

Architecting Web Applications Using PHP

Trainer Guide

For Aptech Centre Use Only

Architecting Web Applications Using PHP

Trainer Guide

© 2022 Aptech Limited

All rights reserved.

No part of this book may be reproduced or copied in any form or by any means – graphic, electronic or mechanical, including photocopying, recording, taping, or storing in information retrieval system or sent or transferred without the prior written permission of copyright owner Aptech Limited.

All trademarks acknowledged.

APTECH LIMITED

Contact E-mail: ov-support@onlinevarsity.com

Edition 1 - 2022



Preface

Hypertext Preprocessor (PHP) is a server-side, open source Web scripting language. It is used for developing dynamic Web pages. PHP supports powerful features for form handling. PHP uses cookies to store information on the user's local computer and sessions to store user information on the Web server.

PHP 3.0 was the first version of PHP developed by Andi Gutmans and Zeev Surakshi in the year 1997. This version supported different databases, protocols, and Application Programming Interfaces (APIs). The developers used the extensibility feature of PHP 3.0 to add new features to it and enhance its functionality. In addition, PHP 3.0 also provided Object-Oriented Programming (OOP) support. In later years, several versions with improved and new features were released. This book covers PHP 8.0.13 and its various features. The book explains essentials of Web application development with PHP and Laravel.

The faculty/trainer should teach the concepts in the theory class using the slides. This Trainer's Guide will provide guidance on the flow of the module and also provide tips and additional examples wherever necessary. The trainer can ask questions to make the session interactive and also to test the understanding of the students.

The knowledge and information in this book is the result of the concentrated effort of the Design Team, which is continuously striving to bring to you the latest, the best and the most relevant subject matter in Information Technology. As a part of Aptech's quality drive, this team does intensive research and curriculum enrichment to keep it in line with industry trends and learner requirements.

Contents

Architecting Web Applications Using PHP

Session 1: Introduction to PHP

Session 2: PHP Basics and Syntax

Session 3: PHP Data Types and Strings

Session 4: Variables and Operators in PHP

Session 5: Conditional and Flow Control Statements in PHP and Arrays

Session 6: Form Handling

Session 7: Working with Functions in PHP

Session 8: Cookies and Sessions Management in PHP

Session 9: Database Management in PHP

Session 10: Advanced Features of PHP

Session 11: File Handling and Exception Handling in PHP

Session 12: Object Oriented Concepts in PHP

Session 13: Methods in PHP and Other OOP Concepts

Session 14: PHP Web Concepts

Session 15: PHP – AJAX

Session 1 – Introduction to PHP

1.1 Pre-Class Activities

Before beginning the session, you should fully familiarize yourself with its topics. Prepare a question or two that will be a key point to relate the current session objectives.

1.1.1 Teaching Skills

To teach this session, you should be well-versed with the concept of internationalization and design patterns in Java. You must also be familiar with the concept of localization.

You must use the given images and teach the concepts in the theory class. For teaching in the class, you are expected to use slides and LCD projectors.

Tips:

It is recommended that you test the understanding of the students by asking questions in between the class.

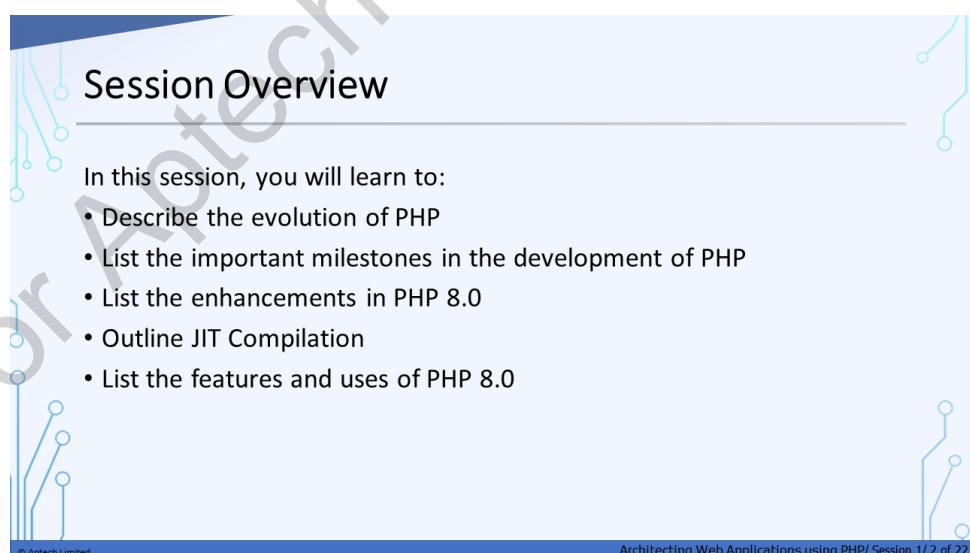
In-Class Activities

Follow the order given here during In-Class activities.

Overview of the Session

Give the students an overview of the current session in the form of session objectives. Read out the objectives on slide 2.

Slide 2



The slide has a blue header bar with the title "Session Overview". Below the title, there is a list of learning objectives: "In this session, you will learn to:" followed by a bulleted list: "• Describe the evolution of PHP", "• List the important milestones in the development of PHP", "• List the enhancements in PHP 8.0", "• Outline JIT Compilation", and "• List the features and uses of PHP 8.0". The slide is decorated with blue circuit board patterns on the left and right sides. At the bottom, it says "© Aptech Limited" and "Architecting Web Applications using PHP/ Session 1 / 2 of 22".

Show slide 2 and give a brief overview of the current session in the form of session objectives. Inform students that this session begins with a brief history of Hypertext Preprocessor (PHP), covering its evolution roadmap from the very beginning to the release of its current version PHP 8.0. The session also covers the enhancements made to the current version of

PHP 8.0 and how it differs from its previous versions. In continuation, the session gives a brief outline of the Just-In-Time (JIT) compilation feature design that has been introduced in PHP 8.0. Lastly, the session explains the features and uses of PHP 8.0.

1.1 In-Class Explanations

Slide 3

The slide has a blue header bar with the title 'History of PHP'. Below the title is an orange box containing the text: 'PHP is a successor of PHP/FI Product, the history of which, dates back to the year 1994.' To the right of the text is a table titled 'Table 1.1: Different Versions of PHP' showing the following data:

PHP Version	Release Date
1.0	October 1995
2.0	April 1996
3.0	June 1997
4.0	June 1999
5.0	July 2004
6.0	
7.0	2019
8.0	2020

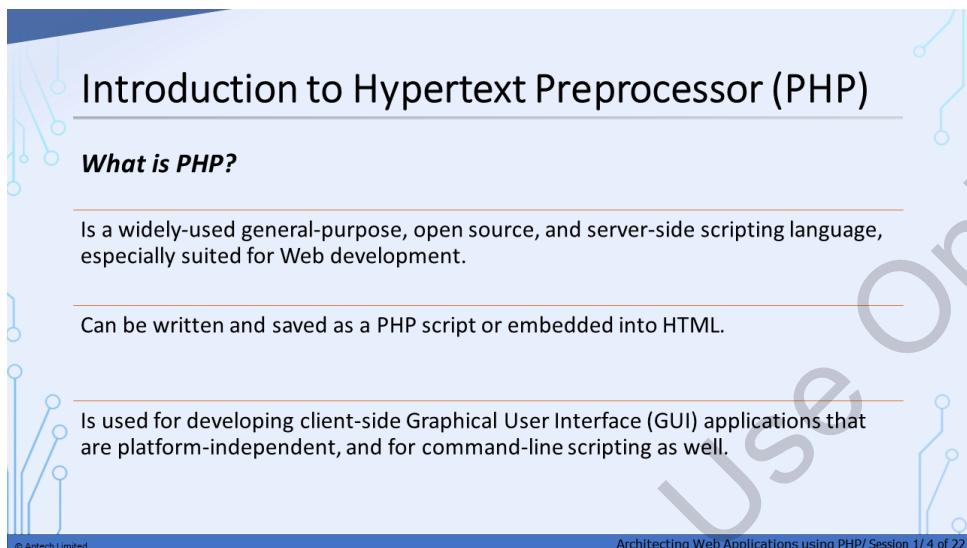
At the bottom left is the copyright notice '© Aptech Limited' and at the bottom right is the page information 'Architecting Web Applications using PHP/ Session 1 / 3 of 22'.

Show slide 3 and explain to students that the history of PHP dates back to the year 1994, which is a successor of PHP/FI Product. To this date, various versions of PHP have been released which are:

- **1.0:** In 1994, Rasmus Lerdorf first created PHP and released it under the name Personal Home Page Construction Kit.
- **2.0:** Rasmus released a complete makeover of the earlier code as PHP/FI. In June 1996, the complete PHP/FI version 2.0 was released.
- **3.0:** The first version, which resembles today's PHP, was PHP 3.0. The inefficiency of PHP/FI 2.0 was realized by Andi Gutmans and Zeev Suraski of Tel Aviv, Israel. This new language was released as PHP and it removed the implication of limited personal use. PHP was the recursive acronym for Hypertext Preprocessor.
- **4.0:** A new engine was introduced here, which improved the codebase and the performance of complex applications. It was dubbed as 'Zend Engine' after the names of Zeev and Andi. Besides the much-improved performance, PHP 4.0 included several other key features.
- **5.0:** The core of PHP 5.0 was the Zend Engine 2.0 with a new object model. PHP 5.0 introduced powerful object-oriented programming support, thereby allowing users to write structured and enterprise-level code.
- **6.0:** Did not release.
- **7.0:** PHP 7.0 was the next major release after PHP 5.0. The core team made several optimizations in the interpreter but did not introduce the JIT compilation in PHP 7.0 version. These optimizations were mainly done to keep the language backwardly

- compatible. Optimized RAM usage and improved syntax in PHP 7.0 boosted performance significantly.
- **8.0:** PHP 8.0 came up with JIT compilation and other features.

Slide 4



The slide has a blue header bar with the title "Introduction to Hypertext Preprocessor (PHP)". Below the title, there is a section titled "What is PHP?". Three bullet points define PHP: 1. Is a widely-used general-purpose, open source, and server-side scripting language, especially suited for Web development. 2. Can be written and saved as a PHP script or embedded into HTML. 3. Is used for developing client-side Graphical User Interface (GUI) applications that are platform-independent, and for command-line scripting as well. The slide is from "Architecting Web Applications using PHP/ Session 1 / 4 of 22".

Show slide 4 and tell the students that PHP is:

- An acronym for Hypertext Pre-Processor.
- A widely used, open-source scripting language.
- Mainly used for Web development.
- Free to download and use.

PHP scripts are executed on the server. PHP files have a `.php` extension, and they are embedded within HTML pages. PHP is used to develop platform-independent applications that have client-side Graphical User Interface (GUI).

Ask the students the following question. Wait for a response before you answer.

In-Class Question: What is meant by client-side scripting?

Answer: Basically, client-side scripts are small programs that are downloaded and processed within the browser on the client device.

Slide 5

Features of PHP [1-2]

Some well-known and important features of PHP:

- Relatively easy for a fresher and also has many advanced features for a professional programmer.
- Runs efficiently on the server-side.
- Works on many operating systems such as Linux, Windows, and Mac OS X.
- Is free and can be downloaded from official PHP resource: www.php.net
- Supports many databases such as Oracle, MySQL, MS SQL Server, Sybase, and so on.

© Aptech Limited Architecting Web Applications using PHP/ Session 1 / 5 of 22

Show slide 5 and tell the students that learning and implementing PHP is comparatively easy for a fresher. In fact, PHP is considered one of the easiest scripting languages. For a professional coder, it offers many advanced features too. On the server-side, it executes with considerable efficiency. It is also compatible with most of the commonly used operating systems such as Microsoft Windows, Linux, Mac OS X, many Unix variants (including Solaris, HP-UX, and OpenBSD), RISC OS, and so on. It is easily downloadable from its official resource, which is www.php.net. PHP supports various databases such as Oracle, MySQL, MS SQL Server, Sybase, and so on.

Today, PHP supports most Web servers, including Apache, IIS, and any Web server utilizing the Fast CGI PHP binary, such as Nginx and Lighttpd. PHP works either as a CGI processor or as a module.

Therefore, with PHP, a user is free to choose an operating system and a Web server. Furthermore, the user can also choose between procedural programming and Object-Oriented Programming (OOP), or a mixture of both.

Slide 6

1.3 More Features of PHP [2-2]

- PHP can dynamically generate content of type HTML, PDF, Text, XML, CSV, and many others.
- Writing programs in PHP is easy and fast, which effectively means that it takes less time to build an application.
- Many popular PHP frameworks such as Zend, Laravel, and Codeigniter are available for PHP.
- There are several Web hosting options available at a fair price for PHP.
- Code deployment is straightforward with PHP.

© Aptech Limited Architecting Web Applications using PHP/ Session 1 / 6 of 22

Show slide 6 and tell the students that using PHP, the user can potentially create contents of various types such as HTML, PDF, Text, XML, CSV, and many more. It also provides the user with ease to write programs with lesser time consumption, to build an application. It also contains various well-known PHP frameworks such as Zend, Laravel, and Codeigniter. Furthermore, it provides several Web hosting alternatives at reasonable costs. Lastly, it offers hassle-free code deployment as well.

Ask the students following question. Wait for a response before you answer.

In-Class Question: What is Web hosting?

Answer: Web hosting is a kind of Internet hosting service that allows users to make their Websites accessible on the World Wide Web (WWW) so that it gets available to anyone using the Internet.

Additional Information:

Refer to following links for more information:

<https://www.geeksforgeeks.org/php-unique-features/>

<https://www.php.net/manual/en/features.php>

<https://www.sitesbay.com/php/php-features-of-php>

Uses of PHP

Areas in which PHP scripts are mainly used:

- Server-side Scripting:** It requires a PHP parser, a Web server, and a Web browser. Software packages such as XAMPP are required for server-side scripting.
- Command-line Scripting:** It allows PHP scripts to run without any server or browser. A PHP parser is required for command-line scripting.
- Writing Desktop:** If a user has ample knowledge of PHP and is using advanced PHP features in their applications, they can use PHP-GTK to write their programs.

© Aptech Limited Architecting Web Applications using PHP/ Session 1 / 7 of 22

Show slide 7 and explain to the students that the PHP scripts are used in following areas:

Server-side scripting: One of the most popular uses of PHP, server-side scripting requires a PHP parser (CGI or server module), a Web server, and a Web browser. The user must run the Web server with the connected PHP installation. The user can access the PHP program output with a Web browser by viewing the PHP page through the server. All these can be run locally through special software packages such as XAMPP if the user is experimenting with PHP programming.

Command-line Scripting: The user can run a PHP script without any server or browser. The user only requires a PHP parser to use it in this manner. This usage is ideal for PHP scripts that are regularly executed using Task Scheduler (on Windows) or cron jobs (on Unix or Linux). The user can refer to command line usage of PHP for detailed information.

Writing Desktop: PHP is not the best language for creating a desktop application with a graphical user interface. However, if the users have expertise in PHP and would want to use some advanced PHP features in their client-side applications, they can also use PHP-GTK to write such programs. Additionally, users can write cross-platform applications in this way. PHP-GTK is a set of language bindings for PHP. It allows GTK GUI applications to be written in PHP. PHP-GTK provides an object-oriented interface to GTK classes and functions. PHP-GTK is an extension to PHP and is not available in the main distribution.

Give following additional information to the students:

TIP: Some famous companies and applications that are using PHP include Facebook, Wikipedia, Tumblr, WordPress, and Slack.

Additional Information:

Refer to following links for more information:

<https://www.php.net/manual/en/intro-whatcando.php>

<https://www.sitesbay.com/php/php-uses-of-php>

<https://www.jobsity.com/blog/8-reasons-why-php-is-still-so-important-for-web-development>

<https://www.geeksforgeeks.org/what-is-php-and-why-we-use-it/>

Slide 8

The slide has a blue header bar with the title 'Basic PHP Syntax'. Below the title is a section labeled 'Code Snippet 1:' containing the following PHP code:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Example</title>
  </head>
  <body>
    <?php
      echo "Hello Students!";
    ?>
  </body>
</html>
```

Below the code, the output is shown as 'Output: Hello Students!'. The slide has a decorative background with blue and white circuit board patterns.

© Aptech Limited Architecting Web Applications using PHP/ Session 1 / 8 of 22

Show slide 8 and tell the students that Code Snippet 1 shows an example of a PHP script inside an HTML file. All the PHP pages have HTML tags with embedded code. The PHP code is enclosed within the `<? php` and `?>` tags, which allows developers to invoke PHP code blocks in an HTML file. `<?php` marks the start processing instruction while `?>` denotes the end processing instruction.

In Code Snippet 1, `echo "Hello Students!"`; is the PHP code which when executed will display the output as "Hello Students!"

Also, advise students that as they are new to PHP, they should start directly from PHP 8.0 to avoid migration costs associated with previous versions. On top of that, the new version offers cleaner code and better performance from the very beginning. If more code is written in older versions, then it would require lots of effort to shift to the newer version (PHP 8.0).

Ask the students the following question. Wait for a response before you answer.

In-Class Question: What is syntax in computer programming?

Answer: Syntax includes the rules to be defined in the structure of a programming language. It controls the symbols and the commands of the language. A code will not compile if the syntax is incorrect. It is basically like what grammar is to English.

Additional Information:

Refer to following links for more information:

<https://www.geeksforgeeks.org/php-basic-syntax/>

https://www.w3schools.com/php/php_syntax.asp

Slide 9

PHP	JavaScript
PHP is a popular scripting language which is used to perform server-side functions.	JavaScript is a high level programming language which is handy for client-side scripting
In PHP, the code is executed on server, not within browser.	In JavaScript, the code is executed within browser (client) itself.
PHP is a back-end language.	JavaScript is used for front-end development.
PHP can only be embedded with HTML.	JavaScript can be combined with HTML, AJAX, XML.
PHP is a multi-threaded language. It means that it blocks I/O to carry out multiple tasks concurrently.	JavaScript is a single-threaded language which is event driven. It never blocks everything and runs concurrently.
PHP helps to build high-level interactive web pages.	JavaScript helps to build user-friendly creative Web pages.

© Aptech Limited Architecting Web Applications using PHP/ Session 1 / 9 of 22

Show slide 9 and tell the students that basic differences between a PHP script and JavaScript are as follows:

- PHP is a server-side scripting language used specifically for Web development. Whereas, JavaScript is one of the most popular programming languages for Web development. Plus, it is used by various non-browser environments. It is mostly used for client-side development.
- In PHP, after the code is executed on the server, HTML is generated and sent to the client. The client receives the results of script execution, but the underlying code is not exposed. In JavaScript, it is totally different.
- PHP is a multithreaded language that enables users to run multiple threads concurrently. On the contrary, JavaScript is single-threaded, that means the codes will be executed in the same thread.

Also, advise students that as they are new to PHP, they should start directly from PHP 8.0 to avoid migration costs associated with previous versions. On top of that, the new version offers cleaner code and better performance from the very beginning. If more code is written in older versions, then it would require lots of efforts to shift to the newer version (PHP 8.0).'

Additional Information:

Refer to following links for more information:

<https://www.geeksforgeeks.org/difference-between-javascript-and-php/>

<https://www.javatpoint.com/javascript-vs-php>

The slide has a blue header bar with the title 'Enhancements in PHP 8.0'. Below the title, a text box states: 'All new PHP 8.0 comes with various new features and optimizations are mentioned as follows:' A central blue circle contains the text 'PHP 8.0'. Surrounding the circle are seven rectangular boxes, each containing a feature name: 'Enhanced error handling, and type system', 'Consistency', 'Attributes', 'Union types', 'Constructor property promotion', 'Named arguments', 'Null-safe operator', and 'Match expression'. The slide has a decorative background of blue circuit-like patterns on the left and right sides. At the bottom, there is a copyright notice: '© Aptech Limited' on the left and 'Architecting Web Applications using PHP/ Session 1/ 10 of 22' on the right.

Show slide 10 and tell the students that PHP 8.0 was released worldwide on November 26, 2020. PHP as an engine constantly changes by either adding new bits or modifying older bits to optimize the code. These changes make PHP easier and uncomplicated to work. PHP 8.0 is regarded as a major update of the PHP language.

PHP 8.0 brings a lot of new features, improvements, functions, and deprecations to the language compared to PHP 7. It has many new features and optimizations, which include Named arguments, Attributes, Constructor property promotion, Union types, Match expression, Null-safe operator, Enhanced error handling and type system, and Consistency.



Named Arguments

Following representation shows how to pass a Named argument to a function in PHP 7.0 and PHP 8.0:

PHP 7.0

```
• htmlspecialchars($string, ENT_COMPAT | ENT_HTML401, 'UTF-8', false);
```

PHP 8.0

```
• htmlspecialchars($string, double_encode: false);
```

© Aptech Limited Architecting Web Applications using PHP / Session 1 / 11 of 22

Show slide 11 and tell the students that named arguments enable the user to pass parameters to methods and functions using an array-like construct. This makes it simpler to skip optional parameters and pass them out of order. This feature also makes function calls with various optional arguments less repetitive and easier to maintain.

In the new PHP feature, the user must specify only the required parameters and skip optional parameters. Arguments are self-documented and are order-independent.

Additional Information:

Refer to following links for more information:

<https://www.phptutorial.net/php-tutorial/php-named-arguments/>

<https://www.tutorialspoint.com/named-arguments-in-php-8>

<https://www.php.net/manual/en/functions.arguments.php>

<https://platform.sh/blog/2020/php-80-feature-focus-named-arguments/>

<https://stitcher.io/blog/php-8-named-arguments>

The slide features a blue header bar with the title 'Attributes'. Below the title, a text block states: 'Following representation shows the syntax for attributes in PHP 7.0 and PHP 8.0 respectively:'. Two code blocks are shown side-by-side. The first code block, labeled 'PHP 7.0', contains:`* class PostsController {
 /**
 * @Route("/api/posts/{id}", methods={"GET"})
 */
 public function get($id) { /* ... */ }`The second code block, labeled 'PHP 8.0', contains:`* class PostsController {
 #[Route("/api/posts/{id}", methods: ["GET"])]
 public function get($id) { /* ... */ }`

At the bottom left is a small copyright notice: '© Aptech Limited'. At the bottom right is the text: 'Architecting Web Applications using PHP / Session 1 / 12 of 22'.

Show slide 12 and tell the students that attributes provide a mechanism to add metadata to a codebase. Attributes can be used with classes, methods, functions, and properties. They are easier to work with, as compared with the previous approaches adapted by many PHP 7 projects. Attributes are defined as simple classes, themselves annotated with PHP's built-in attribute. They can be attached to entities within a codebase. Attributes are likely to be most useful within frameworks and libraries, where they help to abstract one-time mapping of app components.

The user can now use structured metadata with PHP's native syntax, instead of PHP Doc annotations.

Additional Information:

Refer to following links for more information:

<https://www.tutorialspoint.com/attributes-in-php-8>

<https://php.watch/articles/php-attributes>

The slide has a blue header bar with the title 'Constructor Property Promotion'. Below the title, a text block says: 'Following representation shows the example for constructor property in PHP 7.0 and PHP 8.0 respectively:'. Two code blocks are shown side-by-side. The left code block is for PHP 7.0:

```
• class Point {  
    public float $x;  
    public float $y;  
    public float $z;  
  
    public function __construct(  
        float $x=0.0,  
        float $y=0.0,  
        float $z=0.0  
    ) {  
        $this->x=$x;  
        $this->y=$y;  
        $this->z=$z;  
    }  
}
```

The right code block is for PHP 8.0:

```
• class Point {  
    public function __construct(  
        publicfloat $x=0.0,  
        publicfloat $y=0.0,  
        publicfloat $z=0.0,  
    ) {}  
}
```

At the bottom left is a small copyright notice: '© Aptech Limited'. At the bottom right is a page number: 'Architecting Web Applications Using PHP/ Session 1 / 13 of 22'.

Show slide 13 and tell the students that PHP adds support for constructor property promotion. It is a shorthand syntax that lets users combine property definition, type hinting, and population in line with the constructor's signature. The new syntax also removes repetition. It is easier to add additional constructor properties in the future. As a result, the number of code sections that the user would require to modify is reduced.

In the new version, there is less boilerplate code to define and initialize properties. Construction Property Promotion proposes a new and more concise syntax that will simplify the property declaration making it shorter and less redundant. This proposal only relates to those parameters, which have prefix - public, protected, and private keywords. Basically, this enhancement gives the users a new way to promote properties that are shorter, more readable, and less prone to errors.

Additional Information:

Refer to following links for more information:

<https://www.tutorialspoint.com/constructor-property-promotion-in-php-8>

<https://www.geeksforgeeks.org/constructor-property-promotion-in-php-8/>

The slide features a blue header bar with the title 'Union Types'. Below the title, a text block states: 'Following representation shows the example of native union type declarations in PHP 7.0 and PHP 8.0 respectively:'. Two code snippets are shown side-by-side.

PHP 7.0

```
*class Number {  
    /** @var int|float */  
    private $number;  
  
    /**  
     * @param float|int $number  
     */  
    public function __construct($number) {  
        $this->number = $number;  
    }  
  
    newNumber('NaN'); // Ok
```

PHP 8.0

```
*class Number {  
    public function __construct(  
        private int|float $number  
    ) {}  
  
    newNumber('NaN'); // TypeError
```

At the bottom left is the copyright notice '© Aptech Limited' and at the bottom right is the page information 'Architecting Web Applications Using PHP/ Session 1 / 14 of 22'.

Show slide 14 and tell the students that in PHP 8, it is now possible to hint types as a union of two or more types. This is where the types' value can be derived from any of the types in the union.

Instead of the PHP Doc annotations, the user can use native union type declarations that are validated at runtime. Union types accept values that can be of different kinds.

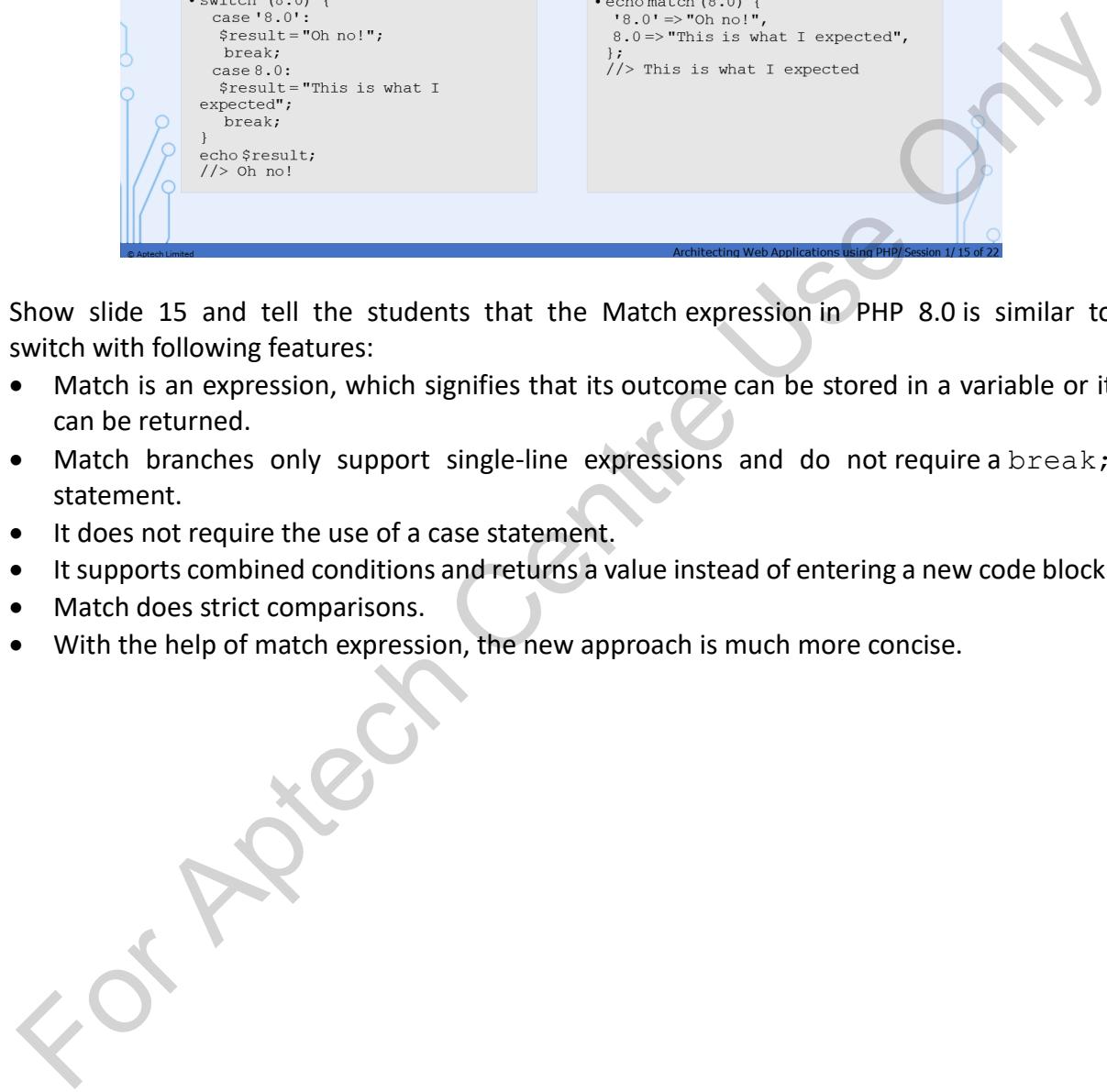
Union types are a collection of two or more types that indicate that either one of those can be used. Another noteworthy point is that void can never be part of a union type since it indicates 'no return value at all'.

Additional Information:

Refer to following links for more information:

<https://www.geeksforgeeks.org/php-8-union-types/>

<https://www.tutorialspoint.com/union-type-in-php-8>



Match Expression

Following representation shows the example of switch v/s match case in PHP 7.0 and PHP 8.0 respectively:

PHP 7.0	PHP 8.0
<pre>• switch (8.0) { case '8.0': \$result = "Oh no!"; break; case 8.0: \$result = "This is what I expected"; break; } echo \$result; //> Oh no!</pre>	<pre>• echomatch (8.0) { '8.0' => "Oh no!", 8.0 => "This is what I expected", }; //> This is what I expected</pre>

© Aptech Limited Architecting Web Applications using PHP / Session 1 / 15 of 22

Show slide 15 and tell the students that the Match expression in PHP 8.0 is similar to switch with following features:

- Match is an expression, which signifies that its outcome can be stored in a variable or it can be returned.
- Match branches only support single-line expressions and do not require a `break;` statement.
- It does not require the use of a `case` statement.
- It supports combined conditions and returns a value instead of entering a new code block.
- Match does strict comparisons.
- With the help of match expression, the new approach is much more concise.

The slide has a blue header bar with the title 'Nullsafe Operator'. Below the title, a text block says: 'Following representation shows the example of the use of nullsafe operator ?- in PHP 7.0 and PHP 8.0:'.

PHP 7.0	PHP 8.0
<pre>\$country = null; if (\$session !== null) { \$user = \$session->user; if (\$user !== null) { \$address = \$user->getAddress(); if (\$address !== null) { \$country = \$address->country; } } }</pre>	<pre>\$country = \$session?->user?->getAddress()?->country;</pre>

At the bottom left is a small copyright notice: '© Aptech Limited'. At the bottom right is the text: 'Architecting Web Applications using PHP/ Session 1 / 16 of 22'.

Show slide 16 and tell the students that the user can use a chain of calls with the new nullsafe operator instead of writing code for the null check conditions. PHP 8.0 provides a simpler solution, which eliminates nesting. This results in considerably less code to test and maintain.

The entire chain execution is aborted when the evaluation of one element in the chain fails. Subsequently, the entire chain evaluates to null.

The Nullsafe operator is not completely reliable, as it does not work on method calls.

Instead, the user requires intermediate checks or relies on optional helpers provided by some frameworks.

Additional Information:

Refer to following links for more information:

<https://php.watch/vendors/8.0/null-safe-operator#:~:text=Null%2Dsafe%20operator%20is%20a,null%20%2C%20without%20causing%20any%20errors.>
<https://www.tutorialspoint.com/nullsafe-operator-in-php-8>

Enhanced Error Handling and Type System [1-2]

Enhancements and modifications in different system types and error handling improvements:

- Stricter type checks for arithmetic/bitwise operators as compared to earlier versions
- Validating abstract trait method
- Magic methods now have correct signatures
- Engine warnings have been reclassified
- Incompatible method signatures give fatal errors

© Aptech Limited Architecting Web Applications using PHP / Session 1 / 17 of 22

Show slide 17 and tell the students that it is very simple in PHP to handle errors.

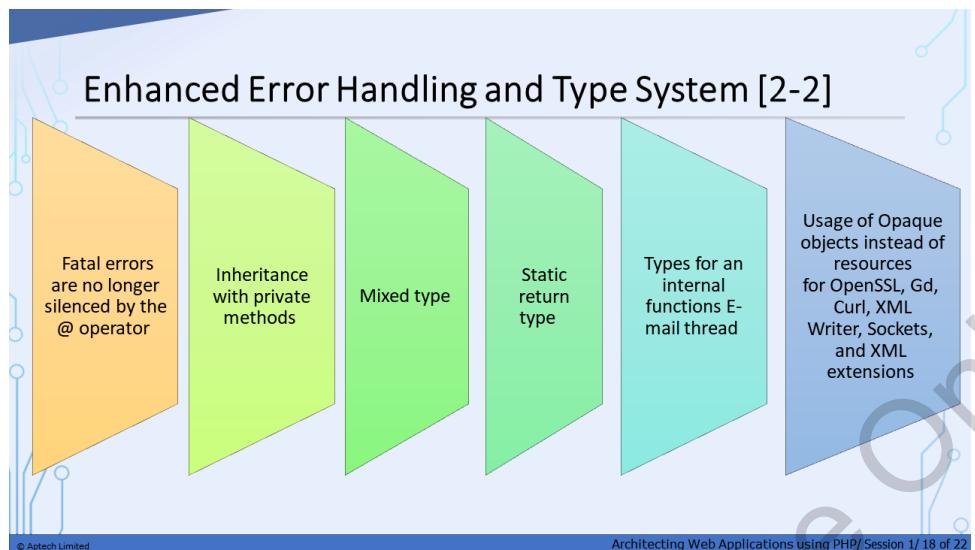
Users can write their own functions to handle any error. PHP provides users a framework to define error-handling functions. Compared to earlier versions, PHP 8.0 now provides stricter type checks for arithmetic or bitwise operators. It also validates the abstract trait method. Magic methods have correct signatures now. The engine warnings have also been reclassified. Fatal errors are given by incompatible method signatures.

Ask the students the following question. Wait for a response before you answer.

In-Class Question: What is error handling?

Answer: Error handling is the process of catching errors raised by a program and then, taking appropriate action. If the user does not handle errors properly, then it may lead to many unforeseen consequences.

Slide 18



Show slide 18 and tell the students that in the previous versions of PHP, fatal errors were silenced by @ operator, which is no longer true in the case of the new version. It also provides inheritance with private methods. Mixed type and static return type there as well. Moreover, there are types for internal functions E-mail thread.

The previous versions used resources for OpenSSL, Gd, Curl, XML Writer, Sockets, and XML extensions, while the new version provides the usage of Opaque objects.

Additional Information:

Refer to following links for more information:

https://www.w3schools.com/Php/php_error.asp
https://www.tutorialspoint.com/php/php_error_handling.htm
<https://www.geeksforgeeks.org/error-handling-in-php/>

Consistency

PHP 8.0 has consistent type errors for internal functions. Most of the internal functions throw an Error exception when there is a failure of validation of the parameters.

© Aptech Limited Architecting Web Applications using PHP/ Session 1 / 19 of 22

Show slide 19 and tell the students that in PHP, user-defined functions throw Type Error, but internal functions do not. They emit warnings instead and return null. The behavior of internal functions has been made consistent throughout PHP 8.

Additional Information:

Refer to following links for more information:

https://wiki.php.net/rfc/consistent_type_errors

<https://externals.io/message/112909>

Just In Time (JIT) Compiler [1-2]

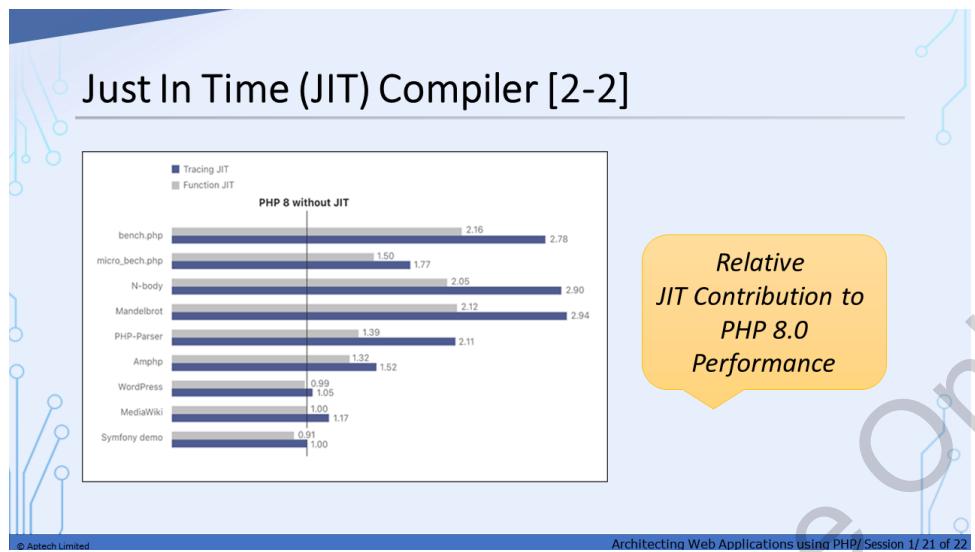
- Being a deciphered or an interpreted language, PHP on compilation, does not run immediately at launch time. It runs in real-time.
- In earlier versions of PHP, during compiling, whenever a PHP code is run, the interpreter must first interpret, then compile, and finally execute the code.
- This is done repeatedly for each request, resulting in slow code execution and wastage of CPU resources.
- This problem has now been overcome with the introduction of the Just In Time (JIT) compiler, in PHP 8.0.

© Aptech Limited Architecting Web Applications using PHP / Session 1 / 20 of 22

Show slide 20 and tell the students that PHP 8 adds support for Just-In-Time (JIT) compilation, which has the potential to significantly enhance performance. Long-running scripts hosting repetitive tasks are most likely to benefit from JIT compilation. The addition of the JIT helps PHP to extend into other areas of programming, enhancing the language's overall appeal and versatility.

The JIT compiler allows users to compile a program into machine code just before it can be executed. As JIT bypasses the compilation stage, it provides flexibility in PHP code and brings out extensive improvements in code execution, memory usage, and performance. However, these improvements apply only to numerical or mathematical calculations, rather than the usual PHP Web applications. JIT is immensely useful in programs that involve a long execution cycle such as 3D rendering, data analysis, or Artificial Intelligence.

Slide 21



Show slide 21 and tell the students that JIT is disabled by default, and if enabled, it compiles and catches native instructions. It does not make a noticeable difference in I/O bound Web applications but provides a performance boost for CPU-heavy applications.

PHP 8.0 introduces two JIT compilation engines, namely Tracing JIT and Function JIT.

The Tracing JIT is the most assuring of the two. On synthetic benchmarks, it has three times better performance. Additionally, there is twice the improvement on specific long-running applications.

The given diagram shows a relative JIT contribution to the performance of PHP 8.0.

Ask the students the following question. Wait for a response before you answer.

In-Class Question: In which version of PHP was Just-In-Time (JIT) Compiler introduced?

Answer: PHP 8.0

Additional Information:

Refer to following links for more information:

<https://thephp.website/en/issue/php-8-jit/>

<https://stackoverflow.com/questions/60581060/what-is-php8-jit-compiler>

Summary

- PHP is a widely-used open source general-purpose scripting language, especially suited for Web development.
- PHP is easy to learn for freshers and includes several advanced features for seasoned developers.
- Users such as Website developers, WordPress plugin, and theme creators must upgrade their knowledge of PHP to stay ahead of their competitors.
- PHP 8.0 is considered as a major update of the PHP language, with many new features and optimizations.
- Just In Time (JIT) compiler is the most significant feature added in PHP 8.0.
- PHP can dynamically generate content of types such as HTML, Flash, PDF, Text, XML, CSV, and many others.
- PHP has useful text processing features, including the Perl Compatible Regular Expressions (PCRE).

© Aptech Limited Architecting Web Applications using PHP/ Session 1 / 22 of 22

Use this slide to summarize the session. You will end the session, with a brief summary of what has been taught in the session. Tell the students pointers of the session. This will be a revision of the current session.

1.3 Post Class Activities for Faculty

You should familiarize yourself with the topics of the next session.

Tips:

You can also check the Articles/Blogs/Expert Videos uploaded on the Online Varsity site to gain additional information related to the topics covered in the next session.

Session 2 – PHP Basic and Syntax

2.1 Pre-Class Activities

Before beginning the session, you should fully familiarize yourself with its topics. Prepare a question or two that will be a key point to relate the current session objectives.

2.1.1 Teaching Skills

To teach this session, you should be well-versed with the concept of internationalization and design patterns in Java. You must also be familiar with the concept of localization.

You must use the given images and teach the concepts in the theory class. For teaching in the class, you are expected to use slides and LCD projectors.

Tips:

It is recommended that you test the understanding of the students by asking questions in between the class.

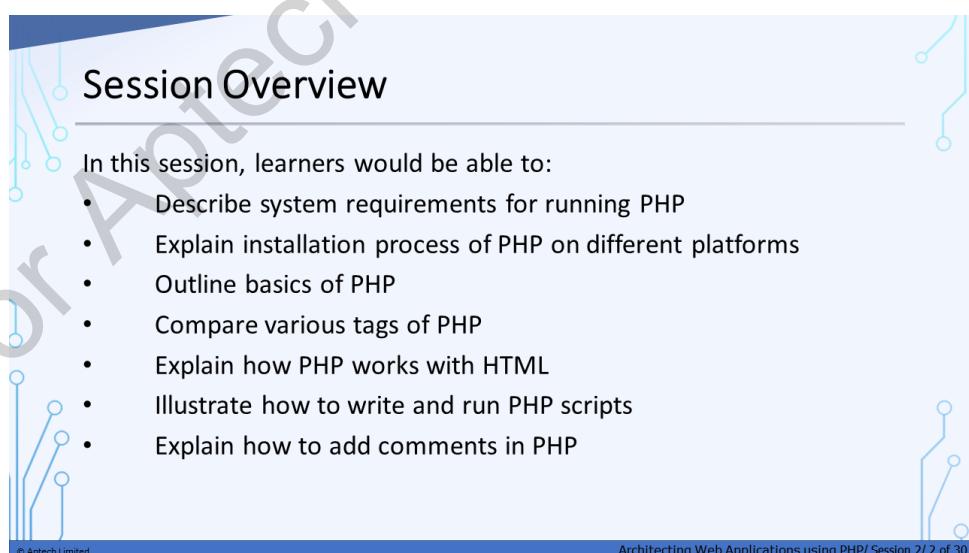
In-Class Activities

Follow the order given here during In-Class activities.

Overview of the Session

Give the students an overview of the current session in the form of session objectives. Read out the objectives on slide 2.

Slide 2



The slide has a blue header bar with the title "Session Overview". The main content area is white with a light blue background. It features a decorative border with blue lines and circles. The text "In this session, learners would be able to:" is followed by a bulleted list of nine items. At the bottom, there is a small copyright notice and a page number.

© Aptech Limited

Architecting Web Applications using PHP/ Session 2 / 2 of 30

- Describe system requirements for running PHP
- Explain installation process of PHP on different platforms
- Outline basics of PHP
- Compare various tags of PHP
- Explain how PHP works with HTML
- Illustrate how to write and run PHP scripts
- Explain how to add comments in PHP

Show slide 2 and give students a brief overview of the current session in the form of session objectives. Inform students that it is necessary for a user to understand how to run PHP on

their system before diving deep into the language. Users must learn how to install PHP on various platforms in accordance with their requirement and begin their PHP coursework.

This session explains the system requirements for running PHP, the installation process on different platforms, and the basics of PHP. The session also covers topics such as various tags of PHP, working of PHP with HTML, adding comments in PHP, and understanding how the PHP scripts are written and executed.

2.2 In-Class Explanations

Slide 3

PHP Installation [1-6]

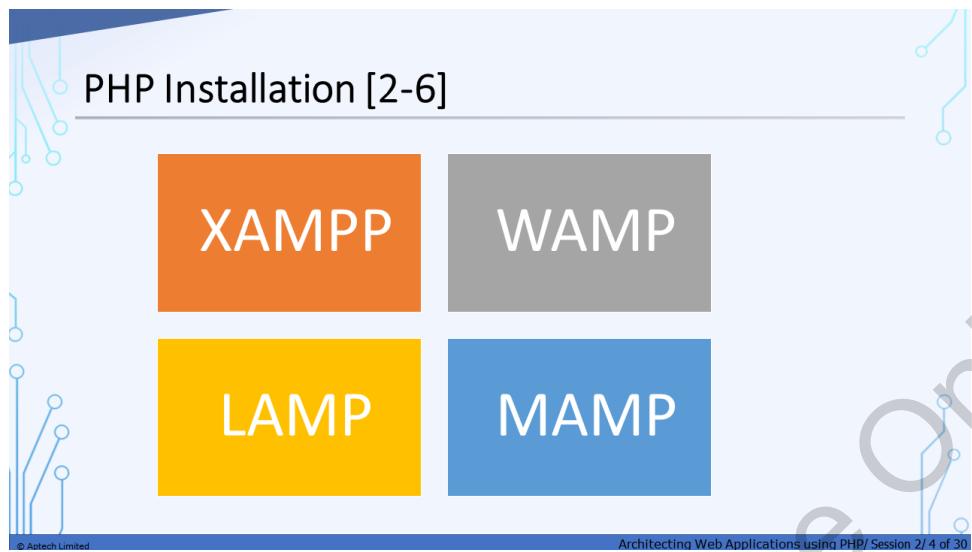
System Requirements to run PHP are:

1. A Web Server such as Apache
2. PHP Interpreter
3. A Database such as MySQL

© Aptech Limited Architecting Web Applications using PHP/ Session 2 / 3 of 30

Show slide 3 and inform students about the system requirements to run PHP.

Installing PHP locally on a computer allows users to develop and test PHP-based apps and Websites in a safe manner. Out of these three, the Database is optional, though it is recommended if one is creating practical, real-world-oriented applications.



Show slide 4 and inform students that the user can download a Web server, a PHP interpreter, and a MySQL database separately from their respective official Websites. Open source developers, on the other hand, have created all-in-one setup packages such as WAMP, LAMP, MAMP, or XAMPP that make these setups simple. The package to select is determined by one's comfort level with the platform. On the user's Windows, Linux, or Mac Consoles, such a package will set up a PHP environment.

Elaborate more regarding these packages.

Most of these packages have three common components:

AMP

A -> Apache HTTP Server

M -> MySQL

P -> PHP

The first alphabet indicates the OS.

L indicates Linux. W indicates Windows. M indicates MacOS and X indicates cross-platform.

LAMP -> Linux, Apache, MySQL, PHP

WAMP -> Windows, Apache, MySQL, PHP. Can handle dynamic Websites. It is available in 32-bit and 64-bit systems.

MAMP -> Equivalent installation of packages such as LAMP/WAMP on MacOS.

XAMPP -> Open-source cross-platform Apache, **MariaDB** database, PHP, Perl.

MariaDB is a community-developed RDMS.

XAMPP helps developers to create and test their programs on a local Web server.

Additional information:

Refer to following links for more information:

<https://www.jetbrains.com/help/phpstorm/installing-an-amp-package.html>

https://www3.ntu.edu.sg/home/ehchua/programming/Webprogramming/AMP_Setup.html

Slide 5

PHP Installation [3-6]

- Apache HTTP Server
 - It runs the open source Web server on Windows or Linux. When Apache Web server is running on a local Windows/Linux machine, it is possible to test Web pages locally by a developer in a browser.
- MySQL
 - Open-source relational database
- PHP
 - Scripting language that can be utilized to access data from the high-speed database such as MySQL.

© Aptech Limited Architecting Web Applications using PHP/ Session 2 / 5 of 30

Show slide 5 to provide more information about AMP (Apache, MySQL, and PHP).

Apache HTTP server can be used to run the Web server on a local machine. It also lets the developers test Web pages locally in a browser.

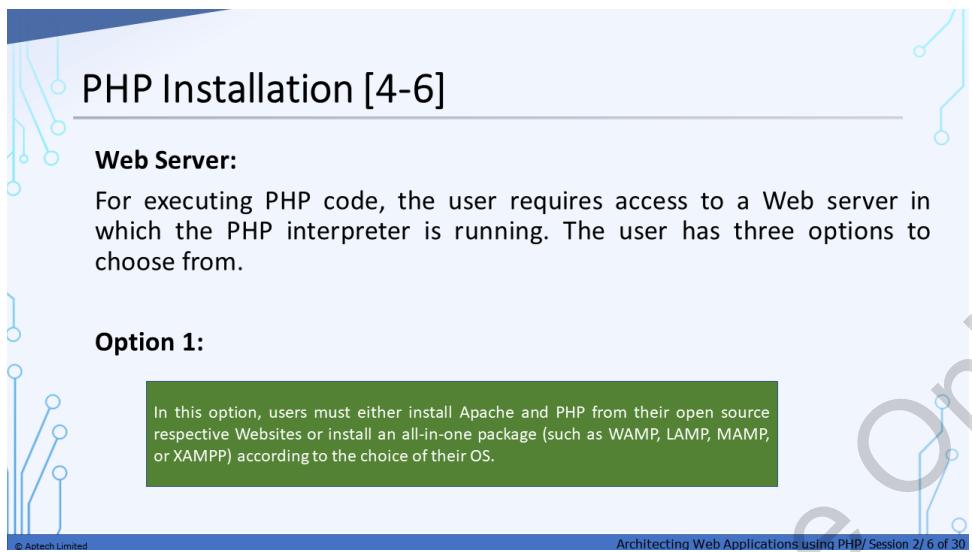
MySQL is an Open Source relational database.

PHP is a scripting language that can be used to access data from a high-speed database such as MySQL.

Ask following question to the students. Wait for the response before you answer.

In-Class Question: What is XAMPP?

Answer: Open source cross platform Apache distribution containing MariaDB, PHP, and Perl.



PHP Installation [4-6]

Web Server:
For executing PHP code, the user requires access to a Web server in which the PHP interpreter is running. The user has three options to choose from.

Option 1:

In this option, users must either install Apache and PHP from their open source respective Websites or install an all-in-one package (such as WAMP, LAMP, MAMP, or XAMPP) according to the choice of their OS.

© Aptech Limited Architecting Web Applications using PHP/ Session 2 / 6 of 30

Show slide 6 and inform the students regarding the Web Server.

Tell the students that, the developers require access to a Web server in which the PHP interpreter is running.

The users have three options to choose from.

Explain option 1, which is using all-in-one packages such as WAMP/LAMP/MAMP or XAMPP.

PHP Installation [5-6]

Post-installation, following actions can be performed:

- Launch Apache Server and PHP from the program list. In the Apache installation folder, the user will find the `www` directory.
- Create and save PHP scripts or create project folders in the `www` directory.
- Open any browser and execute the scripts by typing `http://localhost/filename.php`.

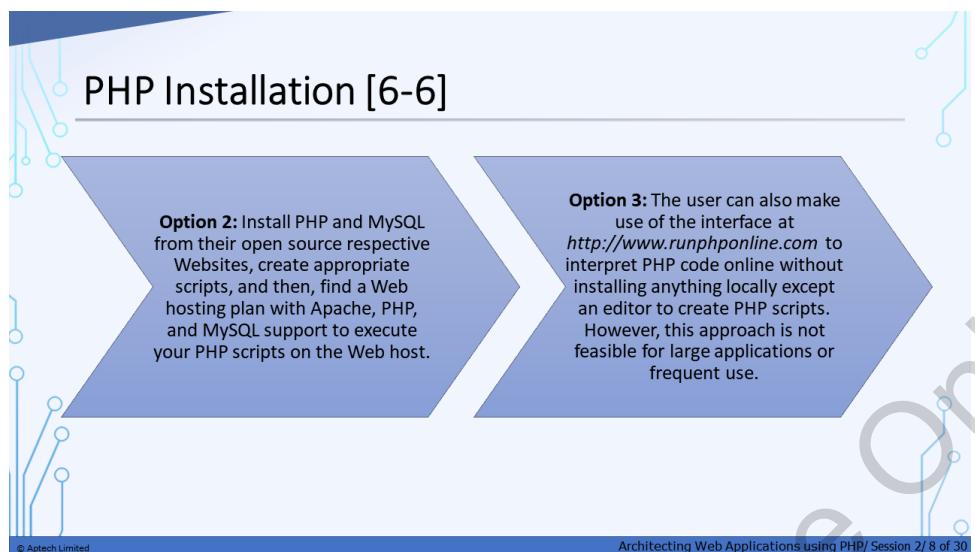
© Aptech Limited Architecting Web Applications using PHP/ Session 2 / 7 of 30

Show slide 7 and explain the actions that can be performed post-installation as per option 1. Apache server and PHP must be launched from the program list. In Apache installation folder, the user will find `www` directory.

Users can create and save PHP scripts and project folders in the `www` directory. The users can open any browser and execute the scripts with the URL:
`http://localhost/filename.php`

filename.php represents the PHP script file name.

Regardless of where the script file is stored locally, the URL to execute it on the browser remains consistent in this format.



Show slide 8 and explain options 2 and 3 for PHP installation.

Option 2 suggests installing PHP and MySQL from their open-source Websites and using them. Users can create appropriate scripts and find a Web hosting plan with Apache, PHP, and MySQL support to execute the PHP scripts on the Web host.

Option 3 suggests that users can also make use of an online interface such as '<http://www.runphponline.com>' to interpret PHP code online without installing anything locally except the editor to create the PHP scripts. However, this option is not feasible for large applications or frequent use.

Also, inform students about Nginx, which is a Web server alternative that can sometimes be used instead of Apache.

It is tricky to set up and configure Nginx on Windows. It also requires more configuration.

The minimum requirement for PHP is at least Windows 2008/Vista, either 32-bit or 64-bit.

Windows 2008 or Vista are not supported from PHP 7.2.0 onwards. PHP requires the Visual C Runtime (CRT). Since many applications require it, the chances are that it may already be installed.

© Aptech Limited

Architecting Web Applications using PHP/ Session 2/ 9 of 30

Show slide 9 and explain the minimum requirements for PHP.

Most of the recent PHP versions work perfectly with the Microsoft Visual C++ Redistributable for Visual Studio 2019. The user must download x64 CRT for PHP x64 builds and x86 CRT for PHP x86 builds.

Give following additional information to the students:

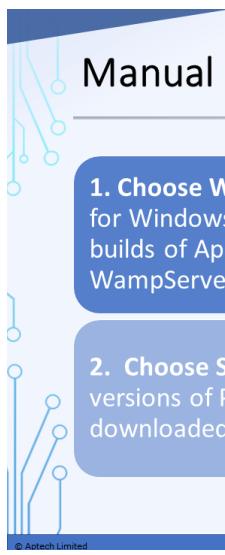
TIP: If a user is using Internet Information Services (IIS) but wants to set up PHP, the easiest technique is to utilize Microsoft's Web Platform Installer (WebPI).

Additional information:

Refer to following links for more information:

<https://php.tutorials24x7.com/blog/how-to-install-php-8-on-windows>

<https://www.w3resource.com/php/installation/install-php-on-windows.php>



Manual PHP Installation on Windows [1-2]

- 1. Choose Web Server Apache:** There exist several builds of Apache 2 for Windows. It is recommended that the user should use the Apache builds of Apache Lounge. Other options include BitNami, XAMPP, and WampServer. These three offer automatic installer tools.
- 2. Choose Source Build of PHP:** Windows-based builds of the latest versions of PHP can be downloaded from <http://windows.php.net/download/>.

© Aptech Limited Architecting Web Applications using PHP / Session 2 / 10 of 30

Show slide 10 and explain the process for manual PHP installation on Windows.

PHP can be used on Apache through mod_fastcgi and mod_php. The mod_php requires a Thread Safe (TS) build of Apache built with the same version of Visual C and the same CPU (x86 or x64).

Also, inform the students regarding the URL to download the source build of PHP.

Give following tip to students:

TIP: In a programming context, a build is a version of a program.

Additional information:

Refer to following links for more information:

<https://www.php.net/manual/en/install.windows.manual.php>

<https://doc.bccnsoft.com/docs/php-docs-7-en/install.windows.manual.html>

Ask following question to the students. Wait for the response before you answer.

In-Class Question: Why is XAMPP used for PHP?

Answer: XAMPP helps a local host or server to test its Website and clients via computers and laptops before releasing it to the main server.

Manual PHP Installation on Windows [2-2]

There are four types of PHP builds:

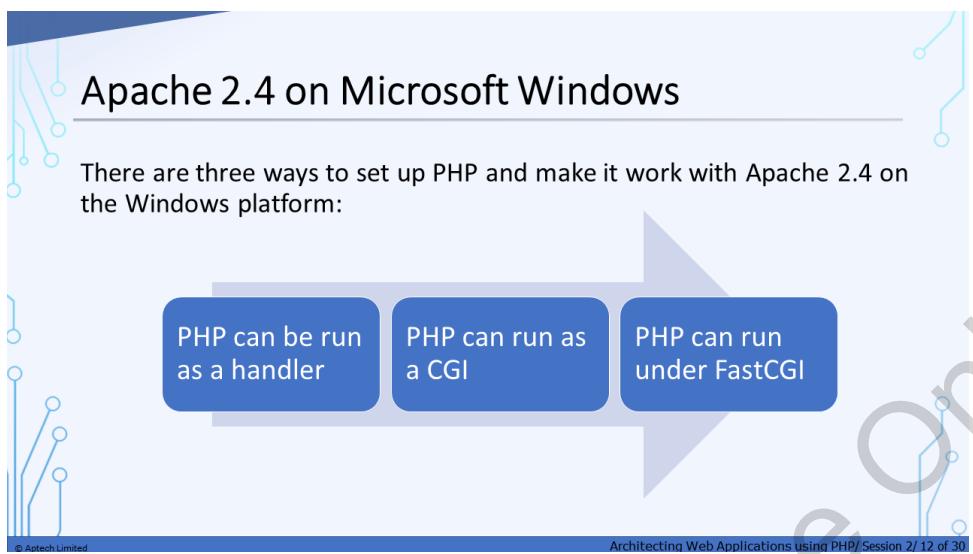
Thread-Safe (TS) PHP build	Non-Thread-Safe (NTS) PHP build
X86	x64

© Aptech Limited Architecting Web Applications using PHP / Session 2 / 11 of 30

Show slide 11 and discuss the four types of PHP builds.

Four types of PHP builds:

- Thread-Safe (TS) PHP Build: This build is for single process Web servers, such as Apache with mod_php.
- Non-Thread-Safe(NTS) PHP build: This build is for IIS and other FastCGI Web servers (Apache with mod_fastcgi) are recommended for command-line scripts.
- X86: This build is for 32-bit systems.
- X64: This build is for 64-bit systems.



The slide has a blue header bar at the top and bottom. The title "Apache 2.4 on Microsoft Windows" is in the header. The footer contains the text "Architecting Web Applications Using PHP / Session 2 / 12 of 30". The main content area features three blue rounded rectangles with white text: "PHP can be run as a handler", "PHP can run as a CGI", and "PHP can run under FastCGI". A large grey arrow points from left to right, containing these three items. The background of the slide has a faint watermark reading "APTECH © 2018 All Rights Reserved Only".

There are three ways to set up PHP and make it work with Apache 2.4 on the Windows platform:

- PHP can be run as a handler
- PHP can run as a CGI
- PHP can run under FastCGI

Show slide 12 and explain the options to set up PHP on Windows and make it work with Apache 2.4. It is highly recommended to consult the official Apache Documentation to get a basic understanding of the Apache 2.4 Server and then, download Apache 2.4.

Post download, the user should first proceed with the manual installation steps as per following URL:

<https://www.php.net/manual/en/install.windows.manual.php>

Then, proceed with the integration of Apache and PHP.

Ask following question to the students. Wait for the response before you answer.

In-Class Question: What are the ways to setup PHP to work with Apache on Windows?

Answer: There are three ways:

- PHP can be run as a handler
- PHP can run as a CGI
- PHP can run under FastCGI.

Note: One must remember that when adding path values in the Apache configuration files on the Windows environment, all backslashes such as c:\directory\file.ext must be converted to forward slashes:

c:/directory/file.ext

Additionally, in the case of directories, the trailing slash may also be necessary.

Additional information:

Refer to following links for more information:

<https://www.php.net/manual/en/install.windows.apache2.php>

<https://httpd.apache.org/docs/2.4/platform/windows.html>

Option 1: Installing as an Apache Handler

Step 1: To load the PHP module for Apache 2.4, following lines in the Apache httpd.conf configuration file must be inserted:

