$_GET["name"];//receiving name field v
alue in $name variable
echo "Welcome, $name";
?>
```

PHP Post Form

Post request is widely used to submit form that have large amount of data such as file upload, image upload, login form, registration form etc.

The data passed through post request is not visible on the URL browser so it is secured. You can send large amount of data through post request.

Let's see a simple example to receive data from post request in PHP.

File: form1.html

```
<form action="login.php" method="post">
<table>
<tr><td>Name:</td><td> <input type="text" name
="name"/></td></tr>
```

```
<tr><td>Password:</td><td> <input type="passwo
rd" name="password"/></td></tr>
<tr><td colspan="2"><input type="submit" value="
login"/>  </td></tr>
</table>
</form>
```

File: login.php

```
<?php
$name=$_POST["name"];//receiving name field
value in $name variable
$password=$_POST["password"];//receiving pas
sword field value in $password variable

echo "Welcome: $name, your password is: $pas
sword";
?>
```

IN PPT SLIDE NO. 28 ONWARDS

TOPIC 5: Difference between PHP and HTML

PHP	HTML
PHP is a server-side programming language.	HTML is a client-side scripting language.

PHP is used in backend development, which interacts with databases to retrieve, store, and modify the information.	HTML is used in frontend development, which organizes the content of the website.
PHP is used to create a dynamic website. The output will depend on the browser.	HTML is used to create a static website. The output of static website remains the same on each time.
PHP can manipulate the data.	It cannot manipulate the data.
PHP code executes on web servers like Apache web server, IIS web server.	HTML code executes on web browsers like Chrome, Internet Explorer, etc.
PHP is scripting language.	HTML is a markup language.
PHP7.3 is the latest version of PHP.	HTML5.2 is the latest version of HTML.
PHP is also easy to learn but not as simple as HTML.	HTML is easy to learn. It can easily learn in a very short time.
PHP files save with .php extension.	HTML files save with .html extension.