```
LoadModule php_module "c:/php/php8apache2_4.dll"
<FilesMatch \.php$>
    SetHandler application/x-httpd-php
</FilesMatch>
```

Step 2: configure the path to php.ini

```
PHPIniDir "C:/php"
```

© Aptech Limited Architecting Web Applications Using PHP / Session 2 / 13 of 30

Show slide 13 and explain the option of installing as an Apache Handler.

Few things to take note of:

- In step one, the name of the module was `php7_module` prior to PHP 8.0.0. You should use `php8apache2_4.dll` for PHP 8.
- In step two, the actual installation path for PHP must be substituted instead of `C:/php/` in the examples.
- Make sure that the file referenced in the `LoadModule` directive is at the specified location.

Option 2: Running PHP as CGI

The PHP-CGI files must be placed in a directory designated as a CGI directory using the `ScriptAlias` directive, to execute PHP as CGI.

Example: PHP and Apache 2.x as CGI

```
# !C:/php/php.exe
<?php
    phpinfo();
?>
```

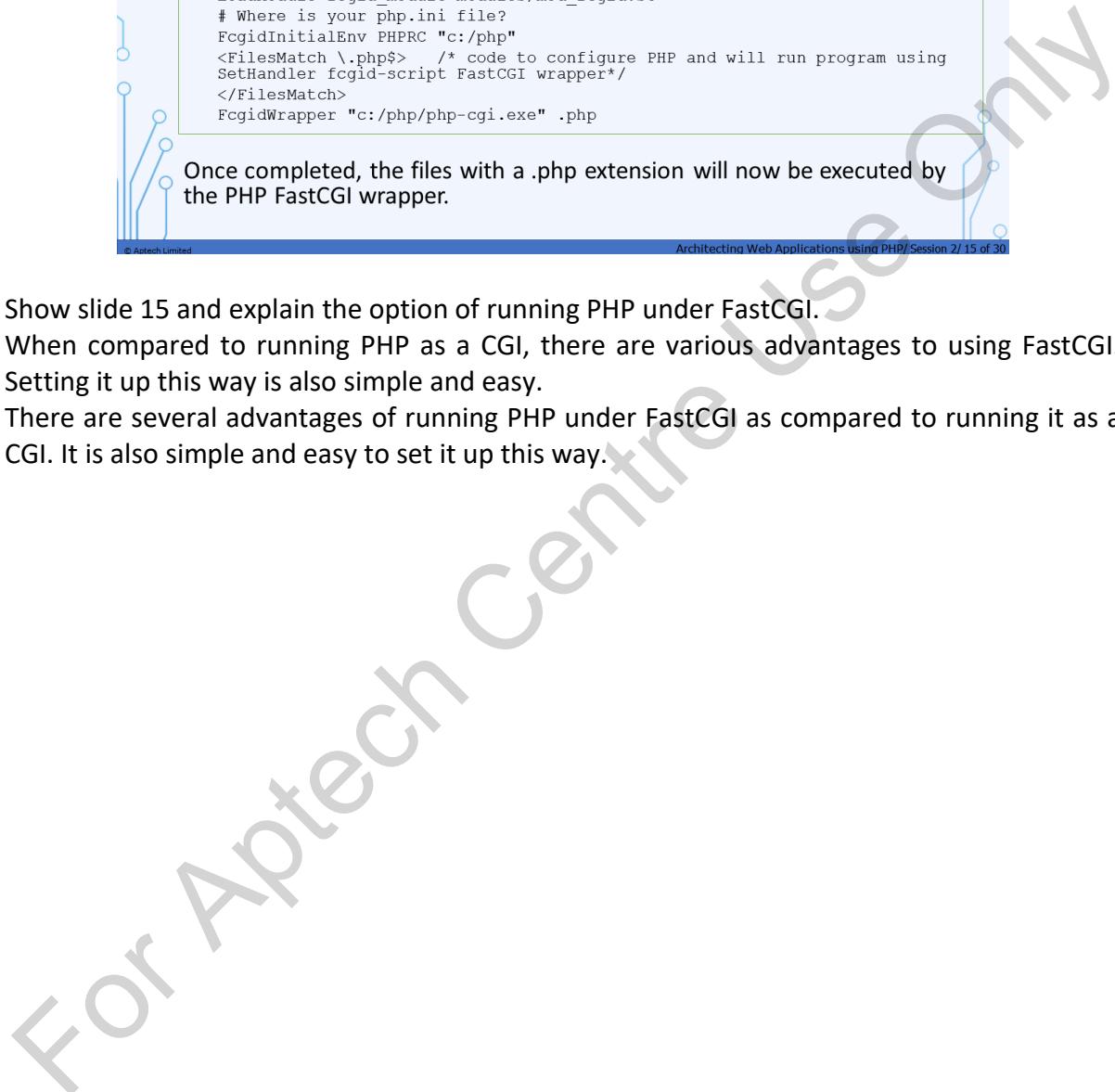
This code shows the location of PHP binary files

© Aptech Limited Architecting Web Applications using PHP/ Session 2/ 14 of 30

Show slide 14 and explain the option of running PHP as CGI along with the code snippet. A `# !` line has to be placed in the PHP files, which must point to the location of the PHP binary. It is recommended that the user must consult Apache CGI documentation for a concise and complete understanding of executing CGI on Apache.

Also, inform the students about following **warning**:

When a server is deployed in CGI mode, it is exposed to many possible vulnerabilities. It is advised that the user must read the CGI security section in the manual to learn how to defend themselves from such attacks.



Option 3: Running PHP under FastCGI

Obtain and install `mod_fcgid` from <https://www.apachelounge.com> and configure the Web server as shown in the following example and also take care to adjust any paths:

```
LoadModule fcgid_module modules/mod_fcgid.so
# Where is your php.ini file?
FcgidInitialEnv PHPRC "c:/php"
<FilesMatch \.php$> /* code to configure PHP and will run program using
SetHandler fcgid-script FastCGI wrapper*/
</FilesMatch>
FcgidWrapper "c:/php/php-cgi.exe" .php
```

Once completed, the files with a .php extension will now be executed by the PHP FastCGI wrapper.

© Aptech Limited Architecting Web Applications Using PHP / Session 2 / 15 of 30

Show slide 15 and explain the option of running PHP under FastCGI.

When compared to running PHP as a CGI, there are various advantages to using FastCGI. Setting it up this way is also simple and easy.

There are several advantages of running PHP under FastCGI as compared to running it as a CGI. It is also simple and easy to set it up this way.

The Apache server can be downloaded from:
www.apache.org/dist/httpd/binaries/win32

It is advised to download the current version of the stable release with the `no_src.msi` extension.

© Aptech Limited

Architecting Web Applications Using PHP / Session 2 / 16 of 30

Show slide 16 and display the URL that can be used to download the Apache server.

Explain the process in detail:

Double-click the installer file to install. C:\Program Files is a common location. The installer will also prompt the user whether to run Apache as a service or from the command line or DOS prompt. Therefore, the user should not install it as a service, as that may cause problems with start up.

The PHP binary archive should be extracted using unzip utility. C:\PHP is a common location. Some of the .dll files should be copied from the PHP directory to the system directory (usually C:\Windows). Refer to the manual to know which files. The user will require `php8ts.dll` for each case. Copy the file corresponding to the Web server module - `C:\PHP\Sapi\php8apache.dll`. to Apache modules directory.

It is recommended to copy either `php.ini-dist` or `php.ini-recommended` to your Windows directory, and rename it to `php.ini`. This file has to be edited to get configuration directives, hence, open this file in a text editor. New users can set error reporting to `E_ALL` within the PHP scripts they will create on their development machines. This will result in all errors being reported by PHP and will help new users debug or troubleshoot faster.

Next, the user has to configure and communicate to the Apache server from where to look for the PHP files and what will be the extension of the PHP files. Usually, `.php` is the standard, but the user can use `.html`, `.phtml`, or so on.

To do this, go to the appropriate HTTP configuration files directory and open `httpd.conf` with a text editor. For example, user may have installed Apache in `C:\Program Files\Apache` directory. In that case, the path for configuration files will be `C:\Program Files\Apache\conf`.

Then, search for the word `DocumentRoot` within the `httpd.conf` file. This word should appear twice. Change both paths to the directory from where to load the PHP files. The user has to add at least one PHP extension directive as shown in the first line of following code:

```
LoadModule php8_module modules/php8apache.dll AddType application/x-httdp-php .php .phtml
```

Additionally, user can also add following line: `AddModule mod_php8.c`

After this, user can stop and restart the WWW service as follows:

- Type **services.msc** in the Run command of Windows or launch Services from **Control Panel**.
- Scroll down the list to World Wide Web Publishing Service, right-click, and select **Stop**.
- Then, right-click and select **Start** to start it again.

If required, user may also restart the computer.

Further, user can launch a text editor and write a basic script to test the PHP installation:

```
<?php  
phpinfo();  
?>
```

Save this file in the Web server's document root as **test.php** and launch it in a browser in the format **http://localhost/test.php** or **http://127.0.0.1/test.php**. User should never launch the PHP file directly from local path, for example, **D:\Php codes\test.php** should not be given in the Address bar of the browser. This is because in that case, it will not be treated as a HTTP request which is required for the file to be processed.

Additional information:

Refer to following link for more information:

<https://www.sitepoint.com/how-to-install-php-on-windows/>



PHP Installation on MacOS X with Apache

Homebrew package management system provides the quickest installation of PHP on MacOS.

- Go to the Homebrew site (<https://brew.sh/>) and install homebrew.
- Use following command to start the installation:
`brew install php`

The user can also refer to following alternative resources for easy-to-install packages and precompiled binaries for PHP on MacOS:

- MacPorts: <http://www.macports.org/>
- Fink: <http://www.finkproject.org/>

© Aptech Limited

Architecting Web Applications Using PHP / Session 2 / 17 of 30

Show slide 17 and display the options to install packages related to PHP installation on Mac OS.

Tell that for the MacOS platform, there are a few pre-compiled and pre-packaged PHP varieties. In a standard configuration setup, these can be quite useful. However, if a database driver or a different secure server are required, users may have to construct PHP and/or the Web server themselves. Whether the user is not really familiar with compiling and building software, it is indeed a good idea to see if there is a pre-built packaged version of PHP with the features users require.

Mac users also have the option of having a pre-built source or a binary installation provided with the platform. In this case, user must only edit the Apache configuration file to update the version and switch on the Web server.

Following steps describe installation process on Mac OS:

Step 1: Open the Apache config file in a text editor as root.

```
sudo open -aTextEdit /etc/httpd/httpd.conf
```

Step 2: Edit the file. Uncomment following lines:

```
LoadModule php8_module  
AddModule mod_php8.c  
AddType application/x-httpd-php .php
```

Step 3: Restart the Web server.

```
sudo apachectl graceful
```

Step 4: Open a text editor and write following basic script:

```
<?php  
phpinfo();  
?>
```

Save it as **test.php** in the Web server's document root.

Step 5: Launch the file in a browser in appropriate format.

Additional information:

Refer to following link for more information:

<https://jasonmccreary.me/articles/install-apache-php-mysql-mac-os-x-catalina/>

Slide 18

The slide features a blue header bar with the title "PHP Installation on Linux and Unix with Apache". Below the title is a light blue rectangular box containing text and a bulleted list. The footer contains copyright and session information.

PHP Installation on Linux and Unix with Apache

PHP can be installed on UNIX variants or Linux. It requires following pre-requisites:

- The PHP source distribution for version 8
<http://www.php.net/downloads.php>
- Latest Apache source distribution
- Database, if required (such as MySQL, Oracle, and so on)
- make utility - you can freely download it at
<http://www.gnu.org/software/make>

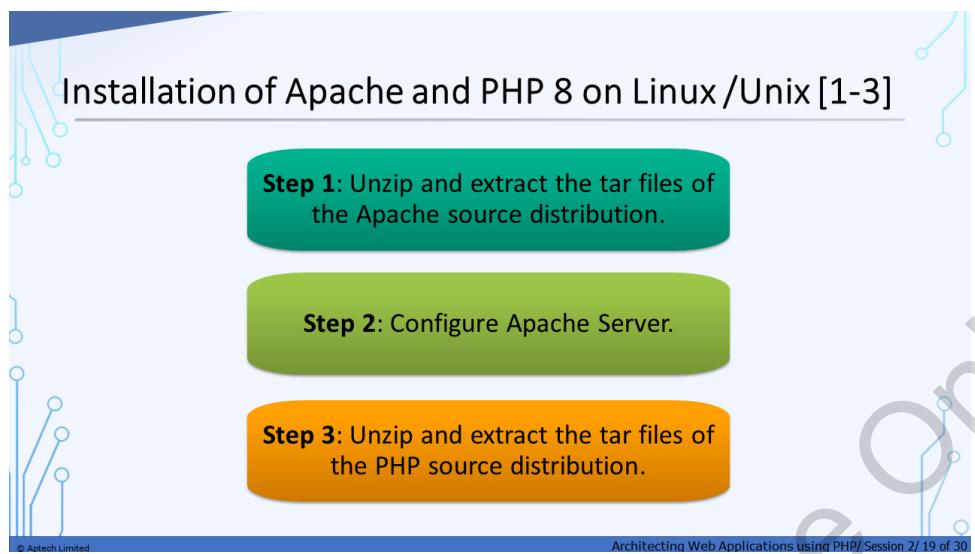
© Aptech Limited Architecting Web Applications using PHP / Session 2 / 18 of 30

Show slide 18 and mention the pre-requisites for PHP installation on Linux or Unix.

Additional information:

Refer to following link for more information:

https://www.tutorialspoint.com/php/php_installation_linux.htm



Show slide 19 and display the installation steps for Apache and PHP 8 on Linux or Unix systems.

Explain the steps to be followed.

First, extract the contents of the downloaded tar files.

Second step to configure Apache server can be done as follows:

```
cd apache_2.4.x  
../configure --prefix=/usr/local/apache --enable-so  
make  
make install
```

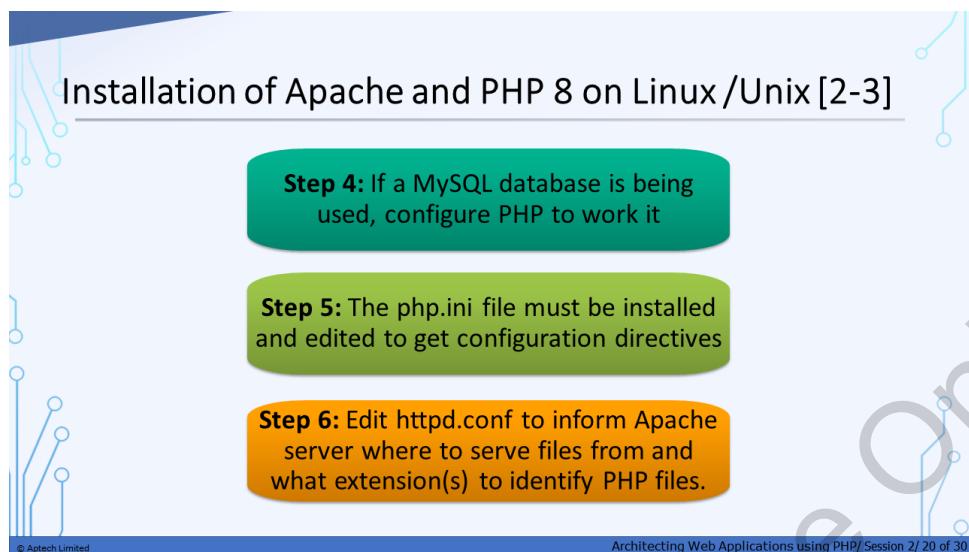
Additional information:

Refer to following links for more information:

<https://linuxize.com/post/how-to-install-php-8-on-ubuntu-20-04/>

<https://computingforgeeks.com/how-to-install-php-on-ubuntu-2/>

<https://www.php.net/manual/en/install.unix.apache2.php>



Installation of Apache and PHP 8 on Linux/Unix [2-3]

Step 4: If a MySQL database is being used, configure PHP to work it

Step 5: The php.ini file must be installed and edited to get configuration directives

Step 6: Edit httpd.conf to inform Apache server where to serve files from and what extension(s) to identify PHP files.

© Aptech Limited Architecting Web Applications using PHP / Session 2 / 20 of 30

Show slide 20 and display the next steps for installation.

Step Four of configuring PHP to work with MySQL database can be done as follows:

```
./configure --with-apxs=/usr/sbin/apxs \
--with-mysql=/usr/bin/mysql
make install
```

Step Five of installing and editing the configuration directives can be done as follows:

```
cp php.ini
/usr/local/lib/php.ini
```

For Step Six, Perform similar steps as the Windows installation for searching DocumentRoot and editing the configuration files.

Installation of Apache and PHP 8 on Linux/Unix [3-3]

Step 7: The user can add at least one PHP extension directive, followed by a second handler to parse all HTML files as PHP.

Step 8: Restart server. Whenever user changes HTTP configuration or php.ini file, the server must be stopped and restarted again.

Step 9: Finally, create and test a simple PHP script similar to how it was described in earlier steps.

© Aptech Limited Architecting Web Applications using PHP/ Session 2/ 21 of 30

Show slide 21 and display the next steps.

Seventh Step can be explained using:

```
AddType application/x-httpd-php .php  
AddType application/x-httpd-php .html
```

After that, users can restart the server and create and test out a simple PHP script.



PHP Basics

PHP is used commonly for building highly dynamic and interactive Web pages for a robust user experience.

It also communicates with the databases and provides with better flexibility and simplicity.

© Aptech Limited

Architecting Web Applications using PHP / Session 2 / 22 of 30

Show slide 22 and elaborate on the working of PHP.

Note: For this course, from this point onwards, it is assumed that XAMPP is installed with PHP 8.0.13 and MySQL on a Windows system and all codes will be run under this environment.

Working of PHP: When a user navigates to a .php page from their Web browser, the browser sends an HTTP request to the Web server. For example, when a user types the URL of the file as index.php in the browser and hits Enter, the browser will send the request to a Web server, and the server will start searching for this file on its file system. If the Web server locates the file, it will send this file to the PHP interpreter. Otherwise, the Web server will generate Error 404 or File Not Found.

The Web server only sends files that have a .php extension to an interpreter. Other files that have extensions such as .html, .htm, and so on are not sent to a PHP interpreter, even if they contain PHP codes inside them.

Once the file is sent to the PHP interpreter, it scans through all the opening and closing PHP tags and then, proceeds with the processing of the PHP code within these tags.

PHP interpreter additionally checks if there is a database connection. If it discovers a database connection, it will send or retrieve data from the database after proper authentication. PHP scripts are interpreted on the Web server and the outcome (HTML) is sent back to the client machine.

Writing PHP Scripts

The basic structure of a PHP script mainly consists of following:

- PHP Start and End Tags
- PHP code with HTML markup
- Comments in PHP (optional)

© Aptech Limited Architecting Web Applications using PHP / Session 2 / 23 of 30

Show slide 23 and explain the basic structure of a PHP script.

Also let students know about various options of text editors which can be used to write PHP code.

There are many good editors available today that provide strong language support and features such as autocomplete, syntax highlighting, and more. Notepad++, Sublime Text, and Atom are some of these editors.

Ask following question to the students. Wait for the response before you answer.

In-Class Question: What is PHP used for?

Answer: PHP is used commonly for building highly dynamic and interactive Web pages for a robust user experience.

Additional information:

Refer to following links for more information:

<https://www.wikihow.com/Write-PHP-Scripts>

https://www.w3schools.com/php/php_syntax.asp

<https://www.lcn.com/support/articles/how-to-create-a-php-script/>

PHP Tags

A PHP code block starts with `<?php` tag and ends with `?>` tag.

Syntax for declaring PHP tags is as follows:

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

© Aptech Limited Architecting Web Applications using PHP/ Session 2/ 24 of 30

Show slide 24 and explain PHP tags.

Additional information:

Refer to following links for more information:

<https://www.php.net/manual/en/language.basic-syntax.phptags.php>

<https://www.w3resource.com/php/syntax/syntax.php>

Ask following question to the students. Wait for the response before you answer.

In-Class Question: What are the PHP tags?

Answer: A set of tags are used to mark the beginning and end of a block of code, in between which any amount of code can be written.

Slide 25

The slide features a decorative background with blue and white circuit-like patterns. At the top center, the title 'PHP with HTML' is displayed. Below the title, a text block states: 'PHP has been designed to work with HTML. Therefore, the user can easily write and embed PHP code within HTML and vice versa.' An 'Example:' label is followed by a code snippet in a code block:

```
<!DOCTYPE html>
<html lang="en">
<title>Hello program in PHP</title>
<body>
<?php echo "Hello, Welcome to PHP"; ?>
</body>
</html>
```

To the right of the code block is a screenshot of a web browser window titled 'Hello program in PHP'. The address bar shows 'localhost/code2_1.php'. The page content displays the text 'Hello, Welcome to PHP'. At the bottom left of the slide is a small copyright notice: '© Aptech Limited'. At the bottom right, it says 'Architecting Web Applications using PHP/ Session 2/ 25 of 30'.

Show slide 25 and elaborate on the code snippet given.

`echo` command of PHP is used, which enables users to write output data to the browser.

Each PHP statement ends with a ; (semicolon). In case you write another statement without completing the first with a semicolon, PHP will report a syntax error.

The output of Code Snippet is shown in the figure.

Additional information:

Refer to following links for more information:

<https://www.javatpoint.com/how-to-use-php-in-html>

<https://www.php.net/manual/en/faq.html.php>

Ask following question to the students. Wait for the response before you answer.

In-Class Question: What is difference between PHP and HTML?

Answer: PHP is a scripting language, whereas **HTML is a markup language**. HTML determines the general structure and content of a Web page, while PHP provides dynamic content through scripts. PHP is typically a server-side language, while HTML is client-side.

Executing PHP Scripts

To run or execute a PHP program, the user must save the code to the Web server under the www or htdocs directory (depending on how the installation is done) with a .php or .html extension (if PHP is embedded within HTML). Once it is done, the server has to be started.

Users can also run or execute a PHP script on the command prompt without the HTML tags.

© Aptech Limited Architecting Web Applications using PHP / Session 2 / 26 of 30

Show slide 26 and explain the execution of PHP scripts with an example.

Assume that XAMPP is installed with PHP 8.0.13. XAMPP will generate an htdocs folder, under which PHP scripts can be placed.

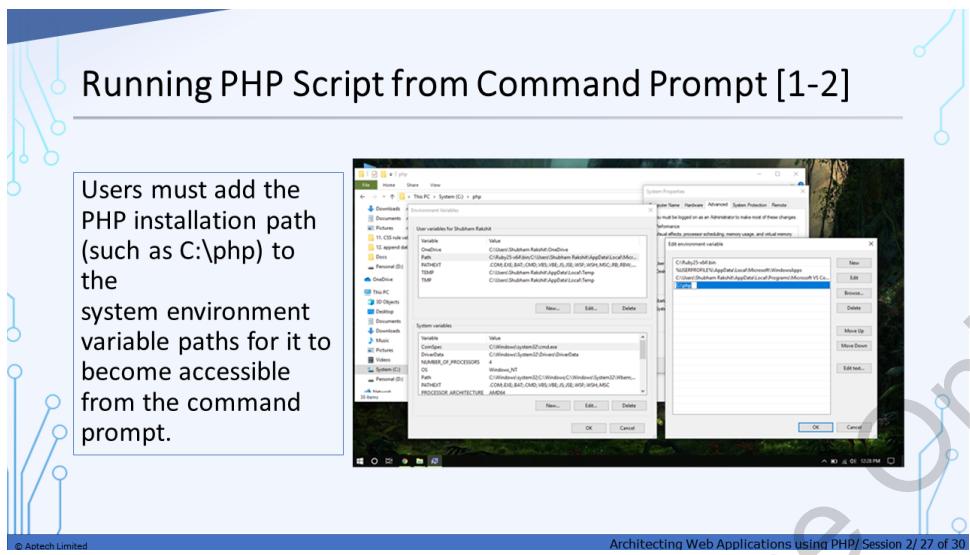
When the server is up and running, the user must open a Web browser, navigate to localhost, and type in the path of the file, for example, <http://localhost/<filename>.php>

The file is saved under the path C:\xampp\htdocs but in the browser, it must be launched as <http://localhost/<filename>.php>.

Ask following question to the students. Wait for the response before you answer.

In-Class Question: What is the difference between JavaScript and PHP?

Answer: PHP is server-side scripting language whereas JavaScript is a **client-side scripting language**. PHP does not execute within browser whereas JavaScript executes within browser. PHP supports database whereas JavaScript does not support databases. PHP accepts both upper case and lower case variables while JavaScript does not.



Users must add the PHP installation path (such as C:\php) to the system environment variable paths for it to become accessible from the command prompt.

Show slide 27 and explain how to add environment variable path in Windows.

To add the path:

- Go to Control Panel. Click **Advanced system settings** link and open **Environment Variables**.
- Click **System Variables** and then, select **Path** to append a new path to the existing Path variable.
- Click **New**, type the path, and click **OK**. Close all the remaining windows after clicking **OK**.

Figure shows adding of path to Environment Variable Path.

Once the Path is set, the php.exe command can be given at the command prompt from any folder, not necessarily the folder where PHP was installed.

The path can be modified to accommodate different installation paths along with PHP installation path by appending to the existing path.

Additional information:

Refer to following links for more information:

<https://stackoverflow.com/questions/12870880/run-php-file-in-windows-command-prompt-cmd>

<https://www.php.net/manual/en/install.windows.commandline.php>

Running PHP Script from Command Prompt [2-2]

Once the environment variable has been set, users can run PHP script from command prompt. Consider the program1.php has a code as shown here:

```
<?php  
echo "Hello, Welcome to PHP!";  
?>
```

The script can be executed from command prompt as follows:

```
D:\php_project> php program1.php  
Hello, Welcome to PHP!
```

© Aptech Limited Architecting Web Applications using PHP/ Session 2/ 28 of 30

Show slide 28 and explain the mentioned example.

Additional information:

Refer to following link for more information:

<https://www.php.net/manual/en/features.commandline.usage.php>

The slide features a decorative background with blue and white circuit board patterns on the left and right sides. The title 'Comments in PHP' is centered at the top. Below the title, a section titled 'Code Snippet:' contains the following PHP code:

```
<html>
<body>
<?php
    // This is a single line commented text, and
    # This is another single line commented text
    /*
        This is
        a Multi-line comment
    */
    echo 'This script made use of comments in PHP';
    ?>
</body>
</html>
```

At the bottom left is the copyright notice '© Aptech Limited' and at the bottom right is the page information 'Architecting Web Applications Using PHP / Session 2 / 29 of 30'.

Show slide 29 and explain multiple ways in which code can be commented in PHP with the help of a code snippet.

Tell that:

Writing comments in a program is an important and good practice, as it makes code readable as well as easy to understand for developers. Consider that a developer Mark employed with a company has written considerable code for some product applications. Now, Mark has left his job and his code has been given to another developer Peter to continue further. However, Peter finds it difficult and cumbersome to understand what Mark has been doing in the code because there was no documentation and no commenting. Had Mark used comments suitably in his code, it would have been a seamless process for Peter to take over the code. Thus, comments play an important role in documenting code.

The PHP interpreter ignores execution of comment blocks, thereby making code readable with no overheads. Therefore, comments can be used anywhere in a program to add information about code blocks.

In PHP, // or # can be used to generate a single-line comment and /* with */ to make a large comment block with multiple lines.

Code Snippet shows an example of a PHP program that shows single and multiple line comments in PHP.

Additional information:

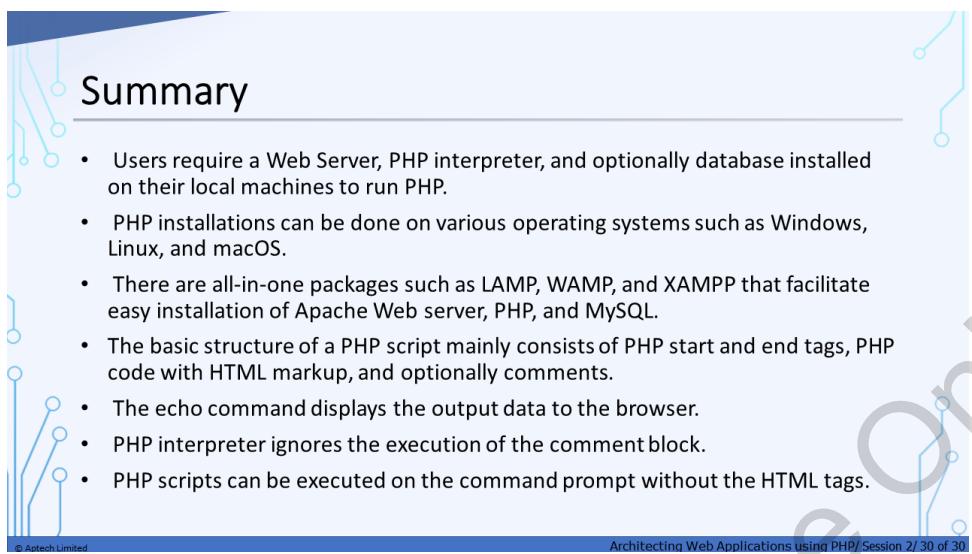
Refer to following links for more information:

https://www.phptutorial.net/php-tutorial/php-comments/#:~:text=PHP%20supports%20both%20one%2Dline,%2F*%20and%20end%20with%20*%2F%20.

https://www.w3schools.com/php/php_comments.asp

<https://www.javatpoint.com/php-comments>

<https://www.geeksforgeeks.org/how-to-write-comments-in-php/>



Summary

- Users require a Web Server, PHP interpreter, and optionally database installed on their local machines to run PHP.
- PHP installations can be done on various operating systems such as Windows, Linux, and macOS.
- There are all-in-one packages such as LAMP, WAMP, and XAMPP that facilitate easy installation of Apache Web server, PHP, and MySQL.
- The basic structure of a PHP script mainly consists of PHP start and end tags, PHP code with HTML markup, and optionally comments.
- The echo command displays the output data to the browser.
- PHP interpreter ignores the execution of the comment block.
- PHP scripts can be executed on the command prompt without the HTML tags.

© Aptech Limited Architecting Web Applications using PHP/ Session 2/ 30 of 30

Use slide 30 to summarize the session. You will end the session with a summary of what has been taught in the session. Tell students the pointers of the session. This will be a revision of the current session.

2.3 Post Class Activities for Faculty

You should familiarize yourself with the topics of the next session.

Tips:

You can also check the Articles/Blogs/Expert Videos uploaded on the Online Varsity site to gain additional information related to the topics covered in the next session.

Session 3 – PHP Data Types and Strings

3.1 Pre-Class Activities

Before beginning the session, you should fully familiarize yourself with its topics. Prepare a question or two that will be a key point to relate the current session objectives.

3.2 Teaching Skills

To teach this session, you should be well-verses with the concept of Data Types and Strings in PHP.

You must use the given images and teach the concepts in the theory class. For teaching in the class, you are expected to use slides and LCD projectors.

Tips:

It is recommended that you test the understanding of the students by asking questions in between the class.

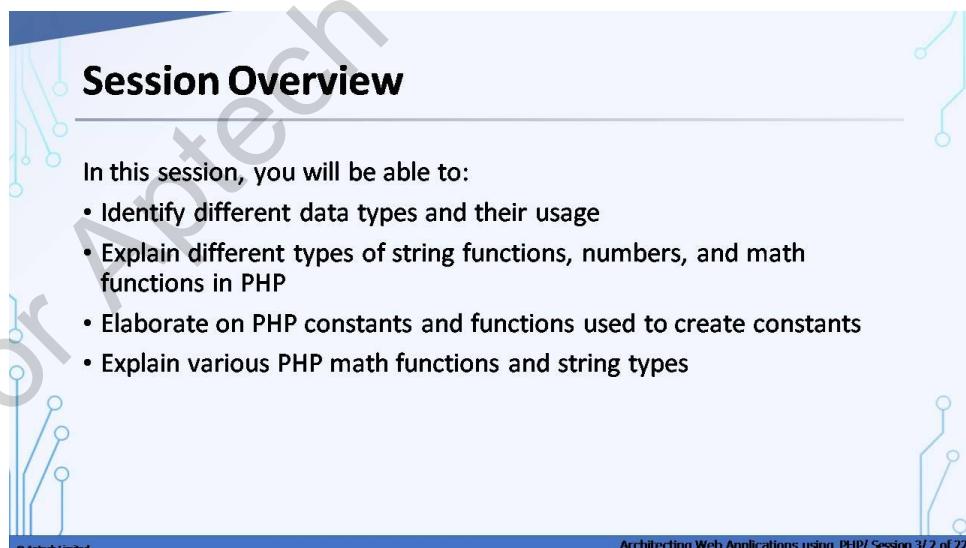
In-Class Activities

Follow the order given here during In-Class activities.

Overview of the Session

Give the students an overview of the current session in the form of session objectives. Read out the objectives on slide 2.

Slide 2



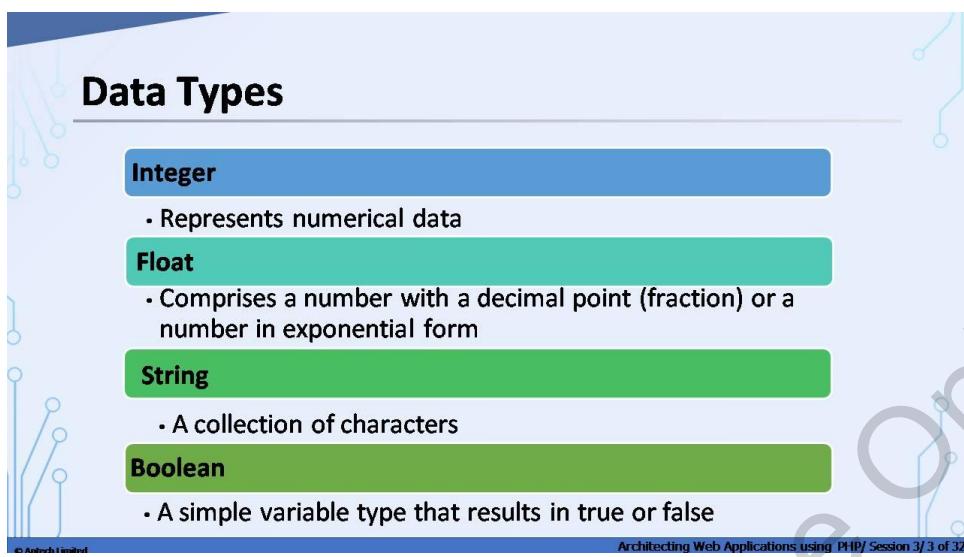
The slide has a blue header bar with the title "Session Overview". The main content area is white with a light gray border. On the left and right sides, there are decorative vertical columns made of blue and cyan lines forming a circuit board-like pattern. At the bottom left is the "Aptech Limited" logo, and at the bottom right is the text "Architecting Web Applications using PHP / Session 3 / 2 of 22".

Session Overview

In this session, you will be able to:

- Identify different data types and their usage
- Explain different types of string functions, numbers, and math functions in PHP
- Elaborate on PHP constants and functions used to create constants
- Explain various PHP math functions and string types

Show slide 2 and give students a brief overview of the current session and the session objectives. Inform students that this session explains the concept of Data Types along with their usage, different types of string functions, numbers, and math functions in PHP, along with the constants and functions which are used to create constants and at last discuss about the string types in PHP.



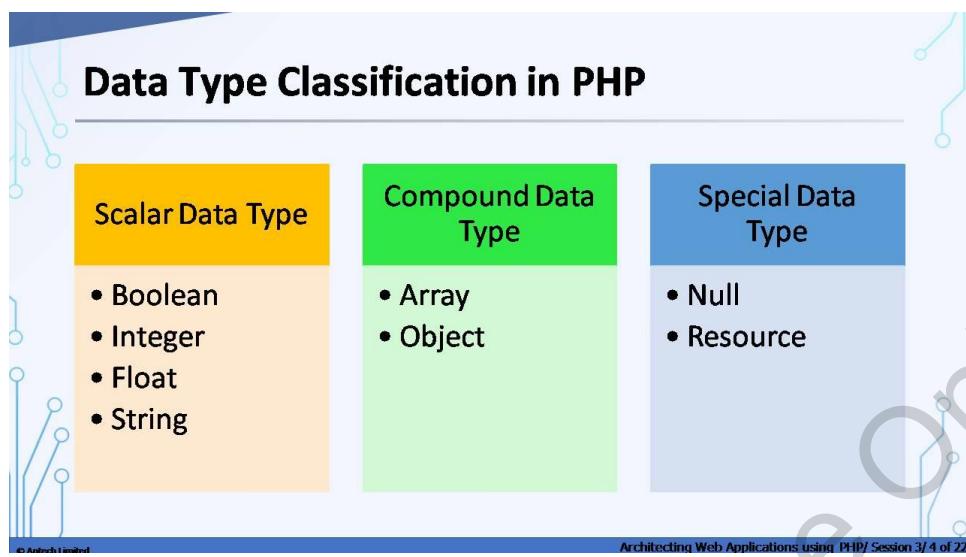
The slide has a blue header bar with the title 'Data Types'. Below the header, there are four colored boxes: blue for Integer, teal for Float, green for String, and light green for Boolean. Each box contains a definition of the data type.

Data Type	Description
Integer	Represents numerical data
Float	Comprises a number with a decimal point (fraction) or a number in exponential form
String	A collection of characters
Boolean	A simple variable type that results in true or false

At the bottom right of the slide, it says 'Architecting Web Applications using PHP/ Session 3 / 3 of 32'.

Show slide 3 and introduce the topic to the students. Tell students that there are four types of Data Types in PHP and give a brief introduction about each one of them.

- Start with the concept of Integer, Inform that this data type represents numerical data. For example 12, 2345, and so on.
- Float data type contains number with a decimal or in fraction form. For example 12.34, 698.78, and so on.
- Explain that String is a collection of characters. For example 'Hello World'.
- Boolean that is a simplest variable type that can results in true or false.



Show slide 4 and tell students about the classification of Data Types in PHP.

Explain students that there are three Data Types in PHP:

1. Scalar Types: The Scalar data type is the most basic data type that holds only a single atomic value at a time. The four scalar data types are integer, float, Boolean, and string.
2. Compound Types: A Compound type is a type that is defined in terms of another type, it means compound types are types like which are using references and pointers. The two compound data types are Array, Object.
3. Special Types: The Special data type is not an actual data type. It is for storing a reference to functions and resources external to PHP. The two Special data types are Null and Resource.

In-Class Question: Give an example of using resource data type.

Answer: A common example of using the resource data type is a database call.

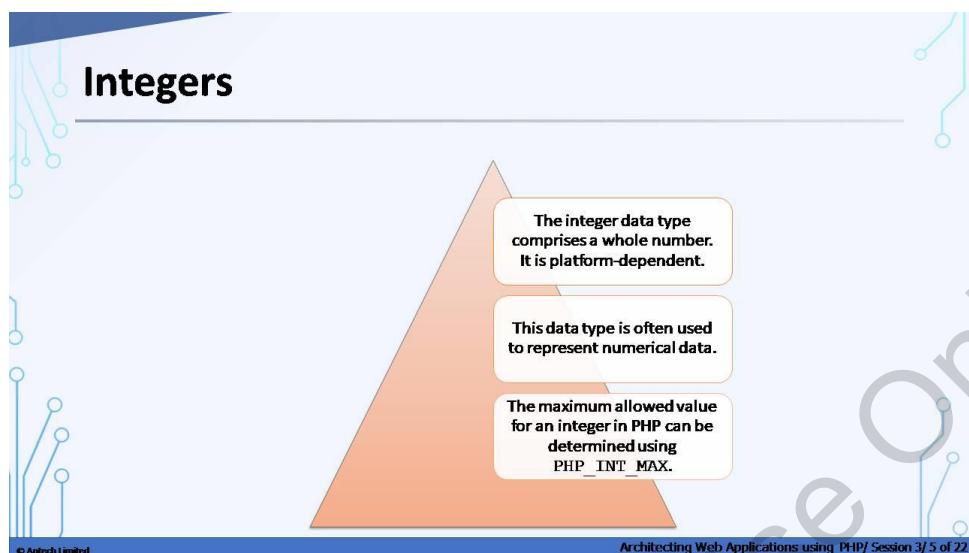
Additional Links:

Refer to following links for more information:

https://www.w3schools.com/php/php_datatypes.asp

<https://www.geeksforgeeks.org/php-data-types/>

Slide 5



Show slide 5 and explain to the students in detail the concept of Integers. A non-decimal number between -2,147,483,648 and 2,147,483,647 is an integer data type.

Integer rules are as follows:

- At least one digit is required for an integer.
- A decimal point cannot be used in an integer.
- A positive or negative integer is a number that can be either positive or negative.
- Decimal (base 10) notation, hexadecimal (base 16), octal (base 8) notation, and binary (base 2) notation can all be used to specify integers.

In following example \$a is an integer.

Note: The PHP var_dump () function returns the data type and value:

```
<?php  
$x = 4000;  
var_dump($a);  
?>
```

Its Output will be: int(4000)

In-Class Question: What do you mean by PHP_INT_MAX?

Answer: The largest integer supported in this build of PHP. Usually int(2147483647) in 32 bit systems and int(9223372036854775807) in 64 bit systems.

Float

- A float data type comprises a number with a decimal point (fraction) or a number in exponential form.
- For example, 256.4, 10.358, 9.8, 7.64E+5, 5.56E-5, and so on.
- It is also called 'double' or 'real number'.
- The float data type can have a value as high as 1.7976931348623E+308, which varies with the platform. The value has a maximum precision of 14 digits.

© Aptech Limited Architecting Web Applications using PHP / Session 3 / 6 of 22

Show slide 6 and explain students the concept of Float in PHP tell students that the float data type in PHP represents any number that can contain a fraction. The fractional component can have digits after the decimal point or be written in scientific notation with the letters e or E. In scientific notation, the number 100 is written as 10e2.

- When decimals or very large numbers are encountered, PHP will automatically convert the number to the float type. The float type is widely used to store numbers having a magnitude of about 1.7976931348623E+308. This, however, is platform-dependent.
- Although 1.7976931348623E+308 appear to be a huge number, floats only have a maximum precision of roughly 14 digits. Any number that has more than that will lose precision. That implies you can hold a large number but not the information about its exact value-in many cases, a float is merely a rough guess.

Explain the given code snippet of float in PHP.

```
<?php  
$x = 11.75;  
var_dump(is_float($x));  
?>
```

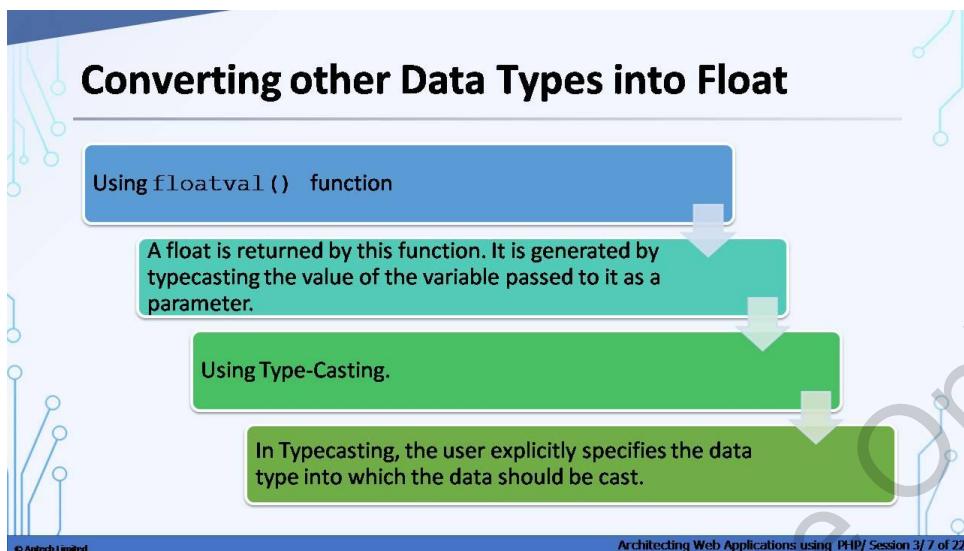
Given code represents \$x a variable which has value in float data type as 11.75 , while running this code get answer as true because yes it is a float.

Additional Information:

Refer to following links for more information:

<https://www.php.net/manual/en/language.types.float.php>

<https://www.javatpoint.com/php-float>



Show slide 7 and explain to the students that there are several ways to convert other data types into float:

- `floatval()` : The `floatval()` method returns the float value of a variable and is an inherent function in PHP.
- Syntax: `float floatval ($var)`
- Parameters: There is only one mandatory parameter for this function, which is as follows: `$var`: This is the variable whose float value will be returned. An object should not be used for this variable. It returns the float value of the provided variable as a return value. Non-empty arrays yield 1, while empty arrays return 0.

The program demonstrate how to utilize PHP `floatval()` function:

```
<?php
$var = '14.75983E4';
$float_value = floatval($var);
echo $float_value;
?>
```

Output: 14759.83

- Type-Casting: The term ‘type casting’ refers to the use of a variable’s value with a different data type. In other words, typecasting is a technique for converting one data type variable into another. Since the user specifies the data type in which user wants to cast, typecasting is an explicit data type conversion.

For Example:

```
<?php
$val = 1;
var_dump($val); // $val is integer
$val = 2.3;
var_dump($val); // $val is float
$val = "php type casting";
var_dump($val); // $val is string
?>
```

In example, understand that variable `$i` type is getting changed on the different type of value assignment. Due to this flexible nature of the PHP, it does not require type cast variable

always. However, sometimes as per the requirement, an extra security in the variable cast type is required.

Slide 8

Boolean

The simplest variable type is Boolean, which acts as a switch. It specifies a truth value that can be either true or false.

Booleans are frequently used in conditional statements; true if the condition is true, false if the condition is false.

To represent a bool literal, use the PHP constants true or false (both are case-insensitive).

© Aptech Limited

Architecting Web Applications using PHP/ Session 3/ 8 of 22

Show slide 8 and explain to the students that Boolean data type in PHP can only have one of two values: true or false. TRUE and FALSE are two Boolean constants provided by PHP and they can be used in the same way.

- The Boolean value operators in programming are as follows: >= – True if a number is greater than or equal to another. == – True if two values are equivalent.

Given code illustrated that how Boolean data type works:

```
<?php  
$height=100;  
$width=50;  
if ($width == 0)  
{  
echo "The width needs to be a non-zero number";  
}  
?>
```

In the given example, it would be false and, therefore, the echo statement will never execute.

PHP Array

- An array is a single variable that contains values of the same data type. In PHP, an array is a predefined map that connects values to keys.
- An array, list (vector), stack, hash table (a map implementation), collection, dictionary, queue, and so on can all be used to represent this type.
- The `array()` language construct can be utilized to create an array.

© Aptech Limited

Architecting Web Applications using PHP/ Session 3/ 9 of 22

Show slide 9 and explain the students about the Arrays in PHP tell students that:

- Arrays are a form of data structure in PHP that allow to store several pieces of the same data type in a single variable, saving the time and effort of having to create a new variable for each data type. The arrays are useful for creating a list of similar-type elements that can be accessed by their index or key.
- Lets assume to save five names and print them out. The usage of five separate string variables makes this simple. However, if the number of variables increases from five to a hundred, the user or developer will find it quite difficult to construct so many different variables. Here, an array comes into play, allowing to store all of the elements in a single variable while also allowing for quick access by an index or key.
- An array is created using an `array()` function in PHP.

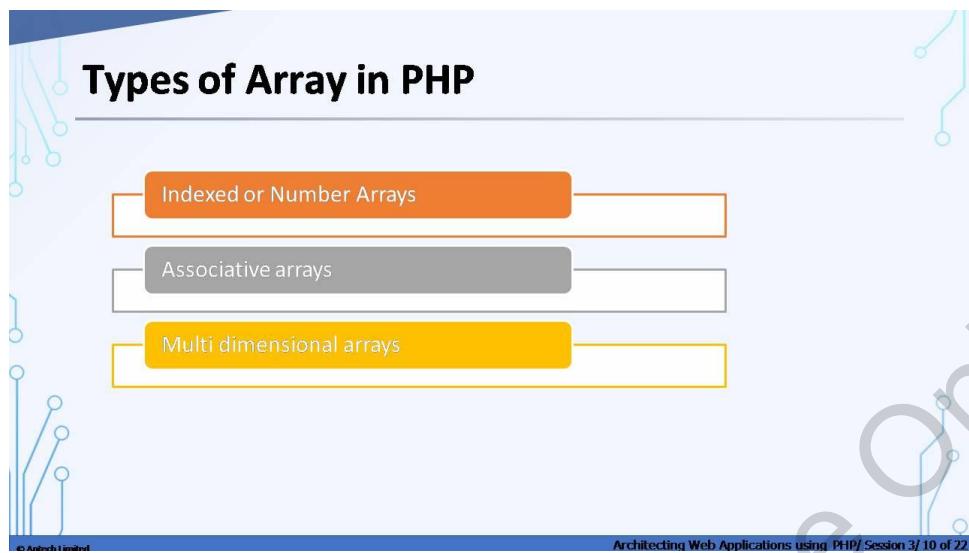
Syntax of Array in PHP:

```
<?php  
$subjects = array("Physics", "Chemistry", "Maths");  
echo "I like " . $subjects[0] . ", " . $subjects[1] . " and " .  
$subjects[2] . ". ";"  
?>
```

Output: I like Physics, Chemistry and Maths.

In-Class Question: What is Index?

Answer: Accessing an array element is the same as indexing it. By referring to the index number of an array element, you can access it. NumPy arrays have indexes that begin with 0, therefore the first element has index 0 and the second element has index 1 and so on.



Show slide 10 and explain the students about the types of Arrays in PHP:

In PHP, there are three primary types of arrays:

- **Indexed or Number Arrays:** Values are stored linearly in an array with a numeric index.
For Example:

```
<?php
$subjects = array("Physics", "Chemistry", "Maths");
echo "I like " . $subjects[0] . ", " . $subjects[1] . " and
" . $subjects[2] . ".";
?>
```

Output: I like Physics, Chemistry and Maths.
- **Associative Arrays:** An array with a string index in which each value can be assigned a unique key instead of linear storage.
For Example:

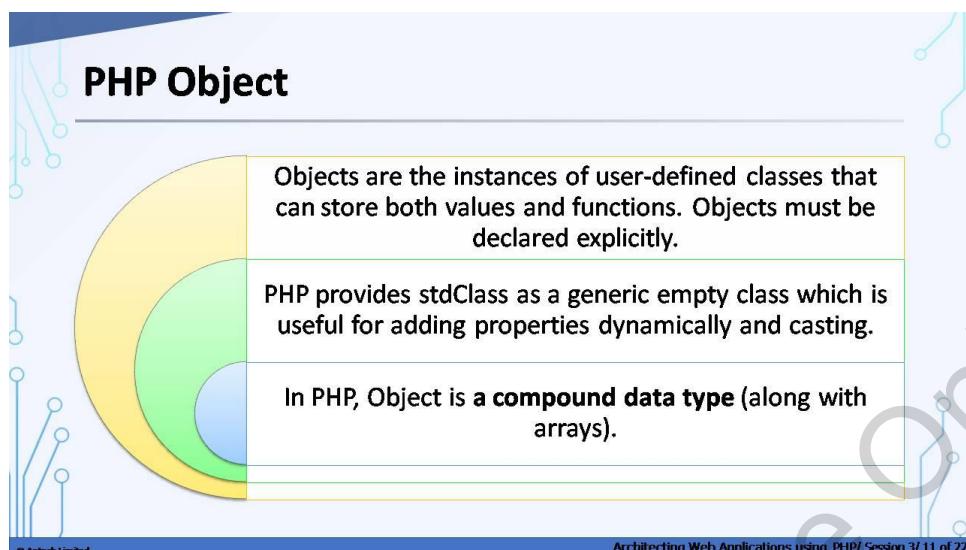
```
<?php
$age = array("Ross"=>"35", "Peter"=>"37", "Leo"=>"43");
echo "Ross is " . $age['Ross'] . " years old.";
?>
```

Output: Ross is 35 years old.
- **Multidimensional Arrays:** An array that may be accessed via several indices and contains a single or numerous arrays.
- The dimension of an array indicates the number of indices required to select an element.
- For a two-dimensional array, two indices are required to select an element.
- For a three-dimensional array, three indices are required to select an element.

For Example:

```
<?php
$myarray = array(
array("Ross", "Peter", "Leo"),
array("Unnao", "Trichy", "London")
);
print_r($myarray)
?>
```

Output: Array ([0] => Array ([0] => Ross [1] => Peter [2] => Leo)
[1] => Array ([0] => Unnao [1] => Trichy [2] => London))



PHP Object

Objects are the instances of user-defined classes that can store both values and functions. Objects must be declared explicitly.

PHP provides stdClass as a generic empty class which is useful for adding properties dynamically and casting.

In PHP, Object is a **compound data type** (along with arrays).

© Aptech Limited

Architecting Web Applications using PHP | Session 3 / 11 of 22

Show slide 11 and explain the concept of Object in PHP:

- An object is a class instance. The new keyword can be used to create a new instance of the class. Multiple instances of the class can be created. These instances can now use the functions and members of the class.

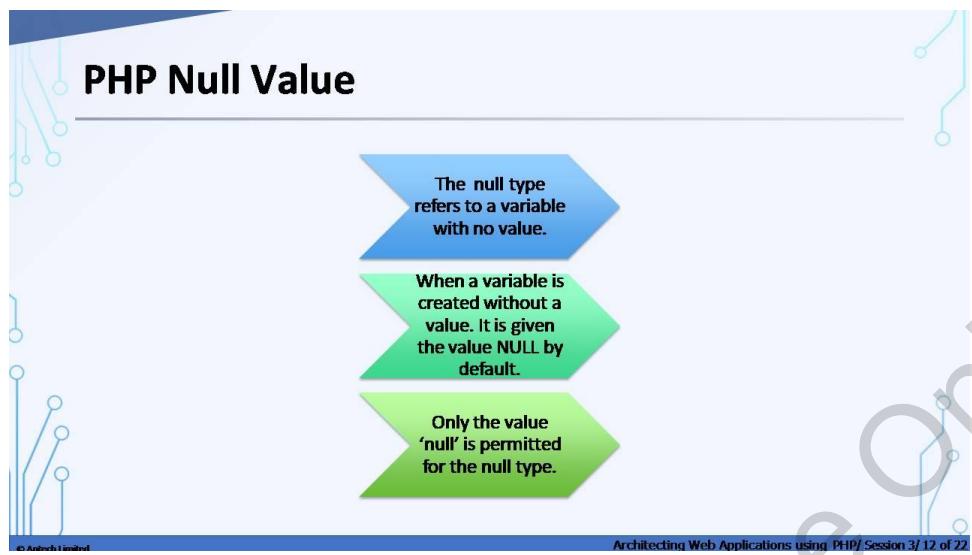
How to Create an Object?

Creating an object is the same as instantiating a class. This instance is created using the new keyword. This process is called instantiation. Since objects are instances of a class and can be created using a new keyword, take a look at how these instances are created.

```
Syntax: objectname = new Classname();  
$parrot = new Bird();  
$pigeon = new Bird();
```

Mentioned are two different objects of the class Bird. Using these objects, properties, and functions of class Bird() can be accessible.

Slide 12



Show students slide 12 and explain the PHP Null Value concept:

- A variable with no value is represented by the null value.
- Type null has only one potential value as null.
- If any of the following conditions apply to a variable, it is deemed null. The constant null has been assigned to it. It has yet to be assigned a value. It has been unset () .

Syntax:

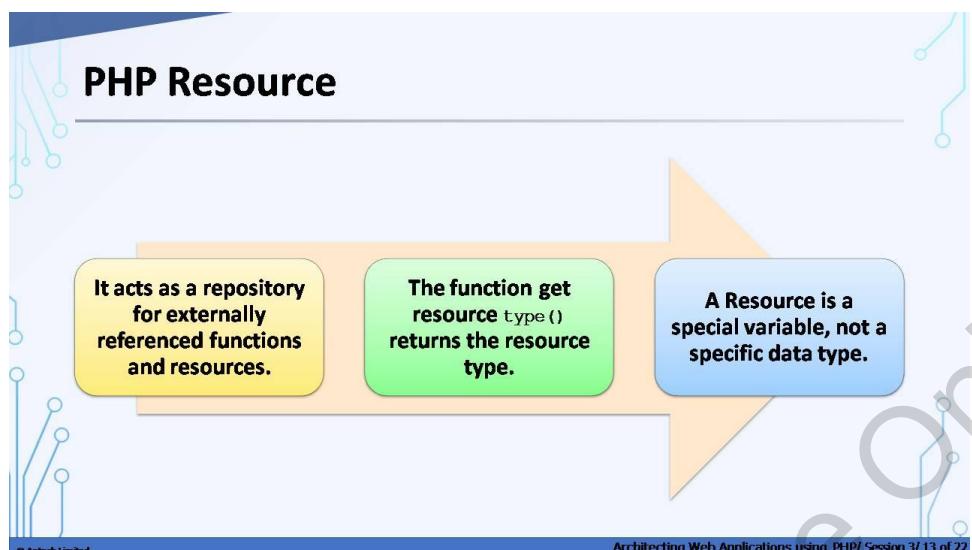
```
<?php  
$var = NULL;  
?>
```

Additional Links:

Refer to following links for more information:

<https://www.php.net/manual/en/language.types.null.php>

https://www.w3schools.com/php/php_datatypes.asp



Show slide 13 and explain them the concept of PHP Resource:

Any external resource is referred to using the data type resource. A resource variable is a pointer to an external data source, such as a stream, file, or database. To construct these resources, PHP employs relevant functions. As a result, memory utilised by resource data types does not require to be manually removed.

PHP String Functions

Functions	Definitions
<code>strlen()</code>	Return the length of the string
<code>str_word_count()</code>	The function returns the number of words in a string.
<code>strrev()</code>	It is a predefined function used to reverse a string.
<code>strpos()</code>	function is used to find a specified text within a string.
<code>str_replace()</code>	Function to replace characters in a string.
<code>ucwords()</code>	Function returns string converting first character of each word into uppercase.

© Aptech Limited Architecting Web Applications using PHP / Session 3 / 14 of 22

Show slide 14 and introduce them with all the String Functions in PHP.

- `strlen()` : A string's length is returned using the `strlen()` method.

For Example: <?php

```
echo strlen("Hello from PHP!");  
?>
```

Output: 15

Note: Whitespaces and exclamation marks are also a part of the string.

- `str_word_count()` : The `str_word_count()` method determines how many words are in a string.

For Example: <?php

```
echo str_word_count("Hello from PHP!", 1));  
?>
```

Output: Array ([0] => Hello [1] => from [2] => PHP)

- `strrev()` : The `strrev()` function reverses a string.

For Example: <?php

```
echo strrev("Hello Jack!");  
?>
```

Output: !kcaj olleH

- `strpos()` : The `strpos()` method determines the position of a string's first occurrence within another string.

It is worth noting that the `strpos()` function is case-sensitive . It's worth noting that this function is binary-safe.

Associated functions include:

- `strrpos()` - Returns the position of a strings last occurrence within another string (case-sensitive)
- `stripos()` – Locates the first occurrence of a string within another string (case-insensitive)
- `strripos()` – Returns the position of a string's last occurrence within another string (case-insensitive)

For Example: Find the position of the first occurrence of 'Chocolates' inside the string.

```
<?php  
echo strpos("I love Chocolates, I love Chocolates too!",  
"Chocolates");  
?>
```

Output: 7

- **str_replace()**: In a string, the `str_replace()` method substitutes certain characters with other characters .

Following are the rules that govern this function:

- It returns an array if the string to be searched is an array.
- If the string to be searched is an array, then, find and replace is applied to each element of the array.
- Replace will be an empty string if both find and replace are arrays and replace has fewer elements than find.
- If replace is a string and find is an array, the replace string will be applied for each find value.
- It is worth noting that this function is case-sensitive.
- To execute a case-insensitive search, use the `str_replace()` function.
- It is worth noting that this function is binary-safe.

For Example: Replace the characters 'Country' in the string 'Hello Country!' with 'Texas':

```
<?php  
echo str_replace("Country", "Texas", "Hello Country!");  
?>
```

Output: Hello Texas!

- **ucwords()**: The `ucwords()` function converts the first character of each word in a string to uppercase. It is worth noting that this function is binary-safe.

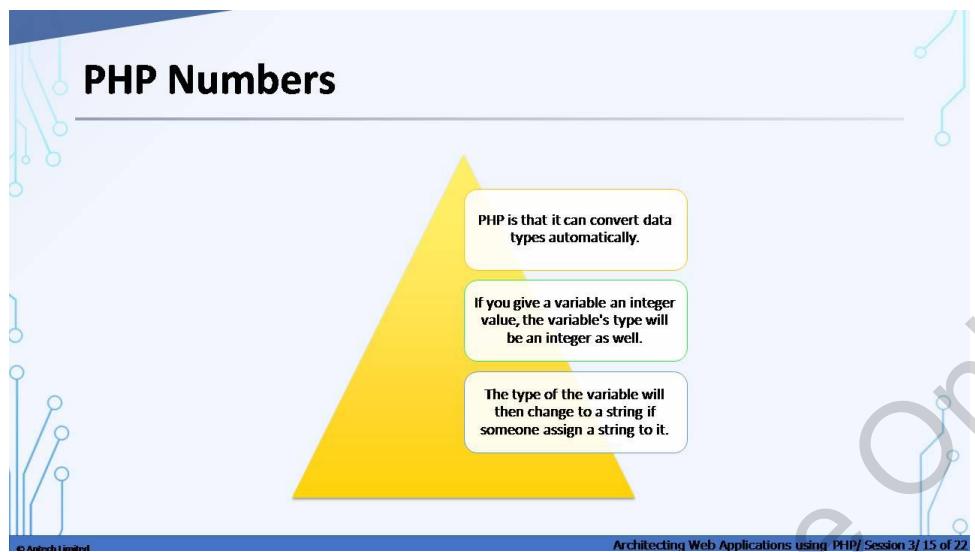
Associated functions include:

- `ucfirst()` - transforms a string's first character to uppercase.
- `lcfirst()` - is a function that lower cases the first character of a string.
- `strtoupper()` - turns a string to uppercase, while `strtolower()` converts it to lowercase.

For Example: Convert the first character of each word to uppercase, with a custom word separator added:

```
<?php  
echo ucwords("hello|texas", " |");  
?>
```

Output: Hello|Texas



Show slide 15 and explain the concept of PHP Numbers:

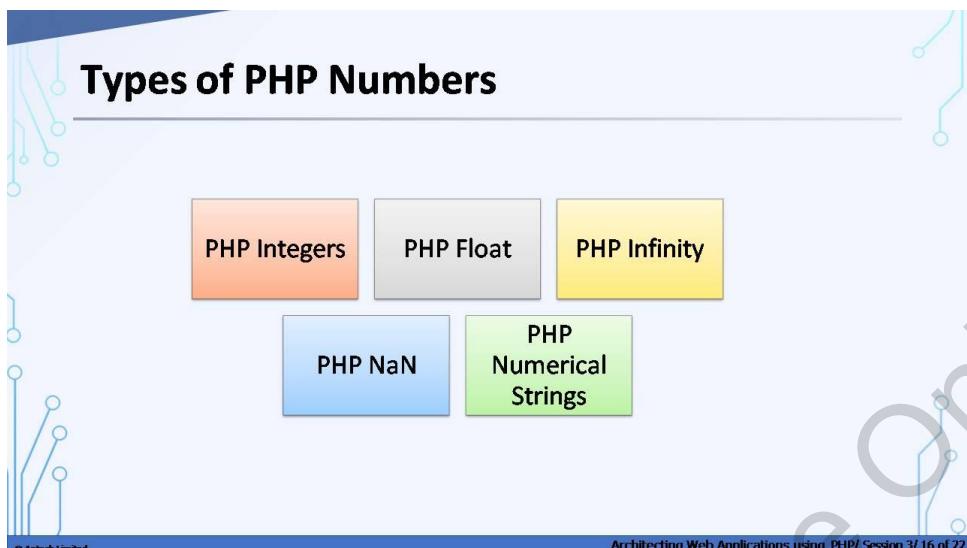
- At first glance, it appears simple because PHP performs automated type conversion. For example, user can give a variable an integer value, and the variable's type will be integer. User can assign a string to the same variable on the next line and the type will change to string. Unfortunately, this automatic conversion can occasionally cause your code to break.
- When user assign an integer value to a variable, that variable's type is immediately changed to an integer. The type of the variable will change to string if user assigns a string to it.
- PHP Numbers includes all the Data Types such as Integer, Float, Infinity, and NaN.
- Numerical Strings in PHP to determine whether a variable is numeric, use the `is_numeric()` function. If the variable is a number or a numeric string, the function returns true; otherwise, it returns false.

Additional Links:

Refer to following links for more information:

https://www.w3schools.com/php/php_numbers.asp

<https://www.php.net/manual/en/language.types.integer.php>



Show slide 16 and tell students that there are five types of PHP Numbers:

- PHP Integers: In PHP, the integer is the most basic type of number. Integers, as user may know, are numbers that have no decimal portion. For example, the numbers 2 and 235298 or -235298 are both integers.
- PHP Float: A Float is the second most common form of number user will encounter. Unlike integers, which are often just numbers with no decimal points, float numbers can be expressed in a variety of ways. All of the numbers are floats: 3.14, 12.0, 5.87E+10, and 3.56E-5. There are two functions that can be used to determine whether a value is a float or not. `is_float()` and `is_double()` are these functions. `Is_double()` is actually simply an alias for `is_float()`, so you can use either one to get the same answer.
- PHP Infinity: In PHP, infinity is not the same as infinity in real life. Any numerical number on a platform that exceeds PHP FLOAT MAX is considered infinite in PHP. On var dump, `1.8e308` will give you `float(INF)`. The `is_finite()` and `is_infinite()` functions can be used to determine whether a numerical value is finite or infinite.
- PHP Nan: Although NaN stands for Not a Number, it does not check whether or not a value is numerical. The value NaN is used to represent the outcome of mathematical operations that are not possible. For instance, `log(-1)` is NaN. In the same way, `acos(5)` will be NaN. The function `is_nan` can be used to determine if a value returned by a mathematical operation is not a number.
- PHP Numerical Strings: In the same way that PHP dynamically changes the type of different numbers depending on how their values are used or allocated, it can also infer the value of different numerical strings so that you can convert them to numbers. The function `is_numeric()` can be used to determine whether or not a string or variable is numeric. For numbers written in octal, binary, or hexadecimal notation, this method returns true. If the numbers are represented in exponential notation, such as `+16.52e39`, it will also return true.

The approximate value of PI is returned by `pi()`.

`min()` function returns the lowest value.

`max()` function is used to find the highest value.

© Aptech Limited

Architecting Web Applications using PHP / Session 3 / 17 of 22

Show slide 17 and explain students about the Math Functions in PHP. The math functions can work with values of both integer and float types.

Following are the Math functions explained:

- `pi()` : The value of PI is returned by the `pi()` function.

For Example: `<?php`

```
echo(pi())  
?>
```

Output: 3.1415926535898

- `min()` : The `min()` function returns the lowest value in an array or the lowest value among a set of values.

For Example: `<?php`

```
echo(min(2, 4, 6, 8, 10))  
?>
```

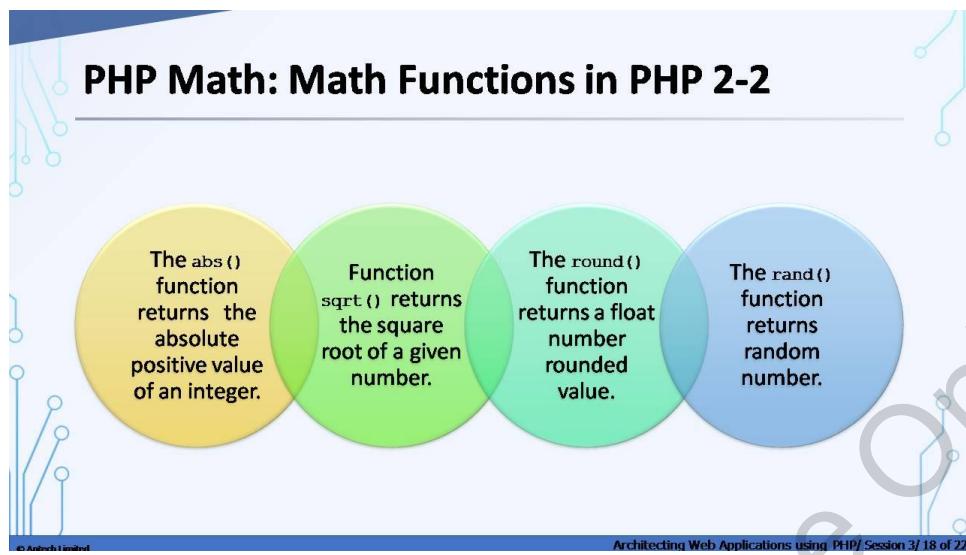
Output: 2

- `max()` : The `max()` function returns the highest value in an array, or the highest value of several specified values.

For Example:

```
<?php  
echo(max(2, 4, 6, 8, 10))  
?>
```

Output: 10



Show slide 18 and continue to explain students about the remaining math functions in PHP:

- `abs()` : The `abs()` function returns the absolute (positive) value of a number.

For Example: <?php

```
echo (abs (-6.7))  
?>
```

Output: 6.7

- `sqrt()` : The `sqrt()` function returns the square root of a number.

For Example: <?php

```
echo (sqrt (2))  
?>
```

Output: 4

- `round()` : The `round()` method rounds a floating-point value.

Look at the `ceil()` function to round a value UP to the closest integer.

Look at the `floor()` function to round a value DOWN to the nearest integer.

For Example: <?php

```
echo (round(0.60))  
?>
```

Output: 1

- `rand()` : A random integer is generated by the `rand()` function.

Use `rand` to generate a random integer between 10 and 100 (inclusive) as an example (10,100).

For Example: <?php

```
echo (rand(10,100));  
?>
```

Output: 78



PHP Constants

Constants are similar to variables, where they cannot be altered or undefined after they have been defined.

A constant is a term that identifies a single value. The value of the constant cannot be changed while the script is running.

The constant name can only begin with a letter or an underscore (\$ sign should not be before the constant name).

Constants, unlike variables, are automatically global over the whole script.

Architecting Web Applications using PHP / Session 3 / 19 of 22

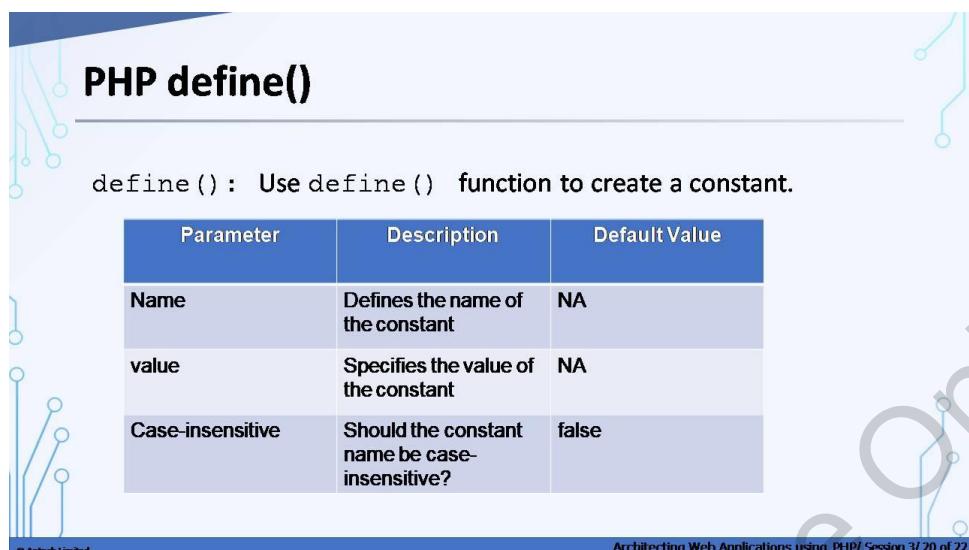
Show slide 19 and explain students the concept of PHP Constants.

- A constant is a term that identifies a single value. During the script, the value cannot be altered.
- A valid constant name begins with a letter or underscore (no \$ sign before the constant name) and ends with a letter or underscore. Constants, unlike variables, are always global over the whole script.
- Constants are Global. Constants are automatically global and can be used across the entire script.

For Example:

```
<?php  
    define("GREETING", "Welcome to Texas!");  
    function myTest() {  
        echo GREETING;  
    }  
    myTest();  
?>
```

Output: Welcome to Texas!



PHP define()

define(): Use define() function to create a constant.

Parameter	Description	Default Value
Name	Defines the name of the constant	NA
value	Specifies the value of the constant	NA
Case-insensitive	Should the constant name be case-insensitive?	false

© Aptech Limited Architecting Web Applications using PHP / Session 3 / 20 of 22

Show slide 20 and explain students how to create constants in PHP:

To create a constant, use the `define()` function.

Syntax:

```
define(name, value, case-insensitive)
```

Parameters:

name: Specifies the name of the constant

value: Specifies the value of the constant

case-insensitive: Specifies whether the constant name should be case-insensitive. Default is false

Magic Constants

There are nine magic constants in PHP. In which eight magic constants start and end with double underscores (__).

Magic Constants:
__LINE__
__FILE__
__DIR__
__FUNCTION__
__CLASS__
__TRAIT__
__METHOD__
__NAMESPACE__
ClassName::class

© Aptech Limited Architecting Web Applications using PHP / Session 3 / 21 of 22

Show slide 21 and explain students the concept of Magic Constants, Magic constants are PHP predefined constants that are used based on their purpose. Various extensions generate these constants. In PHP, there are nine magic constants, all of which are resolved at compile time, unlike conventional constants, which are resolved at run time. There are eight magic constants with double underscores (__) at the beginning and end.

Take students through the list of constants with the example code:

- __line__: This magical constant returns the file's current line number. If you utilize this magic constant somewhere in your software file, it will display the line number during compilation.

For Example: <?php

```
echo "The Line number is : ". __line__;  
?>
```

Output: The Line number is : 2

- __file__: This magic constant return the full path of the executed file with the name of the file.

For Example: <?php

```
echo "the file name is : ". __file__;  
?>
```

Output: The file name is : home/abc.php

- __dir__: This magic constant return the directory of the executed file.

For Example: <?php

```
echo "The directory is : ". __dir__;
```

```
?>
```

Output: The directory is : home

- **__function__**: This magic constant return the name of the function where this magic constant is included.

For Example: <?php

```
function Test() {  
    echo "The function name is : ". __FUNCTION__;  
}  
Test();  
?>
```

Output: The function name is : Test

- **__class__**: This magic constant return the name of the class where this magic constant is included.

For Example: <?php

```
class Test  
{  
    public function getClassName() {  
        return __CLASS__;  
    }  
}  
  
$obj = new Test();  
echo $obj->getClassName();  
?>
```

Output: Test

- **__method__**: This magic constant return the method name where this magic constant is included.

For Example: <?php

```
class MyTest  
{  
    public function Test() {  
        return __METHOD__;  
    }  
}
```

```
    }
}

$obj = MyTest();
echo $obj->Test();
?>
```

Output: Test

- **__namespace__**: This magic constant return the current namespace where this magic constant is included.
- **__trait__**: This magic constant return the trait name where this magic constant is included.

For Example: <?php

```
trait Test{
    function MyTest() {
        echo __trait__;
    }
}

class Trial{
    use Test;
}

$a = new Trial();
$a->MyTest();
?>
```

Output: Test

- **ClassName::class**: This magic constant return the fully qualified class name.

For Example: <?php

```
namespace Computer_Science;
class Test{ }
echo Test::class;//Classname::class
?>
```

Output: Computer_Science\Test



Summary

- A data type is the classification of data based on its attributes.
- PHP is a loosely typed language which means that it does not require defining variables with types. Instead, PHP analyzes given attributes or values and determines appropriate data type.
- There are different data types, mainly Integer, Float, String, Boolean, and so on.
- PHP also supports compound types such as arrays and objects.
- PHP has many functions related to Strings, Numbers, NaN, Math, and so on.
- In PHP, a constant is an identifier (name) for a simple value and its value constant cannot be changed during the script execution.
- Magic constants are predefined constants in PHP and change their values with the context of their use.

Architecting Web Applications using PHP / Session 3 / 22 of 22

Use slide 22 to summarize the session. You will end the session, with a brief summary of what has been taught in the session. Tell the students pointers of the session. This will be a revision of the current session.

3.3 Post Class Activities for Faculty

You should familiarize yourself with the topics of the next session.

Tips:

You can also check the Articles/Blogs/Expert Videos uploaded on the Online Varsity site to gain additional information related to the topics covered in the next session. You can also connect to online tutors on the Online Varsity site to ask queries related to the sessions.

Session 4 – Variables and Operators in PHP

4.1 Pre-Class Activities

Before beginning the session, you should fully familiarize yourself with its topics. Prepare a question or two that will be a key point to relate the current session objectives.

4.1.1 Teaching Skills

To teach this session, you should be well-versed with the concept of internationalization and design patterns in Java. You must also be familiar with the concept of localization.

You must use the given images and teach the concepts in the theory class. For teaching in the class, you are expected to use slides and LCD projectors.

Tips:

It is recommended that you test the understanding of the students by asking questions in between the class.

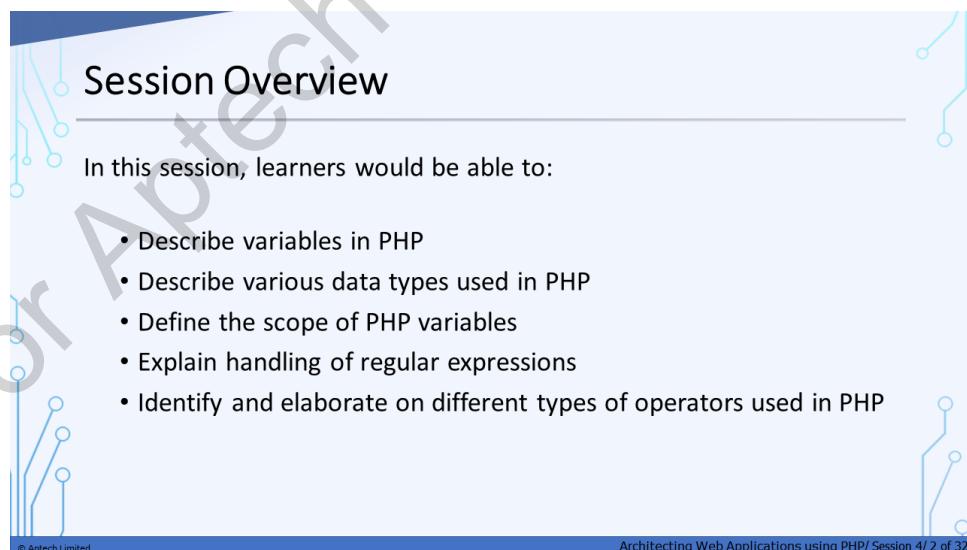
In-Class Activities

Follow the order given here during In-Class activities.

Overview of the Session

Give the students an overview of the current session in the form of session objectives. Read out the objectives on slide 2.

Slide 2



The slide has a blue header bar with the text "Session Overview". The main content area has a light blue background with a decorative border featuring blue lines and circles. The text "In this session, learners would be able to:" is followed by a bulleted list of five items:

- Describe variables in PHP
- Describe various data types used in PHP
- Define the scope of PHP variables
- Explain handling of regular expressions
- Identify and elaborate on different types of operators used in PHP

© Aptech Limited

Architecting Web Applications using PHP / Session 4 / 2 of 32

Show slide 2 and give a brief overview of the current session in the form of session objectives. Inform students that this session begins with the definition of variables in PHP and informs about various data types used in PHP. The scope of these variables has been defined in the upcoming sections. It also covers Portable Operating System Interface for

uniX (POSIX) regular expressions. Lastly, the session explains object-oriented programming in PHP in brief.

A 'Variable' is defined as any name or symbol that takes the place for a value. An 'Operator' is a symbol that performs an operation. Both variables and operators are of utmost importance in PHP. In PHP programming, various types of operators and variables are used. Numerous data types are also supported in PHP. There are different types of variables and operators used in PHP programming.

4.2 In-Class Explanations

Slide 3

Variables

Variables in a program are usually used for storing data or values that can be used anytime during the execution of a program.

A pictorial representation of the concepts of variables is shown in figure. Here, `empNumber` is a variable name.

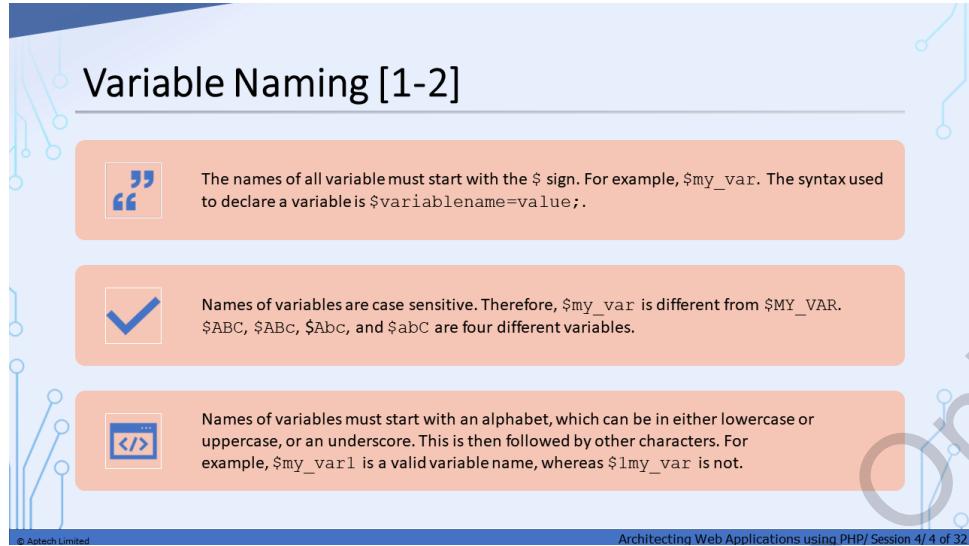
Figure: Variable

Show slide 3 and explain to students that:

- In a program, a variable is a name which is given to computer memory locations, which stores data types and values in the program.
- The value of a particular variable can be changed. This depends on the conditions or the information passed to the program.
- During the execution of a program these stored values can be used anytime, depending on the requirement.
- Based on the data types, variables can store numeric and character values, numeric values, strings, and memory addresses at runtime.

The given Figure represents a variable named as 'empNumber', which stores integer-type values. The variable 'empNumber' has the value 100.

PHP Variables: The data type in PHP is determined by the attributes related to the entered data. Various data types such as floating-point numbers, strings, and so on, have implicit support in PHP.



The slide has a blue header bar with the text "Variable Naming [1-2]" in white. Below the header are three orange callout boxes with icons and text:

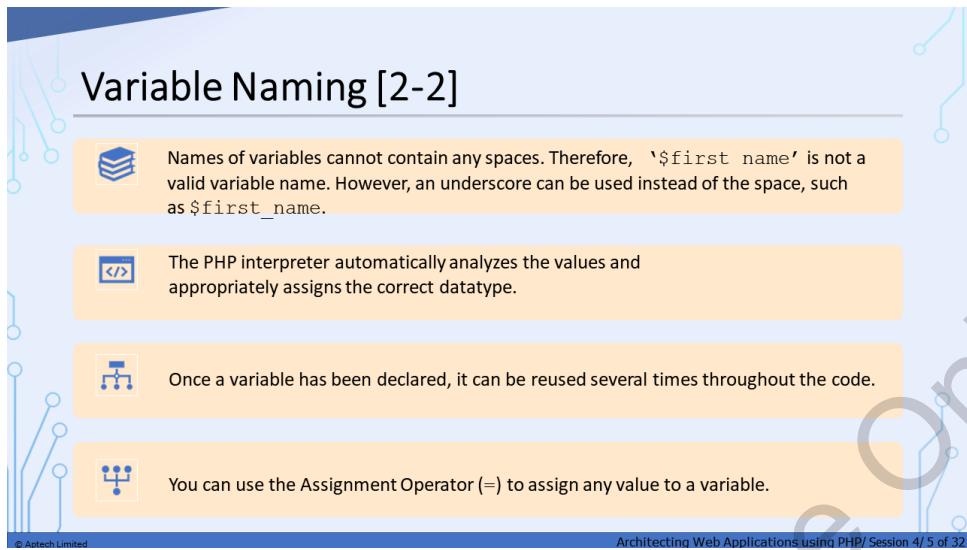
- Icon:** Two blue double quotes ("").
Text: The names of all variable must start with the \$ sign. For example, \$my_var. The syntax used to declare a variable is \$variablename=value;.
- Icon:** A blue checkmark inside a square.
Text: Names of variables are case sensitive. Therefore, \$my_var is different from \$MY_VAR. \$ABC, \$ABC, \$Abc, and \$abC are four different variables.
- Icon:** A blue square containing '</>' symbols.
Text: Names of variables must start with an alphabet, which can be in either lowercase or uppercase, or an underscore. This is then followed by other characters. For example, \$my_var1 is a valid variable name, whereas \$1my_var is not.

At the bottom left is the copyright notice "© Aptech Limited". At the bottom right is the page information "Architecting Web Applications using PHP/ Session 4/ 4 of 32".

Show slide 4 and explain to students that in order to declare a variable, a user must keep some rules in mind. Some of these rules are basic rules that are same as those in other programming languages, while some of them are unique to PHP. They are as follows:

- There should be a dollar sign (\$) sign, before the name of the variable.
- PHP variable names are case-sensitive.
- A variable name must start with a letter or underscore "_". PHP variable names cannot start with a number.

Slide 5



The slide has a blue header bar with the title "Variable Naming [2-2]" and decorative circuit board graphics on the sides. Below the title are four orange callout boxes, each containing an icon and text:

- Names of variables cannot contain any spaces.** Therefore, '\$first name' is not a valid variable name. However, an underscore can be used instead of the space, such as \$first_name.
- The PHP interpreter automatically analyzes the values and appropriately assigns the correct datatype.**
- Once a variable has been declared, it can be reused several times throughout the code.**
- You can use the Assignment Operator (=) to assign any value to a variable.**

At the bottom left is the copyright notice "© Aptech Limited" and at the bottom right is the page information "Architecting Web Applications using PHP/ Session 4/ 5 of 32".

Show slide 5 and continue following points from the previous slide.

- A variable name cannot contain any spaces. It can only contain numeric characters, letters, and underscores (a-z, A-Z, 0-9, and _). All blank spaces must be replaced by an underscore.
- The declaration of a data type for a variable is not required in PHP. The values are analyzed and the correct data type is assigned by the PHP interpreter, depending on the values.
- After declaring a value, the user can use the variable repeatedly in the whole PHP script.
- All the values are assigned to variables by using the assignment operator (=).

Slide 6

The slide features a decorative background with blue and white circuit-like patterns. At the top center is the title "PHP Local Variables". Below the title is a code snippet box containing PHP code. A callout box points to this code with the text: "Code Snippet shows how to declare a variable with local scope and demonstrates how it is used in PHP." Below the code snippet is a screenshot of a browser window showing the execution of the code. The output shows the variable's value inside the function, but an error message for trying to use it outside.

```
<?php
function Test() {
    $x = 5; // local scope
    echo "<p>Variable x inside function is: $x</p>";
}
Test();
// using x outside the function will generate an error
echo "<p>Variable x outside function is: $x</p>";
```

Code Snippet: Declaring a variable with local scope in PHP

Code Snippet shows how to declare a variable with local scope and demonstrates how it is used in PHP.

Variable x inside function is: 5
Following line shows an error since variable x is declared inside the function
Warning: Undefined variable \$x in C:\xampp\htdocs\s04_p0.php on line 10
Variable x outside function is: error

Figure: Output for Code Snippet

© Aptech Limited

Architecting Web Applications using PHP/ Session 4 / 6 of 32

Show slide 6 and tell students that the variables that are declared inside the function of a PHP program are called local variables. The local variables have scope solely in that function in which they are declared. So local variables cannot be accessed outside the functions in which they are declared. Any other variable having the same name as the local variables in a program, outside the function in which they are declared, will be treated as a totally different variable.

Refer to code snippet to understand how to declare a variable with local scope in PHP and how to use it.

Output explanation: Here, since no value for x variable is declared outside the function, it will show error in the output when you try to print value of x outside the function. Local variables are recognized only by the function in which they have been declared. Therefore, local variables can exist with the same name across different functions.

Additional Information:

Refer to following links for more information:

- https://www.w3schools.com/php/php_variables_scope.asp
- <https://www.geeksforgeeks.org/php-variables/>
- https://www.tutorialspoint.com/php/php_local_variables.htm
- <https://www.php.net/manual/en/language.variables.scope.php>

The slide features a decorative background with blue and white circuit board patterns on the left and right sides. At the top center is the title "PHP Global Variables". Below the title is a code snippet box containing PHP code. A callout box points to this code with the text "Code Snippet shows an example of a program to demonstrate a global variable.". To the right of the code is a screenshot of a browser window showing the execution results.

```
<?php  
$x = 5; // global scope  
function Test() {  
    // using x inside this function will generate an error  
    echo "Following line shows an error since variable x is  
    declared outside the function.>";  
    echo "<p>Variable x inside function is: $x</p>";  
}  
Test();  
echo "<p>Variable x outside function is: $x</p>";  
?>
```

Code Snippet: Program to demonstrate a global variable

Code Snippet shows an example of a program to demonstrate a global variable.

Figure: Output for Code Snippet

Output from the browser window:

```
F:\xampp\htdocs\4d\index.php  
+ [index.php] 100%  
+ 0 ① index.php  
② Run  
Following line shows an error since variable x is declared outside the function  
Warning: Undefined variable $x in C:\xampp\htdocs\4d\index.php on line 7  
Variable x inside function is: 5  
Variable x outside function is: 5
```

Architecting Web Applications using PHP/ Session 4 / 7 of 32

Show slide 7 and tell the students that global variables are the variables that are defined outside a function. Contrary to local variables, global variables can be accessed from anywhere, either inside, or outside the function definition. The user must use the keyword 'global' before the variable name, in order to use it as a global variable.

Refer to code snippet to understand how to declare a global variable in PHP and how to use it.

Output explanation: Here, variable x inside function does not contain any value, since x has been declared as the global variable outside the function. The echo keyword is used to print value of x variable outside the function. Therefore, variable x outside the function is 5.

Additional Information:

Refer to following links for more information:

<https://www.geeksforgeeks.org/how-to-declare-a-global-variable-in-php/>

https://www.tutorialspoint.com/php/php_global_variables.htm

<https://stackoverflow.com/questions/13530465/how-to-declare-a-global-variable-in-php>

Slide 8

The slide features a decorative background with blue and white circuit board patterns on the left and right sides. The title 'PHP Static Variables' is centered at the top. Below the title is a code snippet in PHP:

```
<?php
function static_variable()
{
    static $X = 10;      //static variable
    $Y = 20;            // non-static variable

    $X++; //increment in static variable

    $Y++; //increment in non-static variable
    echo "Static: " . $X . "<br>";
    echo "Non-static: " . $Y . "<br>";
}

//first function call
static_variable();

//second function call
static_variable();
?>
```

Code Snippet: Program to demonstrate a static variable

Code Snippet shows how to declare a static variable.

To the right of the code is a screenshot of a web browser window showing the output:

localhost:php_Session04/codesnippet1.php

Static: 11
Non-static: 21
Static: 12
Non-static: 21

Figure: Output for Code Snippet

At the bottom of the slide, there is a footer bar with the text '© Aptech Limited' on the left and 'Architecting Web Applications using PHP/ Session 4/ 8 of 32' on the right.

Show slide 8 and tell the students that the user can create a static variable in a local function to use it, even after a function call, in case if it is required to call it again, since PHP variables are deleted once they have been executed and their memory is released. However, this is not the case with static variables because a static memory associated with it. It is stored in the location where the static variables are stored. Static variables can be created during the start of program execution and can be destroyed at the end of program execution.

The user must use the keyword 'static' before the variable name, in order to use it as a static variable.

Refer to code snippet to understand how to declare a static variable in PHP and how to use it.

Output explanation: Here, after each function call, \$X is regularly incremented. However, \$Y is not. Since \$Y is not a static variable, its memory is released after each function call is executed. In case of static variable, it does not hold the previously incremented value. Each new function call stores the new value assigned to the Y variable. As a result, the incremented value is shown inside the function. Whereas, in the case of \$X, it retains the previous value. Each time the incremented value is printed, the new value is being used in the second function call.

Additional Information:

Refer to following links for more information:

<https://stackoverflow.com/questions/7508284/static-variables-in-php>

https://www.w3schools.com/php/keyword_static.asp

https://www.tutorialspoint.com/php/php_static_variables.htm

Slide 9

The slide has a decorative background with blue and white circuit board patterns. The title 'PHP Function Parameters' is at the top. Below it is a code snippet in a light gray box:

```
<?php
// multiply a value by 20 and return it to the caller
function multiply($value)
{
    $value=$value*20;
    return $value;
}
$retval=multiply(10);
print "Return value is $retval\n";
?>
```

A callout box below the code says 'Code Snippet shows a program with function parameters.' To the right is a screenshot of a browser window showing the output: 'Return value is 200'. At the bottom left is a copyright notice: '© Aptech Limited' and at the bottom right: 'Architecting Web Applications using PHP/ Session 4 / 9 of 32'.

Show slide 9 and explain to the students that for a function there can be either zero or multiple number of parameters. If the function has multiple parameters, the parameters are separated by a comma (,). If the number of parameters defined in the functions are more than the number of arguments that are passed to that function, PHP raises an error. All the function parameters must be declared after the function name, inside parentheses. Their declaration must be same as that of regular variables. Trailing commas were ignored by the PHP interpreter in PHP version 7.0. While in the versions following PHP 8.0, the user can place all the trailing commas differently.

Refer to code snippet to understand function parameters in a better manner.

Output explanation: Here, one parameter is passed to the function 'multiply ()' and then, 't' function is called and the result of '\$value' variable is stored in the 'retval' variable.

Slide 10

Variable Types [1-2]

Based on the attributes, data can be classified into different categories. Those categories are called as data types. There are eight data types that are used to construct variables.

- Integers**
 - Are whole numbers without a decimal point. For example: 4195.
- Doubles**
 - Are floating point numbers. For example: 49.1 or 3.14159.
- Booleans**
 - Have only two possible values. They are either True or False.

© Aptech Limited Architecting Web Applications using PHP/ Session 4/ 10 of 32

Show slide 10 and explain to students that PHP supports the eight data types to construct variables, which are integers, doubles, booleans, NULL, strings, arrays, objects, and resources.

Ask the students following question. Wait for a response before you answer.

In-Class Question: What are floating-point numbers?

Answer: Floating-point numbers are the numbers that contain floating decimal points. For example 5.5, -22.001, and so on.

Additional Information:

Refer to following links for more information:

https://www.tutorialspoint.com/php/php_variable_types.htm

https://www.w3schools.com/php/php_datatypes.asp

Slide 11

Variable Types [2-2]

- NULL**
 - Is a special type that has only one value, that is, NULL.
- Strings**
 - Are sequences of characters. For example, 'PHP supports string operations'.
- Arrays**
 - Are named and indexed collections of other values.
- Objects**
 - Are instances of programmer-defined classes, which can package up both other kinds of values and functions that are specific to the class.
- Resources**
 - Are special variables that hold references to resources that are external to PHP. For example, database connections.

© Aptech Limited Architecting Web Applications using PHP / Session 4 / 11 of 32

Using slide 11, explain the following:

Alphanumeric characters are classified as strings, whole numbers are classified integers, and numbers with decimal points are classified as floating-points.

While the first five data types are simple types, arrays and objects are compound types. This means that they are able to package up other arbitrary values of arbitrary type.

Some of these important data types are going to be discussed in the next slides.

Ask the students following question. Wait for a response before you answer.

In-Class Question: Which function can be used to test the type of any variable?

Answer: The function 'gettype()' can be used to test the type of any variable.

Slide 12

Integers [1-2]

There are three functions supported in PHP, to check if the given variable is an integer:

- is_int()
- is_long() - alias of is_int()
- is_integer() - alias of is_int()

Following are some of the rules that apply to integers:

There must be at least one digit in an integer. For example, \$int_var = 69.

There must not be any decimal point in an integer. For example, \$int_var1 = 87654.

Integer can be specified in three different formats:

- Octal (8-based and prefixed with 0)
- Decimal (10-based)
- Hexadecimal (16-based and prefixed with 0x)

© Aptech Limited Architecting Web Applications using PHP / Session 4 / 12 of 32

Show slide 12 and tell the students that the numbers that do not have a decimal point are known as integers. For example, 2, 256, -256, 10358, and -179567.

There are three functions in PHP to check if the type of the given variable is integer. Those three functions have been mentioned in the slide.

Similar to integers, an integer data type is defined as a non-decimal number. The range is different for 32-bit systems and 64-bit systems. The range has been mentioned. If an integer's limit exceeds, any value more or less than the mentioned range, will be stored in the form of a float. The result will only be stored in the form of a float, even if just one of the operands is a float. For example, in the operation $4 * 2.5=10$, the result is still stored in the form of a float, since one of the operands (2.5) is a float.

There is a set of rules that is applied to integers, which are as follows:

- An integer must contain at least one digit.
- An integer must not contain any decimal point in it.
- An integer can be either a positive one or a negative one.
- Integers are specified in octal, decimal, and hexadecimal formats.

Slide 13

The slide features a decorative background with blue and white circuit board patterns. At the top center is the title "Integers [2-2]". Below the title is a code snippet box containing PHP code. To the right of the code is a callout box with text. Further right is a screenshot of a browser window showing the output of the code.

Code Snippet:

```
<?php
// Check if the type of a variable is integer
$x1 = 1024;
var_dump(is_int($x1));
echo "The given number $x1 is an integer <br>";
// Check again...
$x2= 99.84;
var_dump(is_int($x2));
echo "The given number $x2 is an integer <br>";
echo "<br>";
$y=6987;
var_dump(is_int($y));
echo "The given number $y is an integer <br>";
?>
```

Code Snippet shows a program to check whether the variable is an integer or not.

Figure: Output for Code Snippet

localhost:php_Session04/codesnippet5.php

bool(true) The given number 1024 is an integer
bool(false) The given number 99.84 is an integer
bool(true) The given number 6987 is an integer

Show slide 13 and tell the students to refer to code snippet to understand how to check if a particular variable is an integer or not.

Output explanation: Here, the code checks to see if the number is an integer or not. A built-in function 'var_dump' is used. This returns a type of data variable. Here, it returns int. Another built-in function 'is_int' is used to check if the number is an integer or not. If it is an integer, true will be returned. Since the first number 1024 is an integer, true is returned. The next number 99.84 is not an integer, therefore, false is returned.

The slide features a blue header bar with the title "Doubles". Below the title is a code snippet in a light gray box:

```
<?php  
$many = 2.2888800;  
$many_2 = 2.2111200;  
$few = $many + $many_2;  
print("$many + $many_2 = $few <br>");  
?>
```

Below the code is the text "Code Snippet: Program to print doubles". To the right of the code is a screenshot of a web browser window showing the output:

localhost/php_Session04/codesnippet6.php
2.28888 + 2.21112 = 4.5

Figure: Output for Code Snippet

At the bottom left is the copyright notice "© Aptech Limited" and at the bottom right is the page number "Architecting Web Applications using PHP/ Session 4/ 14 of 32".

Show slide 14 and tell the students that by default, doubles data type prints with the minimum number of decimal places. For example, 3.14159 or 49.1.

In PHP, Floats and doubles are same. With a precision up to 14 digits, floating-points in PHP include the cumulative digits, before and after the period.

Note: When converting to int, certain types have distinguish behavior. This is observed even when converting to float.

Considering the floating-point representation internally, it is problematic scenario to test floating-point values for equality. However, comparisons of floating-point values are performed that over these limitations.

Inform students that for testing floating-point values for equality, a value called machine epsilon or unit roundoff is used which is the smallest acceptable difference in calculations. Machine epsilon is error that occurs due to rounding off.

Refer to code snippet to understand how to print doubles with the help of PHP script.

Figure shows the output for Code Snippet.

Output explanation: In the code, two floating-point numbers are added and the value is stored in variable 'few'. The program shows that double numbers only display a few decimal number places (just one for 4.5).

Slide 15

Boolean

```
<?php  
if (TRUE)  
    echo "This condition is TRUE . . .";  
if (FALSE)  
    echo "This condition is FALSE . . .";  
?>
```

Code Snippet: Program to understand the usage of Boolean data type

Code Snippet shows a program to highlight the usage of Boolean data type.

Booleans are the simplest data type and their operation is similar to an electrical toggle switch.

They can hold only one of the two values at a time, namely, TRUE (1) or FALSE (0).

This condition is TRUE.

Figure: Output for Code Snippet

© Aptech Limited

Architecting Web Applications using PHP/ Session 4/ 15 of 32

Show slide 15 and tell the students that booleans are conditional statements and can have only two values, either TRUE or FALSE. If a specified condition is correct, then they return TRUE. If the condition is not correct, they return FALSE.

Refer to code snippet to understand how to use booleans with the help of PHP script.

Output explanation: Here, the Boolean value is shown as true. By default, if there is no condition is present in the program, the first value true is always printed. Hence, the true block is executed.

Additional Information:

Refer to following links for more information:

<https://www.php.net/manual/en/language.types.boolean.php>

https://www.tutorialspoint.com/php_BOOLEAN-data-type

<https://www.w3resource.com/php/data-types/booleans.php>

Slide 16

NULL

```
<?php  
$nl=NULL;  
echo $nl; //it will not give any output  
?>
```

Code Snippet: Program to show the usage of data type NULL

Code Snippet shows a program that shows the usage of data type NULL.

It is generally written in capital letters, as it is case sensitive.

The syntax is: \$my_var = null;

In Code Snippet, since the variable has been assigned with a null value, there is no output.

© Aptech Limited Architecting Web Applications using PHP/ Session 4/ 16 of 32

Show slide 16 and tell the students that NULL is a special type that has only one value NULL. If a variable is created without any value, it is automatically assigned the NULL value. A variable with NULL value will evaluate to FALSE in a Boolean operations.

A variable is considered to be null if:

- It has been assigned the constant null.
- It has not been set to any value yet.
- It has been `unset()`.

Inform students that other variables can be assigned NULL and utilized in comparisons.

An in-built function `is_null()` in PHP is utilized to determine whether a variable is NULL or not. It results True if the given variable is null, otherwise returns False.

Refer to code snippet to understand how to use NULL with the help of PHP script.

Note: Here, code snippet would not give any output, as it contains the NULL variable.

Strings

- A string is a series of characters, where the size of each character is one byte.
- In PHP, only the 256-character set is supported and not the native Unicode.
- However, in 32-bit systems, a string can have a size of 2 GB (maximum of 2147483647 bytes).

© Aptech Limited Architecting Web Applications using PHP / Session 4 / 17 of 32

Show slide 17 and tell the students that strings are sequences of characters, having one byte as the size of each character. PHP supports only 256-character set and it does not offer native Unicode support. While the size of a string can be 2 GB in 32-bit systems.

In PHP, there are four ways to specify a string literal:

- Single quoted, where the user can create a string in PHP by enclosing the text in a single-quote.
- Double quoted, where the user can create a string in PHP by enclosing the text in double-quotes.
- Heredoc syntax, where the user can define strings with variables that contain numerous words, to create text for Web pages.
- Nowdoc syntax, which the user can define with '<<<' sequence with their identifiers by enclosing it with single-quotes.

Ask the students following question. Wait for a response before you answer.

In-Class Question: What is the use of the strcmp() function in PHP?

Answer: In PHP, the strcmp() function is used to compare the strings, including the case.

Additional Information:

Refer to following links for more information:

https://www.w3schools.com/php/php_string.asp
<https://www.javatpoint.com/php-string>
<https://www.geeksforgeeks.org/php-strings/>

Slide 20

PHP – Static Keyword

```
<?php
function myTest() {
    static $x = 0;
    echo $x;
    $x++;
}
myTest();
echo "<br>";
myTest();
echo "<br>";
myTest();
?>
```

Code Snippet: Program using static keyword

Code Snippet shows a program using the static keyword.

Figure: Output for Code Snippet

© Aptech Limited

Architecting Web Applications Using PHP/ Session 4/ 20 of 32

Show slide 20 and tell the students that static variables are created when the local variable is required multiple times and it is not to be deleted after function execution. To create a static variable, the user can add the static keyword before the variable while declaring it in a function. Each time the function is called, the variable will retain the information from the last instance in which the function was called. The variable still remains local to the function.

Refer to code snippet to understand how to use the 'static' keyword in PHP.

Output explanation: In Code Snippet, increment operator is used to print the value of x, where the value of x is initially zero. Function 'myTest()' is called thrice with echo. Hence, the value of x gets incremented each time the function is called.

Slide 21

Regular Expression

- Generally called regex, regular expressions are a sequence of characters that describe a specific search pattern in the form of text strings.
- Regular expressions are strings composed of delimiters, a pattern, and optional modifiers.
- The common delimiter that is most commonly used is the forward slash (/).
- There are two types of Regular Expressions: POSIX Regular Expressions and PERL Style Regular Expressions.

© Aptech Limited Architecting Web Applications Using PHP / Session 4 / 21 of 32

Show slide 21 and explain to students that regular expressions are nothing but a sequence of characters that describe a specific search pattern in the form of text strings. They are commonly used in programming algorithms aimed at specific tasks. Regular expressions fetch required strings based on the pattern defined.

Regular expressions are used to perform all types of text search and replace operations. They are a compact way to describe string patterns that match a specific amount of text. Regular expressions are strings composed of delimiters, a pattern, and optional modifiers. The syntax for regex is \$exp = "/eliteschools/i";. Where, / is the delimiter, eliteschools is the pattern that is being searched for, and 'i' is a modifier that makes the search case-insensitive.

Any character except an alphabet, number, backslash, or space can be a delimiter. The delimiter that is commonly used is the forward slash (/). However, if the pattern itself contains forward slashes, then one can choose any other character as the delimiters such as # or ~.

Regular Expressions are of two types:

- **POSIX Regular Expressions:** This is defined as set of character where any of its single character is required to match from the input string. These expressions are defined within [].
- **PERL Style Regular Expressions:** These type of Regex patterns are similar to POSIX regex but created with meta characters and identifiers. Syntax for this Regexes are interchangeable with POSIX style.

Slide 22

POSIX Regular Expressions

Portable Operating System Interface for uniX (POSIX) is a collection of standards that are used to define some of the functionality that should be supported by UNIX operating system.

One such standard defines two types of regular expressions. POSIX regular expressions are to be used only with textual data.

If data contains a NUL-byte (\x00), the regular expression will interpret that as the end of the string and matching will not happen beyond that point.

Brackets ([]) can be used to find a range of characters. For example, if a match is to be found for any digit between 0 and 9, then, the expression is [0-9].

To get a match for a lowercase character between a and z, use the expression [a-z].

To get a match for a character between uppercase A and uppercase Z, use the expression [A-Z].

To find a match for a character between lowercase a and uppercase Z, use the expression [a-zA-Z].

© Aptech Limited Architecting Web Applications using PHP / Session 4 / 22 of 32

Show slide 22 and explain to students that Portable Operating System Interface for uniX (POSIX) regular expressions are combination of standards, that define functionalities that the UNIX operating system supports. Regular expressions deal only with textual data. The regular expression will interpret the data as the end of the string, if it contains a NUL-byte (\x00), matching will not be executed beyond that point.

A POSIX regular expression's structure almost resembles an arithmetic expression. Numerous elements combine and form highly complex expressions. Any expression that is a match to a single character is the simplest regular expression.

To find a range of characters, brackets [] can be used. For example, the expression to find a match for any digit between 0 and 9, is [0-9]. The expression [a-z] is to get a match for a lowercase character between a and z. To get a match for a character between uppercase A and uppercase Z, the expression is [A-Z]. To find a match for a character between lowercase a and uppercase Z, the expression to be used is [a-zA-Z].

The very basic regex is the one that matches a single character.

Additional Information:

Refer to following links for more information:

https://www.tutorialspoint.com/php/php_regular_expression.htm#:~:text=The%20structure%20of%20a%20POSIX,g%2C%20haggle%2C%20or%20bag.

<https://www.geeksforgeeks.org/php-regular-expressions/>

PHP's Regexp POSIX Functions

At present, PHP has seven different functions to search for strings using POSIX-style regular expressions. They are as follows:

ereg()	ereg_replace()	eregi()	eregi_replace()
split()	spliti()	sql_regcase()	

© Aptech Limited Architecting Web Applications using PHP / Session 4 / 23 of 32

Show slide 23 and explain to students that PHP currently offers seven functions in order to search for strings using POSIX-style regular expressions. They are described here:

- `ereg ()` : This function searches a string for a pattern and returns true if a pattern is found, else false.
- `ereg_replace ()` : This function searches for a pattern and replaces it with the desired replacement, if it is found.
- `eregi ()` : This function searches for a string-specified string throughout a pattern-specified string. This is non-case sensitive.
- `eregi_replace ()` : This function operates exactly as `ereg_replace ()` except that this is not case sensitive.
- `split ()` : This function divides a string into various elements, the boundaries of the elements being based on the occurrence of string pattern.
- `spliti ()` : This function operates exactly like `split()` except that this is not case sensitive.
- `sql_regcase ()` : This function helps in converting each character in the input parameter string into a bracketed expression containing two characters.

Additional Information:

Refer to following links for more information:

https://www.tutorialspoint.com/php/php_regular_expression.htm
<http://php.adamharvey.name/manual/en/ref.regex.php>
<https://www.oreilly.com/library/view/programming-php/1565926102/ch04s09.html>



Operators in PHP

An operator is used to perform a variety of operations on variables and values.

Operator Types: There are different types of operators which perform different operations.

Architecting Web Applications using PHP / Session 4 / 24 of 32

Show slide 24 and explain to students that an operator is a character that comes handy in performing various operations on values and variables. For example, take the expression $6+5=11$. Here, 6 and 5 are the operands and + is the operator.

The operators in PHP are divided into following groups:

- Arithmetic Operators
- Comparison Operators
- Logical or Relational Operators
- Assignment Operators
- Ternary or Conditional Operators

Additional Information:

Refer to following links for more information:

https://www.w3schools.com/php/php_operators.asp

<https://www.geeksforgeeks.org/php-operators/>

<https://www.javatpoint.com/php-operators>

Slide 25

Arithmetic Operators

Arithmetic operators perform different arithmetic operations.

Operator	Name	Description
+	Addition	Returns the sum of the operands
-	Subtraction	Returns the difference between the two operands
*	Multiplication	Returns the product of two operands
/	Division	Returns the quotient after dividing the first operand by the second operand
%	Modulus	Returns the remainder after dividing the first operand by the second operand

Table: Arithmetic Operators in PHP

© Aptech Limited Architecting Web Applications using PHP / Session 4 / 25 of 32

Show slide 25 and explain to students that the PHP arithmetic operators perform common arithmetical operations, such as addition, subtraction, multiplication, and so on, of numerical values. For example, if variable A holds the value 30 and variable B holds the value 10, then the arithmetic operator * is used to multiply both and given the result as 300.

Refer to table to understand the function of the arithmetic operators supported by PHP.

Slide 26



The slide has a blue header bar with the title 'Comparison Operators' in white. Below the title is a green bar containing the text 'Comparison operators compare two operands.' The main content is a table with 9 rows, each containing an operator, its name, and a description. The table has three columns: 'Operator', 'Name', and 'Description'. The operators and their descriptions are:

Operator	Name	Description
<code>==</code>	Equal to	Returns true if both the operands are equal
<code>===</code>	Identical	Returns true if both the operands are equal and are of the same data type
<code>!=</code>	Not equal to	Returns true if the first operand is not equal to the second operand
<code><></code>	Not equal to	Returns true if the first operand is not equal to the second operand
<code>!==</code>	Not Identical	Returns true if the first operand is not equal to the second operand or they are not of the same data type
<code><</code>	Less than	Returns true if the first operand is less than the second operand
<code><=</code>	Less than or equal to	Returns true if the first operand is less than or equal to the second operand
<code>></code>	Greater than	Returns true if the first operand is greater than the second operand
<code>>=</code>	Greater than or equal to	Returns true if the first operand is greater than or equal to the second operand

Table: Comparison Operators in PHP

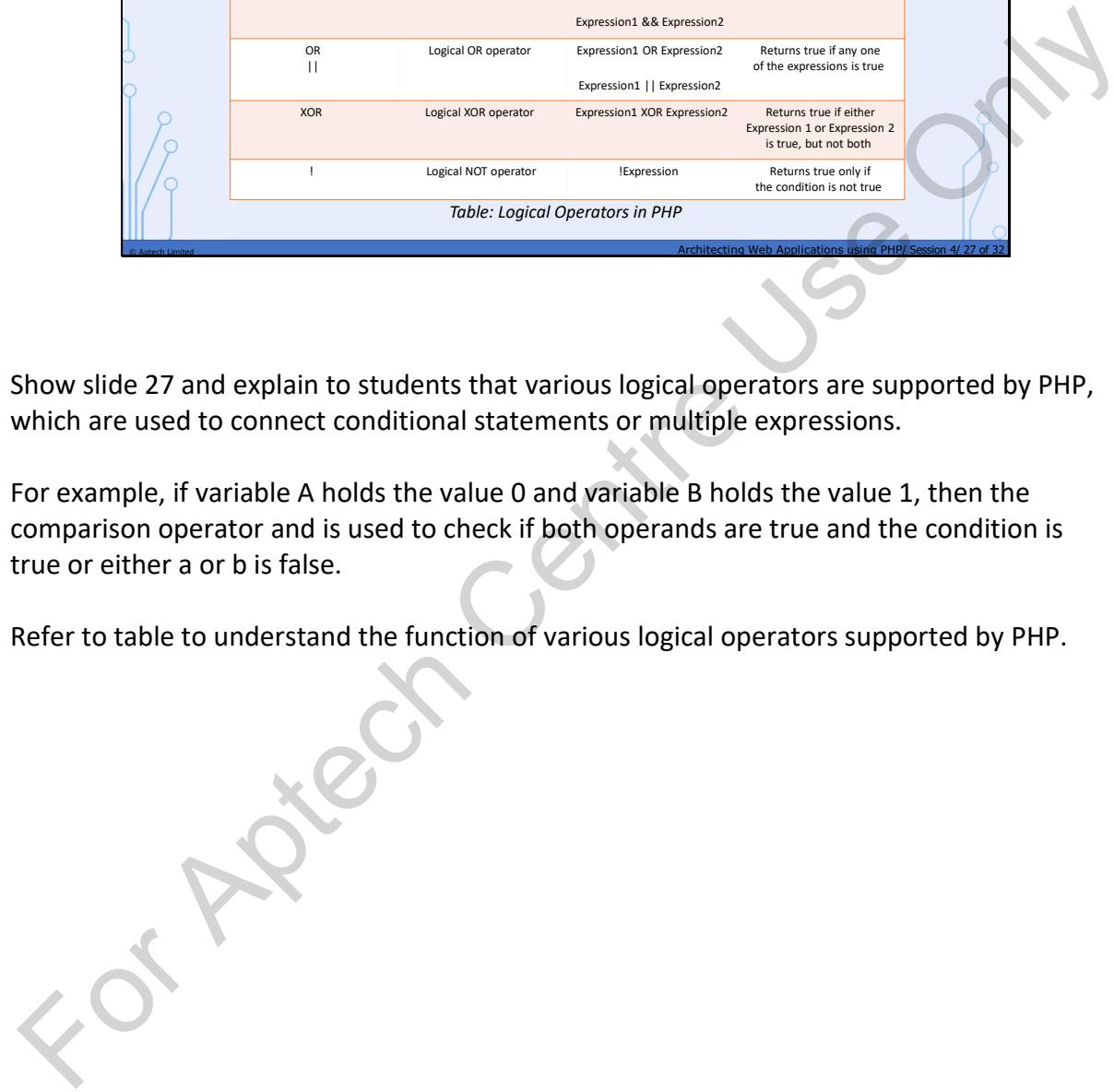
© Aptech Limited Architecting Web Applications using PHP/ Session 4/ 26 of 32

Show slide 26 and explain to students that the PHP comparison operators help to compare two operands (numbers or strings).

For example, if variable A holds the value 10 and variable B holds the value 40, then the comparison operator `==` is used to check if the value of the operands is equal and give the result of `(A==B)` as not true.

Refer to table to understand the function of the comparison operators supported by PHP.

Slide 27



Logical Operators

Logical operators connect multiple expressions. PHP supports various logical operators.

Operator	Name	MSO	General Form	Description
AND &&	Logical AND operator		Expression1 AND Expression2 Expression1 && Expression2	Returns true only if both the expressions are true
OR	Logical OR operator		Expression1 OR Expression2 Expression1 Expression2	Returns true if any one of the expressions is true
XOR	Logical XOR operator		Expression1 XOR Expression2	Returns true if either Expression 1 or Expression 2 is true, but not both
!	Logical NOT operator		!Expression	Returns true only if the condition is not true

Table: Logical Operators in PHP

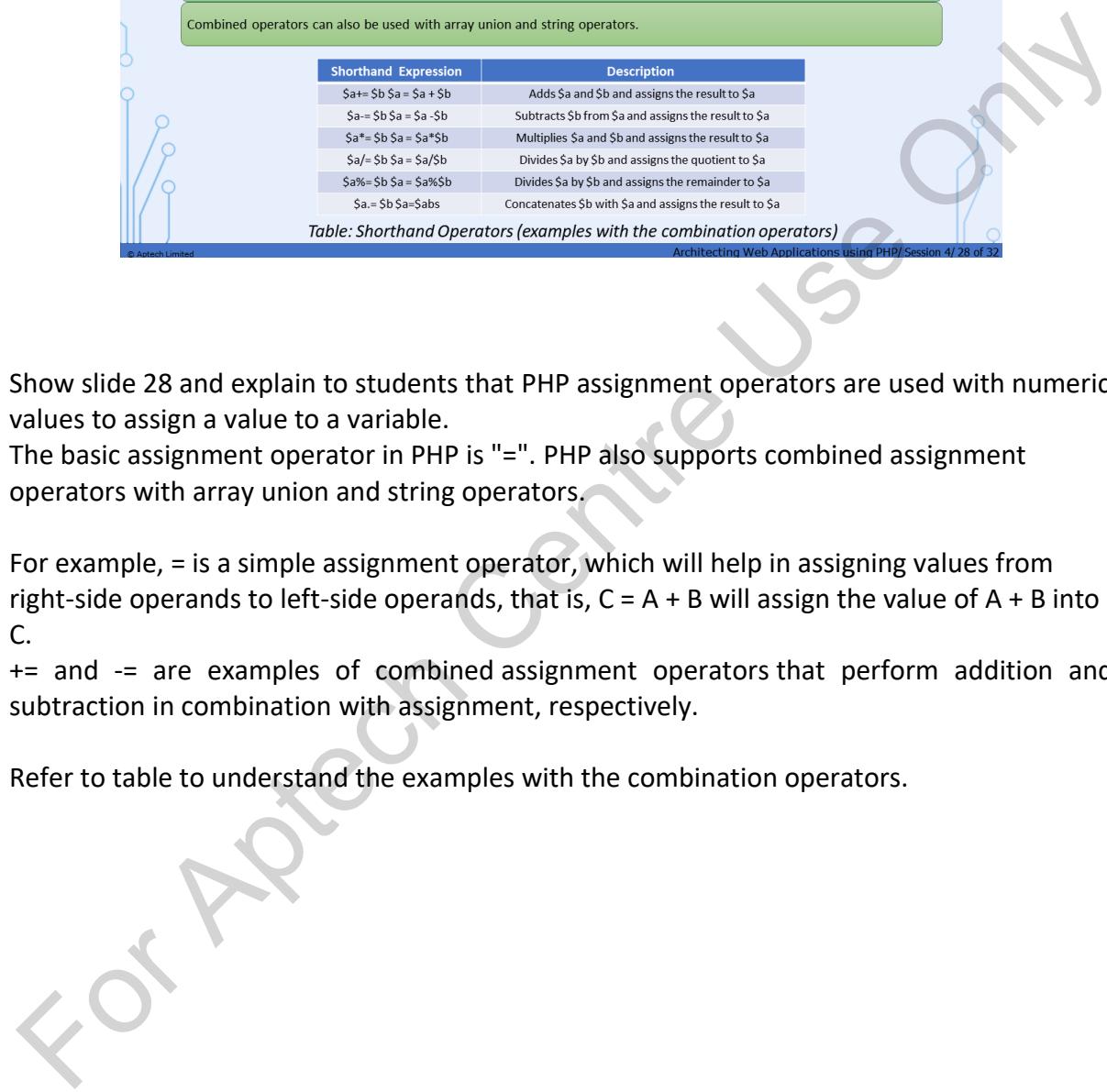
Aptech Limited | Architecting Web Applications using PHP / Session 4 / 27 of 32

Show slide 27 and explain to students that various logical operators are supported by PHP, which are used to connect conditional statements or multiple expressions.

For example, if variable A holds the value 0 and variable B holds the value 1, then the comparison operator and is used to check if both operands are true and the condition is true or either a or b is false.

Refer to table to understand the function of various logical operators supported by PHP.

Slide 28



Assignment Operators

Assignment operators are used to assign values to variables.

Besides the primary assignment operator represented by the symbol =, PHP also supports combined assignment operators.

+= and -= are examples of combined assignment operators that perform addition and subtraction in combination with assignment, respectively.

Combined operators can also be used with array union and string operators.

Shorthand Expression	Description
\$a += \$b \$a = \$a + \$b	Adds \$b to \$a and assigns the result to \$a
\$a -= \$b \$a = \$a - \$b	Subtracts \$b from \$a and assigns the result to \$a
\$a *= \$b \$a = \$a * \$b	Multiples \$a and \$b and assigns the result to \$a
\$a /= \$b \$a = \$a / \$b	Divides \$a by \$b and assigns the quotient to \$a
\$a %= \$b \$a = \$a % \$b	Divides \$a by \$b and assigns the remainder to \$a
\$a .= \$b \$a = \$a . \$b	Concatenates \$b with \$a and assigns the result to \$a

Table: Shorthand Operators (examples with the combination operators)

Architecting Web Applications using PHP/ Session 4/ 28 of 32

Show slide 28 and explain to students that PHP assignment operators are used with numeric values to assign a value to a variable.

The basic assignment operator in PHP is "=". PHP also supports combined assignment operators with array union and string operators.

For example, = is a simple assignment operator, which will help in assigning values from right-side operands to left-side operands, that is, $C = A + B$ will assign the value of $A + B$ into C.

+= and -= are examples of combined assignment operators that perform addition and subtraction in combination with assignment, respectively.

Refer to table to understand the examples with the combination operators.



Conditional Operator

The conditional operator is used to perform various operations that are based on different conditions.

PHP supports the conditional operator `? :`. It is used to first evaluate an expression for a TRUE or FALSE value and then, execute one of the two given statements depending upon the result of the evaluation.

It is commonly used to evaluate the `if-then-else` operation.

© Aptech Limited Architecting Web Applications using PHP / Session 4 / 29 of 32

Show slide 29 and explain to students that in PHP, conditional operator (`:?`) is used to perform simple comparisons or operations to check whether a condition is TRUE or FALSE.

It decreases the length of the code performing conditional operations. It is mainly used to evaluate the 'if-then-else operation' in a left to right order. In short, it contains a condition, a result statement for TRUE and a result statement for FALSE. It then executes the result statement based on if the condition is TRUE or FALSE.

The conditional operators are also called comparison operators.

Additional Information:

Refer to following links for more information:

<https://www.geeksforgeeks.org/php-ternary-operator/>

https://www.tutorialspoint.com/php/php_conditional_operator_examples.htm

Operator Categories

On the basis of the number of operands used, all the operators that have been discussed so far, can be categorized into different types as shown in Table.

Operator Category	Description
Unary prefix operators	Precede a single operand
Binary operators	Take two operands and execute different arithmetic and logical operations
Conditional or ternary operators	Take three operands. Based on how the first expression is evaluated, either the second or third expression will be evaluated
Assignment operators	Assign a value to a variable

Table: Operator Categories

© Aptech Limited Architecting Web Applications Using PHP / Session 4 / 30 of 32

Show slide 30 and tell the students that as per the discussion so far, all the operators can be divided into different types that are based on the number of operands used. Those are:

- **Unary prefix operators:** These operators accept only one value.
- **Binary operators:** These operators accept two values.
- **Conditional or ternary operators:** The conditional operator is used to perform various operations that are based on different conditions. PHP supports the conditional operator `?:`. It is used to first evaluate an expression for a TRUE or FALSE value and then, execute one of the two given statements depending upon the result of the evaluation.
- **Assignment operators:** Assignment operators are used in writing values to variables. Besides the primary assignment operator represented by the symbol `=`, PHP also supports combined assignment operators. For example, `=` is a simple assignment operator, which will help in assigning values from right-side operands to left-side operands, that is, `C = A + B` will assign the value of `A + B` into `C`. `+=` and `-=` are examples of combined assignment operators that perform addition and subtraction in combination with assignment respectively. Combined operators can also be used with array union and string operators.

Refer to table to know more about the operator categories.

Additional Information:

Refer to following links for more information:

https://www.tutorialspoint.com/php/php_operator_types.htm
<https://www.geeksforgeeks.org/php-operators/>
https://www.w3schools.com/php/php_operators.asp
<https://www.php.net/manual/en/language.operators.php>
<https://www.javatpoint.com/php-operators>

Slide 31

Precedence of PHP Operators

The precedence of operators decides the order of execution of operators in an expression. Some operators have a higher precedence than others.

To override precedence and force an operation, use parentheses.

A few operators have an equal level of precedence. In such a case, the order of operations is decided by the order of associativity, that is left or right.

© Aptech Limited Architecting Web Applications using PHP / Session 4 / 31 of 32

Show slide 31 and tell the students that operator associativity and precedence determines how operators are grouped (ordered) and executed in an expression, but the order of evaluation is not specified by it. The order of operations depends on the order of associativity, since some of the operators have a higher or a lower precedence than the others. For example, in the expression $2+6*3$, the addition operator (due to its lower precedence) will be executed after the multiplication operator (higher precedence), giving the answer as 20. Similarly, a division operator will have a higher precedence over the addition operator.

To override or force the precedence of operators, a parentheses is used. For example, if the previous expression is rewritten as $(2+6)*3$, then the addition operator will be executed before the multiplication operator because the addition operator has the precedence over multiplication operator due to the use of parentheses.

Operators can have same level of precedence as well. In such cases, the order of associativity (from left or right) decides the order of operations. For example, the order of associativity will be from left to right if the user is using additive operators such as $+-$.

Additional Information:

Refer to following links for more information:

<https://www.php.net/manual/en/language.operators.precedence.php>
https://www.tutorialspoint.com/php/php_operator_types.htm

Summary

- Variables can be used to store and access data during program execution.
- Three types of variables, local, global, and static are supported by PHP.
- There are different rules to be followed while naming a variable.
- Variable types are integer, Boolean, doubles, character strings, and floating-point numbers.
- A string literal can be specified in different ways such as single quoted, double quoted, heredoc, or Nowdoc.
- You can categorize operators as unary prefix, binary, ternary, or assignment operators.
- PHP language provides support to different operator types such as arithmetic, comparison, logical, assignment, and conditional operators.
- A regular expression is a character sequence that forms a search pattern.

© Aptech Limited Architecting Web Applications using PHP/ Session 4/ 32 of 32

Use slide 32 to summarize the session. You will end the session with a summary of what has been taught in the session. Tell students the pointers of the session. This will be a revision of the current session.

4.3 Post Class Activities for Faculty

You should familiarize yourself with the topics of the next session.

Tips:

You can also check the Articles/Blogs/Expert Videos uploaded on the Online Varsity site to gain additional information related to the topics covered in the next session.

Session 5 – Conditional and Flow Control Statements and Arrays

5.1 Pre-Class Activities

Before beginning the session, you should fully familiarize yourself with its topics. Prepare a question or two that will be a key point to relate the current session objectives.

5.1.1 Teaching Skills

To teach this session, you should be well-versed with the concept of internationalization and design patterns in Java. You must also be familiar with the concept of localization.

You must use the given images and teach the concepts in the theory class. For teaching in the class, you are expected to use slides and LCD projectors.

Tips:

It is recommended that you test the understanding of the students by asking questions in between the class.

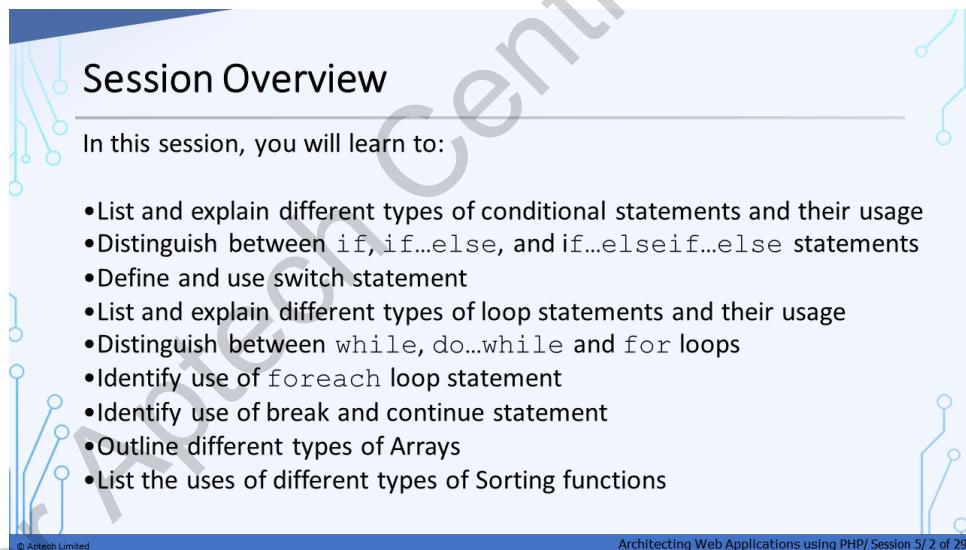
In-Class Activities

Follow the order given here during In-Class activities.

Overview of the Session

Give the students an overview of the current session in the form of session objectives. Read out the objectives on slide 2.

Slide 2



The slide has a blue header bar with the title "Session Overview". Below the title, there is a sub-section titled "In this session, you will learn to:" followed by a bulleted list of 12 items. The slide features a decorative background with abstract blue and white line art. At the bottom, there is a footer bar with the text "© Aptech Limited" and "Architecting Web Applications using PHP/Session 5/2 of 29".

Session Overview

In this session, you will learn to:

- List and explain different types of conditional statements and their usage
- Distinguish between if, if...else, and if...elseif...else statements
- Define and use switch statement
- List and explain different types of loop statements and their usage
- Distinguish between while, do...while and for loops
- Identify use of foreach loop statement
- Identify use of break and continue statement
- Outline different types of Arrays
- List the uses of different types of Sorting functions

© Aptech Limited Architecting Web Applications using PHP/Session 5/2 of 29

Show slide 2 and give a brief overview of the current session in the form of session objectives. Inform students that this session begins with a brief explanation of various kinds of conditional statements and their usage in PHP. It introduces different types of conditional and loop statements, explaining how and where to use them. Then, it explains which type of constructs are to be used in different scenarios. Further, it introduces the concept of arrays in PHP and continues to explain different types of arrays. The session finally concludes after explaining various functions in PHP used for sorting arrays.

5.2 In-Class Explanations

Slide 3

PHP Conditional Statements

Conditional statements in PHP help the user to arrive at a decision based on certain conditions. When a user writes code in PHP, there will be scenarios where different actions must be performed for different conditions.

Conditional statements are used in such scenarios. These conditions are defined by a set of conditional statements which contain expressions that are evaluated to a Boolean value of true or false.

PHP language supports the following conditional statements:

- if
- if...else
- If...elseif...else
- switch

© Aptech Limited Architecting Web Applications using PHP/ Session 5/ 3 of 29

Show slide 3 and explain to students that when a user wants to perform different actions for different conditions, he/she can use conditional statements in their code to be able to do this. Based on certain conditions, the user can arrive at a decision. A set of conditional statements define these conditions. These statements contain expressions that are evaluated to a Boolean value of true or false.

Following conditional statements are supported by PHP:

- if : Executes some code when one condition is true. In this case, the `if` statement only executes the code when the condition is **true**.
- if...else: When the condition is true, the statement/statement block following `if..else` will be executed. In case condition is false, then statement after the `else` will be executed.
- if...elseif...else: If there are more than two conditions, then `if...elseif...else` statement executes different codes.
- Switch: Selects one of many code blocks to be executed.

Ask the students following question. Wait for a response before you answer.

In Class Question: What is the basic syntax for calling a function?

Answer: `function_name (expression1, expression2)`

Additional Information:

Refer to following links for more information:

<https://www.includehelp.com/php/conditional-statements.aspx>

https://www.tutorialspoint.com/php/php_decision_making.htm

https://www.w3schools.com/php/php_if_else.asp

if Statement [1-2]

An if statement allows you to execute one or more statements after evaluating a specified logical condition.

It starts with the `if` keyword and is followed by the condition.

If the condition evaluates to *true*, the block of statements following the `if` statement is executed.

If the condition evaluates to *false*, the block of statements following the `if` statement is ignored and the statement after the block is executed.

```
graph TD; START([START]) --> Init[/num = -4/]; Init --> Cond{num < 0 ?}; Cond -- Yes --> Print[/PRINT<br>"The number is negative"/]; Print --> STOP([STOP]); Cond -- No --> STOP;
```

Figure 1:
Flowchart Representing if Statement

Show slide 4 and explain to students following example:

Consider a real-life analogy where you want to go out. However, it is rainy outside, so you decide to check the weather first. If you feel it will rain, you must bring an umbrella or raincoat. On the contrary, you do not have to bring them if the weather forecast is sunny and clear. As a result, carrying or not carrying rain gear is determined by the weather condition. Conditional statements are used in programming to achieve this decision-making sequence of action.

Similar to the given example, an `if` statement allows the user to execute one or more statements after analyzing a logical condition. The `if` keyword comes first, followed by the condition. The block of statements following the `if` statement is executed if the condition evaluates to true. If the condition is false, the block of statements after the `if` statement is skipped, and the statement after the block is executed.

Figure 1 contains a flowchart representing an if statement. In the figure, the variable `num` is assigned by value `-4`. It checks for the condition whether `num` is less than zero. If this condition is true, it prints 'the number is negative'. However, it ends the search if the condition evaluates false.

Additional Information:

Refer to following links for more information:

<https://www.php.net/manual/en/control-structures.if.php>
<https://www.phptutorial.net/php-tutorial/php-if/>

Slide 5

if Statement [2-2]

Code Snippet:

```
<?php  
$timeofday = date("H");  
if ($timeofday < "12") {  
    echo "Happy Morning!";  
}  
?>
```

Output
Happy Morning!

Code Snippet shows a sample program that uses if statement to display 'Happy Morning' if the current time is less than 12 o'clock.

In Code Snippet, the program is executed when current time is less than 12 o'clock. Hence, the output is 'Happy Morning!'

© Aptech Limited Architecting Web Applications using PHP/ Session 5/ 5 of 29

Show slide 5 and explain the syntax of *if* statement:

Syntax

```
if (condition) {  
code to be executed when the condition is true;  
}
```

Moreover, show Code Snippet to the students and explain how an if statement works. The code contains an if statement with a condition `$timeofday < "12"`. It executes the program only if the value of the current time is less than 12 o'clock. Thus, 'Happy Morning' is displayed on the screen if the condition is true. Otherwise, the code is not executed if the condition is false.

Ask the students the following question. Wait for a response before you answer.

In-Class Question: What are the three arguments of the *if* function?

Answer: logical_test, value_if_true, value_if_false.

Slide 6

if...else Statement [1-2]

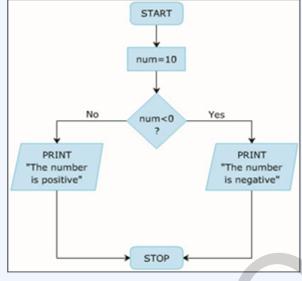
The if statement executes a block of statements only if the specified condition is true.

However, in some situations, it is required to define an action for a false condition. This is done using an if...else construct.

The if...else construct starts with if block followed by an else block.

The else block starts with else keyword followed by a block of statements.

If the condition specified in the if statement evaluates to false, the statements in the else block are executed.



```
graph TD; START([START]) --> num10[num=10]; num10 --> cond{num<0 ?}; cond -- No --> printPos[PRINT "The number is positive"]; printPos --> STOP([STOP]); cond -- Yes --> printNeg[PRINT "The number is negative"]; printNeg --> STOP;
```

Figure 2: Flowchart for if...else Statement

Architecting Web Applications using PHP/ Session 5/ 6 of 29

Show slide 6 and tell the students that if the supplied condition is true, the if statement runs a block of statements. In some cases, though, it is necessary to define an action for a false condition. An if...else construct is used to do this. The if...else statement begins with an if block and ends with an else block. The else block begins with the keyword else and is followed by a series of statements. The instructions in the else block are performed if the condition indicated in the if statement evaluates to false.

Consider the flowchart in Figure 2, depicting the if...else statement. The num variable is set to 10 in this case. If num is less than zero, the condition is checked. If the condition is true, the message 'The number is positive' will be printed, and the execution will terminate. If the condition evaluates to false, the sentence 'The number is negative' is printed, and the procedure exits.

Additional Information:

Refer to following links for more information:

<https://www.javatpoint.com/php-if-else>
<https://www.educba.com/if-else-statement-in-php/>
<https://www.phptutorial.net/php-tutorial/php-if-else/>
<https://linuxhint.com/php-if-else-statements/>

Slide 7



if...else Statement [2-2]

Code Snippet:

```
<?php  
$hourOfDay = date("H");  
if ($hourOfDay < "18") {  
    echo "Have a Nice Day ahead!";  
}  
else {  
    echo "Good Night!";  
}  
?>
```

Output
Good Night!

In Code Snippet, the else statement will be executed and the output displayed is **Good Night!** as the current time hour is greater than 18.

However, if the time hour is less than 18, then if statement executes to display **Have a Nice Day ahead!**

Architecting Web Applications using PHP/ Session 5 / 7 of 29

Show slide 7 and explain the syntax of the *if...else* statement:

Syntax

```
if (condition) {  
    Code to be executed when the condition is true;  
}  
else {  
    Code to be executed when the condition is false;  
}
```

Show Code Snippet 2 to the students and demonstrate the use of the *if...else* statement.

Output explanation: The *else* statement will be executed in Code Snippet and the output displayed will be **Good Night!** if the current hour would be greater than 18.

On the other hand, if the time hour is less than 18, then *if* statement is executed to display **Have a Nice Day ahead!**

Ask the students the following question. Wait for a response before you answer.

In-Class Question: What is the difference between *if else* and multiple *if* statements in PHP?

Answer: *if...else* statement executes some code if a condition is true and another code if that condition is false. While multiple *if* statements evaluate each statement separately and if the conditions are satisfied, the code in all of them might be executed.

if...elseif...else Statement [1-2]

Statement if...elseif...else is used when the user wants to handle multiple conditions.

In other words, if...elseif...else statement executes different codes when there are more than two conditions.

```

graph TD
    START([START]) --> Init[/num=13/]
    Init --> Cond{num<0 ?}
    Cond -- No --> Odd{num%2=0 ?}
    Odd -- No --> PrintOdd[Print "The number is odd"]
    Odd -- Yes --> PrintEven[Print "The number is even"]
    Cond -- Yes --> PrintNeg[Print "The number is negative"]
    PrintEven --> STOP([STOP])
    PrintOdd --> STOP
    PrintNeg --> STOP
  
```

Figure: Flowchart for if...elseif...else Statement

Show slide 8 and tell the students that when a user is required to handle numerous situations, the if...elseif...else statement is implemented. When there are more than two criteria, the if...elseif...else statement runs separate codes.

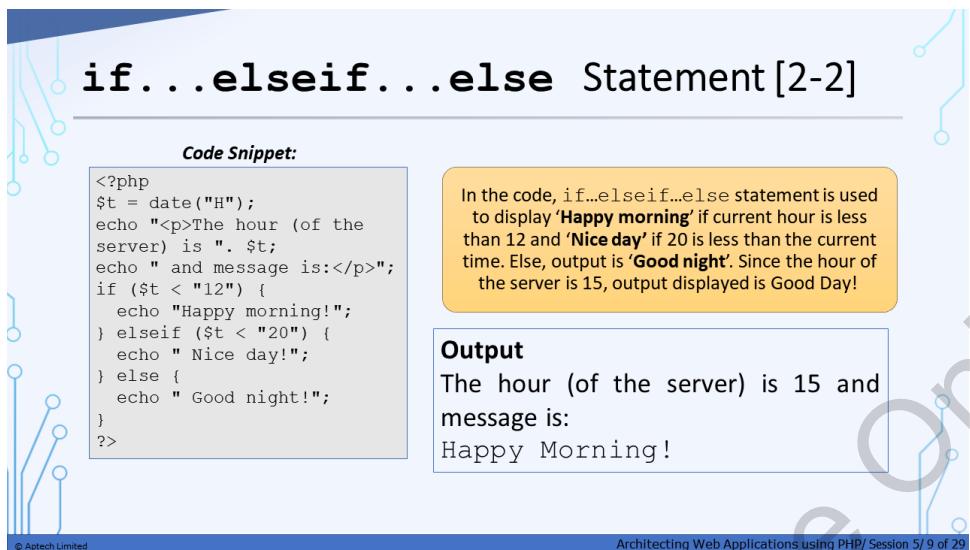
Figure depicts an if...elseif...else statement flowchart. The number 13 has been allocated to the num variable in this case. If the condition is true where num is less than 0, the code prints 'The number is negative' and terminates the program. If the condition evaluates false, the elseif statement is checked, which is num%2=0. If the number is divisible by 2, this statement is true. It displays 'The number is even' if it evaluates true and 'The number is odd' if it evaluates false.

Additional Information:

Refer to following links for more information:

- <https://www.tutorialrepublic.com/php-tutorial/php-if-else-statements.php>
- <https://www.php.net/manual/en/control-structures.elseif.php>
- <https://www.tutorialrepublic.com/php-tutorial/php-if-else-statements.php>
- <https://tutorials.supunkavinda.blog/php-conditionals>

Slide 9



if...elseif...else Statement [2-2]

Code Snippet:

```
<?php
$t = date("H");
echo "<p>The hour (of the server) is ". $t;
echo " and message is:</p>";
if ($t < "12") {
    echo "Happy morning!";
} elseif ($t < "20") {
    echo " Nice day!";
} else {
    echo " Good night!";
}
?>
```

In the code, if...elseif...else statement is used to display 'Happy morning' if current hour is less than 12 and 'Nice day' if 20 is less than the current time. Else, output is 'Good night'. Since the hour of the server is 15, output displayed is Good Day!

Output

The hour (of the server) is 15 and message is:
Happy Morning!

© Aptech Limited Architecting Web Applications using PHP/ Session 5/ 9 of 29

Show slide 9 and explain the syntax of the if...elseif...else statement:

Syntax

```
if (condition) {
    Executes when this condition of the code is true;
} elseif (condition) {
    Executes when the first condition of the code is false and
    the second condition is true;
} else {
    Executes when all conditions of the code are false;
}
```

Also, show Code Snippet to the students and describe the use of the if...elseif...else statement.

Output explanation: The if...elseif...else statement is used in the code to display 'Happy morning' if the current hour is less than 12. Similarly, 'Nice day' gets displayed if the current time is less than 20. Otherwise, the output is 'Good night.' As the server's hour is 15, the output returned is Good Day!

Ask the students the following question. Wait for a response before you answer.

In-Class Question: Why is *else if* required in a code?

Answer: The main reason to use *else if* is to avoid excessive indentation.

Slide 10

switch Statement

Code Snippet:

```
<?php  
$language = "PHP 8";  
switch ($language) {  
    case "C":  
        echo "Your favorite language is C ";  
        break;  
    case "PHP 8":  
        echo "Your favorite language is PHP 8";  
        break;  
    case "C++":  
        echo "Your favorite language is C++";  
        break;  
    default:  
        echo "Your favorite language is neither C, PHP 8, nor C++";  
}  
?>
```

Output

Your favorite language is PHP 8

Code Snippet shows how to use the switch statement.

Show slide 10 and explain to the students that `switch` statements are used in PHP to perform various actions based on different criteria. It is frequently used as a replacement for several `if...elseif` conditions.

Explain the syntax of `switch` statement:

Syntax

```
switch (n) {  
    case label1:  
        execute the code block when label1=n;  
        break;  
    case label2:  
        execute the code block when label2=n;  
        break;  
    case label3:  
        execute the code block when label3=n;  
        break;  
    ...  
    default:  
        execute the code block when n differs from all labels;  
}
```

The expression `n` is evaluated once in the first step and multiple code blocks are run depending on the value of `n`. The label values for each case in the structure are then, equated to the value of the expression. If they are the same, the code block for that condition is run. The `break` at the end of the code block prevents the code from moving on to the next case automatically. The `default` statement is used if there is no match.

Further, provide an output explanation for the Code Snippet. The value in the variable `language` is set to `PHP 8` in Code Snippet, and the control does not correspond to the first

case. The control then matches the variable value and the second case. As a consequence, here is the output:

Your favorite language is PHP 8

The value for language is hardcoded in this code. In real-life conditions, however, the value would be accepted from the user.

Ask the students the following question. Wait for a response before you answer.

In-Class Question: What are the two main structures for branching?

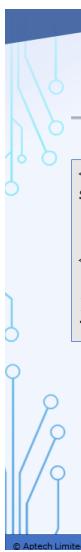
Answer: if and switch

Additional Information:

Refer to following links for more information:

https://www.w3schools.com/php/php_switch.asp

<https://www.javatpoint.com/php-switch>



PHP while Loop

Code Snippet:

```
<?php  
$n = 1;  
while($n <= 10) {  
    echo "The number is: $n  
<br>";  
    $n++;  
}  
?>
```

Code Snippet shows a sample program for displaying numbers from 1 to 10 using a while loop.

Output

```
The number is: 1  
The number is: 2  
The number is: 3  
The number is: 4  
The number is: 5  
The number is: 6  
The number is: 7  
The number is: 8  
The number is: 9  
The number is: 10
```

Architecting Web Applications Using PHP / Session 5 / 11 of 29

Show slide 11 and tell the students that loops in PHP are used to repeat the same block of code until a specified condition is met, just like in any other programming language.

`while`, `do...while`, `for`, and `foreach` are the four different types of loops supported by PHP.

The condition is checked first in the `while` loop and if it is met, the code block is executed until the condition becomes false.

Explain the syntax of the `while` loop:

Syntax

```
while (condition is true) {  
    code to be executed;  
}
```

Using a `while` loop, Code Snippet illustrates a sample program for printing numbers from 1 to 10.

Explain the Output: Since the value of `n` in Code Snippet is originally assigned as 1, the first number in the output is 1.

The `++` operator is then used to increment `n`, resulting in `n++`. As, the given `while` condition is `n <= 10`, the next numerical value will be 2 and so on until the number 10 is displayed.

Ask the students the following question. Wait for a response before you answer.

In-Class Question: Which loop evaluates the conditional expression before processing the loop?

Answer: `while` loop

Additional Information:

Refer to following links for more information:

https://www.w3schools.com/php/php_looping_while.asp

<https://www.javatpoint.com/php-while-loop>



PHP do...while Loop

Code Snippet:

```
<?php  
$n = 15;  
do {  
    echo "The number is:  
$n <br>";  
    $n++;  
} while ($n <= 20);  
?>
```

Output

```
The number is: 15  
The number is: 16  
The number is: 17  
The number is: 18  
The number is: 19  
The number is: 20
```

Code Snippet shows a sample program to display numbers from 1 to 5 using the do...while loop.

© Aptech Limited

Architecting Web Applications using PHP/ Session 5/ 12 of 29

Show slide 12 and tell the students that before testing the condition, the do...while loop runs the code block and executes it. If the condition is found to be true, the procedure is repeated until it is false.

Explain the syntax of the do...while loop:

Syntax

```
do {  
    code to be executed;  
} while (condition is true);
```

Furthermore, explain the output of Code Snippet. The value of n is initially set to 15, hence, the first number in the output is 15. Then, by incrementing n as n++ using the ++ operator, the next number value is increased to 16, and the condition is checked again until number 20 is reached, as the while condition is n <= 20.

There is a significant distinction between while and do-while. The while loop checks the condition first and then performs instruction, whereas the do-while loop runs instruction first and then checks condition.

Ask the students the following question. Wait for a response before you answer.

In-Class Question: What is the main difference between while and do while loops?

Answer: Both while and do while loops are nearly similar, the only difference being that the do while loop checks for the condition after executing statements, and therefore, is an example of Exit Control Loop.

Additional Information:

Refer to following links for more information:

https://www.w3schools.com/php/php_looping_do_while.asp

<https://www.javatpoint.com/php-do-while-loop>

Slide 13



PHP for Loop

Code Snippet:

```
<?php
for ($n = 11; $n <= 20;
$n++) {
    echo "The number is:
$n <br>";
}
?>
```

Output

```
The number is: 11
The number is: 12
The number is: 13
The number is: 14
The number is: 15
The number is: 16
The number is: 17
The number is: 18
The number is: 19
The number is: 20
```

Code Snippet shows a sample program to display numbers from 11 to 20 using the **for loop**.

Following parameters are used in for loop:

- init counter is used to initialize the counter value of loop.
- test counter is used to evaluate iteration for each loop. The loop continues if it evaluates to TRUE. The loop ends if it evaluates to FALSE.
- increment counter is used to increase the counter value of loop.

© Aptech Limited Architecting Web Applications using PHP / Session 5 / 13 of 29

Show slide 13 and tell the students that a **for** loop is used to repeat a code block a definite number of times. Simply put, a **for** loop can be used when the user already knows how frequently the code should run.

Explain the syntax of the **for** loop:

Syntax

```
for (init counter; test counter; increment counter) {
    execute the code block for each iteration;
}
```

Explain the working of the program given in Code Snippet. The value of **n** is originally set to 11, thus, the first number in the output is 11. Following, the value of **n** is increased using the **++** operator, which is **n++**, so the next number value is 12. The condition is then checked again until number 20.

In contrast to the **while** and **do-while** loops, the **for** loop assigns variable values, checks conditions, and increments variables in a single line.

The parameters used in the **for** loop are: **init counter**, **test counter**, and **increment counter**.

Additional Information:

Refer to following links for more information:

https://www.w3schools.com/php/php_looping_for.asp
<https://www.javatpoint.com/php-for-loop>



foreach Loop Statement

Code Snippet:

```
<?php
$flower = array("Rose"=>"10",
"Lotus"=>"30");
foreach($flower as $f => $value) {
    echo "$f= $value<br>";
}>
```

Output

```
Rose=10
Lily=20
Lotus=30
```

Code Snippet shows a sample program to display both keys and values of the array using **foreach** loop.

© Aptech Limited Architecting Web Applications Using PHP / Session 5 / 14 of 29

Show slide 14 and tell the students that when it comes to iterating over an array of elements, the **foreach** loop comes in handy. It only works with arrays and loops through all of their key/value pairs.

Explain the syntax of **foreach** loop:

Syntax

```
foreach ($array as $value) {
    code to be executed;
}
```

For each loop iteration, the current array element's value is assigned to `$value` and the array pointer is moved by one until it reaches the last array element.

Using the **foreach** loop, Code Snippet displays flower names and numbers. The `=>` operator is used in the array and **foreach** loop to display both keys and values.

TIP: The double arrow operator is used as an access mechanism for arrays. It indicates that whatever is on the left of it, we will have a corresponding value of whatever is on the right in array context. This can be used to set values of any acceptable type into a corresponding index of an array.

Ask the students the following question. Wait for a response before you answer.

In-Class Question: What are bounded loops?

Answer: Loops that are iterated for a fixed number of times are known as bounded loops. A range or a collection could be used to determine the number of times.

Additional Information:

Refer to following link for more information:

https://www.w3schools.com/php/php_looping_foreach.asp

break Statement

Code Snippet:

```
<?php
for ($n = 5; $n < 15; $n++) {
    if ($n == 9) {
        break;
    }
    echo "Number is: $n <br>";
}
?>
```

Output

```
Number is: 5
Number is: 6
Number is: 7
Number is: 8
```

Code Snippet shows a sample program to display the use of break statement and how to jump out of a loop.

© Aptech Limited

Architecting Web Applications using PHP / Session 5 / 15 of 29

Show slide 15 and tell the students that using the `break` statement enables a user to exit a loop construct. When `x` equals 4, for example, the loop ends. The `break` statement is also applied in `switch case` statements to prevent the code from running into the next case automatically. It is mostly required to get out of a loop or exit a code block in a program. When the program detects `break`, it exits the corresponding block or loop and runs the code after the loop or block.

Code Snippet contains a sample program that demonstrates how to apply the `break` statement and how to exit a loop.

Provide the following output explanation to the students:

The `for` loop is used in Code Snippet to display numbers from 5 to 14, but there is a `break` statement within the `if` block. The loop iterates from 5 to 9, but when the number crosses 9, the program control exits the `for` loop, and no more numbers are displayed. As a result, only 5 to 8 are printed in the output.

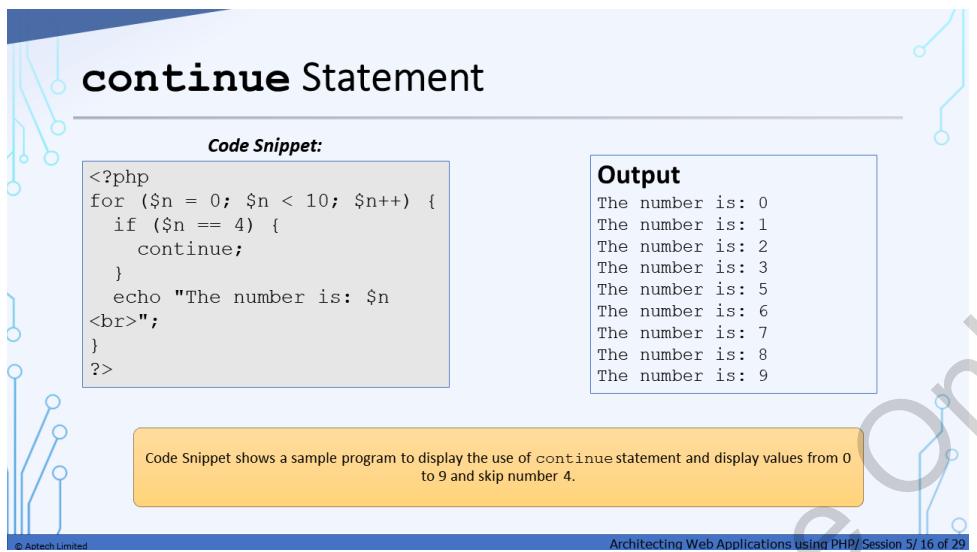
Additional Information:

Refer to following links for more information:

<https://www.javatpoint.com/php-break>

<https://www.php.net/manual/en/control-structures.break.php>

<https://www.tutorialspoint.com/php-break-statement>



continue Statement

Code Snippet:

```
<?php
for ($n = 0; $n < 10; $n++) {
    if ($n == 4) {
        continue;
    }
    echo "The number is: $n
<br>";
}
?>
```

Output

```
The number is: 0
The number is: 1
The number is: 2
The number is: 3
The number is: 5
The number is: 6
The number is: 7
The number is: 8
The number is: 9
```

Code Snippet shows a sample program to display the use of `continue` statement and display values from 0 to 9 and skip number 4.

© Aptech Limited Architecting Web Applications Using PHP/ Session 5/ 16 of 29

Show slide 16 and tell the students that the `continue` statement breaks an iteration in the loop and continues with the following iteration when a given condition is met. If the loop has one or more statements after the `continue`, they will all be skipped because control will skip them and go straight to the next iteration.

Give an output explanation to the students:

In Code Snippet, the `for` loop is used to display numbers from 0 to 9, however, the `if` block contains a `continue` statement. As a result, when the value approaches 4, the program control does not exit the `for` loop. Instead, it skips the remaining statements and moves on to the next iteration. As a result, the values 0–3 and 5–9 are shown in the output.

Ask the students the following question. Wait for a response before you answer.

In-Class Question: What is the difference between `break` and `continue` statements?

Answer: The `break` statement is used to exit from the loop constructs while the `continue` statement is not used to exit from the loop constructs.

Additional Information:

Refer to following link for more information:

<https://www.javatpoint.com/php-continue>

array() function is used to create an array in PHP.

An array can store multiple values with the same name and the values can be accessed by referencing an index number.

For example, if you want to hold 300 numbers, then instead of defining 300 variables, it is easy to define an array of length 300.

Arrays are classified into three types:

- Numerical array
- Associative array
- Multi-dimensional array

© Aptech Limited Architecting Web Applications using PHP / Session 5 / 17 of 29

Show slide 17 and tell the students to assume that the user must save and print five names. Using five separate string variables, this task is simple to perform. However, if the number increases from five to a hundred, the user or developer will find it difficult to create so many separate variables.

Consider another example. If you have a list of items, such as car names, storing the cars in single variables might look like this:

```
$cars1 = "Ford";  
$cars2 = "Aston Martin";  
$cars3 = "Toyota";
```

However, if the car list contains 300 names, having 300 variables becomes problematic. Repeating the same block of code in the program and running a common block of code for all 300 names is impractical. In such cases, an array is useful as it allows users to keep all of the elements in a single variable. An array is a special kind of data structure that can hold one or more values of the same type. Arrays are used to hold a list of elements of similar types in a single variable that can be retrieved by their index or key. This avoids the time and effort of generating a unique variable for each data set.

There are three different types of arrays:

- Numerical array: This array, as the name implies, has numerical index. The values are stored and accessed in a linear manner.
- Associative array: This is an array with strings as an index. rather than a strict linear index order, this array stores element values in association with specific key values.
- Multi-dimensional array: An array containing one or more arrays and values are accessed using multiple indices.

Ask the students the following question. Wait for a response before you answer.

In-Class Question: Which in-built function is used to add a value to the end of an array?

Answer: `array_push()`

Additional Information:

Refer to following links for more information:

<https://www.geeksforgeeks.org/php-arrays/#:~:text=Arrays%20in%20PHP%20is%20a,different%20variable%20for%20every%20data.>
https://www.w3schools.com/php/php_arrays.asp
https://www.tutorialspoint.com/php/php_arrays.htm

Slide 18

The slide features a blue header bar with the title "Numerical Array". Below the title is a "Code Snippet:" section containing PHP code. To the right is an "Output" section showing the execution results. A note at the bottom left explains the purpose of the code snippet.

Code Snippet:

```
<?php
/* Method for creating an array. */
$num = array(21, 22, 23, 24, 25, 26);
foreach( $num as $value )
{
echo "Value is $value <br />";
}
$num[0]="one";
$num[1]="two";
$num[2]=23;
foreach( $num as $value )
{
echo "Value is $value <br />";
}
?>
```

Output

```
Value is 21
Value is 22
Value is 23
Value is 24
Value is 25
Value is 26
Value is one
Value is two
Value is 23
```

Code Snippet shows a sample program that shows how to create and access numeric arrays.

© Aptech Limited Architecting Web Applications using PHP/ Session 5/ 18 of 29

Show slide 18 and tell the students that numbers, strings, or any other object can be stored in numerical arrays, but their indexes are represented by numbers. By default, the array index is zero.

Provide an explanation of the output to students:

In Code Snippet, an array is generated, and the numbers 21 to 26 are displayed using a `foreach` loop.

Additional Information:

Refer to following links for more information:

https://www.tutorialspoint.com/php/php_arrays.htm
<https://tutorialink.com/php/php-numeric-array.php>

Slide 19

The slide features a decorative background with blue lines and circles. At the top center is the title "Associative Array". Below it is a "Code Snippet:" section containing two blocks of PHP code. The first block creates an associative array with string keys ("John", "Roger", "Susan") and integer values (20, 10, 60). The second block uses the array keyword to create an associative array with string keys ("John", "Roger", "Susan") and string values ("Adult", "Child", "Senior Citizen"). Both blocks include echo statements to print the array elements. To the right of the code is an "Output" box showing the resulting text. At the bottom left is a copyright notice for "Aptech Limited", and at the bottom right is the text "Architecting Web Applications Using PHP/ Session 5/ 19 of 29".

```
<?php
/* One approach to create an associative array. */
$Age = array(
"John" => 20,
"Roger" => 10,
"Susan" => 60
);
echo "Age of John is ". $Age['John'] . "<br />";
echo "Age of Roger is ". $Age['Roger']. "<br />";
echo "Age of Susan is ". $Age['Susan']. "<br />";
/* Another approach to create an associative array. */
$Age['John'] = "Adult";
$Age['Roger'] = "Child";
$Age['Susan'] = "Senior Citizen";
echo " John is ". $Age['John'] . "<br />";
echo " Roger is ". $Age['Roger']. "<br />";
echo " Susan is ". $Age['Susan']. "<br />";
?>
```

Code Snippet shows a sample program to create an associative array.

Output

Age of John is 20
Age of Roger is 10
Age of Susan is 60
John is Adult
Roger is Child
Susan is Senior Citizen

© Aptech Limited Architecting Web Applications Using PHP/ Session 5/ 19 of 29

Show slide 19 and tell the students that associative arrays are pretty similar to numeric arrays in terms of functionality, although they differ index-wise. Using an associative array with a string as an index, users can construct a strong link between the key and the values.

Use Code Snippet as a referring point, where John is a string key or index with the value 20, that is, Age ['John']=20.

Numbers are used as indexes in numerical arrays. In Code Snippet, an array's index is expressed as a number, such as num[0]="one," num[2]="two", and so on.

A numerically indexed array is not the best solution for storing employee wages in an array. Alternately, the names of the employees should be the key in the associative array, with their individual salaries as the value. There are two methods for storing array values. The first approach uses the array keyword to construct an array. All of the values and their related keys can be stored in a single statement.

```
$Age=array("John" => 20, "Roger" => 10, "Susan" => 60);  
Here, John is the Key and the Value is 20.
```

The array keyword is not used to create arrays in the second method. Since a single line is required to hold a single value, values are stored within multiple lines. Only a single quote is used to store and show the values for the string index.

That is, \$Age ['John']= "Adult";
Here, John is the String index and the value is Adult.

Note: The associative array should not be kept inside double quotes while printing, otherwise it will not return any value.

Provide a detailed output explanation to the students:

The values in this code are stored in arrays. The array is assigned to a variable called Age, and the values of the array are shown using the variable name. The variable and key associated with a certain value are shown. That is, the key in the first statement is John, and the value associated with it is 20. As a result, the output - Age of John is 20. The key linked with the value 10 is Roger, and so on.

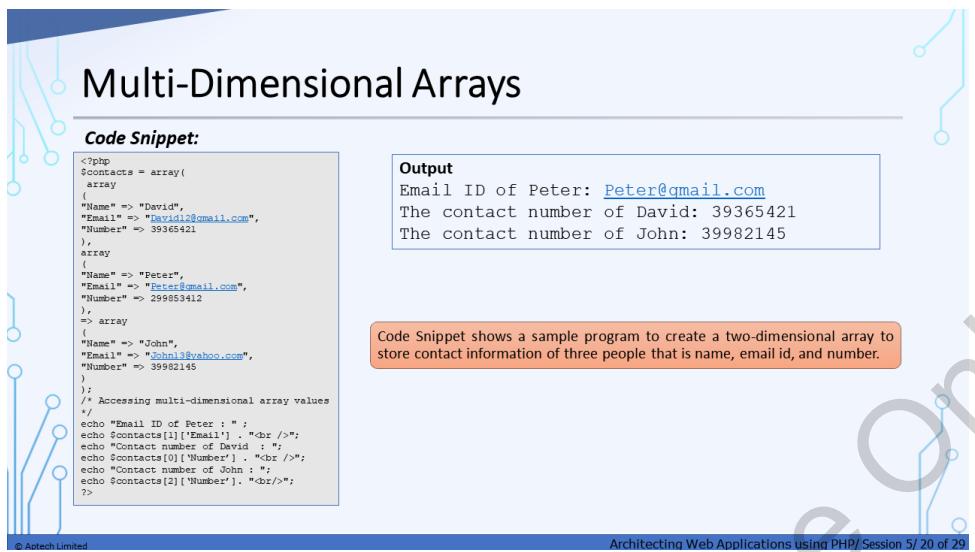
Additional Information:

Refer to following links for more information:

https://www.w3schools.com/php/php_arrays_associative.asp

<https://www.geeksforgeeks.org/associative-arrays-in-php/>

Slide 20



Multi-Dimensional Arrays

Code Snippet:

```
<?php  
$contacts = array(  
    array(  
        "Name" => "David",  
        "Email" => "David12@gmail.com",  
        "Number" => 39365421  
    ),  
    array(  
        "Name" => "Peter",  
        "Email" => "Peter@gmail.com",  
        "Number" => 299853412  
    ),  
    array(  
        "Name" => "John",  
        "Email" => "John13@yahoo.com",  
        "Number" => 39982145  
    )  
);  
/* Accessing multi-dimensional array values */  
echo "Email ID of Peter : " ;  
echo $contacts[1]['Email'] . "<br />";  
echo "Contact number of David : " ;  
echo $contacts[0]['Number'] . "<br />";  
echo "Contact number of John : " ;  
echo $contacts[2]['Number'] . "<br />";  
?>
```

Output

```
Email ID of Peter: Peter@gmail.com  
The contact number of David: 39365421  
The contact number of John: 39982145
```

Code Snippet shows a sample program to create a two-dimensional array to store contact information of three people that is name, email id, and number.

Architecting Web Applications using PHP/ Session 5/ 20 of 29

Show slide 20 and tell the students that every element in the main array can be an array in a multi-dimensional array. Each sub-array element can be an array, and so on.

The values in the multi-dimensional array are accessed using multiple numbers of indexes.

To access the values of an array, the name of a variable must be used, such as contacts in which array values are assigned. Thus, the variable's name, array index, and array sub-index must all be written first, as shown here:

```
$contacts[1] ['Name'] ="Peter";
```

Here, '1' is the Array Index and 'Name' is the Sub Index for accessing Peter.

Further, provide an output explanation for the given Code Snippet:

Multi-dimensional arrays are used to store the names, emails, and phone numbers of three people in Code Snippet. The email ID and contacts of a person are retrieved from the array and printed in the output. The data can be retrieved by giving the array's location and name.

`$contacts[0]` has David's Name, Email, and Number whereas `$contacts[1]` contains Peter's Name, Email, and Number. John's Name, Email, and Number are stored in the array `$contacts[2]`.

Additional Information:

Refer to following links for more information:

https://www.w3schools.com/php/php_arrays_multidimensional.asp
<https://www.geeksforgeeks.org/multidimensional-arrays-in-php/>

Slide 21

The slide has a decorative background with blue and white circuit board patterns on the left and right sides.

Indexed Arrays

Code Snippet:

```
<?php
$student=array("Peter", "John",
"David", "Sean");
echo "Names of the students
are: ".
$Student[0]. ",".$Student[1]. "
,".$Student[2].
", and ".$Student[3]. ".";
?>
```

Output

```
Names of the students
are: Peter, John, David, and
Sean.
```

Note: Code Snippet shows a sample program for creating an indexed array named \$Student, assigning it with four elements. The output is a text that contains the array values.

© Aptech Limited Architecting Web Applications using PHP/ Session 5/ 21 of 29

Show slide 21 and explain to students that an indexed array is defined by an index number. The index starts from zero by default. An index is used to represent each element in an array. These index arrays are used to hold strings, numeric values, or any objects in PHP.

To construct index arrays, any one of the following two methods can be used:

Method 1

Automatically assign the index (index always starts at 0):

```
$Student=array("Peter", "John", "David");
```

Method 2

Manually assign the Index:

```
$Student[0]= "Peter"
$Student[1]= "John"
$Student[2]= "David"
```

In the first method, to create an array and store the values, the keyword `array` is used. Here, a variable name that is assigned with an array can be declared only in one line that is: `$Student=array("Peter", "John", "David")`. There is no requirement to write index numbers 0, 1, 2, and so on.

In the second method, `array` keyword is not used, instead variable name and index numbers are written, that is:

```
$Student[0]= "Peter"
```

```
$Student[1] = "John"  
$Student[2] = "David"
```

In the second method, array values are declared in multiple lines.

Show the Code Snippet and explain its output:

The student names in Code Snippet are presented using array indexes. Method 1 of automatically assigning indexes is applied in this code. The keyword `array` is used to declare an array in a single line. When a programmer wants to declare an array in a single line, this method is applied.

Additional Information:

Refer to following links for more information:

https://www.w3schools.com/php/php_arrays_indexed.asp

<https://www.javatpoint.com/php-indexed-array>

Slide 22

Sorting Arrays

The most popular functions for sorting arrays are:

- **sort()** and **rsort()**: Indexed arrays are sorted using this array.
- **ksort()** and **krsort()**: Associative arrays are sorted by key using this array.
- **asort()** and **arsort()**: These are used to sort associative arrays by value.

Sort Functions	Description
sort()	Sort arrays in ascending order
rsort()	Sort arrays in descending order
asort()	Sort associative arrays in ascending order, according to the value
ksort()	Sort associative arrays in ascending order, according to the key
arsort()	Sort associative arrays in descending order, according to the value
krsort()	Sort associative arrays in descending order, according to the key

Table: Array Sort Functions

© Aptech Limited Architecting Web Applications using PHP/ Session 5/ 22 of 29

Show slide 22 and tell students that PHP comes with a number of built-in functions for sorting array elements alphabetically or numerically in ascending or descending order. It also improves the efficiency of searching. Familiarize them with the most prominent functions for sorting arrays.

Sorting changes the internal order of elements in an array and optionally rewrites the keys to reflect this new order. For example, users might use sorting to arrange a list of scores from biggest to smallest, to alphabetize a list of names or to order a set of users based on how many messages they posted.

PHP provides three ways to sort arrays—sorting by keys, sorting by values without changing the keys, or sorting by values and then changing the keys.

Using sorting users can analyze a list in a more effective way. In most situations, users must sort an array by value.

Sorting significantly improves the ability to search and also helps in raising or decreasing patterns based on a linear relationship between the data components. Describe the Sort functions for arrays in PHP listed in Table.

Additional Information:

Refer to following links for more information:

https://www.w3schools.com/php/php_arrays_sort.asp
<https://www.php.net/manual/en/array.sorting.php>

Slide 23

Sort Array in Ascending Order - **sort ()**

Code Snippet:

```
<?php  
$Student = array("Peter", "John", "David");  
sort($Student);  
$clength = count($Student);  
for($x = 0; $x < $clength; $x++) {  
    echo $Student[$x];  
    echo "<br>";  
}  
?>
```

Output

```
David  
John  
Peter
```

In Code Snippet, the elements of the \$Student array are sorted in ascending alphabetical order.

© Aptech Limited Architecting Web Applications using PHP / Session 5 / 23 of 29

Show slide 23 and tell students that the elements of the \$Student array are sorted in ascending alphabetical order in Code Snippet.

Explain the output of Code Snippet:

The built-in function `sort ()` is used to sort student names in alphabetical order.

Ask the students the following question. Wait for a response before you answer.

In-Class Question: Which function(s) can be used to sort dates in ascending order in PHP?

Answer: `strtotime()` function, which changes a given date string into timestamp (large integer value) and `usort()` function which sorts the given array as per a user defined comparison function.

Additional Information:

Refer to following link for more information:

https://www.w3schools.com/php/func_array_sort.asp

Slide 24

Sort Array (Ascending Order), According to Value - **asort()**

Code Snippet:

```
<?php
$age = array("Peter"=>"35", "John"=>"37", "David"=>"43");
asort($age);
foreach($age as $x => $x_value) {
    echo "Key=" . $x . ", Value=" . $x_value;
    echo "<br>";
}
?>
```

Output

Key=Peter, Value=35
Key=John, Value=37
Key=David, Value=43

Using `asort()`, Code Snippet demonstrates an example for sorting an associative array in ascending order according to the value.

© Aptech Limited Architecting Web Applications using PHP / Session 5 / 24 of 29

Show slide 24 and tell students that in Code Snippet, an example of sorting an associative array in ascending order by value is shown.

It is an inbuilt function in PHP that is used to sort an array according to values. It sorts in a way that relation between indices and values is maintained. By default, it sorts in ascending order of values. The keys are preserved, i.e. the key-to-value mapping will remain unchanged by the sort operation.

Next, provide an output explanation for the code:

The `foreach` loop is used in the code to display a person's name as a key and their age as a value. The built-in function `asort()`, which sorts an associative array of values in ascending order, is used here.

Additional Information:

Refer to following links for more information:

https://www.w3schools.com/php/func_array_asort.asp

<https://www.php.net/manual/en/function.asort.php>

https://www.tutorialspoint.com/php/php_function_asort.htm

<https://www.javatpoint.com/post/php-array-asort-function>

Slide 25

Sort Array (Ascending Order), According to Key - `ksort()`

Code Snippet:

```
<?php
$age = array("Peter"=>"35", "John"=>"37", "David"=>"43");
ksort($age);
foreach($age as $x => $x_value) {
    echo "Key=" . $x . ", Value=" . $x_value;
    echo "<br>";
}
?>
```

Output

```
Key=David, Value=43
Key=John, Value=37
Key=Peter, Value=35
```

Code Snippet shows an example that sorts an associative array by key in ascending order.

© Aptech Limited Architecting Web Applications using PHP/ Session 5/ 25 of 29

Show slide 25 and demonstrate to students an example of sorting an associative array by key in ascending order is shown in Code Snippet.

This is an inbuilt function in PHP, which is used to sort an array in ascending order according to its key values. It sorts in a way that the relationship between the indices and values is maintained.

Explain the output of the given code: The `foreach` loop is used in the code to display a person's name as a key and their age as a value. The built-in PHP function `ksort()` is being used here to sort an associative array of values. In this case, the key is sorted according to the array's values. The keys David, John, and Peter are alphabetically sorted.

Additional Information:

Refer to following links for more information:

https://www.w3schools.com/php/func_array_ksort.asp
<https://www.php.net/manual/en/function.ksort.php>
<https://www.javatpoint.com/post/php-array-ksort-function>
<https://www.phptutorial.net/php-tutorial/php-ksort/>

Slide 26

Sort Array (Descending Order), According to Value - **arsort ()**

Code Snippet:

```
<?php  
$age = array("Peter"=>"35", "John"=>"37", "David"=>"43");  
arsort($age);  
  
foreach($age as $x => $x_value) {  
    echo "Key=" . $x . ", Value=" . $x_value;  
    echo "<br>";  
}  
?>
```

Output

Key=David, Value=43
Key=John, Value=37
Key=Peter, Value=35

In Code Snippet, **arsort ()** array is used to sort the associative array in descending order, according to the value.

© Aptech Limited Architecting Web Applications Using PHP / Session 5 / 26 of 29

Show slide 26 and tell students that the **arsort ()** array is used in Code Snippet to sort the associative array in descending order by value.

In PHP, this function is used to sort an array according to values. It sorts in a way that relation between indices and values is maintained. By default, it sorts in descending order of values.

Provide an output explanation for the students: The `foreach` loop is used in the code to display a person's name as a key and their age as a value. This is where PHP's built-in function **arsort ()** comes in handy, as it sorts an associative array of values in descending order by key values of an array.

Additional Information:

Refer to following links for more information:

[https://www.w3schools.com/PHP/func_array_arsort.asp#:~:text=The%20arsort\(\)%20function%20sorts,order%2C%20according%20to%20the%20key.](https://www.w3schools.com/PHP/func_array_arsort.asp#:~:text=The%20arsort()%20function%20sorts,order%2C%20according%20to%20the%20key.)

<https://www.php.net/manual/en/function.arsort.php>

<https://www.tutorialspoint.com/arsort-function-in-php>

Slide 27

Sort Array (Descending Order), According to Key - `krsort()`

Code Snippet:

```
<?php  
$age = array("Peter"=>"35", "John"=>"52", "David"=>"43");  
krsort($age);  
foreach($age as $x => $x_value) {  
    echo "Key=" . $x . ", Value=" . $x_value;  
    echo "<br>";  
}  
?>
```

Output

```
Key=Peter, Value=35  
Key=John, Value=52  
Key=David, Value=43
```

Code Snippet shows an example for sorting an associative array in descending order, according to the key.

© Aptech Limited Architecting Web Applications using PHP/ Session 5/ 27 of 29

Show slide 27 and tell students that in Code Snippet, one can see how to sort an associative array in descending order according to the key.

It is an inbuilt function in PHP, which is used to sort an array by key in reverse order according to its index values. It sorts in a way that relation between indices and values is maintained.

Explain the output of the code to the students:

The name is displayed as a key and the age of a person is displayed as a value in using the `foreach` loop. The built-in PHP method `arsort()` is used to sort an associative array of values in reverse alphabetical order according to the array's key.

Additional Information:

Refer to following links for more information:

[https://www.w3schools.com/php/func_array_krsort.asp#:~:text=The%20krsort\(\)%20function%20sorts,order%2C%20according%20to%20the%20value.](https://www.w3schools.com/php/func_array_krsort.asp#:~:text=The%20krsort()%20function%20sorts,order%2C%20according%20to%20the%20value.)

<https://www.php.net/manual/en/function.krsort.php>

<https://www.tutorialrepublic.com/php-reference/php-krsort-function.php>

Slide 28

Sort Array in Descending Order - `rsort()`

Code Snippet:

```
<?php
$Student = array("Peter", "John", "David");
rsort($Student);
$clength = count($Student);
for($x = 0; $x < $clength; $x++) {
    echo $Student[$x];
    echo "<br>";
}
?>
```

Output

Peter
John
David

Code Snippet shows an example for sorting the elements of the `$Student` array in descending alphabetical order.

© Aptech Limited Architecting Web Applications using PHP / Session 5 / 28 of 29

Show slide 28 and inform the students that `rsort()` function is used to sort an array in descending order.

Refer to code snippet to understand sorting the elements of the `$Student` array in descending alphabetical order through an example.

Provide an output explanation:

In this code, the `rsort()` built-in function of the array is used which sorts the names of students in descending alphabetical order.

Ask the students the following question. Wait for a response before you answer.

In-Class Question: Which function will sort an array in reverse order?

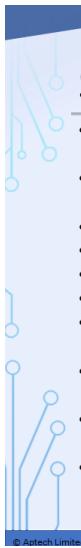
Answer: `reverse()` function

Additional Information:

Refer to following links for more information:

<https://www.php.net/manual/en/function.rsort.php>

<https://stackoverflow.com/questions/2045401/sort-array-in-desc-order>



Summary

- There are three types of conditional statements in PHP, namely `if`, `if...else`, and `if...elseif...else` statements.
- If a test condition can give multiple values, then the `switch` statement is used for testing multiple conditions and explaining different types of conditional statements and using them.
- Loops are used for executing the same block of code multiple times till a condition is satisfied.
- There are three types of loop statements, namely `while`, `do...while`, and `for` loops.
- `foreach` loop statement loops through a block of code for each element in an array.
- `break` statement is used to jump out of a particular code block.
- The `continue` statement skips the iteration in the loop and proceeds to the next iteration when a specified condition is satisfied.
- There are three different types of arrays, namely numeric, associative, and multi-dimensional array.
- The elements in an array can be sorted in alphabetical or numerical order and in descending or ascending manner.
- There are six functions for sorting in PHP namely, `sort()`, `rsort()`, `asort()`, `ksort()`, `arsort()`, and `krsort()`.

© Aptech Limited

Architecting Web Applications using PHP/ Session 5/ 29 of 29

Use slide 29 to summarize the session. You will end the session, with a brief summary of what has been taught in the session. Tell the students pointers of the session. This will be a revision of the current session.

5.3 Post Class Activities for Faculty

You should familiarize yourself with the topics of the next session.

Tips:

You can also check the Articles/Blogs/Expert Videos uploaded on the Online Varsity site to gain additional information related to the topics covered in the next session.

Session 6 – Form Handling

6.1 Pre-Class Activities

Before beginning the session, you should fully familiarize yourself with its topics. Prepare a question or two that will be a key point to relate the current session objectives.

6.1.1 Teaching Skills

To teach this session, you should be well-versed with the concept of internationalization and design patterns in Java. You must also be familiar with the concept of localization.

You must use the given images and teach the concepts in the theory class. For teaching in the class, you are expected to use slides and LCD projectors.

Tips:

It is recommended that you test the understanding of the students by asking questions in between the class.

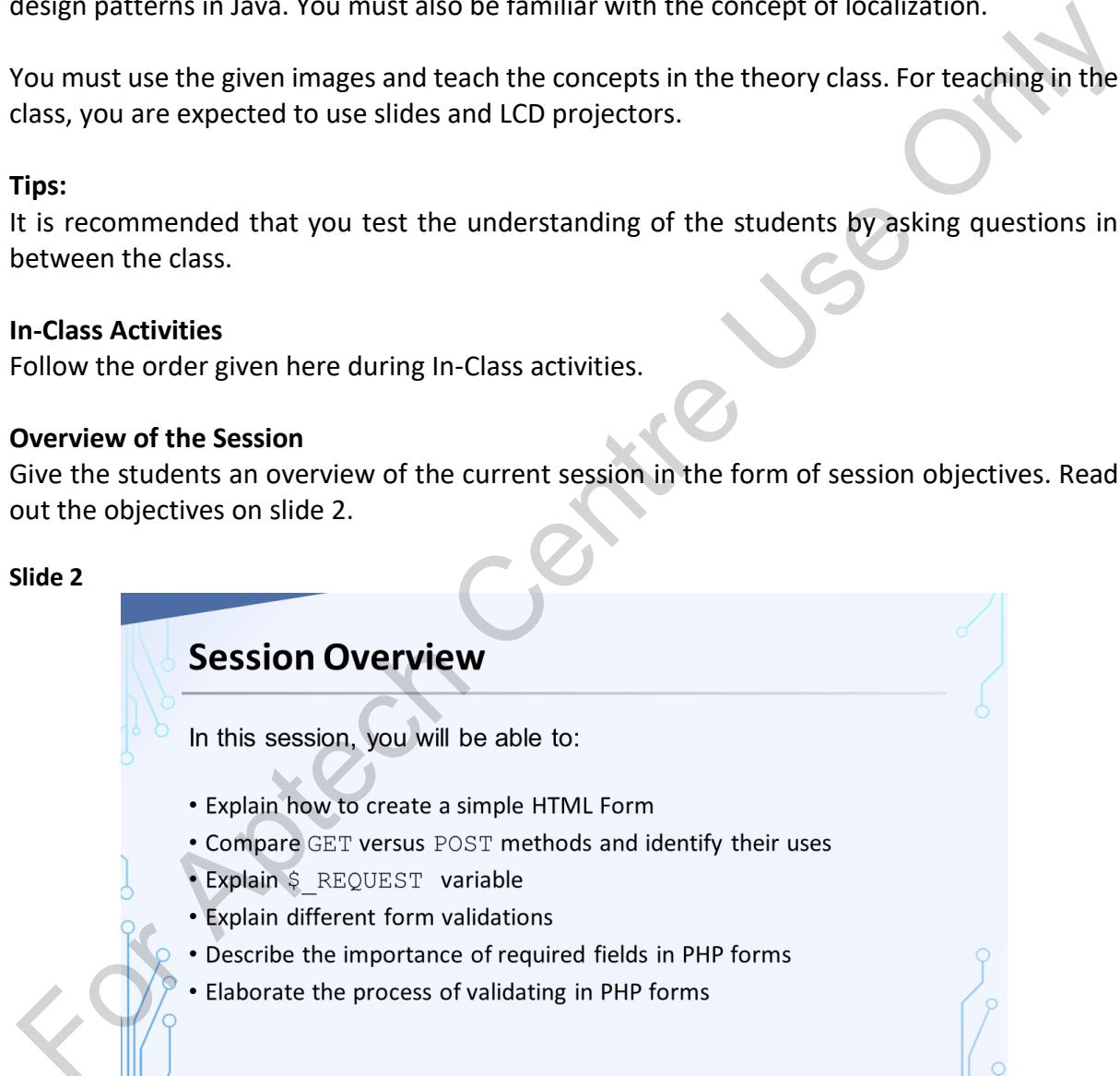
In-Class Activities

Follow the order given here during In-Class activities.

Overview of the Session

Give the students an overview of the current session in the form of session objectives. Read out the objectives on slide 2.

Slide 2



Session Overview

In this session, you will be able to:

- Explain how to create a simple HTML Form
- Compare GET versus POST methods and identify their uses
- Explain `$_REQUEST` variable
- Explain different form validations
- Describe the importance of required fields in PHP forms
- Elaborate the process of validating in PHP forms

© Aptech Limited

Architecting Web Applications using PHP/ Session 6 / 2 of 22

Show slide 2 and give a brief overview of the current session in the form of session objectives. Inform students that this session explains how to develop a simple HTML form, as well as various GET and POST methods and their use in HTML forms. Later on, the session informs regarding form validation for other elements using PHP form security, mandatory fields in a PHP form and error messages, validating names, and passwords.

6.2 In-Class Explanations

Slide 3

Simple HTML Form [1-3]

A form is used to get data from the users and pass it to the Web server for processing. It is created using HTML tags.

Different types of form controls for collecting data include Check boxes, Radio buttons, Submit and Reset buttons, Clickable controls, and so on.

Data can be submitted to the server for processing using the following methods:

PHP POST method → PHP GET method

© Aptech Limited Architecting Web Applications using PHP/ Session 6 / 3 of 22

Show slide 3 and explain to students that a form is used to collect information from users and send it to a Web server for processing. HTML tags are used to create a form. Check boxes, radio buttons, submit and reset buttons, clickable controls, and more are various types of form controls to collect data.

Two techniques can be used to send data to the server for processing – the PHP POST method and the PHP GET method.

PHP POST method: The POST method instructs the Web browser to deliver all of the user's data to the processing agent via the HTTP request's message body. As there is no physical limit to the amount of data that may be provided through the body of an HTTP request, this approach can deliver more information.

PHP GET method: The GET method instructs the Web browser to transmit the encoded user information added to the end of the Uniform Resource Locator (URL), to the processing agent. A question mark is appended to the end of the URL in this approach. The URL and the form information are separated by this question mark. A Name tag and a value are assigned to each input variable in a form by the user.

Ask the students the following question. Wait for a response before you answer.

In-ClassQuestion: What is the key difference between GET and POST methods?

Answer: The main difference between GET and POST methods is that GET displays the submitted data as part of the URL. However, while using POST, this information is not shown.

Additional Information:

Refer to following links for more information:

- https://www.w3schools.com/php/php_forms.asp
- <https://www.php.net/manual/en/tutorial.forms.php>
- <https://www.javatpoint.com/php-form>

Slide 4

The slide features a blue header bar with the title 'Simple HTML Form [2-3]'. Below the header, there are two sections: 'Code Snippet:' and 'OUTPUT:'. The 'Code Snippet:' section contains the following HTML code:

```
<html>
  <body>
    <form action="welcome.php" method="post">
      Name: <input type="text" name="name"><br>
      Password: <input type="password" name="password">
      <br>
      <input type="submit" value="Submit">
    </form>
  </body>
</html>
```

The 'OUTPUT:' section shows a screenshot of a web browser window displaying the form. The URL is 'localhost/php_Session06/codesnippet1.php'. The form has two input fields: 'Name:' and 'Password:', and a submit button labeled 'Submit'.

Figure: Generating a Simple Form

Code Snippet shows a simple HTML form with two input fields and a submit button.

© Aptech Limited

Architecting Web Applications using PHP/ Session 6/ 4 of 22

Show slide 4 and explain to students that PHP has a concept known as superglobal variables. These are built-in variables that are available in all scopes at all times. The superglobal variables `$_GET` and `$_POST` are used to get data from an HTML form that has been submitted using the GET or POST methods.

`$_GET` is internally represented as an array of variables passed to the current PHP script through URL parameters. Meanwhile, `$_POST` is an array of variables passed to the current PHP script using the HTTP POST method.

The syntax for the GET and POST methods is as follows:

```
<?php
$_GET['variable_name'];
?>

<?php
$_POST['variable_name'];
?>
```

Where, the superglobals `$_GET` and `$_POST` are always accessible, regardless of the scope. They are accessible across any function, class, or file.

Explain how Code Snippet demonstrates code for rendering a form with two input fields for a user's name and password. When the user hits the Submit button after filling the form, the code transfers the user to another page, a PHP file named, that shows a welcome message with the username and password of the user.

The HTTP POST method is used to deliver form data to `welcome.php` present on the server, for processing.

Show them Figure depicting the output of Code Snippet.

Slide 5

The slide features a blue header bar with the title 'Simple HTML Form [3-3]'. Below the header, there are two sections, each containing a 'Code Snippet:' heading and a code block. To the right of each snippet is a 'OUTPUT:' section showing a screenshot of a web browser displaying the results.

Code Snippet:

```
Welcome <?php echo $_POST["name"]; ?> <br>
Your password is: <?php echo $_POST["password"]; ?>
```

Code Snippet 2 shows statements to retrieve and display name and password of the user that were submitted through the form.

Code Snippet:

```
<html>
<body>
<form action="codesnippet3.php" method="get">
First name: <input type="text" name="firstname">
<input type="submit" value="Submit">
</form>
Welcome
<?php
// Turn off all error reporting
error_reporting(0);
echo $_GET["firstname"]
?>
</body>
</html>
```

OUTPUT:

localhost/php_Session06/welcome.php
Welcome Robin
Your password is: abc@123

Figure: Displaying Form Data

OUTPUT:

localhost/php_Session06/codesnippet3.php
First name: David [Submit]
Welcome David

Figure: Displaying Form Data with GET

Second Code Snippet demonstrates use of GET to display the name entered by the user with a personalized greeting.

© Aptech Limited

Architecting Web Applications using PHP/ Session 6/ 5 of 22

Show slide 5 and tell students that statements to display and retrieve the user's name and password that were submitted through the form are shown in first Code Snippet. The `echo` function is used to create the output with the submitted data. These statements will be saved as `welcome.php`.

Moreover, describe the output to the students:

When the user submits the form in first Code Snippet with data, the `$_POST` superglobal variable is invoked to recollect the user's name and password submitted in the form.

First Figure illustrates the output for first Code Snippet, which is dependent on data given when first Code Snippet is run.

Furthermore, show second Code Snippet and provide an explanation for its output:

The code for creating input form and generating output is included within a single file called `codesnippet3.php`. After the code has been deployed, the user may fill out the form and submit it. The user is then led to a new version of the same page that uses the GET method to extract data from the textbox and displays it with a customized message.

When PHP finds undefined variables such as `firstname` before the form is submitted, it raises errors in the browser. The `error_reporting(0);` statement, which is shown in second Code Snippet, can be used to turn this OFF.

Show them the output of second Code Snippet using second Figure.

GET versus POST

Both GET and POST methods each create an array (Example: array(key1 => value1, key2 => value2, key3 => value3, ...)).

Uses of GET	Uses of POST
<ul style="list-style-type: none"> GET method displays all the values being sent in the URL itself. Hence, this method is recommended only for sending non-sensitive data. One should not use the GET method to send passwords or other sensitive information. This method also has a limit on the amount of information that can be sent. The limitation is about 2048 characters. However, it is possible to bookmark the page. That means, a browser can remember the page and it will be easy for you to access it next time. Unlike POST method, GET method can be bookmarked because the data required for the request is stored in the URL. 	<ul style="list-style-type: none"> POST method does not display values in the URL because data sent is in the package form, and maintained in a separate communication processor. It has no limit on the amount of information sent because it is submitted via the HTTP's body. Moreover, the POST method supports advanced functionality such as multi-part binary input while uploading files to a server. It also supports other data types such as string, numeric, and so on. However, as variables are not displayed in the URL, it is not possible to bookmark the page.

© Aptech Limited

Architecting Web Applications using PHP/ Session 6/ 6 of 22

Show slide 6 and explain to students that arrays are created by both the GET and POST methods. The array is made up of key and value pairs, where the keys are the names of the form controls and the values are the user's input data. The browser converts each name/value pair into a URL, replacing spaces with the + sign and non-alphanumeric characters with hexadecimal values. After that, the encoded data is transferred to the server.

Uses of GET:

- GET is a method that sends information by appending them to the page request.
- Using the GET method to bookmark data is fairly simple.
- The GET method has a maximum length constraint, which is about 2048 characters.
- This method can only be used to retrieve data from the browser's address bar.
- This approach makes it simple to save data.
- It is possible to bookmark the page using this method.

Uses of POST:

- The request body and query string are used to create the POST method request.
- The query parameters in the browser URL will not show data sent via the POST method.
- POST method parameters are not stored in the browser history.
- The length of data that can be sent is unrestricted.
- It allows users to securely send sensitive and secret information to the server, such as login credentials.
- It is not possible to bookmark the page using this method.

Additional Information:

Refer to following links for more information:

https://www.w3schools.com/tags/ref_httpmethods.asp

https://www.tutorialspoint.com/php/php_get_post.htm

<https://www.javatpoint.com/get-vs-post>

Slide 7

The slide features a blue header with the title '\$_REQUEST Variable'. Below the title is a section labeled 'Code Snippet:' containing the following PHP code:

```
<?php
error_reporting(0);
if( $_REQUEST["name"] || $_REQUEST["weight"] )
{
echo "Welcome ". $_REQUEST['name']. "<br />";
echo "You are ". $_REQUEST['weight']. " kgs .";
exit();
}
?>
<html>
<body>
<form action="<?php $_PHP_SELF ?>"""
method="POST">
Name: <input type="text" name="name" />
Weight: <input type="text" name="weight" />
<input type="submit" />
</form>
</body>
</html>
```

To the right of the code is a screenshot of a web browser window showing the output:

Welcome David Smith
Your weight is 200 pounds.

A green callout box below the browser window states: 'Code Snippet shows the use of \$_REQUEST variable.'

At the bottom right of the slide, it says 'Architecting Web Applications using PHP/ Session 6/ 7 of 22'.

Show slide 7 and tell students that `$_GET`, `$_POST`, and `$_COOKIE` are the values contained in the PHP `$_REQUEST` variable.

The `$_REQUEST` variable in PHP is used to get the result of form data sent via the GET and POST methods.

The `$_REQUEST` variable is used in Code Snippet.

Explain the code output to the students:

The input form and display form are both written in the same program. The user's name and weight are entered as input. When the button is submitted, the `$_PHP_SELF` variable is used to send the submitted form data to the same page. The values of the user's name and weight are returned by `$_REQUEST`.

Figure illustrates the output of Code Snippet after input data has been submitted via the form.

Note: A cookie is a text file created on a user's computer to store user information for tracking that can be used by Web servers. `$_COOKIE` is an array of variables passed to the current script via HTTP cookies.

Additional Information:

Refer to following links for more information:

https://www.w3schools.com/php/php_superglobals_request.asp
<https://www.php.net/manual/en/reserved.variables.request.php>
<https://stackoverflow.com/questions/29195602/request-in-php>

Slide 8

The slide has a decorative background with blue lines and circles. At the top center is the title **Form Fields Validation**. Below it is a box containing the text: "In general, commonly used input fields in PHP form include: Required and optional text fields, Radio buttons, A submit button". To the left of this text is a screenshot of a PHP form titled "PHP Form Validation". The form has several fields: "Name" (text input), "e-mail" (text input), "Website" (text input), "Comment" (text area), and "Gender" (radio buttons for Female, Male, Other). The "Name" field is highlighted in red, indicating it is a required field. Below the screenshot is the caption **Figure: Form Validation**. To the right of the text is a table titled **Table: Form Validation Rules**. The table has three columns: Field, Validation Rules, and Condition. The data is as follows:

Field	Validation Rules	Condition
Name	It must only contain letters and whitespace. The user must provide the input.	Mandatory
e-mail	It must contain a valid e-mail address (with '@' and '.'). The user must provide the input.	Mandatory
Website	If present, it must contain a valid URL.	Optional
Comment	Multi-line input field (textarea). If present user has to write comments in text area given.	Optional
Gender	The user must select one option.	Mandatory

At the bottom right of the slide is the footer: "Architecting Web Applications using PHP/ Session 6/ 8 of 22".

Show slide 8 and tell students that before submitting the data provided by the user in a form to the server/database, form validation is an essential process. The scripts compare the data entered in the fields to the developer-defined validation rules.

Following are some of the most regularly used input fields in PHP forms:

- **Required and optional text fields:** A text input can be entered into text fields in a form. Text input can consist of things such as a person's name, address, occupation, and so on. Short and lengthy text is accepted in text boxes and text fields, respectively.
- **Radio buttons:** Radio buttons are used when a user is presented with numerous options and must pick one.
- **A submit button:** Last but not least is the Submit button. The value of its value element is displayed as the button's text. Clicking on it submits the form.

Show them Figure that illustrates a validation-enabled sample form with necessary fields highlighted in red.

Then, describe the contents of Table illustrating some of the validation criteria that can be applied for a form.

Ask the students the following question. Wait for a response before you answer.

In-ClassQuestion: What will happen if a user tries to submit a form without entering the text area or required field?

Answer: An error message will be displayed and the form will not get submitted.

Slide 9

Text Fields

Code Snippet:

```
<form action="login.php" method="get">
UserID: <input type="text" name="userID"><br><br>
Password: <input
type="password" name="password"><br><br>
Company: <input type="text" name="company"><br><br>
Mobile Number: <input type="text" name="Mobile
Number"><br><br>
Comments: <textarea name="comments" rows="6" cols="50">
</textarea>
</form>
```

Code Snippet shows how to create a simple form that utilizes text fields for accepting Username, Password, Company, and Comment.

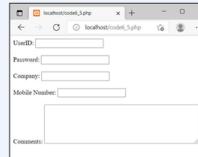


Figure: Using Text Fields in a Form

Architecting Web Applications using PHP/ Session 6/ 9 of 22

Show slide 9 and describe to students that a text input can be entered into text fields in a form. Text input can consist of things such as a person's name, address, occupation, and so on. Short and lengthy text is accepted in text boxes and text fields, respectively.

By referring to Code Snippet, one can learn how to build a simple form with text fields that take Username, Password, Company, and Comment.

Explain the Output of the code to students:

A simple form is generated in Code Snippet where users may input their name, password, company, and comments. The user fills in his or her information in the text boxes provided for each field. As comments might include product information or feedback, the comment field has larger a text field than the other text boxes in the available fields.

The form with text fields is shown in Figure.

Ask the students the following question. Wait for a response before you answer.

In-ClassQuestion: What is the syntax for the GET method?

Answer: The syntax for the GET method is: `$varname = $_GET['variable']`

Slide 10

Radio Buttons

Code Snippet:

```
<form action="form1.php" method="get">
<input type="radio" name="gender" value="female">Female
<input type="radio" name="gender" value="male">Male
<input type="radio" name="gender" value="other">Other
</form>
```

Code Snippet creates gender fields in a form to be selected using radio buttons.

Figure: Using Radio Buttons in a Form

localhost/php_Session06/codesnippet6.php

© Aptech Limited

Architecting Web Applications using PHP/ Session 6/ 10 of 22

Show slide 10 and tell students that radio buttons are used when a user is presented with numerous options and must pick one.

Radio buttons are used when there is a list of two or more options that are mutually exclusive and the user must select exactly one choice. In other words, clicking a non-selected radio button will deselect whatever other button was previously selected in the list. A radio button has two states: checked and unchecked. When users link a label with a radio button, they can check the radio button by clicking the label or the radio button itself. Hence, the label increases the usability of the radio button because it expands the selection area.

Explain the output of Code Snippet in detail. The code prompts the user to pick the right gender using radio buttons. The user may choose between male, female, and other options. Figure depicts a form with radio buttons.

In a real-world scenario, the code in snippets in slides 9 and 10 may be coupled with other statements to make a full and proper form.

Ask the students the following question. Wait for a response before you answer.

Additional Information:

Refer to following links for more information:

<https://www.phptutorial.net/php-tutorial/php-radio-button/>
<https://www.homeandlearn.co.uk/php/php4p10.html>

Slide 11

Using htmlspecialchars() with Form Elements

Form elements are the elements which are used inside the `form` tag.

In a form, it is recommended that methods such as GET or POST, variables such as `$_SERVER["PHP_SELF"]`, and `htmlspecialchars()` function be used.

© Aptech Limited

Architecting Web Applications using PHP/ Session 6/ 11 of 22

Show slide 11 and tell the students that the elements used within the form tag are known as form elements. Methods such as GET and POST, variables such as `$_SERVER["PHP_SELF"]`, and the `htmlspecialchars()` function are all suggested for implementation in forms.

Explain syntax for form elements to the students:

```
<form method="post" action="<?
php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>">
```

where,

`htmlspecialchars()` is a function used to convert special characters into HTML entities. It replaces HTML characters such as < and > with < and >. This prevents attackers from manipulating code by inserting HTML or JavaScript code (Cross-site Scripting attacks) in forms.

The form data is delivered to the server via the POST method when it is submitted.

In-ClassQuestion: How to enter text in PHP Forms?

Answer: `input type= "text"`

Slide 12

Validating Form Data with PHP [1-5]

- As a first step in validation, one should pass all variables through the PHP's `htmlspecialchars()` function to validate form data.
- Consider a scenario where a malicious user tries to submit a hyperlink within a text field that is meant to receive user names or some such text. When the hyperlink is posted to the redirected page, which can lead to harmful action.
- When `htmlspecialchars()` function is used, if a user submits data such as following in a text field, it will not be executed as a hyperlink: `<script> location.href('http://www.attack.com') </script>`.
- This is because the `htmlspecialchars()` function caused the data to be saved internally as HTML escaped code: `<script&gtlocation.href('http://www.attack.com') </script&gt`. Thus, this code can now be displayed safely on a page or within an e-mail because it is treated as text instead of a hyperlink. On the other hand, if `htmlspecialchars()` function had not been used with the text field, the outcome would have been unsafe and potentially dangerous.

© Aptech Limited

Architecting Web Applications using PHP/ Session 6/ 12 of 22

Show slide 12 and tell the students that in order to validate form data, the first step in validation is to pass all variables via PHP's `htmlspecialchars()` function.

Consider the case when a malicious user tries to submit a URL into a text box intended to collect the user's name or other content. When the hyperlink is posted to a redirected Website, it might lead to a potentially hazardous activity.

If a user enters data like this in a text field with the `htmlspecialchars()` method, it will not be executed as a hyperlink:

```
<script> location.href('http://www.attack.com') </script>
```

This is because the `htmlspecialchars()` function stored the data as HTML escaped code internally:

```
&lt;script&ampgtlocation.href('http://www.attack.com') &lt;/script&ampgt;
```

As a result of being recognized as text rather than a hyperlink, this code can now be safely displayed on a page or within an e-mail. The result would have been hazardous and potentially harmful if the `htmlspecialchars()` method had not been used with the text field.

When the user submits the form, two further actions can be taken:

- Remove unwanted characters (extra spaces, tabs, and newlines) from the user input data using the PHP `trim()` function.
- Remove backslashes (\) using the PHP `stripslashes()` function, from user-supplied data.

The next step is to create a function that will perform the data sanitizing process, which is handier than repeatedly writing the same code.

Slides 13 and 14

Validating Form Data with PHP [2-5]

Code Snippet:

```
<!DOCTYPE html>
<head>
</head>
<body>
<?php
// Initialize variables during creation to empty values
$name = $email = $gender = $comment = $website = "";
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $name = sanitize_data($_POST["name"]);
    $password = sanitize_data($_POST["password"]);
    $company = sanitize_data($_POST["company"]);
    $comment = sanitize_data($_POST["comment"]);
    $gender = sanitize_data($_POST["gender"]);
    echo "Data is sanitized in appropriate format <br>";
}

function sanitize_data($data) {
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data, ENT_QUOTES);
    return $data;
}
?>
<form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>">
Name: <input type="text" name="name">
<br><br>
```

© Aptech Limited

Code Snippet shows a program to check each `$_POST` variable with a `sanitize_data()` function.

Architecting Web Applications using PHP/ Session 6/ 13 of 22

Validating Form Data with PHP [3-5]

```
Password: <input type="password" name="password">
<br><br>
Company: <input type="text" name="company">
<br><br>
Comment: <textarea name="comment" rows="5" cols="40"></textarea>
<br><br>
Gender:
<input type="radio" name="gender" value="female">Female
<input type="radio" name="gender" value="male">Male
<input type="radio" name="gender" value="other">Other
<br><br>
<input type="submit" name="submit" value="Submit">
</form>
<?php
if (empty($_POST)) {
    echo "Name";
    echo "<br>";
    echo $password;
    echo "<br>";
    echo $company;
    echo "<br>";
    echo $comment;
    echo "<br>";
    echo $gender;
}
?>
</body>
</html>
```

© Aptech Limited

Architecting Web Applications using PHP/ Session 6/ 14 of 22

Using slides 13 and 14, tell the students that Code Snippet demonstrates a program that uses the `sanitize_data()` function to validate each `$_POST` variable.

Slide 15

The figure consists of three screenshots illustrating form validation in PHP:

- Figure: Using a Function to Validate Data in a Form**: Shows a browser window with a form containing fields for Name, Password, Company, and Comment. The Company field contains the value "AlkalineLimitedInc". The Comment field contains the value "Chemicals & Pharmaceuticals" are our lifeline.
- Figure: Output After Clicking Submit**: Shows the browser output after clicking the Submit button. The Company field now contains "Data is sanitized in appropriate format". The Comment field contains "Data is sanitized in appropriate format".
- Figure: Page Source Showing Encoded Data**: Shows the page source code. It includes the original input values and the sanitized output. The company value is shown as "AlkalineLimitedInc" and the comment value is shown as "Chemicals& Pharmaceuticals" are our lifeline.

Show slide 15 and tell the students that first Figure shows the form with data before clicking Submit. Special characters, such as back slashes and quotes, are used in the company and comment data.

Second Figure displays the output after clicking the Submit button. It can be seen that the output still includes single and double quotes in the data. This is because the browser's internal engine decodes the output before displaying it, even after `htmlspecialchars` has finished the encoding. The data that is submitted to the server, on the other hand, will be transmitted in its original encoded form. This can be seen in third Figure, which shows the .php file's Page source view.

Since this form is without any mandatory fields currently, there will be no issues even if the user submits empty data or blank fields. Additional validation actions are required to overcome this.

Validating Form Data with PHP [5-5]

Code Snippet:

```
<?php
// define variables and set to empty values
$nameErr = $pwdErr = $genderErr = $companyErr = "";
$name = $password = $gender = $comment = $company = "";

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (empty($_POST["name"])) {
        $nameErr = "Name is required";
    } else {
        $name = sanitize_data($_POST["name"]);
    }

    if (empty($_POST["password"])) {
        $pwdErr = "Password is required";
    } else {
        $password = sanitize_data($_POST["password"]);
    }

    if (empty($_POST["comment"])) {
        $comment = "";
    } else {
        $comment = sanitize_data($_POST["comment"]);
    }

    if (empty($_POST["gender"])) {
        $genderErr = "Gender is required";
    } else {
        $gender = sanitize_data($_POST["gender"]);
    }
}
?>
```

Code Snippet shows how error variables store values.

© Aptech Limited Architecting Web Applications Using PHP / Session 6 / 16 of 22

Show slide 16 and tell the students that the fields which are compulsory for users to fill out can be found in forms. These fields are referred to as required fields collectively. As illustrated in Code Snippet, new variables `$nameErr`, `$pwdErr`, `$genderErr`, and `$companyErr` are introduced to the previous code. The error warnings for required fields in the slide 13's Code Snippet form are stored in these variables. For each `$_POST` variable, `if` `else` statements are added, which verify if the `$_POST` variable is empty, by using PHP `empty()` function. If the `$_POST` variable is empty, an error message is recorded in various error variables; if the variable is not empty, the user input data is sent to the `sanitise_data()` function.

Slides 17 and 18

Display Error Messages [1-3]

Code Snippet:

```
<html>
<head>
<style>
.error {color: #FF0000;}
</style>
</head>
<body>
<?php
error_reporting(0);
// Initialize variables during creation to empty values
$nameErr = $pwdErr = $genderErr = $companyErr = "";
$name = $pwd = $gender = $comment = $company = "";
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (empty($_POST["name"])) {
        $nameErr = "Name is required";
    } else {
        $name = sanitize_data($_POST["name"]);
    }
    if (empty($_POST["password"])) {
        $pwdErr = "password is required";
    }
}
else {
    $password = sanitize_data($_POST["password"]);
}
if (empty($_POST["company"])) {
    $company = "";
}
else {
    $company = sanitize_data($_POST["company"]);
}
if (empty($_POST["comment"])) {
    $comment = "";
}
else {
    $comment = sanitize_data($_POST["comment"]);
}
if (empty($_POST["gender"])) {
    $genderErr = "Gender is required";
}
else {
    $gender = sanitize_data($_POST["gender"]);
}
}
function sanitize_data($data) {
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}
}

```

Code Snippet shows a program to display an error message when the user submits the form without filling the required fields.

© Aptech Limited

Architecting Web Applications using PHP/ Session 6/ 17 of 22

Display Error Messages [2-3]

```
?>
<p><span class="error">* indicates required field</span></p>
<form method="post" action="?>
echo htmlspecialchars($_SERVER["PHP_SELF"]);?>">
Name: <input type="text" name="name">
<span class="error">*</span> <?php echo $nameErr;?></span>
<br><br>
Password: <input type="password" name="password">
<span class="error">*</span> <?php echo $pwdErr;?></span>
<br><br>
Company: <input type="text" name="company">
<span class="error">*</span> <?php echo $companyErr;?></span>
<br><br>
Comment: <textarea name="comment" rows="5" cols="40"></textarea>
<br><br>
Gender:
<input type="radio" name="gender" value="female">Female
<input type="radio" name="gender" value="male">Male
<input type="radio" name="gender" value="other">other
<span class="error">*</span> <?php echo $genderErr;?></span>
<br><br>
<input type="submit" name="submit" value="Submit">

```

```
</form>
<?php
echo $name;
echo "<br>";
echo $password;
echo "<br>";
echo $company;
echo "<br>";
echo $comment;
echo "<br>";
echo $gender;
?>
</body>
</html>
```

© Aptech Limited

Architecting Web Applications using PHP/ Session 6/ 18 of 22

Use slides 17 and 18 to tell the students that depending on the necessity, extra code is added after each mandatory field in the form to generate the appropriate error message. When a user tries to submit a form without filling out all of the required fields, an error message appears.

Show Code Snippet to the students for better understanding.

Slide 19

The image displays two side-by-side screenshots of a web browser window, both titled "localhost/code6_9.php". The left screenshot shows a form with four fields: Name, Password, Company, and Comment. Below the fields is a gender selection section with radio buttons for Female, Male, and Other, followed by a required field indicator (*). The right screenshot shows the same form after a submission attempt. The Name and Password fields now have error messages: "Name: * Name is required" and "Password: * password is required". The gender section also has an error message: "Gender: * Gender is required". Both screenshots include a "Submit" button at the bottom.

Figure: Page Source Showing Encoded Data

Figure: Page Source Showing Encoded Data

Show slide 19 and tell the students that both Figures demonstrate the output of Code Snippets in slides 17 and 18, before and after clicking Submit button respectively. As it can be seen, empty fields will not be permitted.



Validating Against Special Characters [1-2]

In a text field such as name, user can only enter letters and whitespaces. Numbers and special characters are not allowed in the name field.

Following statements check the name field for letters, dashes, apostrophes, and whitespaces:

```
$name = sanitize_data($_POST["name"]);  
if (!preg_match("/^[\w- ]*$/", $name)) {  
    $nameErr = "Only letters and white space allowed";  
}
```

An error message is displayed if the value entered for the name field is not valid. The `preg_match()` function is used here to search a string for a pattern. It returns `true` if the pattern exists and `false` if the pattern does not exist.

Aptech Limited

Architecting Web Applications Using PHP/ Session 6/ 20 of 22

Show slide 21 and tell the students that the user can only enter letters and whitespaces in a text field such as name. In the name field, no numbers or special characters are permitted.

Following statements check for letters, dashes, apostrophes, and whitespaces in the name field:

```
$name = sanitize_data($_POST["name"]);  
if (!preg_match("/^[\w- ]*$/", $name)) {  
    $nameErr = "Only letters and white space allowed";  
}
```

If the value provided for the name field is invalid, an error message appears. The `preg_match()` function is used to find a pattern in a string. If the pattern exists, it returns true; otherwise, it returns false.

The user's password must be at least eight characters long, with one uppercase character, one lowercase character, one number, and one special character, according to the password validation criteria.

If all of the requirements or conditions are met, the user password is considered strong; otherwise, a message stating that the password is weak appears. A message will appear on the form, requesting the user to provide a password of at least eight characters, including one uppercase, one lowercase, one number, and one special character.

Code Snippet:

```

$password = $_POST['password'];
if( strlen($password) < 8 ) {
    $pwdErr .= " Password too short. ";
}
if( strlen($password) > 20 ) {
    $pwdErr .= " Password too long! ";
}
if( !preg_match("#[0-9]#", $password) ) {
    $pwdErr .= " Password must include at least one number. ";
}
if( !preg_match("#[a-z]+#", $password) ) {
    $pwdErr .= " Password must include at least one letter. ";
}
if( !preg_match("#[A-Z]+#", $password) ) {
    $pwdErr .= " Password must include at least one uppercase letter. ";
}
if( !preg_match("#[^\\w]#", $password) ) {
    $pwdErr .= " Password must include at least one symbol. ";
}
if( !empty($pwdErr) ) {
    if( preg_match("#^.^(?=.{8,20})(?=.*[a-z])(?=.*[A-Z])(?=.*[0-9])(?=.*[^\\w]).*$#", $password) ){
        echo "Your password is strong.";
    } else {
        echo "Your password is not safe.";
    }
}

```

Code Snippet shows how to check if a password entered by the user is valid or not. An error message is displayed if the password entered is not valid.

Architecting Web Applications Using PHP/ Session 6/ 21 of 22

Show slide 21 and tell the students that Code Snippet demonstrates how to determine whether a password supplied by the user is valid or not. If the password provided is invalid, an error message appears.

The password provided by the user is validated in this code. `if` condition of the code checks the length of the password. If it is less than eight characters, it will generate an error message stating that the password is too short.

The next `if` condition determines if the password input is more than 20 characters. If this is the case, an error message stating that the password is too long will be displayed. The code's remaining `if` conditions verify that the password has at least one number, one lowercase, one uppercase, and at least one special character. If all of these requirements are met in the `preg_match` function, the code displays '**Your password is strong.**'

To validate a form password, this code can be embedded in Code Snippet in slides 17 and 18.



Summary

- Forms can be created using HTML tags and input elements to accept user data and send to a Web server for processing.
- GET and POST methods in PHP can be used in forms to send form data.
- \$_GET and \$_POST are superglobal variables and are always accessible, regardless of function, class, or file scope.
- The PHP \$_REQUEST variable contains values of \$_GET, \$_POST, and \$_COOKIE and is used to get result from form data sent with the GET and POST methods.
- The PHP form input fields have validation rules.
- PHP's htmlspecialchars() function can help to encode special characters into HTML entities and can be used to sanitize form data.
- An error message can be generated at the form level or field level through code if user tries to submit a form without filling mandatory fields.
- The preg_match() function searches a string for a pattern, returning true if the pattern exists and false otherwise. It is used to validate the name entered.

Architecting Web Applications Using PHP/ Session 6/ 22 of 22

Use slide 22 to summarize the session. You will end the session with a summary of what has been taught in the session. State the most important pointers of the session. This will be a revision for the students.

6.3 Post Class Activities for Faculty

You should familiarize yourself with the topics of the next session.

Tips:

You can also check the Articles/Blogs/Expert Videos uploaded on the Online Varsity site to gain additional information related to the topics covered in the next session.

Session 7 – Working with Functions in PHP

7.1 Pre-Class Activities

Before beginning the session, you should fully familiarize yourself with its topics. Prepare a question or two that will be a key point to relate the current session objectives.

7.1.1 Teaching Skills

To teach this session, you should be well-versed with the concept of internationalization and design patterns in Java. You must also be familiar with the concept of localization.

You must use the given images and teach the concepts in the theory class. For teaching in the class, you are expected to use slides and LCD projectors.

Tips:

It is recommended that you test the understanding of the students by asking questions in between the class.

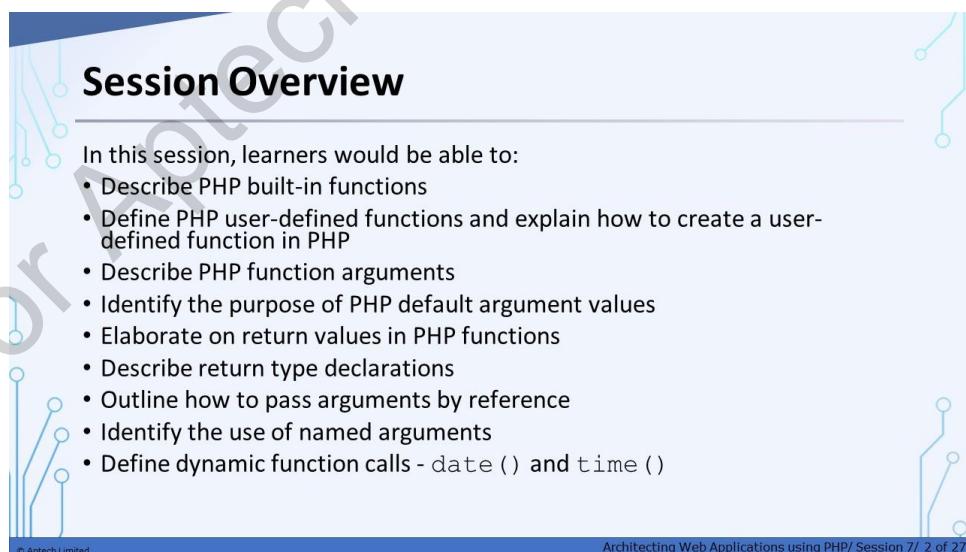
In-Class Activities

Follow the order given here during In-Class activities.

Overview of the Session

Give the students an overview of the current session in the form of session objectives. Read out the objectives on slide 2.

Slide 2



The slide has a blue header bar with the title "Session Overview". Below the title is a horizontal line. To the left of the text area are decorative blue circuit board patterns. At the bottom left is the copyright notice "© Aptech Limited" and at the bottom right is the page information "Architecting Web Applications using PHP / Session 7 / 2 of 27".

In this session, learners would be able to:

- Describe PHP built-in functions
- Define PHP user-defined functions and explain how to create a user-defined function in PHP
- Describe PHP function arguments
- Identify the purpose of PHP default argument values
- Elaborate on return values in PHP functions
- Describe return type declarations
- Outline how to pass arguments by reference
- Identify the use of named arguments
- Define dynamic function calls - date () and time ()

Show slide 2 and give the students a brief overview of the current session in the form of session objectives. Inform students that the session begins with an overview of PHP's built-in functions. In addition, the session gives an overview of user-defined functions and how to

construct them. PHP function arguments, default argument value, returning values, and return type declarations are also covered in this session. It explains named arguments, a new feature introduced in PHP 8, and describes how to pass arguments by reference. The session also covers the dynamic function calls - date() and time().

7.2 In-Class Explanations

Slide 3

The slide has a blue header bar with the title 'Functions in PHP'. Below the title are three colored boxes: blue, green, and green. The blue box contains the text 'Reusable code'. The green boxes contain the text 'Minimal code' and 'Clarity of code'. To the right of each green box is a bulleted list of benefits:

- PHP functions are reusable and can be invoked many times within the same program.
- Using functions gives the flexibility to write code once and reuse the same whenever required. This reduces the amount of written code and also reduces time taken to write code.
- PHP functions segregate the programming logic. This allows the user to comprehend application flow as code is split into different functions.

At the bottom left is the copyright notice '© Aptech Limited' and at the bottom right is the slide footer 'Architecting Web Applications using PHP/ Session 7/ 3 of 27'.

Show slide 3 and inform the students about PHP functions. A PHP function is a self-contained program segment that carries out a specific, well-defined task and can be invoked by users, multiple times as per their requirement.

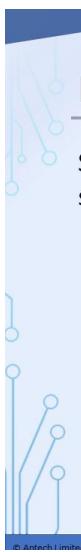
Explain students the benefits of using PHP functions: reusable code, minimal code, and clarity of code. Refer to the slide to describe these.

PHP functions can be categorized as built-in functions and user-defined functions.

Additional Information:

Refer to following links for more information:

<https://zetcode.com/lang/php/functions/>
<https://www.javatpoint.com/php-functions>



PHP Built-in Functions

Some of the important PHP functions which will be covered in the session are as follows:

- abs()
- gettype()
- var_dump()

Aptech Limited Architecting Web Applications using PHP/ Session 7/ 4 of 27

Show slide 4 and inform the students that built-in functions are readymade functions provided by PHP. Some are core functions that are automatically available whereas, some require additional extensions in order to use them.

PHP has over 1000 built-in functions that help users to complete common tasks. This makes it easy for users to execute and rerun common tasks in a program.

Three of the PHP built-in functions are as follows:

- abs()
- gettype()
- var_dump()

Additional Information:

Refer to following links for more information:

https://www.w3schools.com/php/php_functions.asp

<https://www.javatpoint.com/php-functions>

Ask the students following question. Wait for a response before you answer.

In-Class Question: What is the difference between `settype()` and `gettype()` functions in PHP?

Answer: The `gettype()` function gets the type of variable; `gettype()` is a function that displays a data type. The `settype()` function sets the type of variable; the `settype()` function changes the data type.

Slide 5

The screenshot shows a slide titled "abs () Function". It features a "Code Snippet" box containing the following PHP code:

```
<?php  
echo abs(-5.4);  
?>
```

Below the code is a browser window displaying the output: "5.4". The browser title bar says "localhost/code7_1.php". A watermark "Aptech Centre for Reuse Only" is diagonally across the slide.

Figure: Output for Code Snippet

Architecting Web Applications using PHP/ Session 7 / 5 of 27

Show slide 5 and tell the students that Code Snippet shows how to use a built-in function, `abs()`.

The built-in function `abs()` is used to calculate absolute (positive) value of any number passed to the function. Here, in the code, the value -5.4 is passed to the function which then, returns 5.4 as the outcome. `echo` is used to display the result.

Figure shows the output of Code Snippet.

Additional Information:

Refer to following links for more information:

[https://www.geeksforgeeks.org/php-abs-function/#:~:text=The%20abs\(\)%20function%20is,a%20negative%20number%20is%20positive.](https://www.geeksforgeeks.org/php-abs-function/#:~:text=The%20abs()%20function%20is,a%20negative%20number%20is%20positive.)
https://www.w3schools.com/php/func_math_abs.asp
<https://www.tutorialspoint.com/php-abs-function>
<https://www.javatpoint.com/php-math-abs-function>

Slide 6

The slide features a decorative background with blue and white circuit board patterns. At the top center, the title 'gettextype () Function' is displayed in a large, bold, sans-serif font. Below the title, the subtitle 'Code Snippet:' is shown in a smaller, regular font. A code block follows, containing PHP code that demonstrates the use of the `gettextype()` function. The code defines three variables (\$var1, \$var2, \$var3) with integer, double, and string values respectively, and then prints their types using `gettextype()`. Below the code block is a screenshot of a web browser window titled 'localhost/code7_2.php'. The browser displays the output of the code, which consists of three lines of text: 'integer', 'double', and 'string', each preceded by an echo statement. At the bottom right of the slide, there is a small footer note: 'Architecting Web Applications using PHP/ Session 7 / 6 of 27'.

Show slide 6 and tell the students that a user can input a variable as an argument to this function and the output is a string value denoting the data type of the argument. Code Snippet shows an example for `gettextype ()` function.

Here, note that the definition for the `gettextype ()` function is not written by the user, instead it is built into PHP. As the function is already built in, users can just call it anytime within their PHP code, any number of times.

Code Snippet retrieves and displays the type of data assigned implicitly to each variable namely, `var1`, `var2`, and `var3`. Based on the data assigned, `var1` is an integer, `var2` is a double, and `var3` is a string. Therefore, the output displayed is: Integer, Double, and String.

Figure shows the output of Code Snippet.

Additional Information:

Refer to following links for more information:

[https://www.geeksforgeeks.org/php-gettype-function/#:~:text=The%20gettextype\(\)%20function%20is,the%20type%20of%20existing%20variable.&text=Parameter%3A%20This%20function%20accepts%20a,checked%20for%20type%20of%20variable](https://www.geeksforgeeks.org/php-gettype-function/#:~:text=The%20gettextype()%20function%20is,the%20type%20of%20existing%20variable.&text=Parameter%3A%20This%20function%20accepts%20a,checked%20for%20type%20of%20variable)

https://www.w3schools.com/php/func_var_gettype.asp

<https://www.php.net/manual/en/function.gettype.php>

<https://www.w3resource.com/php/function-reference/gettype.php>

Slide 7

The slide features a blue header with the title 'var_dump() Function'. Below the title is a section labeled 'Code Snippet:' containing the following PHP code:

```
<?php  
$var1=13;  
$var2="Hello";  
var_dump($var1);  
echo "<br>";  
var_dump($var2);  
?>
```

Below the code snippet is a screenshot of a web browser window displaying the output of the code. The browser title bar says 'localhost/code7_3.php'. The page content shows:

```
int(13)  
string(5) "Hello"
```

At the bottom of the slide, there is a footer with the text '© Aptech Limited' and 'Architecting Web Applications using PHP/ Session 7 / 7 of 27'.

Show slide 7 and explain that this function also accepts a variable name as an argument and output is the details about the argument passed to it. Code Snippet shows an example for `var_dump()` function.

In the code, the statement `var_dump($var1);` displays details about the variable `$var1`. First, the type and the variable are displayed `int (13)`. Next, `var_dump($var2);` displays details about the variable `$var2`. The type and the length are displayed along with the string value stored in `var2`. Figure depicts the output.

Ask the students following question. Wait for a response before you answer.

In-Class Question: What is the difference between `gettype()` and `var_dump()` functions?

Answer: While the function `gettype()` returns the type of the variable, the function gives complete details about the variable.

Additional Information:

Refer to following links for more information:

[https://www.javatpoint.com/php-var_dump-function#:~:text=The%20var_dump\(\)%20function%20is,string%20passed%20inside%20the%20function.](https://www.javatpoint.com/php-var_dump-function#:~:text=The%20var_dump()%20function%20is,string%20passed%20inside%20the%20function.)

https://www.w3schools.com/php/func_var_var_dump.asp

https://www.geeksforgeeks.org/php-var_dump-function/

Slide 8

The slide features a decorative background with blue and white circuit board patterns on the left and right sides. The title 'PHP Array Functions' is at the top. Below it, a 'Code Snippet:' label is followed by a code block:

```
<?php  
$names=array("David","Charlie","George","Peter");  
echo "Names are: $names[0], $names[1], $names[2] and $names[3]";  
?>
```

Below the code is a screenshot of a web browser window titled 'localhost/code7_4.php'. The page content displays the output: 'Names are: David, Charlie, George and Peter'. A caption below the screenshot reads 'Figure: Output for Code Snippet'.

Text on the slide states: 'There are three types of arrays that can be created in PHP: Indexed Arrays, Associative Arrays, Multidimensional Arrays'.

At the bottom right, there is a small footer: 'Aptech Limited' and 'Architecting Web Applications using PHP/ Session 7/ 8 of 27'.

Show slide 8 and explain that in PHP, arrays are a part of built-in functions. An array can be generated using `array()`. This function allows users to work with arrays in multiple ways. Arrays help users to store, manage, and operate on sets of variables.

Explain the syntax for arrays:

Syntax

```
array();  
Or  
array("value 1", "value 2", ...);
```

Code Snippet shows how to use the array function.

This code displays the names of a person using the array function. This function stores the values according to different memory positions. That is, the first value is stored in the zero place, the second value in the first place, and so on. The variable names is assigned to the function array and the values of the array are printed utilizing echo. For example `$names[0]=David`, `$names[1]=Charlie`, and so on. Figure depicts the corresponding output.

PHP supports both simple arrays and multidimensional arrays. They can be created either by the users or by another function.

Following are the three types of arrays that can be created:

- Indexed Arrays
- Associative Arrays
- Multidimensional Arrays

Additional Information:

Refer to following links for more information:

https://www.w3schools.com/php/php_ref_array.asp

<https://www.javatpoint.com/php-array-functions>

https://www.tutorialspoint.com/php/php_array_functions.htm

Slide 9

Function	Description
<code>array_change_key_case(array \$array, int \$case)</code>	This function switches all the keys in an array to lowercase or uppercase.
<code>array_chunk(array \$array, int \$length, bool \$preserve_keys)</code>	This function breaks an array into smaller chunks of arrays.
<code>array_column(array \$array, int string null \$column_key, int string null \$index_key)</code>	This function displays values from the column mentioned.
<code>array_combine(array \$keys, array \$values)</code>	This function generates an array utilizing elements from keys array and values array.
<code>array_count_values(array \$array)</code>	This function counts all the values of an array.
<code>array_diff(array \$array, array ...\$arrays)</code>	This function displays differences after comparing the arrays. It compares only values.
<code>array_diff_assoc(array \$keys, array \$values)</code>	This function also displays differences, but compares both the keys and values.
<code>array_diff_key(array \$array, array ...\$arrays)</code>	This function also displays differences, but compares only the keys.
<code>array_fill(int \$start_index, int \$count, mixed \$value)</code>	This function helps in inserting values to an array.
<code>array_filter(array \$array, ?callable \$callback, int \$mode)</code>	This function helps in filtering the array values utilizing a callback function.
<code>array_flip(array \$array)</code>	This function interchanges all the keys with their corresponding values in an array.
<code>array_intersect(array \$array, array ...\$arrays)</code>	This function compares the values in arrays and displays the matches.
<code>array_map(?callable \$callback, array \$array, array ...\$arrays)</code>	This function transfers each value of an array to a user-made function, which in turn displays new values.
<code>array_key_exists(string int \$key, array \$array)</code>	This function checks the array for the mentioned key.
<code>array_keys(array \$array)</code>	This function displays all the keys of an array.
<code>array_merge(array ...\$arrays)</code>	This function combines one or more arrays into a single array.

Table: Array Functions

Architecting Web Applications using PHP/ Session 7 / 9 of 27

Show slide 9 and inform the students that the named variables in an array for which users can assign values are called keys. For example, `$age = array ("Sam"=>"36", "John"=>"37", "Peter"=>"43")`; In the example, an array is assigned to the variable age and the array has number of other variables. Sam, John, and Peter are the keys and 36, 37, and 43 are the values, respectively.

Table shows some of the functions that can be used with arrays. They are part of PHP's core functions.

PHP string Functions [1-3]

PHP string functions help the user to perform different operations on strings. They are as follows:

- strrev() function
- strtolower() function
- str_repeat() function

© Aptech Limited Architecting Web Applications using PHP / Session 7 / 10 of 27

Show slide 10 and inform the students that the string functions are also a core part of PHP. They help the user to perform different operations on strings. To perform common string-related tasks, user can utilize various built-in string functions saved in the PHP installation package. These functions can be utilized without having to install anything new.

Following are the three PHP string functions:

- strrev() function
- strtolower() function
- str_repeat() function

Additional Information:

Refer to following links for more information:

https://www.w3schools.com/PHP/php_ref_string.asp

<https://www.javatpoint.com/php-string-functions>

Slide 11

Show slide 11 and explain the following:

strrev() function:

This function takes a string as an input and reverses the order of the original string. Mention the example given in the slide.

Users do not require to save values displayed by the functions into variables. Instead, they can use such values directly. In the example, echo is employed to display the value obtained by executing the `strrev()` function, for which "Hello, World!" is the argument.

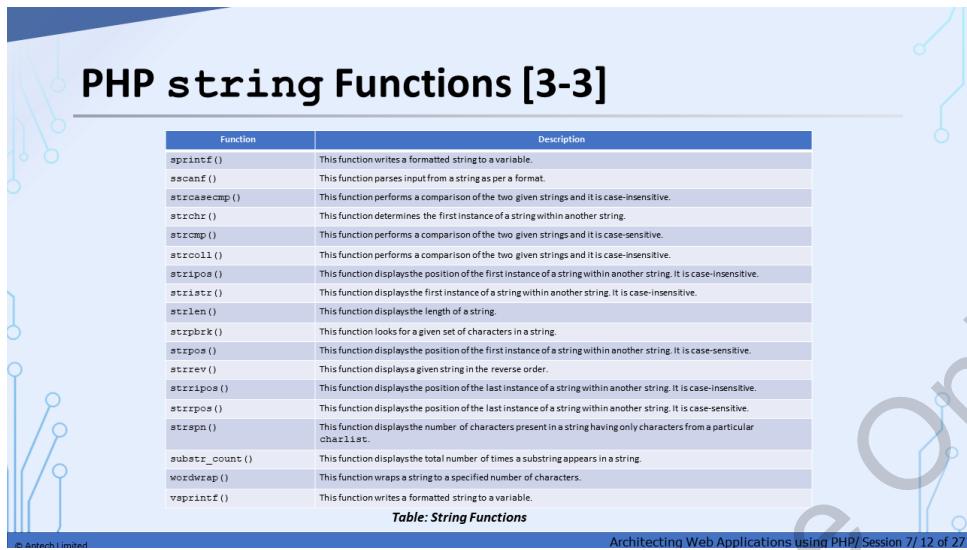
strtolower() function:

This built-in PHP function allows users to change the case of a string. Users can use the `strtolower()` function to convert an input string to entirely lowercase letters. Mention the example given in the slide.

str repeat() function:

The function accepts two arguments, the first of which is a string and the second of which is a number. It returns the argument string after iterating through it by the count specified in the second argument. Mention the example given in the slide.

Slide 12



Function	Description
<code>sprintf()</code>	This function writes a formatted string to a variable.
<code>sscanf()</code>	This function parses input from a string as per a format.
<code>strcasecmp()</code>	This function performs a comparison of the two given strings and it is case-insensitive.
<code>strchr()</code>	This function determines the first instance of a string within another string.
<code>strcmp()</code>	This function performs a comparison of the two given strings and it is case-sensitive.
<code>strcoll()</code>	This function performs a comparison of the two given strings and it is case-insensitive.
<code>strpos()</code>	This function displays the position of the first instance of a string within another string. It is case-insensitive.
<code>stripos()</code>	This function displays the first instance of a string within another string. It is case-insensitive.
<code>strlen()</code>	This function displays the length of a string.
<code>strpbrk()</code>	This function looks for a given set of characters in a string.
<code>strpos()</code>	This function displays the position of the first instance of a string within another string. It is case-sensitive.
<code>strrev()</code>	This function displays a given string in the reverse order.
<code> strrpos()</code>	This function displays the position of the last instance of a string within another string. It is case-insensitive.
<code> strrpos()</code>	This function displays the position of the last instance of a string within another string. It is case-sensitive.
<code>strspn()</code>	This function displays the number of characters present in a string having only characters from a particular charlist.
<code>substr_count()</code>	This function displays the total number of times a substring appears in a string.
<code>wordwrap()</code>	This function wraps a string to a specified number of characters.
<code>vfprintf()</code>	This function writes a formatted string to a variable.

Table: String Functions

© Aptech Limited Architecting Web Applications using PHP/ Session 7/ 12 of 27

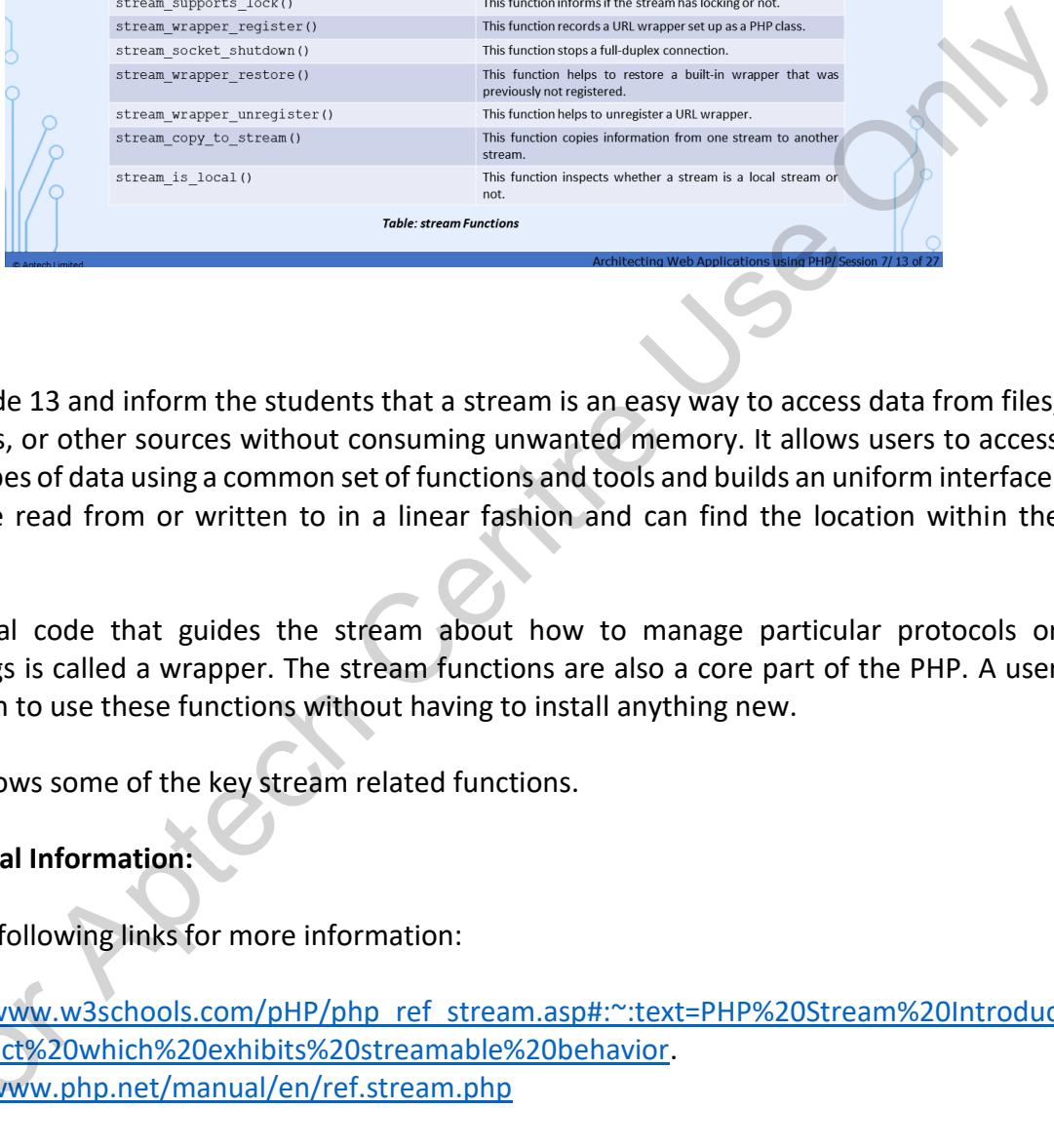
Show slide 12 and tell that table shows some of the essential string related functions available in PHP.

Ask the students following question. Wait for a response before you answer.

In-Class Question: What is the use of `strpos()` function in PHP?

Answer: The `strpos()` function is used to find the position of first occurrence of a string inside another string.

Slide 13



PHP stream Functions [1-2]

Function	Description
<code>stream_bucket_append()</code>	This function joins the bucket to the brigade.
<code>stream_bucket_make_writeable()</code>	This function displays a bucket object from the brigade to further work on.
<code>stream_socket_server()</code>	This function generates an Internet or Unix domain server socket.
<code>stream_supports_lock()</code>	This function informs if the stream has locking or not.
<code>stream_wrapper_register()</code>	This function records a URL wrapper set up as a PHP class.
<code>stream_socket_shutdown()</code>	This function stops a full-duplex connection.
<code>stream_wrapper_restore()</code>	This function helps to restore a built-in wrapper that was previously not registered.
<code>stream_wrapper_unregister()</code>	This function helps to unregister a URL wrapper.
<code>stream_copy_to_stream()</code>	This function copies information from one stream to another stream.
<code>stream_is_local()</code>	This function inspects whether a stream is a local stream or not.

Table: stream Functions

Architecting Web Applications Using PHP / Session 7 / 13 of 27

Show slide 13 and inform the students that a stream is an easy way to access data from files, networks, or other sources without consuming unwanted memory. It allows users to access many types of data using a common set of functions and tools and builds an uniform interface. It can be read from or written to in a linear fashion and can find the location within the stream.

Additional code that guides the stream about how to manage particular protocols or encodings is called a wrapper. The stream functions are also a core part of the PHP. A user can begin to use these functions without having to install anything new.

Table shows some of the key stream related functions.

Additional Information:

Refer to following links for more information:

https://www.w3schools.com/pHP/php_ref_stream.asp#:~:text=PHP%20Stream%20Introduction,object%20which%20exhibits%20streamable%20behavior.
<https://www.php.net/manual/en/ref.stream.php>

PHP stream Functions [2-2]

Code Snippet:

```
<?php
$fp = fsockopen("www.education.com", 80);
if (!$fp) {
    echo "Unable to open\n";
}
else {
    fwrite($fp, "GET / HTTP/1.0\r\n\r\n");
    stream_set_timeout($fp, 2);
    $res = fread($fp, 2000);
    $info = stream_get_meta_data($fp);
    fclose($fp);
    if ($info['timed_out']) {
        echo 'Connection timed out!';
    }
    else {
        echo $res;
    }
}
var_dump(stream_is_local("http://education.com"));
echo "<br>";
var_dump(stream_is_local("/etc"));
?>
```

Figure: Output for Code Snippet

localhost/code7_5.php

Warning: fsockopen(): unable to connect to www.education.com:80 (A connection attempt failed because the connected party did not properly respond after a period of time, or established connection failed because connected host has failed to respond.) in C:\xampp\htdocs\code7_5.php on line 2

Unable to open
bool(false)
bool(true)

Architecting Web Applications using PHP/Session 7/ 14 of 27

Show slide 14 and explain that given Code Snippet shows an example of how stream functions can be utilized. The objective of the code is to check whether given streams can be opened or not and set the timeout period when the request to open the streams fails.

This code sets the timeout value on stream expressed in terms of seconds and microseconds. When the function `stream_set_timeout` cannot open the intended file, it displays `Unable to open`. The function `var_dump` returns the type of variable stored, which is Boolean type. Then, the function `stream_is_local` checks whether a stream or a URL is a local one or not. If it is local, the function returns true, if not, it returns false.

Figure depicts the corresponding output.



PHP User-defined Functions [1-2]

- While PHP has numerous built-in functions, it also gives users flexibility to create their own functions.
- Unlike built-in functions which are readily available, user-defined functions are created by users.

Architecting Web Applications using PHP/ Session 7/ 15 of 27

Show slide 15 and inform the students that user-defined functions can be defined as those functions that are generated by the users depending on their requirements to complete a particular task. However, there may arise situations wherein none of the built-in functions can serve the purpose. In such cases, with user-defined functions, the user can create functions according to the task to be performed.

Mention the steps to create a user-defined function in PHP.

Some of the rules that users must adhere to while creating a user-defined function are as follows:

- The name of a function must include only alphabets numbers and underscores. Special characters cannot be a part of the function name.
- A function name must begin with either an alphabet or an underscore. A name cannot be used at the beginning of the name.
- Function names are case-insensitive.
- After the function name, use the opening curly brace { to indicate the beginning of the function code and end it with the closing curly brace }.

Additional Information:

Refer to following links for more information:

https://www.w3schools.com/php/php_functions.asp

<https://www.tutorialspoint.com/php-user-defined-functions>

<https://www.javatpoint.com/php-functions>

PHP User-defined Functions [2-2]

Code Snippet:

```
<?php
function even_number()
{
    for( $i=0; $i<=10; $i++ )
    {
        if( $i%2 == 0 ){
            echo "<br>", $i;
        }
    }
    even_number();
?>
```

Figure: Output for Code Snippet

localhost/code7_5.php

Warning: fopen(): unable to connect to www.education.com:90 (A connection attempt failed because the connected party did not properly respond after a period of time, or established connection failed because connected host has failed to respond.) in C:\xampp\htdocs\code7_5.php on line 2

Unable to open
bool(false)
bool(true)

Architecting Web Applications using PHP/ Session 7/ 16 of 27

Show slide 16 and explain the syntax to create a user-defined function:

Syntax:

```
function functionName() {  
    <code block comprising one or more statements>  
}
```

Code Snippet shows how to display a message using a user-defined function.

This code displays even numbers using `for` loop and `if` statements. `for` loop assigns variable `i` to 0 and then, increments it till 10 (`i<=10`). The `if` condition checks whether `i` is divisible by 2 or not. If yes, it displays `i`, otherwise it does not display any number. Note that `%` is the modulus operator that returns the remainder.

Figure depicts the output of the code demonstrating a user-defined function.

The slide features a decorative background with blue and white circuit board patterns. At the top center, the title 'PHP Function Arguments and Parameters' is displayed in a bold, black font. Below the title, the heading 'Code Snippet:' is shown in a smaller, bold, black font. A code block follows, containing PHP code that defines a function 'numbers' with three parameters (\$num1, \$num2, \$num3) and calculates their product. It then outputs the result. Below the code is a screenshot of a web browser window titled 'localhost/code7_7.php'. The browser shows the output of the code: 'The product of all the numbers is: 60'. At the bottom right of the slide, there is a small text 'Aptech Limited' and a footer note 'Architecting Web Applications Using PHP/ Session 7/ 17 of 27'.

Show slide 17 and explain that for a function, users can write one or more parameters to which they can pass arguments during a function call. Arguments can be used to feed data to the functions. An argument is similar to a variable. To pass an argument to a function, users can specify it within the parentheses soon after the function name. Users can pass as many parameters as required by separating each one of them with a comma.

Code Snippet shows how to display the names of people by passing one argument to a function.

In the code, the name of the function is `numbers()` and it has three parameters `num1`, `num2`, and `num3`. Now, when the `numbers()` function is called, all the three arguments are passed to the function parameters. Therefore, the program enters into a function body, calculates the product of all the three numbers 4, 3, and 5 and displays the output as 60.

Figure depicts the output of the code.

Additional Information:

Refer to following links for more information:

<https://www.phptutorial.net/php-tutorial/php-function-parameters/>
<https://www.tutorialspoint.com/php-function-arguments>

Ask the students following question. Wait for a response before you answer.

In-Class Question: What is the difference between parameters and arguments?

Answer: The parameters are used to accept inputs during runtime. While passing the values during a function call, they are called arguments. An **argument is a value passed to a function** and a parameter is used to hold those arguments.

Slide 18

The slide features a blue header bar with the title 'PHP Default Argument Value'. Below the title is a section labeled 'Code Snippet:' containing the following PHP code:

```
<?php declare(strict_types=1); // strict requirement ?>
<body>
<?php
function value(int $val_default = 100) {
    echo "The value is : $val_default <br>";
}
value(300);
value();
value(135);
value(150);
?>
```

Below the code is a screenshot of a browser window showing the output of the code execution. The output is:

The value is : 300
The value is : 100
The value is : 135
The value is : 150

At the bottom of the slide, there is a footer with the text '© Aptech Limited' and 'Architecting Web Applications using PHP / Session 7 / 18 of 27'.

Show slide 18 and explain that function arguments play a vital role in executing the function code. There are instances when a user might not remember to pass the argument while calling the function, which expects arguments. This may lead to unexpected results. Therefore, to avoid such instances, a default value is assigned to the arguments. Such a value is called default argument value. This value will be utilized if no value is entered for the argument when the function is executed.

Code Snippet shows how to use a default argument.

In the code, when the function `value()` is called, the numbers are passed to the function during function call and the respective numbers are displayed. However, when the function is called without an argument, it considers the default value 100 as the argument.

Figure depicts the output of the code.

Additional Information:

Refer to following link for more information:

<https://easytolearning.com/default-argument-values-function-in-php>

PHP Functions - Returning Values

Code Snippet:

```
<?php
function Add_Numbers(int $a, int $b) {
    $c = $a + $b;
    return $c;
}
echo "15 + 10 = " . Add_Numbers(15,10) . "<br>";
echo "17 + 13 = " . Add_Numbers(17,13) . "<br>";
echo "12 + 14 = " . Add_Numbers(12,14);
?>
```

Figure: Output for Code Snippet

localhost/code7_9.php

15 + 10 = 25
17 + 13 = 30
12 + 14 = 26

© Aptech Limited

Architecting Web Applications Using PHP / Session 7 / 19 of 27

Show slide 19 and explain that an important aspect of PHP functions is that they can also return results. For example, if there is a function defined to perform certain mathematical calculations, users might prefer to return the result of the calculations to the calling program. In such instances, when the function is called, the outcome of the function can be obtained in the calling statement.

To do this, users can utilize the return statement within the function body as the last statement of the function. It helps in returning any variable or value from a function. Note that only one variable/value can be returned. The return keyword cannot be followed by multiple values.

Code Snippet shows how to return a value to a function using the return statement.

The code shows how to add two given numbers and store the sum in variable c. The function named `Add_Numbers` is defined to accept two parameters namely, a and b and then, calculates the sum of variables a and b. The function returns the value of c to the calling program. In the calling program, the statements calling the function are embedded within echo statements.

Figure depicts the output of the code.

Slide 20

The slide features a decorative background with blue circuit board patterns on the left and right sides. The title 'PHP Return Type Declarations [1-2]' is centered at the top. Below the title, a section labeled 'Code Snippet:' contains the following PHP code:

```
<?php declare(strict_types=1); // strict requirement
function AddNumbers(float $n1, float $n2) : float {
    return $n1 + $n2;
}
echo AddNumbers(1.5, 7.2);
?>
```

Below the code is a screenshot of a web browser window showing the output of the code execution. The browser title bar says 'localhost/code7_10.php'. The main content area of the browser shows the number '8.7'.

Figure: Output for Code Snippet

Architecting Web Applications using PHP/Session 7/ 20 of 27

Show slide 20 and explain that PHP 8 offers a new feature called type declarations for return statements. Similar to type declaration for function arguments, if strict requirement is enabled, the code displays a 'Fatal Error' in case of a type mismatch.

In order to do a type declaration for a function return, while creating the function, users must insert a colon (:) and type before the opening curly ({) bracket.

Code Snippet shows an example of how to specify return type for a function.

In the code, a user-defined function `AddNumbers` is declared. It adds two floating-point numbers saved in variables `n1` and `n2` respectively. It then returns the sum of the numbers utilizing the keyword `return`. The function `AddNumbers` is called and passed two floating-point values. The return value is displayed using `echo`. All these actions seem similar to actions being performed in general in earlier programs. The difference here is that the return type is limited to a floating-point value through the use of colon and `float` keyword.

Therefore, the output is sum of 1.5 and 7.2, which is 8.7, as shown in figure.

Additional Information:

Refer to following links for more information:

<https://stackoverflow.com/questions/5104564/php-return-type/38970809>
https://www.tutorialspoint.com/php7/php7_returntype_declarations.htm

Slide 21

The slide features a decorative background with blue circuit board patterns on the left and right sides. The title 'PHP Return Type Declarations [2-2]' is centered at the top. Below the title, a 'Code Snippet:' label is followed by a code block:

```
<?php declare(strict_types=1); // strict requirement
function SumNumbers(float $x1, float $x2) : int {
    return (int)($x1 + $x2);
}
echo SumNumbers(5.2, 7.2);
?>
```

Below the code is a screenshot of a web browser window showing the output: 'localhost/code7_11.php' displays '12'. A caption below the browser window reads 'Figure: Output for Code Snippet'.

At the bottom right of the slide, the text 'Architecting Web Applications using PHP/Session 7/21 of 27' is visible.

Show slide 21 and explain that Code Snippet shows how to set a return type that is different from the argument types by ensuring the return is the correct type.

In the code, two floating-point numbers are added, but the sum displayed is an integer. This is because, in the function declaration `SumNumbers()`, `int` is entered after the `:`. The two arguments and their data types passed are `float $x1` and `float $x2` respectively, but an integer value of sum is returned. `int` datatype specified after the keyword `return` displays an integer as the output while adding values of two variables `x1` and `x2`. Thus, the sum of `(5.2, 7.2)` is returned as `12` as shown in figure.

Ask the students following question. Wait for a response before you answer.

In-Class Question: What is void return type in PHP?

Answer: Functions declared with void as their return type must either omit their return statement altogether, or use an empty return statement.

Slide 22

The slide has a blue header bar with the title "Passing Arguments by Reference". Below the title is a "Code Snippet:" section containing the following PHP code:

```
<?php
function add(&$value1) {
    $value1 += 10;
}
$num1 = 2;
add($num1);
echo $num1;
?>
```

To the right of the code is a screenshot of a web browser window showing the output: "localhost/code7_12.php" and "12".

Below the code snippet is a diagram illustrating the difference between "pass by reference" and "pass by value". It shows two columns separated by a vertical dashed line.

- pass by reference:** Shows a variable "cup" pointing to a coffee mug icon. Below it is the code "fillCup(cup)".
- pass by value:** Shows a variable "cup" pointing to a coffee mug icon. Below it is the code "fillCup(cup)".

Figure: Pass By Reference and Pass By Value

Figure: Output for Code Snippet

Architecting Web Applications Using PHP/Session 7/22 of 27

Show slide 22 and explain that In PHP, arguments can be passed using one of two approaches:

- Pass by Value
- Pass by Reference

Generally, in PHP, by default, arguments are entered by value. This means that a replica of the value is utilized in the function and the actual variable that was entered into the function is not modified.

However, if a function argument is passed by reference, any modification done to the argument within the function will automatically modify the variable passed. In order to change a function argument into a reference, users can use the & operator.

Code Snippet shows how to pass function arguments by reference.

In the code, a user-defined function named `add()` is created and an argument is passed named `value1`, which stores the number passed. Later, another variable `num1=2` is declared. Then, the function `add()` is called by passing the reference of `num1` variable. Therefore, it calculates the value as $2+10=12$. Here, it can be seen that the `value1` variable is replaced by `num1` when the function `add($num1)` is called. In this example, the original argument passed to the function is itself modified and hence, the modified value is reflected in the echo statement. Thus, in this approach, pass by reference, the original argument does not remain unchanged after the function has been called. Instead, it reflects the change made within the function body. Figure on the top depicts the output.

The bottom figure shows the concept of pass by reference and pass by value.

In case of pass by reference, if any changes are made to the function definition the values also change. As seen in Figure, when a coffee mug is filled, that is, `cup=fillCup`, the changes

reflect outside the function when `fillcup()` function is called. However, in case of pass by value, the value is not affected by any changes made to `fillcup()` function.

Additional Information:

Refer to following links for more information:

<https://www.tutorialspoint.com/php-passing-by-reference#:~:text=In%20PHP%2C%20arguments%20to%20a,value%20of%20actual%20argument%20variable>.

<https://www.php.net/manual/en/language.references.pass.php>

Slide 23

The slide has a blue header bar with the title 'Dynamic Function Calls'. Below the title is a section labeled 'Code Snippet:' containing the following PHP code:

```
<?php
function Test()
{
echo "Statement is displayed by a dynamic function call<br />";
}
$function_holder = "Test";
$function_holder();
?>
```

Below the code is a screenshot of a web browser window displaying the output: 'Statement is displayed by a dynamic function call'. The browser title bar shows 'localhost/code7_13.php'. At the bottom of the slide, there is a footer bar with the text '© Aptech Limited' and 'Architecting Web Applications using PHP/ Session 7/23 of 27'.

Show slide 23 and explain that PHP allows users to allocate function names as strings to variables and later utilize these variables in the same way as the function name.

Code Snippet shows how to call a function dynamically.

In the code, a function `Test()` is created, which is used to display a message. The function name is assigned as a string value to the variable `$function_holder` using the concept of dynamic function call. Later, variable `function_holder` is called similar to how a regular function would be called. This will result in function `Test()` being called indirectly.

This concept is used when users prefer not to call the function with the same name used during function declaration and definition. Therefore, the name is assigned to another variable and the function is called by that name during a function call.

Ask the students following question. Wait for a response before you answer.

In-Class Question: What is dynamic function call?

Answer: Dynamic function calls are **useful when users want to alter program flow according to changing circumstances.**

Slide 24

The slide has a decorative background with blue and white circuit board patterns. The title 'PHP Named Arguments' is at the top. Below it, a 'Code Snippet:' section contains the following PHP code:

```
<?php
function named_arguments($number = 11, $value1 = 5){
echo "Number: "; $number;
echo " ";
echo "Value: "; $value1;
}
named_arguments (value1: 5, number: 20); //Named arguments in // different order
?>;
```

Below the code is a screenshot of a browser window showing the output: 'Number: 20 Value: 5;'. The browser title bar says 'localhost/code7_14.php'. A caption below the screenshot reads 'Figure: Output for Code Snippet'. At the bottom right, there is a footer: 'Architecting Web Applications using PHP/ Session 7/ 24 of 27'.

Show slide 24 and explain that named arguments is a new feature introduced in PHP 8.0. It permits users to pass arguments to a function considering only the parameter names and not the parameter positions. This allows users to pass the arguments without being concerned about the position of the parameter. They can simply specify the correct names of the parameters and pass values to the function.

Code Snippet shows how to display numbers using named arguments of a function.

In the code, while defining the function, number parameter is written first and then, value1 parameter.

However, while calling the function `named_arguments()`, value1 variable is passed first and then, the number variable. That is, value 5 is passed first and then, 20 is passed. Note that the output is according to the function definition itself that is, the updated value of number variable (20) is displayed first and then, the value1 variable value (5). Figure depicts this output.

From the example, it is evident that the order of the arguments during a function call does not matter. The arguments can be written in any order, but the name of the arguments during the function call must match with the parameters written during the function definition.

Slide 25

The slide is titled "date () Function". It contains four colored boxes with bullet points:

- d**: This character indicates the day of the month in two digits, for example, 01 to 31.
- m**: This character indicates the month from 01 to 12.
- Y**: This character indicates the year in four digits.
- l**: This character, which is the lowercase of the letter L, indicates the day of the week.

Code Snippet:

```
<?php  
echo "Today's date is :";  
$today1 = date("d/m/Y");  
echo $today1;  
?>
```

Output:

localhost/code7_15.php

localhost/code7_15...

Today's date is :24/12/2021

Figure: Output for Code Snippet

Architecting Web Applications using PHP/ Session 7/ 25 of 27

Show slide 25 mention that PHP also provides support for date and time.

The PHP `date()` function displays the current date and/or time of the server. This function helps users to format a given timestamp such that the date and time are more readable. A timestamp can be defined as a sequence of characters, representing the date and/or time during which a particular event took place.

Following is the syntax for the same:
`date(format, timestamp)`

The format parameter indicates how the date must be formatted. Some commonly utilized characters to represent the date are mentioned in the slide.

To perform additional formatting, users can utilize characters such as '/', '.', or '-' between the characters. Code Snippet shows how to format today's date in various ways.

In the code, today's date is displayed using the built-in function `date()`. Arguments are passed to the date function to display the date as shown in the output. Figure depicts the output.

Additional Information:

Refer to following links for more information:

https://www.w3schools.com/php/func_date_date.asp
https://www.tutorialspoint.com/php/php_function_date.htm

Slide 26

The slide features a decorative background with blue and white circuit board patterns. At the top center, the title 'time() Function' is displayed. Below the title, there is a 'Code Snippet' box containing the following PHP code:

```
<?php  
$currentTimeInSeconds = time();  
echo $currentTimeInSeconds;  
?>
```

Below the code snippet, five colored boxes (orange, orange, pink, brown, grey) list the meanings of specific characters used in the date function:

- H: This character indicates a 24-hour format of an hour from 00 to 23.
- h: This character indicates a 12-hour format of an hour from 01 to 12.
- i: This character indicates the minutes from 00 to 59.
- s: This character indicates the seconds from 00 to 59.
- A: This character indicates Ante Meridiem (AM) and Post Meridiem (PM).

Below these boxes is another 'Code Snippet' box containing the following PHP code:

```
<?php  
echo "The time is " . date("h:i:sA");  
?>
```

To the right of the slide, two screenshots of browser outputs are shown:

- The first screenshot shows the output of the first code snippet: '1640345382'. It is captioned 'Figure: Output for Code Snippet'.
- The second screenshot shows the output of the second code snippet: 'The time is 11:30:02am'. It is captioned 'Figure: Output for Code Snippet'.

At the bottom right of the slide, the text 'Architecting Web Applications using PHP/ Session 7/ 26 of 27' is visible.

Show slide 26 mention that the PHP `time()` function displays the current time in terms of seconds. The `date()` function helps in converting the number of seconds to the current date.

First Code Snippet shows how to print the current time in seconds.

In the code, the built-in function `time()` is used to display the current time in seconds. The function `time()` is assigned to a variable `currentTimeInSeconds` and then, the result is displayed using the `echo` keyword.

Some commonly utilized characters to represent time are mentioned in the slide.

Second Code Snippet shows how to output the current time in the specified format.

The code prints the current time using the commonly utilized characters in hour, minutes, and seconds format as shown in the output. Second figure shows the output.

Additional Information:

Refer to following links for more information:

https://www.w3schools.com/php/func_date_time.asp
<https://www.tutorialspoint.com/time-function-in-php>

Ask the students following question. Wait for a response before you answer.

In-Class Question: What is the difference between ECHO and print?

Answer: They are both used to output data to the screen. The differences are small: **echo has no return value while print has a return value of 1** so it can be used in expressions. echo can take multiple parameters while print can take one argument. echo is marginally faster than print.



Summary

- PHP has over 1000 built-in functions that help users to complete common tasks. This makes it easy for the users to execute and rerun common tasks in a program.
- Some of the advantages of using PHP functions are reusable code, minimal code, and clarity of code.
- PHP gives users the flexibility to create their own functions. User-defined functions can be created according to the task user wants to perform.
- An argument is similar to a variable and is used to pass input data into functions.
- Default argument value will be utilized if no value is entered for the argument when the function is executed.
- If a function argument is entered by reference, any modification done to the argument will automatically modify the variable that was entered.
- An important aspect of PHP functions is that they can also return results during a later stage of the program.
- PHP allows users to allocate function names as strings to variables and later utilize these variables in the same way as the function name.
- Named arguments is a new feature introduced in PHP 8, which permits users to pass arguments to a function considering only the parameter names and not the parameter positions.
- While the PHP date() function displays the current date and/or time of the server, the time() function displays the current time in terms of seconds.

Architecting Web Applications Using PHP/ Session 7/ 27 of 27

Use slide 27 to summarize the session. You will end the session with a summary of what has been taught in the session. Tell students the pointers of the session. This will be a revision of the current session.

7.3 Post Class Activities for Faculty

You should familiarize yourself with the topics of the next session.

Tips:

You can also check the Articles/Blogs/Expert Videos uploaded on the Online Varsity site to gain additional information related to the topics covered in the next session.

Session 8 – Cookies and Sessions Management in PHP

8.1 Pre-Class Activities

Before beginning the session, you should fully familiarize yourself with its topics. Prepare a question or two that will be a key point to relate the current session objectives.

8.1.1 Teaching Skills

To teach this session, you should be well-versed with the concept of cookies and session management in PHP. You must also be familiar with the concept of localization.

You must use the given images and teach the concepts in the theory class. For teaching in the class, you are expected to use slides and LCD projectors.

Tips:

It is recommended that you test the understanding of the students by asking questions in between the class.

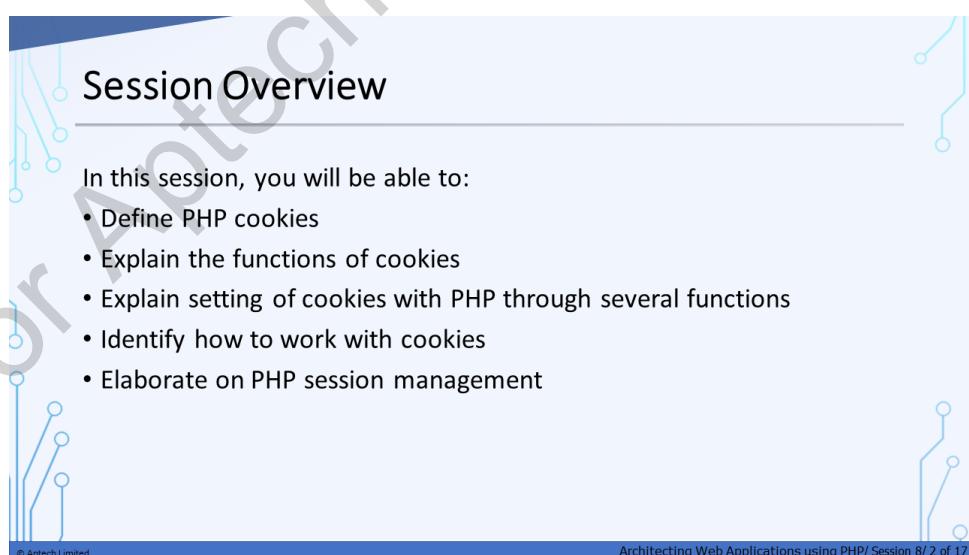
In-Class Activities

Follow the order given here during In-Class activities.

Overview of the Session

Give the students an overview of the current session in the form of session objectives. Read out the objectives on slide 2.

Slide 2



The slide has a blue header bar with the title "Session Overview". The main content area is white with a light blue background. On the left and right sides, there are decorative vertical columns made of blue lines and small circles. The text in the center reads: "In this session, you will be able to:" followed by a bulleted list of six items. At the bottom, there is a footer bar with the text "© Aptech Limited" and "Architecting Web Applications using PHP/ Session 8/ 2 of 17".

In this session, you will be able to:

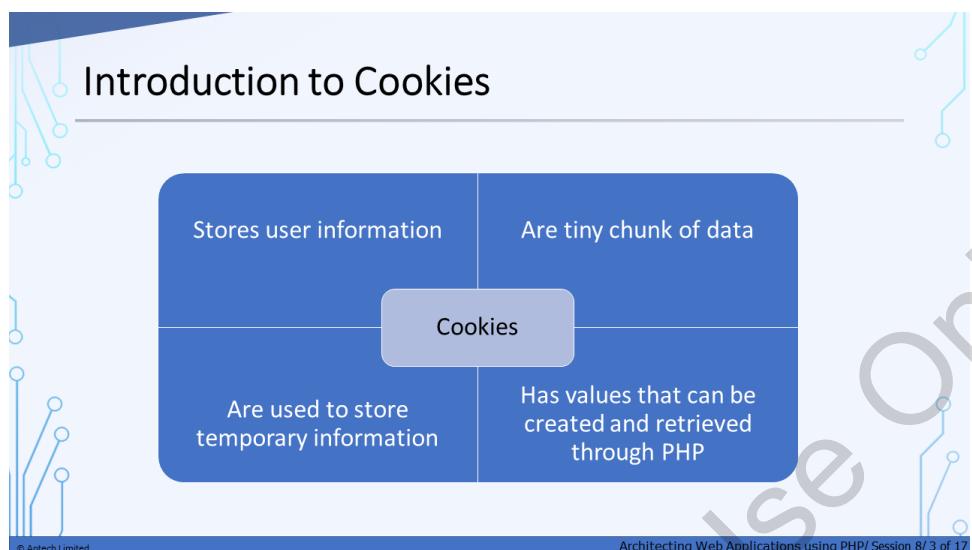
- Define PHP cookies
- Explain the functions of cookies
- Explain setting of cookies with PHP through several functions
- Identify how to work with cookies
- Elaborate on PHP session management

Show slide 2 and give the students a brief overview of the current session in the form of session objectives. Inform students that this session explains the functions of cookies and settings of cookies with PHP through several functions.

Additionally, tell students that they will learn elaborately on PHP session management.

8.2 In-Class Explanations

Slide 3



Show slide 3 and explain to students that, cookies in the world of the Web and networks, are a tiny chunk of data-carrying information that comes in handy. They enable Websites to store user information.

Cookies are used to store temporary information of nonregistered frequent Website visitors. The values in cookies can be both created and retrieved through PHP. They are created at the Web server-side and stored at the client Web browser.

Ask the following question to the students. Wait for the response before you answer.

In-Class Question: What are cookies used for?

Answer: Cookies are used to store temporary information of nonregistered frequent Website visitors.

Types of Cookies [1-2]	
Persistent	Non-persistent
Exist in the Web browser for a period specified at the time of its creation	Deleted from the Web browser as soon as the user exits the browser

© Aptech Limited Architecting Web Applications using PHP / Session 8 / 4 of 17

Show slide 4 and explain the difference between persistent and non-persistent cookies. Say that the persistent or temporary information is stored in the cookies for a stipulated period. The non-persistent or permanent data is stored in the cookies for a certain period, and then, the required information is saved in the database.

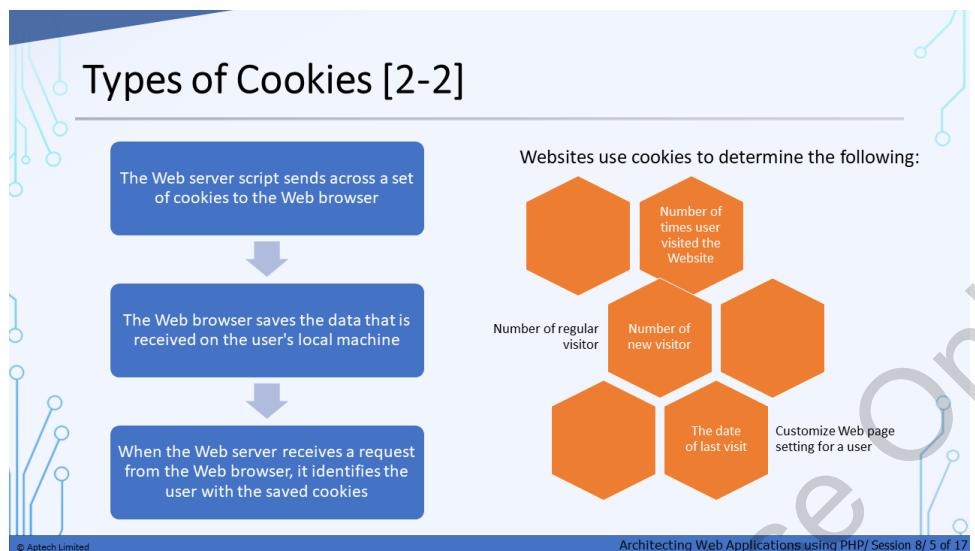
As cookies are saved on the client computer, they can be used for tracking purposes.

Additional Information:

Refer to following links for more information:

- <https://www.studytonight.com/php/php-cookies#:~:text=There%20are%20two%20types%20of,it%20with%20an%20expiration%20time.>
- <https://www.cookiepro.com/knowledge/what-is-a-persistent-cookie/>

Slide 5



Show slide 5 and explain the steps involved in identifying returning users shown in the flow diagram.

Websites use cookies to determine the following:

- Number of times the user has visited the Website
- Number of new visitors
- Number of regular users
- The date on which the user had last visited the Website
- Customized the Web page settings for a user

Ask following question to the students. Wait for the response before you answer.

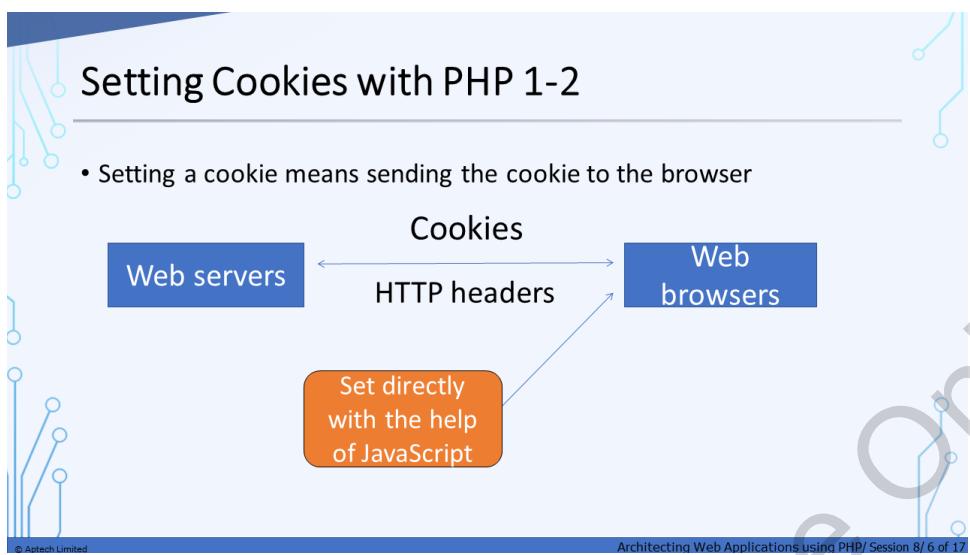
In-Class Question: List some uses of cookies.

Answer: To know the number of new visitors, number of regular visitors, the date of last visit, number of times user visited the Website, and to customize the Web page setting for a user.

Additional Information:

Refer to following links for more information:

<https://www.geeksforgeeks.org/php>
https://www.tutorialspoint.com/php/php_cookies.htm
https://www.w3schools.com/php/php_cookies.asp



Show slide 6 and explain that setting cookies means sending cookies to the browser. It can be done with an HTTP header or else directly using the JavaScript.

Web servers and Web browsers send cookies to each other in HTTP headers. The Web server sends the cookie to the browser in the `Set_cookie` header field. This field is a part of the HTTP response. The Web browser stores the cookie and uses the same in subsequent requests to the same Web server.

Generally, HTTP headers are used to set cookies. They can also be set directly on a Web browser with the help of JavaScript.

Additional Information:

Refer to following link for more information:

<https://www.geeksforgeeks.org/php-cookies/#:~:text=Setting%20Cookie%20In%20PHP%3A%20To,%2C%20path%2C%20domain%2C%20security%3B>

Setting Cookies with PHP 2-2

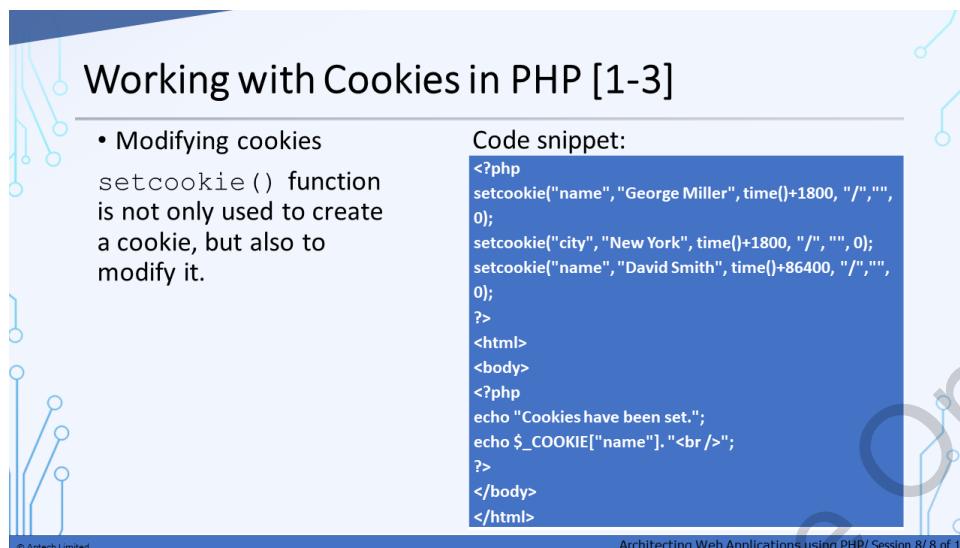
**PHP
setcookie()
function**

**Retrieving Cookie
values using PHP
\$_COOKIE**

© Aptech Limited

Architecting Web Applications using PHP / Session 8 / 7 of 17

Show slide 7 and explain that in the PHP `setcookie()` function, the cookies are set with the HTTP response and `Expires` field contains the date until the data remain in the browser. Then, explain that `$_COOKIE` is a superglobal variable in PHP and is used to retrieve the value of a cookie.



Working with Cookies in PHP [1-3]

• Modifying cookies
setcookie() function is not only used to create a cookie, but also to modify it.

Code snippet:

```
<?php
setcookie("name", "George Miller", time() + 1800, "/", "", 0);
setcookie("city", "New York", time() + 1800, "/", "", 0);
setcookie("name", "David Smith", time() + 86400, "/", "", 0);
?>
<html>
<body>
<?php
echo "Cookies have been set.";
echo $_COOKIE["name"]. "<br />";
?>
</body>
</html>
```

© Aptech Limited Architecting Web Applications using PHP / Session 8 / 8 of 17

Show slide 8 and tell that to update an existing cookie with a new value, it is recommended to add '/' in the fourth argument which is the 'path' argument. This prevents creation of another cookie with the same name.

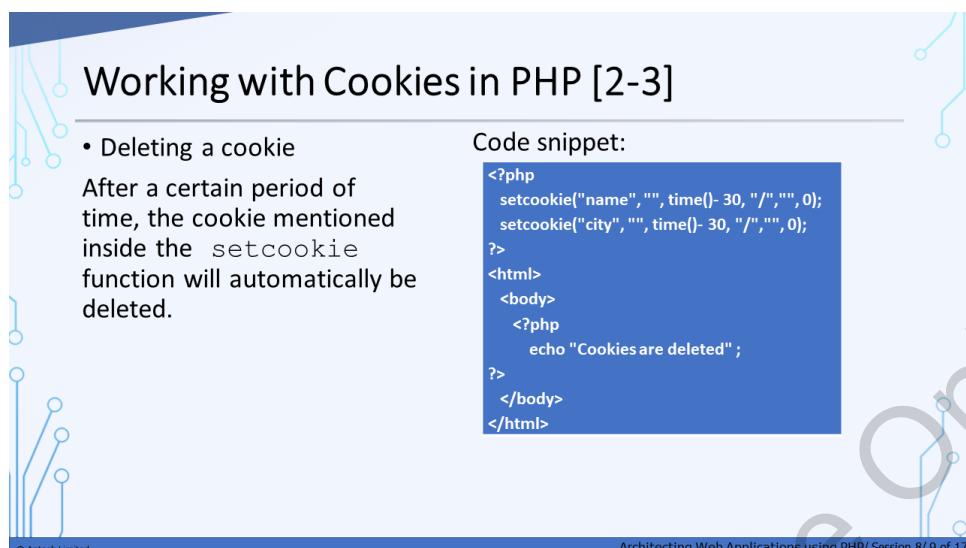
In this code snippet, tell that the output will show David Smith instead of George Miller. Ask the students to explain the reason.

Additional Information:

Refer to following links for more information:

<https://code.tutsplus.com/tutorials/how-to-work-with-cookies-in-php--cms-36575>

<https://www.guru99.com/cookies-and-sessions.html>



Working with Cookies in PHP [2-3]

• Deleting a cookie
After a certain period of time, the cookie mentioned inside the `setcookie` function will automatically be deleted.

Code snippet:

```
<?php
    setcookie("name", "", time() - 30, "/", "", 0);
    setcookie("city", "", time() - 30, "/", "", 0);
?>
<html>
<body>
<?php
    echo "Cookies are deleted";
?>
</body>
</html>
```

© Aptech Limited Architecting Web Applications using PHP / Session 8 / 9 of 17

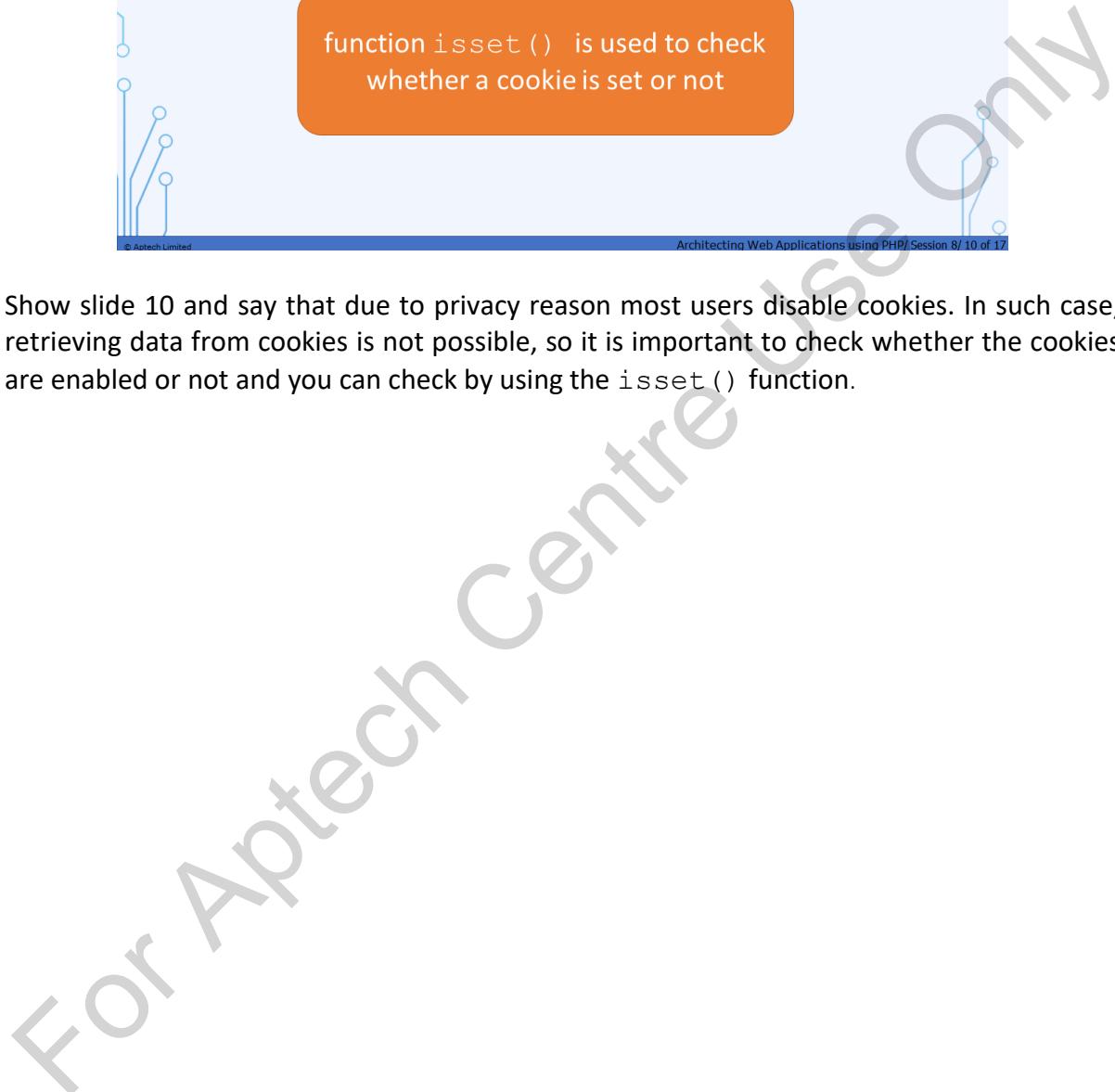
Show slide 9 and explain the code snippet. Say that as per the code after 30 minutes the cookie will be deleted.

You can delete a cookie by calling the `setcookie()` function with the cookie name and any value (such as an empty string), however, you must set the expiration date in the past. In this code, the expiration date and time are set to half an hour before current time.

Additional Information:

Refer to following links for more information:

- [https://www.geeksforgeeks.org/remove-a-cookie-using-php/#:~:text=Deleting%20Cookie%3A%20There%20is%20no,using%20the%20setcookie\(\)%20function.](https://www.geeksforgeeks.org/remove-a-cookie-using-php/#:~:text=Deleting%20Cookie%3A%20There%20is%20no,using%20the%20setcookie()%20function.)
- https://www.w3schools.com/php/php_cookies.asp



Working with Cookies in PHP [3-3]

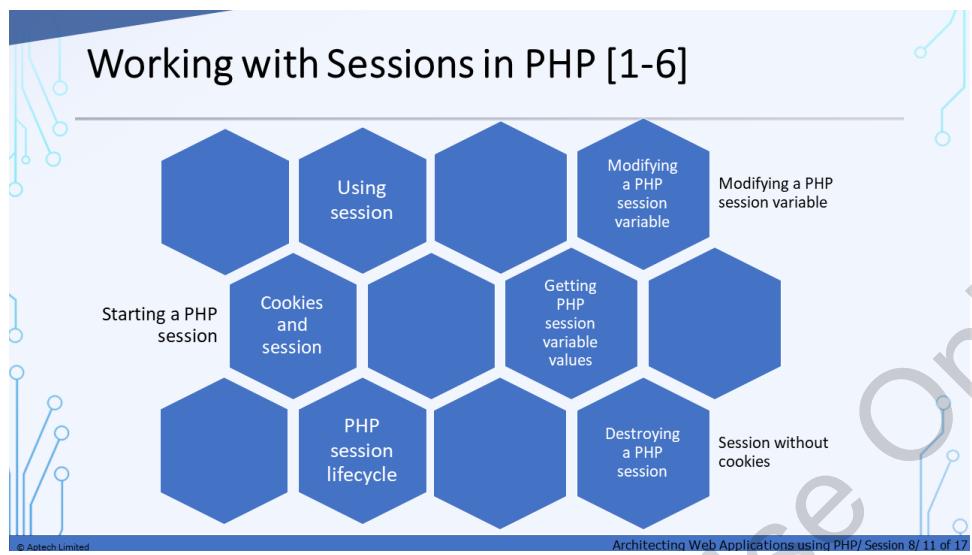
- Checking whether cookies are enabled
Browsers can be configured to disable cookies due to privacy concerns.

function `isset()` is used to check whether a cookie is set or not

© Aptech Limited

Architecting Web Applications using PHP/ Session 8/ 10 of 17

Show slide 10 and say that due to privacy reason most users disable cookies. In such case, retrieving data from cookies is not possible, so it is important to check whether the cookies are enabled or not and you can check by using the `isset()` function.



Show slide 11 and tell that, a Session in PHP temporarily stores and passes the information from one Web page to another, until the Website is closed.

Then, give the students an over view about sessions listing:

Cookies and session, using session, PHP session lifecycle, starting a PHP session, getting PHP session variable values, destroying a PHP session, modifying a PHP session variable, turning ON auto session, session without cookies.



Working with Sessions in PHP [2-6]

Disadvantages of using cookies:

- Deletion of cookies
- Multiple cookies to the same user
- Size of the cookies
- Cookies disabled
- Cookies are prone to risk

© Aptech Limited Architecting Web Applications using PHP/ Session 8/ 12 of 17

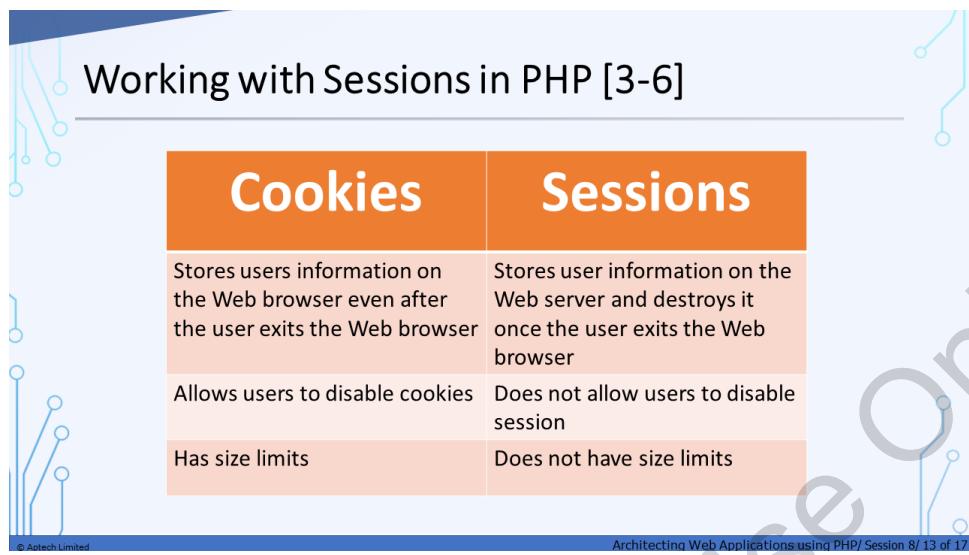
Show slide 12 and state the disadvantage of cookies. Say that cookies are more prone to risk when a Website is accessed through a public computer, cookies can be deleted easily from a client system, and it usually records a new user entry for the same person login through a different computer, users can even disable cookies as the cookies are stored in clients hard disk and it takes huge space and reduces the performance of the computer. The size of the Web page increases when a large amount of information is stored in a cookie, increased size reduces the performance.

Ask the following question to the students. Wait for the response before you answer.

In-Class Question: What are the disadvantages of using cookies? Mention any three.

Answer: Disadvantages of cookies are:-

- Users can delete cookies.
- Security risk as anyone can open and tamper with cookies.
- Limited number of cookies only can be used by a browser or domain.



Working with Sessions in PHP [3-6]

Cookies	Sessions
Stores users information on the Web browser even after the user exits the Web browser	Stores user information on the Web server and destroys it once the user exits the Web browser
Allows users to disable cookies	Does not allow users to disable session
Has size limits	Does not have size limits

© Aptech Limited Architecting Web Applications using PHP/ Session 8/13 of 17

Using slide 13, enumerate the difference between cookies and session. Explain to the students that cookies are client side files on a local computer that holds user information. It can only store a certain amount of information and cookies end on the lifetime set by the user. Whereas, sessions are server-side files that contain user data. It can hold an indefinite quantity of data and when the user quits the browser or logs out of the programmed, the session is over.

Ask the following question to the students. Wait for the response before you answer.

In-Class Question: What is the difference between cookies and sessions?

Answer: 1. Cookies stores user's information on the Web browser.

Sessions stores user information on the Web server.

2. There is a size limit for cookies data.

Sessions can store any amount of data.

3. Users can disable cookies.

Users cannot disable session.

Working with Sessions in PHP [4-6]

- Using Session

PHP creates a unique identifier for the session.
This is a random string of 32 hexadecimal digits.

The PHPSESSID cookie is automatically sent to the user's computer to save the unique session identification string.

An automatically created file on the Web server in the designated temporary directory bears the name of the unique identifier.

© Aptech Limited

Architecting Web Applications using PHP/ Session 8/ 14 of 17

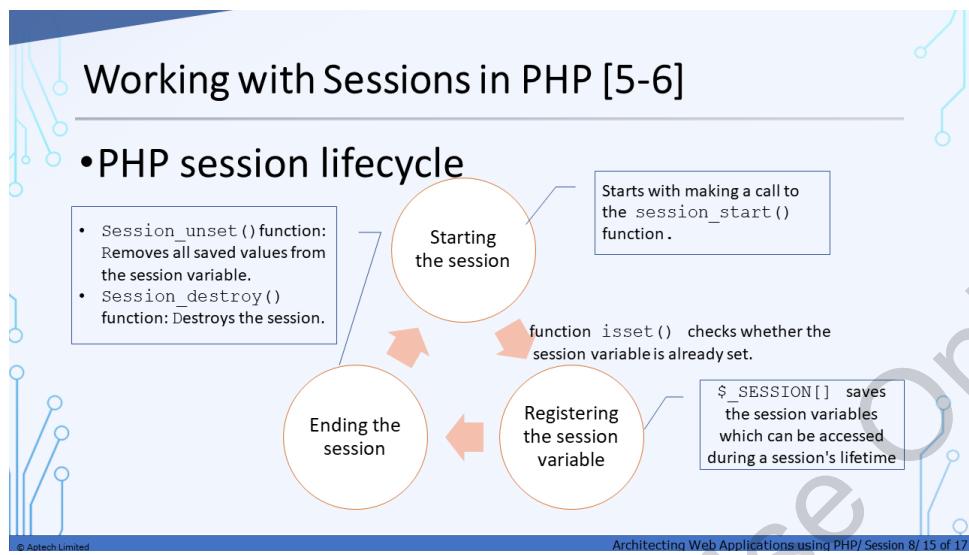
Show slide 14 and explain that the unique ID for the session is a random string of 32 hexadecimal digits. This ID is sent to the user's computer to save the session ID and a file on the Webserver is created which bears the information of the user. PHP automatically retrieves the unique session identifier string from the PHPSESSID cookie when a PHP script requires a value from the session variable. It then checks in its temporary directory for the file with the same name so that verification can be done by collating both the values.

A session is also terminated when a user loses the Web browser connection or leaves the Web page.

Additional Information:

Refer to following links for more information:

- <https://blog.eduonix.com/Web-programming-tutorials/learn-working-with-sessions-in-php/>
- https://www.w3schools.com/php/php_sessions.asp
- <https://code.tutsplus.com/tutorials/how-to-use-sessions-and-session-variables-in-php--cms-31839>



Show slide 15 and explain the lifecycle of the PHP session. Say that the PHP session lifecycle involves starting a PHP session using the `session_start()` function, then getting the PHP session variable values, and finally, destroying a PHP session.

Working with Sessions in PHP [6-6]

- Modifying a PHP Session Variable**
 - Simply overwrite the new value over the old one in the session variable.
- Turning ON Auto Session**
 - PHP facilitates automatic start of a session through `session.auto_start`
- Using sessions Without Cookies**
 - Session ID is sent to the browser through an alternate method when user does not want to save the data
 - Current session id is retrieved using `session_id()`

© Aptech Limited

Architecting Web Applications using PHP/ Session 8/ 16 of 17

Show slide 16 and explain how to modify a PHP session variable, turn ON auto session and using sessions without cookies.

Summary

- A cookie is a small file created at the Web server side and stored at the client Web browser.
- PHP cookies can be created, modified, retrieved, and even deleted.
- The `setcookie()` function in PHP enables to create and modify cookies.
- Cookies have some disadvantages; hence, PHP sessions are sometimes used in place of them.
- PHP sessions temporarily store and pass the information from one Web page to another, until the Website is closed.
- The major difference between cookies and sessions is that cookies store information on local computer, whereas a session stores information on the Web server.
- User data is stored in session variables to be utilized across multiple Web pages and they last until the Web browser is closed by the user.
- There are three stages in the lifecycle of a session based on communication between Web browser and Web server, namely, Starting the session, registering the session variable, and ending the session.
- A PHP session starts by making a call to the `session_start()` function.
- The `session_unset()` function removes all the saved values from the session variable and cleans the session variable.
- Session information can be sent from one page to another without cookies, by using SID.

Use slide 17 to summarize the session. You will end the session, with a brief summary of what has been taught in the session. Tell the students pointers of the session. This will be a revision of the current session.

8.3 Post Class Activities for Faculty

You should familiarize yourself with the topics of the next session.

Tips:

You can also check the Articles/Blogs/Expert Videos uploaded on the Online Varsity site to gain additional information related to the topics covered in the next session.

Session 9 – Database Management in PHP

9.1 Pre-Class Activities

Before beginning the session, you should fully familiarize yourself with its topics. Prepare a question or two that will be a key point to relate the current session objectives.

9.1.1 Teaching Skills

To teach this session, you should be well-versed with the concept of internationalization and design patterns in Java. You must also be familiar with the concept of localization.

You must use the given images and teach the concepts in the theory class. For teaching in the class, you are expected to use slides and LCD projectors.

Tips:

It is recommended that you test the understanding of the students by asking questions in between the class.

In-Class Activities

Follow the order given here during In-Class activities.

Overview of the Session

Give the students an overview of the current session in the form of session objectives. Read out the objectives on slide 2.

Session Overview

In this session, you will be able to:

- Describe PHP support for MySQL
- Identify prerequisites for databases in PHP
- Elaborate how to establish MySQL database connection
- Describe how to create and delete MySQL Database and tables using MySQL in PHP
- Explain data insertion and data retrieval to/from MySQL database
- Explain how to update data into MySQL database
- List and explain the three ways to backup MySQL database
- Explain selecting and filtering data from MySQL database using WHERE, ORDER BY, and LIMIT clauses

© Aptech Limited Architecting Web Applications using PHP/ Session 9/ 2 of 29

Show slide 2 and give a brief overview of the current session in the form of session objectives. Inform students that the session begins with a brief overview of PHP's MySQL support. The session walks the user through setting up a MySQL database connection. The session also describes how to use PHP to construct and erase MySQL databases and tables. Users will then learn how to input and retrieve data to and from the MySQL database. Finally, this session describes how to extract and filter data from a MySQL database using different clauses.

9.2 In-Class Explanations

Slide 3

PHP and MySQL

PHP Web applications will involve data handling at some point or the other. This data will be stored in databases to persist them.

To store, retrieve, and maintain such data through scripts, PHP supports many database management systems including MySQL, MariaDB, Db2, MongoDB, Oracle, PostgreSQL, and SQLite.

What is MySQL?

- First developed by Oracle, MySQL is a Relational Database Management System (RDBMS) that is a free and open-source software.
- MySQL mostly uses the standard SQL and runs on the server, along with being used in Web application development.
- Being exceptionally fast, easy-to-use, and reliable, it helps organize the data in databases and is an ideal choice for small and large applications. PHP, along with MySQL, is cross-platform.

© Aptech Limited Architecting Web Applications using PHP/ Session 9/ 3 of 29

Show slide 3 and explain to students that at some point, data handling will be required in a PHP Web application. A shopping cart application, a login form display and processing application, and a ticket booking application are all examples of data-driven apps. This data will be saved in databases in order to preserve it. PHP supports a variety of database management systems, including MySQL, MariaDB, Db2, MongoDB, Oracle, PostgreSQL, and SQLite, for storing, retrieving, and maintaining data via scripts.

Versions 5.5 and newer of MySQL are supported. Versions 5.5 and newer of MariaDB are supported.

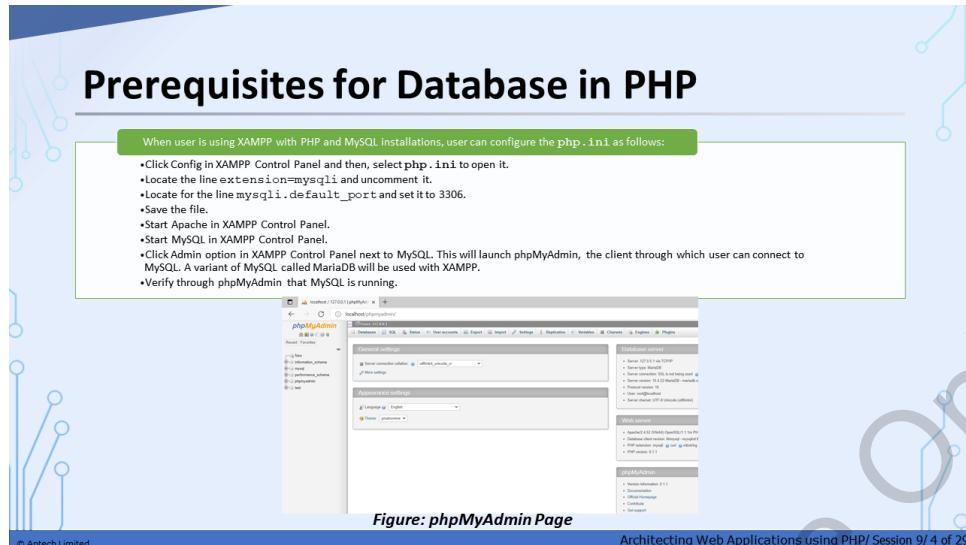
MySQL: MySQL is a free and open-source Relational Database Management System (RDBMS) that was first created by Oracle. MySQL, which mostly uses the standard SQL, is a database management system that runs on a server and is used to construct Web applications. It helps to organize data in databases and is a perfect solution for small and large applications since it is extremely fast, easy-to-use, and robust. PHP, including MySQL, is cross-platform.

Tables are used to store data in MySQL. A table is a data collection that consists of rows and columns. Databases can categorize the provided information. For example, a company's database may have tables categorized as Employees, Customers, Products, Orders, and so on.

Give the following additional information to the students:

TIP: The 'My' in MySQL stands for the cofounder Monty Widenius's daughter's name and SQL stands for Structured Query Language.

Slide 4



The screenshot shows the phpMyAdmin interface. On the left, there's a sidebar with navigation links like Home, Databases, Structure, etc. The main area has tabs for Databases, Structure, and SQL. Under Databases, there are two entries: 'Information_schema' and 'test'. Under 'test', there are several tables listed: 'category', 'customer', 'order', 'order_details', 'product', and 'supplier'. The top of the page has a title 'Prerequisites for Database in PHP' and a note about XAMPP configuration.

When user is using XAMPP with PHP and MySQL installations, user can configure the `php.ini` as follows:

- Click Config in XAMPP Control Panel and then, select `php.ini` to open it.
- Locate the line `extension=mysqli` and uncomment it.
- Locate for the line `mysqli.default_port` and set it to 3306.
- Save the file.
- Start Apache in XAMPP Control Panel.
- Start MySQL in XAMPP Control Panel.
- Click Admin option in XAMPP Control Panel next to MySQL. This will launch phpMyAdmin, the client through which user can connect to MySQL. A variant of MySQL called MariaDB will be used with XAMPP.
- Verify through phpMyAdmin that MySQL is running.

Figure: phpMyAdmin Page

Architecting Web Applications using PHP/ Session 9/ 4 of 20

Show slide 4 and explain to students that the inclusion of mysqli support is a key prerequisite for using MySQL databases in PHP 8. When using XAMPP with PHP and MySQL installations, the `php.ini` file may be configured as follows:

- In the XAMPP Control Panel, go to Config and choose `php.ini` to access it.
- Remove the comment from the line `extension=mysqli`.
- Set `mysqli.default_port` to 3306 in the line `mysqli.default_port`.
- Save the document.
- In the XAMPP Control Panel, start Apache.
- In the XAMPP Control Panel, start MySQL.
- In the XAMPP Control Panel, next to MySQL, choose the Admin option. This will start phpMyAdmin, a client that allows the user to connect to MySQL. With XAMPP, a MySQL variant named MariaDB will be used.
- Validate that MySQL is running using phpMyAdmin.

Given figure illustrates the phpMyAdmin page. Databases and tables are listed on the left panel. The database and query editor are in the center.

MySQL Database Connection Through PHP Script [1-2]

PHP mysqli_connect(): The PHP mysqli_connect() function is used to connect with the MySQL database. It can return a resource, depending on whether the connection is null or established successfully.

```
graph TD; A[Working with MySQL and PHP] --> B[MySQL]; A --> C[PHP Data Object]; C --> D[Object-oriented]; C --> E[Procedural];
```

© Aptech Limited Architecting Web Applications using PHP/ Session 9/ 5 of 29

Show slide 5 and tell students that a user will require a username and password in order to connect to MySQL successfully and work with PHP. Different ways may be used to write PHP scripts with MySQL. The major approaches to deal with MySQL and PHP are as follows:

- **MySQLi:** It is an enhanced or updated extension for accessing and working with the MySQL database. Object-oriented (based on the concept of objects) and procedural (based on structured programming or procedures) interfaces are available in MySQLi.
- **PHP Data Object (PDO):** This is a PHP extension that supports MySQL using a broad database abstraction layer and provides a user interface for working with multiple databases. It provides more flexibility in the way data is returned. This method seeks to create a single API for all database access.

In-Class Question: Which function is used to access a MySQL database in PHP?

Answer: mysqli_connect() function

Additional Information:

Refer to following links for more information:

[https://phoenixnap.com/kb/connect-mysql-with-php#:~:text=%3C%3F-,php%20%24servername%20%3D%20%22localhost%22%3B%20%24database%20%3D%20%22,PDO%3A%3AERRMODE_EXCEPTION\)%3B%20echo%20%E2%80%9C](https://phoenixnap.com/kb/connect-mysql-with-php#:~:text=%3C%3F-,php%20%24servername%20%3D%20%22localhost%22%3B%20%24database%20%3D%20%22,PDO%3A%3AERRMODE_EXCEPTION)%3B%20echo%20%E2%80%9C)
https://www.w3schools.com/php/php_mysql_connect.asp
<https://www.geeksforgeeks.org/php-database-connection/>

Slide 6

MySQL Database Connection Through PHP Script [2-2]

Code Snippet:

```
<?php  
$host = "localhost:3306";  
$username = "root";  
$password = "root";  
$phpconn = mysqli_connect($host, $username, $password);  
if(! $phpconn) {  
    die("Could not connect to database: Please verify the  
privileges" . mysqli_error());  
}  
echo "Connection to database is successful";  
mysqli_close($phpconn);  
?>
```

In the Code Snippet, PHP's built-in function `mysqli_connect` is used to connect with the database. The variables `host`, `username`, and `passwd` are passed.

Figure: Output for Code Snippet

The screenshot shows a browser window with the URL `http://localhost/testscript.php`. The page content is "Connection to database is successful." Below the screenshot, the caption "Figure: Output for Code Snippet" is displayed.

Show slide 6 and explain the following:

PHP `mysqli_connect()`: To deal with PHP scripts prior to PHP 8, the extension `mysqli_connect()` was used. However, this has been deprecated since version 8. It is now advised to use the MySQL Improved (MySQLi) Extension, which is a database driver that acts as a bridge between MySQL databases and PHP scripts.

Following is the syntax used for the `mysqli_connect()` function:

Syntax:

```
resource mysqli_connect (username, server, password)
```

where, `username` is the account name of the user that will be used to connect to the database, `server` is the host name or IP address, and `password` is the MySQL password to authenticate the connection.

The `mysqli` extension was introduced from PHP 5.0.0 onwards. On Linux, `sudo apt-get install php-mysql` may be used to install the new package. On Windows, the `mysqli` extension is already enabled for PHP 5.3 onwards. It uses the MySQL Native Driver by default.

Connection string refers to the parameters passed to the `mysqli_connect()` function. Username, server name, and password are included in this. If necessary, the connection string can be additionally supplied with the port number and database name.

Example of a connection string:

```
"localhost",    "root",    "root123",    "sampledb",    3306,    null,  
"utf8mb4"
```

Here, `localhost` is the server name, `root` and `root123` are username, and password respectively for the MySQL account, `sampledb` is the database name, `3306` is the port number, `null` is the socket number, and `utf8mb4` is the character set encoding.

The Code Snippet demonstrates how to use PHP `mysqli_connect()`. Since no specific database is specified in the connection string, a connection to the database client is made.

If the values are accurate, database connection will be made, and the output connection to database is successful will be shown. Otherwise, the if statement's die function is called and an error message 'Could not connect to database: Please verify the privileges' will be shown.

PHP includes a built-in function called `die()`. Its purpose is to print a message and exit from the current PHP script. In PHP, it is equivalent to the `exit()` function. The user must have account credentials in order to connect to the database effectively.

Given figure depicts the output of the Code Snippet.

Slide 7

Creating and Deleting a MySQL Database [1-3]

Code Snippet:

```
<?php
$servername = "localhost";
$username = "root";
$password = "root";
// Create a connection
$conn = new mysqli($servername, $username, $password);
// Verify whether connection was successful or not
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
// Create database
$sql = "CREATE DATABASE sampleDB";
if ($conn->query($sql) === TRUE) {
    echo "Task complete. You have successfully created a database";
} else {
    echo "You have an error while creating database: " . $conn->error;
}
$conn->close();
?>
```

The Code Snippet shows an example of creating a database called sampleDB while working with MySQLi Object-oriented.

Figure: Output for Code Snippet

© Aptech Limited

Architecting Web Applications using PHP/ Session 9/ 7 of 29

Show slide 7 and tell the students that Using commands in PHP scripts, a user-defined MySQL database can be created or erased.

Creating a MySQL Database Using PHP

The `CREATE DATABASE` statement is used to construct a MySQL database. The given Code Snippet explains how to use MySQLi Object-oriented to create a database called `sampleDB`.

An object `conn` is being created in the given Code Snippet. To check for a Boolean condition, PHP's built-in function `query` is called, using this object.

The `$conn->query($sql)` call to the MySQLi query function is performed in the code line. On failure, `MySQLi::query()` returns the Boolean FALSE. On a successful query with no result sets, the Boolean TRUE will be returned. On success, a `mysqli_result` containing a result set will be returned.

If the condition is true, `sampleDB` will be created as a database, and an output 'Task complete. You have successfully created a database.' will be generated. Otherwise, an error message 'You have an error while creating database:' will appear. The database name never allows spaces in the command shown in the code snippet. The `die()` function is called if the connection is not set, and `connect_error` displays an error message indicating that the connection has failed. The output for Code Snippet is shown in figure.

Ask the students the following question. Wait for a response before you answer.

In-Class Question: What is the type of array that refers to the array with strings as an index?

Answer: Associative array

Additional Information:

Refer to following links for more information:

<https://www.geeksforgeeks.org/php-mysql-creating-database/>

https://www.w3schools.com/php/php_mysql_delete.asp

https://www.w3schools.com/php/php_mysql_create.asp

https://www.tutorialspoint.com/php/create_mysql_database_using_php.htm

<https://www.tutorialrepublic.com/php-tutorial/php-mysql-delete-query.php>

<https://www.studentstutorial.com/php/php-mysql-data-delete.php>

Creating and Deleting a MySQL Database [2-3]

Code Snippet:

```

<?php
$servername = "localhost";
$username = "root";
$password = "root";
// Initially you should create connection
$conn = mysqli_connect($servername, $username, $password);
// Check connection to return the status of connection
if (!$conn) {
    die("Error in Connection: " . mysqli_connect_error());
}
// SQL command is given to create database
$sql = "CREATE DATABASE sample2DB";
if (mysqli_query($conn, $sql)) {
    echo "Task complete. You have successfully created a database";
} else {
    echo "Error while creating database: Please check" .
        mysqli_error($conn);
}
mysqli_close($conn);
?>

```

Code Snippet shows an example of using MySQLi procedural.

© Aptech Limited

Architecting Web Applications using PHP/ Session 9/ 8 of 29

Figure: Output for Code Snippet

Show slide 8 and tell the students that when creating a new database, make sure to include the username, server name, and password in the first three arguments. If the MySQL port is different from the default of 3306, the user must specify the port number after the database name as a parameter. For example, in `$conn = new mysqli_connect("localhost", "username", "password", "database name", 3307)`, the user must specify port number as 3307.

Code Snippet demonstrates an example of using MySQLi procedural.

The code in Code Snippet looks similar to the code in Code Snippet of slide 6. However, the function parameters are specified using the MySQLi procedural method. To create a database, the user can use any of these methods. In this example, a database named sample2DB is created and stored in the variable sql.

SQL queries are executed using the `mysqli_query` function. It can be used to run a variety of query types, including insert, select, update, and delete. The `mysqli_query` function is used to connect to the database in the `if` statement, and it is passed the `sql` variable, which is a string containing the SQL command for creating the database. The output 'Task complete. You have successfully created a database' is printed. If not, the message 'Error while creating database: Please check' appears. The output for this Code Snippet is identical to the output for Code Snippet in slide 6.

Slide 9

Creating and Deleting a MySQL Database [3-3]

Code Snippet:

```
<?php
$dbhost = "localhost";
$username = "root";
$password = "root";
$conn = mysqli_connect($dbhost, $username, $password);
if(! $conn) {
    die("Could not connect: " . mysqli_error());
}
$sql = "DROP DATABASE sampleDB";
$retval = mysqli_query($conn, $sql);
if(! $retval) {
    die("Due to SQL error, deleting database sampleDB is not possible: " .
        mysqli_error());
}
echo "Database deleted successfully here";
mysqli_close($conn);
?>
```

Code Snippet shows an example of deleting a database.



Figure: Output for Code Snippet

© Aptech Limited

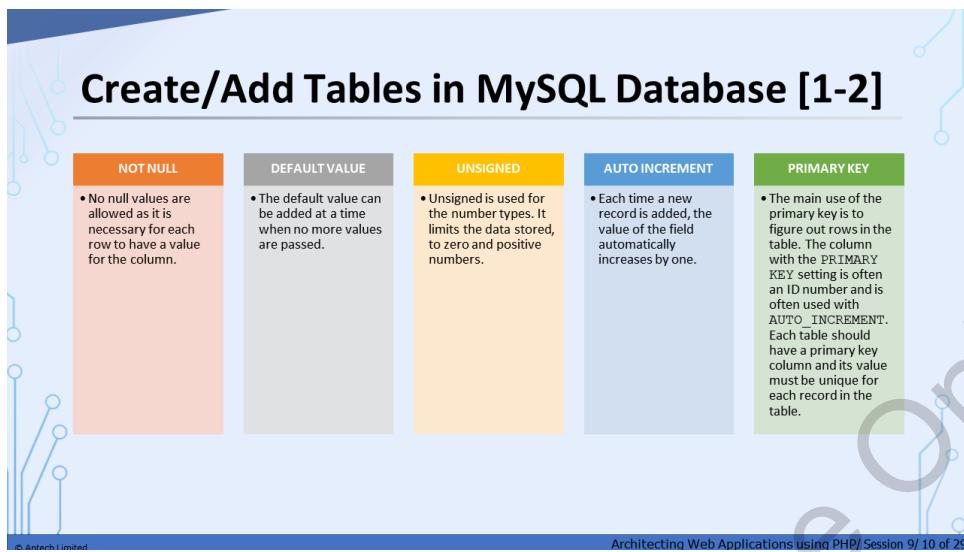
Architecting Web Applications using PHP/ Session 9/ 9 of 29

Show slide 9 and tell the students that if there is no longer a requirement for a database, it can be permanently deleted. To delete a database, pass an SQL command to `mysqli_query`.

A database deletion is demonstrated in this Code Snippet.

The statement `DROP DATABASE sampleDB` is used to delete an existing database in Code Snippet. The `mysqli_query()` function will be used to execute the statement. The query's return value is assigned to the variable `retval`. If this is the case, the database has been successfully deleted. It was not deleted in some other situation. The output for Code Snippet is shown in figure.

Slide 10



Show slide 10 and tell the students that a database by itself is useless. It must contain at least one table with data records. A database table has rows and columns, as well as a unique name.

MySQL Table Creation Using MySQLi

To create a table in MySQL, use the CREATE TABLE statement.

The data type specifies the type of data that can be stored in the column. Aside from the data type, each column has a number of optional attributes that should be specified. They are given on the slide.

Following is the MySQL code used to create a table:

```
CREATE TABLE Passengers (
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
firstname VARCHAR(30) NOT NULL,
lastname VARCHAR(30) NOT NULL,
email VARCHAR(50),
reg_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
)
```

Here, the primary key is the id column.

Ask the students the following question. Wait for a response before you answer.

In-Class Question: Which mysqli function frees the memory associated with a result?

Answer: mysqli_result.free.php

Additional Information:

Refer to following links for more information:

https://www.w3schools.com/php/php_mysql_create_table.asp

<https://www.tutorialrepublic.com/php-tutorial/php-mysql-create-table.php>

<https://www.geeksforgeeks.org/php-mysql-creating-table/>

<https://www.tutorialspoint.com/mysql/mysql-create-tables.htm>

<https://www.javatpoint.com/php-mysql-create-table>

Slide 11

The screenshot shows a web page with a blue header and footer. The main content area has a light gray background with a decorative border featuring blue lines and circles. At the top, it says 'Create/Add Tables in MySQL Database [2-2]'. Below that is a 'Code Snippet:' label followed by a code block. The code is PHP code for creating a MySQL table named 'Passengers' with columns: id (INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY), firstname (VARCHAR(30) NOT NULL), lastname (VARCHAR(30) NOT NULL), email (VARCHAR(50)), and reg_date (TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP). It includes error handling and connection closing logic. A blue arrow points from the code block to a text box below it that reads: 'Code Snippet shows how to create a table in PHP (MySQLi Object-oriented).'. To the right of the code block is a screenshot of a terminal window showing the output: 'Table Passengers created successfully'. Below the terminal screenshot is a caption: 'Figure: Output for Code Snippet'. The footer of the page includes the text '© Aptech Limited' and 'Architecting Web Applications using PHP/ Session 9/ 11 of 29'.

Show slide 11 and tell the students that this code snippet demonstrates how to make a table in PHP (MySQLi Object-oriented). It illustrates how to create a Passengers table with five columns: id, firstname, lastname, email, and reg_date.

In the given Code Snippet, the MySQL query `CREATE TABLE Passengers` is used to create a table called Passengers. It contains all of the necessary table information.

An output Table for passengers is created when the code in Code Snippet is executed. By creating an object of `mysqli` in the code snippet, the MySQLi (object-oriented) method is used to write the code. The output for Code Snippet is shown in figure.

Code Snippet:

```
<?php
$dbhost = "localhost";
$username = "root";
$password = "root";
$dbname = "sample2DB";
$conn = mysqli_connect($dbhost, $username, $password);
if(! $conn) {
die("Issue in the connection. Please check the details again: " .
mysqli_error());
}
$sql = "INSERT INTO Passengers (id,firstname,
lastname,email,reg_date) ". "VALUES (101,'Derek', 'Houston',
'derek@example.com', NOW())";
mysqli_select_db($conn,"sample2DB");
$retval = mysqli_query($conn,$sql);
if(! $retval) {
die("Could not enter data: " . mysqli_error());
}
echo "Entered data successfully in the table\n";
mysqli_close($conn);
?>
```

Figure: Output for Code Snippet

Architecting Web Applications using PHP/ Session 9/ 12 of 29

Show slide 12 and tell the students that users can use MySQLi to insert or add data to, and retrieve data from, a MySQL database after the table and database have been created in MySQL.

Database Queries

A query can be thought of as both a request and a question. An example of a standard SQL query is as follows:

```
SELECT LastName FROM Employees
```

This query selects all of the data in the Employees table's LastName column.

Inserting Data into MySQL Database

The `INSERT INTO` statement can be used to insert new records into the MySQL table. The `INSERT INTO` statement has the following syntax:

Syntax:

```
INSERT INTO table_name (column1, column2, column3,...)
VALUES (value1, value2, value3,...)
```

Here are some syntax rules:

- The SQL query should be quoted in PHP.
- The string values in the SQL query must be quoted.
- Numeric values should be quoted.
- The word `NULL` should not be quoted.

The `INSERT` statement is used to insert data into the passengers table in Code Snippet. The table's database is chosen first and then the query is run using the `mysqli query()` function. When the query is executed, an output 'Entered data successfully in the table' is displayed. Otherwise, the message 'Could not enter data:' will appear. The output for Code Snippet is shown in figure.

Ask the students the following question. Wait for a response before you answer.

In-Class Question: Which client utilities helps create logical database backups?

Answer: mysqldump

Additional Information:

Refer to following links for more information:

<https://www.geeksforgeeks.org/how-to-insert-form-data-into-database-using-php/>

https://www.w3schools.com/php/php_mysql_insert.asp

<https://www.tutsmake.com/php-code-insert-data-into-mysql-database-from-form/>

Slide 13

Code Snippet:

```
<?php
$servername = "localhost";
$username = "root";
$password = "root";
$dbname = "sample2DB";
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
$sql = "INSERT INTO Passengers (firstname, lastname, email)VALUES ('Hugh', 'Sheperd', 'hugh@sample.com')";
if ($conn->query($sql) === TRUE) {
    echo "Successfully created new record";
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}
$conn->close();
?>
```

Code Snippet shows how to add a new record to the table (MySQLi Object-oriented).

Figure: Output for Code Snippet

Architecting Web Applications using PHP/ Session 9/ 13 of 29

Show slide 13 and tell the students that instead of being hard-coded, input values will be taken using HTML forms in a real-world application. These values will be collected by a PHP script and stored in MySQL tables. Logging into PhpMyAdmin and looking through the database tables will reveal the rows that have been inserted into the table.

The given code snippet demonstrates how to add a new record to the table (MySQLi Object-oriented).

In Code Snippet, data is being inserted into a table using MySQL query "INSERT PASSENGERS INTO (firstname, lastname, email) VALUES ('Hugh', 'Sheperd', 'hugh@sample.com')". The first name, last name, and email ID values are all being inserted here. When the code in Code Snippet is run, it prints the output 'Successfully created new record'. If this is not the case, an error message will appear. The output for Code Snippet is shown in figure.

Multiple Records Insertion

Code Snippet:

```
<?php
$servername = "localhost";
$username = "root";
$password = "root";
$dbname = "sample2DB";
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
$sql = "INSERT INTO Passengers (firstname, lastname, email) VALUES ('Luke', 'Thompson', 'Luke@example.com');";
$sql .= "INSERT INTO Passengers (firstname, lastname, email) VALUES ('Ben', 'Smith', 'ben@example.com');";
$sql .= "INSERT INTO Passengers (firstname, lastname, email) VALUES ('Ruby', 'Cookies', 'ruby@example.com');";
if ($conn->multi_query($sql) === TRUE) {
    echo "New records created successfully";
} else {
    echo "Error: " . $sql . " " . $conn->error;
}
$conn->close();
?>
```

Code Snippet shows an example of adding three new records to the Passengers table using MySQLi Object-oriented.

© Aptech Limited

Architecting Web Applications using PHP/ Session 9/ 14 of 29

Figure: Output for Code Snippet

Show slide 14 and tell the students that when the user requires to execute multiple SQL statements, he or she can use the `mysqli_multi_query()` function. The given Code Snippet shows how to use MySQLi Object-oriented to add three new records to the Passengers table.

Using the `multi_query()` function in Code Snippet, multiple records are inserted into the table Passengers. A semicolon must be used to separate each SQL statement. If the query is successful, the result is displayed as New records created successfully. If this is not the case, an error message will appear. The output for Code Snippet is shown in figure.

Additional Information:

Refer to following links for more information:

- <https://stackhowto.com/how-to-insert-multiple-rows-in-mysql-using-php/>
- https://www.w3schools.com/php/php_mysql_insert_multiple.asp
- <https://www.geeksforgeeks.org/php-inserting-into-mysql-database/>

Getting Last Inserted ID

Code Snippet:

```
<?php
$servername = "localhost";
$username = "root";
$password = "root";
$dbname = "sampleDB";
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
$sql = "INSERT INTO Passengers (firstname, lastname, email) VALUES
('Mark', 'Mortan', 'Mark@example.com')";
if ($conn->query($sql) === TRUE) {
    $last_id = $conn->insert_id;
    echo "Created successfully new record in Table-Passengers. Last
    inserted ID is: " . $last_id;
} else {
    echo "Error: " . $sql . "  
" . $conn->error;
}
$conn->close();
?>
```

Code Snippet shows how to retrieve the ID of the last inserted record (MySQL Object-oriented).

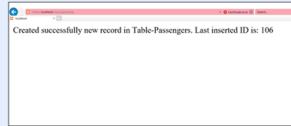


Figure: Output for Code Snippet

© Aptech Limited

Architecting Web Applications using PHP/ Session 9/ 15 of 29

Show slide 15 and tell the students that when an `INSERT` or `UPDATE` on a table is performed with an `AUTO INCREMENT` field, the ID of the last inserted or updated record can be quickly accessed.

The given code snippet demonstrates how to get the ID of the last inserted record (MySQLi Object-oriented).

Code to retrieve and show the ID of the last inserted record has been added to Code Snippet. `insert id` is being referenced in a `mysqli` object that has been generated. If the code in Code Snippet is run, an output 'Created successfully new record in Table-Passengers. Last inserted ID is: 3' is shown. The output of Code Snippet is displayed in the figure.

Additional Information:

Refer to following links for more information:

https://www.w3schools.com/php/php_mysql_insert_lastid.asp
https://www.w3schools.com/php/php_mysql_insert_lastid.asp
<https://www.tutorialrepublic.com/php-tutorial/php-mysql-last-inserted-id.php>

Code Snippet:

```
<?php
$dbhost = "localhost";
$username = "root";
$password = "root";
$dbname = "sample2DB";
$conn = mysqli_connect($dbhost, $username, $password);
if (!$conn) {
    die("Something went wrong!!Could not connect: " .
    mysqli_error());
}
$sql = "SELECT id,firstname, lastname, email FROM passengers";
mysqli_select_db($conn, "sample2DB");
$result = mysqli_query($conn, $sql);
if(!$result) {
    die("Error encountered while fetching the data: " .
    mysqli_error());
}
while($row = mysqli_fetch_array($result)) {
    echo "Passenger ID: " . ($row["id"]) . "<br>" .
    "First Name : " . ($row["firstname"]) . "<br>" .
    "Last Name: " . ($row["lastname"]) . "<br>" .
    "Email: " . ($row["email"]) . "<br>" .
    "-----<br>";
}
echo "Data successfully retrieved\n";
mysqli_close($conn);
?>
```

Code Snippet shows an example for fetching records from the Passengers table.

Figure: Output for Code Snippet

The screenshot shows the browser output of the PHP script. It displays a table-like structure with columns for Passenger ID, First Name, Last Name, and Email. The data is as follows:

	First Name	Last Name	Email
Passenger ID: 101	Derek	Houston	Email: derek@example.com
Passenger ID: 102	Morgan	Hugh	Email: hugh@example.com
Passenger ID: 103	Luke	Thompson	Email: luke@example.com
Passenger ID: 104	Benji	Sarah	Email: benji@example.com
Passenger ID: 105	Ruby	Stokes	Email: ruby@example.com
Passenger ID: 106	Mark	Morgan	Email: mark@example.com

Data successfully retrieved

Architecting Web Applications using PHP/ Session 9/ 16 of 29

Show slide 16 and tell the students that Users can retrieve data from a MySQL database with PHP. The SELECT statement can be used within a PHP script to select and fetch data from one or more tables.

The syntax for implementing the SELECT statement is as follows:

Syntax:

```
SELECT column_name(s) FROM table_name
SELECT * FROM table_name
```

The given Code Snippet demonstrates how to get records from the Passengers table. In this code snippet, the function `mysqli_fetch_array()` is used. The row is returned as a numeric or associative array in this function. The row is sometimes returned as both a numeric and an associative array. In the scenario that there are no more rows, MySQL returns false.

All passenger information is retrieved from the passengers table using a SELECT query in Code Snippet. Passenger details will be presented in the output along with the message 'Data successfully retrieved' if the code in Code Snippet is executed and a connection with the database sample2DB is established.

Here, one argument namely, `$retval`, is passed to a function `mysqli_fetch_array()`. The `mysqli_fetch_array()` function is used to display all the records from the passengers table. It can fetch the next row of a result set as an associative, a numeric array, or both. Note that `$retval` is a `mysqli_result` object returned by `mysqli_query()`, `mysqli_store_result()`, `mysqli_use_result()` or `mysqli_stmt_get_result()`. Hence, this is passed as

parameter to the `mysqli_fetch_array()` function, in order to iterate the records of the table.

Figure illustrates a section of the Code Snippet output.

Additional Information:

Refer to following links for more information:

https://www.tutorialspoint.com/php/mysql_select.php#:~:text=Data%20can%20be%20fetched%20from,a%20numeric%20array%2C%20or%20both.

<https://www.formget.com/read-mysql-data-using-php/>

<https://www.studenttutorial.com/php/php-mysql-data-retrieve.php>

Retrieving Data from MySQL Database [2-3]

Code Snippet:

```
<?php
$dbhost = "localhost";
$username = "root";
$password = "root";
$database = "sample2DB";
$conn = mysqli_connect($dbhost, $username, $password);
if(!$conn) {
    die("Could not connect: " . mysqli_error());
}
$sql = "SELECT id,firstname, lastname, email FROM passengers";
mysqli_select_db($conn, "sample2DB");
$result = mysqli_query($conn, $sql);
if($result) {
    die("Could not get data: " . mysqli_error());
}
while($row = mysqli_fetch_assoc($result)) {
    echo "Passenger ID: " . $row['id'] . "<br> ";
    echo "First Name: " . $row['firstname'] . "<br> ";
    echo "Last Name: " . $row['lastname'] . "<br> ";
    echo "Email: " . $row['email'] . "<br> ";
}
echo "Data successfully retrieved\n";
mysqli_close($conn);
?>
```

Code Snippet shows example to display all the records from the Passengers table using the `mysqli_fetch_assoc()` function.

Figure: Output for Code Snippet

The screenshot shows a browser window displaying the output of the PHP script. The output is a list of passengers with their ID, first name, last name, and email address. The output is as follows:

Passenger ID	First Name	Last Name	Email
104	Bert	Smith	ben@sample.com
105	Ruby	Stokes	ruby@sample.com
106	Mark	Mortan	Mark@sample.com
103	Luke	Thompson	

Data successfully retrieved

Show slide 17 and tell the students that `mysqli_fetch_assoc()` is another PHP function that returns the particular row or rows as an associative array.

The given code snippet demonstrates how to use the `mysqli_fetch_assoc()` function to display all records from the Passengers table.

A SELECT query is being used in Code Snippet to retrieve passenger details from the table. If the query is executed and a connection is made with the database `sample2DB`, passenger data will be shown in the output, along with the message Data successfully retrieved. The `mysqli_fetch_assoc()` function is used to show the data from the passengers table. The output for Code Snippet is shown in the figure.

Retrieving Data from MySQL Database [3-3]

Code Snippet:

```
<?php
$dbhost = "localhost";
$username = "root";
$password = "root";
$dbname = "sampleDB";
$conn = mysqli_connect($dbhost, $username, $password);
if (!$conn) {
    die("Encountered an error, could not connect: " . mysqli_error());
}
$sql = "SELECT id,firstname, lastname, email FROM passengers";
mysqli_select_db($conn, "sampleDB");
$stmt = mysqli_query($conn, $sql);
if (!$stmt) {
    die("Error while fetching data: " . mysqli_error());
}
while ($row = mysqli_fetch_array($stmt, MYSQLI_NUM)) {
    echo "Passenger ID.:{$row[0]} <br> ".
        "First Name: {$row[1]} <br> ".
        "Last Name: {$row[2]} <br> ".
        "Email: {$row[3]} <br> ";
}
echo "Data successfully fetched\n";
mysqli_close($conn);
?>
```

Code Snippet shows an example where all records from the passengers table are displayed using the `MYSQLI_NUM` argument.

Passenger ID.	First Name	Last Name	Email
101	Derek	Houston	derek@sample.com
102	Hugh	Shepard	hugh@sample.com
103	Luke	Thompson	luke@sample.com

Data successfully fetched

Figure: Output for Code Snippet

Architecting Web Applications using PHP/ Session 9/ 18 of 29

Show slide 18 and tell the students that constant `MYSQLI_NUM` can also be used as the second parameter to the `mysqli_fetch_array()` function. `MYSQLI_NUM` causes the function to act similarly to the `mysqli_fetch_row()` function, which fetches a numeric array. The given code snippet explains how to use the `MYSQLI_NUM` argument to display all records from the passengers table.

Passenger details are being retrieved from the passengers table in Code Snippet. If the query is executed and a connection is made with the database `sampleDB`, then the message 'Data successfully fetched' will be printed in the output, along with the passenger details. Using the `mysqli_fetch_assoc()` method, all records from the Passengers table are shown. In Code Snippet, the function `mysqli_fetch_array()` takes an extra argument `MYSQLI_NUM`. As a result, the array data are retrieved using numeric indexes rather than column names this time.

The output for Code Snippet is shown in the figure.

Note: Code Snippets of this slide and slide 16 will yield the same output. This is because the data in both code snippets is obtained from the same table and database with the same set of rows.

Data Selection with MySQLi

Code Snippet:

```
<?php
$servername = "localhost";
$username = "root";
$password = "root";
$dbname = "sampleDB";
// Let us create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection status
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
$sql = "SELECT id, firstname, lastname FROM Passengers";
$result = $conn->query($sql);
// The given code iterates through all the records in table
if ($result->num_rows > 0) {
    echo "<table><thead><tr><th>Name</th></tr>";
    // output data of each row
    while($row = $result->fetch_assoc()) {
        echo "    <tr><td>" . $row["id"] . "</td>" . $row["firstname"] . "</td>" . $row["lastname"] . "</td></tr>";
    }
    echo "</table>";
} else {
    echo "0 results";
}
$conn->close();
?>
```

Code Snippet shows selecting the firstname, lastname, and id columns from the Passengers table and showcasing them on to the page.

Figure: Output for Code Snippet

ID	Name
101	Derek Houston
102	Hugh Shepherd
103	Luke Thompson
104	Ben Smith
105	Ruby Stokes
106	Mark Morton

Architecting Web Applications using PHP/ Session 9/ 19 of 29

Show slide 19 and tell the students that the code snippet displays selecting the firstname, lastname, and id columns from the Passengers table and displaying them on the page. This makes the use of MySQLi Object-oriented.

In Code Snippet, the data is being retrieved based on the number of rows in the table. The SQL query in this case is set to get the firstname, lastname, and id fields from the Passengers table. The code in the next line executes the query and stores the data in the variable \$result.

The function `num_rows()` then checks whether there are more than zero rows returned. If more than two rows are returned, the function `fetch_assoc()` places them all in an associative array that can be looped through. The `while()` loop iterates through the result set, returning data from the firstname, lastname, and id columns. The output for Code Snippet is shown in the figure. In the shown table, the Name column combines first and last names.

Ask the students the following question. Wait for a response before you answer.

In-Class Question: How can data be retrieved from database in PHP using MySQLi?

Answer: To successfully fetch data from MySQL using mysqli extension in PHP, users can perform more or less three actions: **connect, execute prepared statement, and fetch data.**

Additional Information:

Refer to following links for more information:

https://www.w3schools.com/php/php_mysql_select.asp#:~:text=Select%20Data%20With%20MySQLi&text=Code%20lines%20to%20explain%20from,into%20a%20variable%20called%20%24result.

https://www.tutorialspoint.com/mysql/mysql_select_query.htm

<https://www.php.net/manual/en/mysqli.query.php>

Updating Data into a MySQL Table

Code Snippet:

```
<html>
<head>
    <title>Update a Record in mysql Database</title>
</head>
<body>
    <?php
        if(isset($_POST["update"])){
            $dbhost = "localhost";
            $username = "root";
            $password = "root";
            $conn = mysqli_connect($dbhost, $username, $password);
            if(!$conn){
                die("Could not connect: " . mysqli_error());
            }
            $passngr_id = filter_input(INPUT_POST, 'id');
            $pemail = filter_input(INPUT_POST, 'email');
            $sql = "UPDATE passengers SET email = '$pemail'
                    WHERE id = $passngr_id ";
            mysqli_query($conn, $sql);
            if(mysqli_query($conn, $sql)){
                die("Could not update data: " . mysqli_error());
            }
            echo "Updated data successfully\n";
            mysqli_close($conn);
        }
    ?>
    <form method = "post" action = "<?php $_PHP_SELF ?>">
        <table width = "400" border = "0" cellspacing =
                    cellpadding = "2">
            <tr>
                <td width = "100"> Passenger ID:</td>
                <td width = "100"> Email:</td>
                <td><input name = "email" type = "text"
                        id = "email"></td>
            </tr>
            <tr>
                <td width = "100"> </td>
                <td></td>
            </tr>
            <tr>
                <td width = "100"> </td>
                <td><input name = "update" type = "submit" id =
                    "update" value = "Update">
                </td>
            </tr>
        </table>
    </form>
</body>
</html>
```

Code Snippet shows an example of the update operation.

Figure: Output for Code Snippet

Architecting Web Applications using PHP/ Session 9/ 20 of 20

Show slide 20 and tell the students that in order to update data in MySQL tables, the user can use the PHP function `mysqli_query` to run the UPDATE statement.

An example of an update operation is shown in the Code Snippet.

The code snippet explains how to update data in the Passengers table depending on input from a form. To update records in a table, use the WHERE conditional statement to locate the specific record.

Taking into consideration that the passenger IDs are unique and only a specific record will be chosen for the operation, a passenger ID must be given as criterion to update the email in this scenario. As a result, the passenger email is updated using UPDATE MySQL query with the passenger ID as the WHERE clause criteria. When the user clicks the Update button, the message 'Updated data successfully' appears in the output. The form generated through Code Snippet is shown in the figure.

After entering data and selecting Update, when the message appears indicating that the data has been successfully updated, navigate to the table in phpMyAdmin to validate the updated record.

Additional Information:

Refer to following links for more information:

https://www.w3schools.com/php/php_mysql_update.asp
https://www.tutorialspoint.com/php/mysql_update_php.htm

Using SQL Command Through PHP [1-2]

Code Snippet:

```
<?php
$dbhost = "localhost";
$username = "root";
$password = "root";
$conn = mysqli_connect($dbhost, $username, $password);
if(! $conn) {
    die("Error while connecting" . mysqli_error());
}

$table_name = "passenger";
$backup_file = "E:\tmp\passenger.sql";
$sql = "SELECT * INTO OUTFILE '$backup_file' FROM $table_name";
mysqli_select_db($conn, "sample2DB");
$stmt = mysqli_query($conn,$sql);
if(!$stmt) {
    die("Error while attempting data backup: Sorry this operation is not successful ". mysqli_error());
}
echo "Backed up data successfully";
mysqli_close($conn);
?>
```

Code Snippet shows an example of using `SELECT INTO OUTFILE` query for creating back up of a table.

Figure: Output for Code Snippet

Architecting Web Applications using PHP/ Session 9/ 21 of 29

Show slide 21 and explain students the following topics:

Using PHP to Back Up MySQL Database

Users should back up their databases on a regular basis. There are three ways for backing up a MySQL database that users can select from. They are as follows:

- Using SQL Command through PHP
- Using MySQL binary mysqldump through PHP
- Using phpMyAdmin user interface

Using SQL Command through PHP

It is a wise decision to archive a backup using the SQL `SELECT` command. It will be necessary to write a separate query for different tables in order to take a whole database dump. Each of these tables will be saved in a unique text file.

The given code snippet demonstrates how to use the `SELECT INTO OUTFILE` query to create a backup of a table.

A MySQL query is being used in Code Snippet to backup the passengers table from database sample2DB. A string containing the name of a local backup file with the extension sql is specified and then substituted into the SQL query.

If the query is successful, the output 'Backed up data successfully' is presented. Otherwise, if an error occurs, the error message 'Error while attempting data backup: Sorry this operation is not successful' appears. The output for Code Snippet is shown in the figure.

Ask the students the following question. Wait for a response before you answer.

In-Class Question: How many ways can users interact with SQL in PHP?

Answer: There are three types of methods in PHP to connect MySQL database through backend:

- MySQL.
- MySQLi.
- PDO.

Additional Information:

Refer to following links for more information:

<https://stackoverflow.com/questions/147821/loading-sql-files-from-within-php>

<https://www.oreilly.com/library/view/php-mysql/9781449318857/ch04.html>

Slide 22

Using SQL Command Through PHP [2-2]

Code Snippet:

```
<?php
$dbhost = "localhost";
$username = "root";
$password = "root";
$conn = mysqli_connect($dbhost, $username, $password);
if(!$conn) {
    die("Could not connect: " . mysqli_error());
}
$table_name = "passengers";
$backup_file = "E:\\tmp\\passengers.sql";
$sql = "LOAD DATA INFILE '$backup_file' INTO TABLE $table_name";
mysqli_select_db($conn, "sample2DB");
$retval = mysqli_query($conn, $sql);
if(!$retval) {
    die("Encountered an error, Could not load data: " . mysqli_error());
}
echo "loaded data successfully\n";
mysqli_close($conn);
?>
```

In case data that has already been backed up must be restored, run the LOAD DATA INFILE query. Code Snippet shows the same.



Figure: Output for Code Snippet

© Aptech Limited

Architecting Web Applications using PHP/ Session 9/ 22 of 29

Show slide 22 and explain students the following:

If previously backed-up data has to be restored, use the LOAD DATA INFILE query. The same is shown in Code Snippet.

Code Snippet contains the code to restore the backed up data from the database that is now present in the local .sql file. The MySQL query LOAD DATA INFILE '\$backup file' INTO TABLE \$table name is used, with the local file name substituted.

Output 'Loaded data successfully' is printed when the query is successfully executed. The output for Code Snippet is shown in the figure.

The passengers table must be empty for the code to execute correctly, otherwise, the backup data restoration will attempt to add duplicate records into the table, which is not allowed. Prior to executing Code Snippet, make sure the table Passengers is empty.

Using MySQL Binary mysqldump Through PHP

Code Snippet:

```
<?php
$dbhost = "localhost:3036";
$username = "root";
$password = "root";
$dbname = "sample2DB";
$backup_file = $dbname. date("Y-m-d-H-i-s") .
'.gz';
$command = "mysqldump --opt -h $dbhost -u
$username -p $password ". "lct2DB | gzip >
$backup_file";
system($command);
echo "Data backed up successfully";
?>
```

Code Snippet shows an example to perform a full database dump.



Figure: Output for Code Snippet

© Aptech Limited

Architecting Web Applications using PHP/ Session 9/ 23 of 29

Show slide 23 and tell the students that in order to perform a database backup, MySQL provides the command-line tool mysqldump. The user can use this binary to do a full database dump in a single command. The given Code Snippet demonstrates how to do a full database dump.

In Code Snippet, use mysqldump to backup a database in a single command. The host, user, and password of database sample2DB are passed here, and the database data is backed up to a file. system() is a PHP function that allow users to run external commands. As mysqldump is an external command-line tool that is generally launched from the command line, the user must call system() to invoke it via PHP. The backup file created in the string backup_file will be substituted in the command string, which will then be passed to the system.

When the code in Code Snippet is run, the message 'Data backed up successfully' is shown. The output for Code Snippet is shown in figure.

Ask the students the following question. Wait for a response before you answer.

In-Class Question: Which database is best for PHP?

Answer: MySQL

Additional Information:

Refer to following links for more information:

[https://stackoverflow.com/questions/6750531/using-a-php-file-to-generate-a-mysql-dump#:~:text=php%20use%20fopen%5CMysqldump%20as,%2D%3EgetMessage\(\)%3B%20%7D%20%3F%3E](https://stackoverflow.com/questions/6750531/using-a-php-file-to-generate-a-mysql-dump#:~:text=php%20use%20fopen%5CMysqldump%20as,%2D%3EgetMessage()%3B%20%7D%20%3F%3E)

<https://gist.github.com/micc83/fe6b5609b3a280e5516e2a3e9f633675>

https://www.tutorialspoint.com/php/perform_mysql_backup_php.htm

If the phpMyAdmin user interface is available, then it is easy to take backup of the database. To do this, click the **export** link on the phpMyAdmin main page.

From the options available, choose any of the database options that user wishes to backup, he/she should go through the different SQL options, and enter the backup file name.

© Aptech Limited Architecting Web Applications using PHP/ Session 9/ 24 of 29

Show slide 24 and tell the students that it is simple to back up the database if the phpMyAdmin user interface is accessible. To do so, go to phpMyAdmin's main page and select the export link. Choose any of the database options that the user wants to backup, from the list of options available. Then, walk through the SQL options and enter the name of the backup file.

Ask the students the following question. Wait for a response before you answer.

In-Class Question: How many database are use in PHP?

Answer: PHP has support for **over 20 databases**, including the most popular commercial and open source varieties.

Additional Information:

Refer to following links for more information:

<https://www.javatpoint.com/phpmyadmin>

<https://www.phpmyadmin.net/>

<https://www.techrepublic.com/article/creating-and-managing-a-mysql-database-with-phpmyadmin/>



Selecting and Filtering Data from a MySQL Table

Adding filters to a query ensures that only the required data is displayed in the query result.

Sorting helps arrange the rows in the query result in an order which gives meaning to the data that will be used.

The `LIMIT` clause makes it easy to code multi page results or pagination with SQL and is very useful while working on large tables.

The three clauses used for these purposes are as follows:

- `WHERE`
- `ORDER BY`
- `LIMIT`

© Aptech Limited

Architecting Web Applications using PHP/ Session 9/ 25 of 29

Show slide 24 and tell the students that by adding filters to a query, one can ensure that only the required data appears in the query result. Sorting allows users to organize the rows in the query result in a way that makes sense for the data they will be using. When working with large tables, the `LIMIT` clause makes it simple to code multi-page results or pagination using SQL.

Following are the three clauses that are used for these purposes:

- `WHERE`
- `ORDER BY`
- `LIMIT`

Ask the students the following question. Wait for a response before you answer.

In-Class Question: What is the use of `ORDER BY` clause in MySQL?

Answer: MySQL `ORDER BY` is used in conjunction with the `SELECT` query to sort data in an orderly manner. The MySQL `ORDER BY` clause is used to **sort the query result sets in either ascending or descending order**.

Additional Information:

Refer to following links for more information:

<https://www.sqlshack.com/learn-mysql-sorting-and-filtering-data-in-a-table/>

<https://stackoverflow.com/questions/28854671/how-to-filter-data-from-a-mysql-database-table-with-php>

Slide 26

The screenshot shows a browser window with the URL 'http://localhost/test/index.php'. The page content is a simple HTML structure with a title 'WHERE Clause' and a 'Code Snippet' section. The code snippet is a PHP script that connects to a MySQL database named 'sample208', selects rows where the last name is 'Shepard', and then loops through the results to print each row's ID and name. The output of the code is displayed in the browser, showing a single row: 'Passenger Id: 102 - Name: Hugh Sheperd'.

Figure: Output for Code Snippet

Show slide 26 and tell the students that in a MySQL query, the WHERE clause is used to filter records and hold a primary key in a table. When a specific field of data is required to be retrieved, another field name with a table's value is provided. This guarantees that the database can search for and select the row element that it must fetch. The WHERE clause extracts records only if they fulfil certain criteria.

Following is the syntax used for WHERE clause:

Syntax:

```
SELECT column_name(s) FROM table_name WHERE column_name operator  
value
```

The given code snippet demonstrates how to display the firstname, lastname, and id columns from the Passengers table on the page.

Using the WHERE clause in Code Snippet, just one row of a table is fetched. The first step is to create a SQL query that selects the firstname, lastname, and id columns from the Passengers table, where the last name is Morse. The next line of code executes the query and stores the data in the variable \$result. The method num_rows() then checks whether there are more than zero rows returned. If more than zero rows are returned, the function fetch_assoc() places the results into an associative array that can be looped through. The output for Code Snippet is shown in the figure.

Ask the students the following question. Wait for a response before you answer.

In-Class Question: What happens if WHERE clause is not given in query?

Answer: The WHERE clause is very useful when you want to fetch the selected rows from a table, especially when the MySQL Join is used. If the given condition does not match any record in the table, **then the query would not return any row.**

Additional Information:

Refer to following links for more information:

https://www.w3schools.com/php/php_mysql_select_where.asp

<https://www.geeksforgeeks.org/php-mysql-clause/>

For Aptech Centre Use Only

The screenshot shows a browser window with the title "Output for Code Snippet". The content of the page is a table-like structure displaying passenger information:

ID	Name
101	Derek Houston
106	Mark Mortan
102	Hugh Sheperd
104	Ben Smith
105	Ruby Stokes
103	Luke Thompson

Code Snippet:

```

<html>
<body>
<?php
$servername = "localhost";
$username = "root";
$password = "root";
$dbname = "sampleDB";
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
$sql = "SELECT id, firstname, lastname FROM Passengers ORDER BY lastname";
$result = $conn->query($sql);
//The given if condition checks if any rows are fetched or not
if ($result->num_rows > 0) {
    //output data of each row
    while($row = $result->fetch_assoc()) {
        echo "<br> ID: ". $row["id"]. " - Name: ". $row["firstname"]. " " .
        $row["lastname"] . "<br>";
    }
} else {
    echo "0 results";
}
$conn->close();
?>
</body>
</html>

```

Code Snippet shows an example that selects the Firstname, lastname, and id columns from the Passengers table and displays it on the page.

© Aptech Limited

Architecting Web Applications using PHP/ Session 9/ 27 of 29

Show slide 27 and tell the students that the ORDER BY clause can be used to shorten the extracted results in either ascending or descending order. By default, this clause can only sort the records in ascending order. The DESC keyword, on the other hand, may be used to sort results in descending order.

Following is the syntax to use the ORDER BY clause:

Syntax:

SELECT column_name(s) FROM table_name ORDER BY column_name(s)
ASC | DESC

Given Code Snippet demonstrates how to use MySQLi to get the firstname, lastname, and id columns from the Passengers table.

The ORDER BY clause of MySQL is used in Code Snippet to display the records in ascending order of rows 1, 2, and 3. Setting up the SQL query that selects the firstname, lastname, and id values from the Passengers table is the first step. The record will be ordered through the lastname column. The next line of code develops the data in the \$result variable and executes the query. The function num_rows() then, checks if the number of rows returned is more than zero.

If there are more than two rows returned, the function fetch_assoc() places all of the obtained results into an associative array. The while() loop runs through the result set, printing the firstname, lastname, and id columns' values. The output for Code Snippet is shown in figure.

Ask the students the following question. Wait for a response before you answer.

In-Class Question: How many columns can be included in the ORDER BY clause?

Answer: More than one column can be included in the ORDER BY clause.

Additional Information:

Refer to following links for more information:

https://www.w3schools.com/php/php_mysql_select_orderby.asp

<https://www.tutorialrepublic.com/php-tutorial/php-mysql-order-by-clause.php>

<https://www.geeksforgeeks.org/php-mysql-order-clause/>

Slide 28

LIMIT Clause

- When a large number of records are returned, it can impact the overall performance.
- The **LIMIT** clause is used to limit returned rows in a MySQL query to a specific number.
- The **LIMIT** clause is extremely useful while working with large tables, since it is easier to code multiple page results simultaneously.

© Aptech Limited Architecting Web Applications using PHP/ Session 9/ 28 of 29

Show slide 28 and tell the students that when a large amount of records are returned, it might slow down the performance. The **LIMIT** clause is used to limit the amount of rows returned in a MySQL query. When working with vast tables, the **LIMIT** clause comes in handy since, it makes it easy to code many page results at once.

Assume the user wishes to choose only 30 entries from a table called Orders that contains one hundred records. This is how the SQL query will be written:

```
$sql = "SELECT * FROM Orders LIMIT 30";
```

The first 30 records will be returned when this SQL query is executed.

Assume that the user wishes to choose records 16–25 (inclusive). Use **OFFSET** in this scenario. Following SQL query indicates that only 10 records should be returned, starting with record 16 (**OFFSET** 15):

```
$sql = "SELECT * FROM Orders LIMIT 10 OFFSET 15";
```

To achieve the same outcome, a shorter syntax might be adopted:

```
$sql = "SELECT * FROM Orders LIMIT 15, 10";
```

When a comma is used, notice that the numbers are reversed.

Ask the students the following question. Wait for a response before you answer.

In-Class Question: What is the use of LIMIT clause with SELECT query?

Answer: The LIMIT clause can **restrict the result set of the query to some maximum number of rows**. If this clause specifies a value smaller than the number of qualifying rows, the query returns only a subset of the rows that satisfy the selection criteria.

Additional Information:

Refer to following links for more information:

<https://www.geeksforgeeks.org/php-mysql-limit-clause/>

<https://www.tutorialrepublic.com/php-tutorial/php-mysql-limit-clause.php>

<https://www.javatpoint.com/mysql-limit>

<https://www.mysqltutorial.org/mysql-limit.aspx>

Slide 29



Summary

- MySQL is an open-source relational database system and is often used with Web applications.
- PHP `mysqli_connect()` function is used to connect with the MySQL database.
- Using PHP, a MySQL database can be created and deleted.
- The `CREATE TABLE` statement is used to create a table in MySQL and can be used in a PHP script.
- Data can be both inserted into and retrieved from a MySQL database through PHP scripts.
- The SQL `UPDATE` statement through PHP function `mysqli_query` is used to update data in to a MySQL table.
- Data can be selected from a MySQL database using the `SELECT` statement.
- The `WHERE` clause is used to filter data based on specific criteria.
- The `ORDER BY` clause is used to display extracted results either in ascending or descending order.
- The `LIMIT` clause is used to specify the numbers of returning records.
- MySQL database can be backed up in three ways, namely, using SQL Command through PHP, using MySQL binary `mysqldump` through PHP, and using phpMyAdmin user interface.

© Aptech Limited

Architecting Web Applications using PHP/ Session 9/ 29 of 29

Use slide 29 to summarize the session. You will end the session with a summary of what has been taught in the session. Tell students the pointers of the session. This will be a revision of the current session.

9.3 Post Class Activities for Faculty

You should familiarize yourself with the topics of the next session.

Tips:

You can also check the Articles/Blogs/Expert Videos uploaded on the Online Varsity site to gain additional information related to the topics covered in the next session.

Session10 – Advanced Features of PHP

10.1 Pre-Class Activities

Before beginning the session, you should fully familiarize yourself with its topics. Prepare a question or two that will be a key point to relate the current session objectives.

10.1.1 Teaching Skills

To teach this session, you should be well-versed with the concepts and advance features of PHP. You must be familiar with the procedure for setting up a local server for email sending. You must use the given images and teach the concepts in the theory class. For teaching in the class, you are expected to use slides and LCD projectors.

Tips:

It is recommended that you test the understanding of the students by asking questions in between the class.

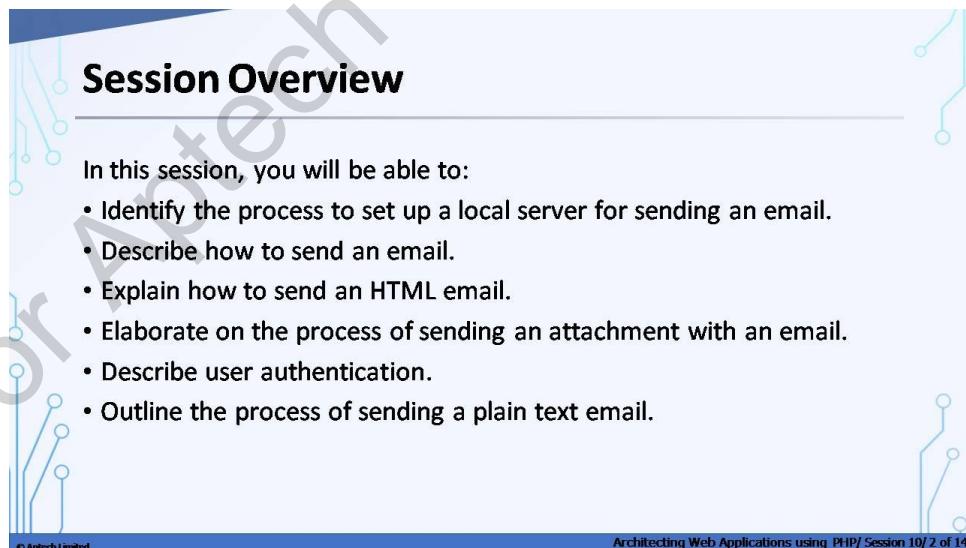
In-Class Activities

Follow the order given here during In-Class activities.

Overview of the Session

Give the students an overview of the current session in the form of session objectives. Read out the objectives on slide 2.

Slide 2



The slide has a blue header bar with the title "Session Overview". Below the title is a horizontal line. To the left and right of the slide are decorative blue and white circuit board patterns. At the bottom left is the copyright notice "© Aptech Limited". At the bottom right is the footer text "Architecting Web Applications using PHP / Session 10 / 2 of 14".

In this session, you will be able to:

- Identify the process to set up a local server for sending an email.
- Describe how to send an email.
- Explain how to send an HTML email.
- Elaborate on the process of sending an attachment with an email.
- Describe user authentication.
- Outline the process of sending a plain text email.

Show slide 2 to the students and give them a quick overview of the current session and its goals. This session will explain the procedure for setting up a local server for email sending. The session also covers topics such as how to send an email, how to send an HTML email, and how to send an email with attachments. Moreover, it introduces the students to the user authentication process in PHP.

10.2 In-Class Explanations

Slide 3

Set Up a Local Server for Email

- Simple Mail Transfer Protocol (SMTP) is a set of communication guidelines (protocol) that allow software applications to transmit an email over the Internet.
- The key objective of SMTP is used to set up communication rules between servers.
- Users can create PHP scripts to send and receive emails using SMTP. To verify and test the functionality of these scripts, users will require an SMTP server.
- One of the easiest and affordable ways to use a local SMTP server is Paper-cut SMTP.

© Aptech Limited Architecting Web Applications using PHP / Session 10 / 3 of 14

Show slide 3 and explain to students how to set up a local server for email.

- An SMTP server enables sending messages to other computer users based on email addresses.
- Users can use SMTP to send and receive emails by using PHP scripts. Users require an SMTP server to validate and test the functionality of these scripts. Paper-cut SMTP is one of the simplest and most cost-effective ways to use a local SMTP server.
- Paper-cut SMTP is a local SMTP server that is used to receive emails. Users can check verified emails in their Paper-cut boxes directly. This lightweight SMTP server can be used by beginners to test how their emails are received. They can have access to the whole contents of an email, including the headers, body, attachments, and even the raw coded data.
- It is simple to set up and configure this software, and it is completely free. To get the software, go to <https://github.com/ChangemakerStudios/Papercut-SMTP>.
- After downloading the zip file, unzip it to a convenient location and run Papercut.exe. It is now configured and ready to use. Users do not have to worry about their test emails ending up in spam folders. Users can utilize the Forward button to send test emails to certain email addresses.

Sending an Email

- In order to send an email, a user must ensure that the `php.ini` file includes the corresponding configurations for PHP.
- The file must also have precise details on how emails are managed by the system.
- Since user is using XAMPP, the file can be opened by right-clicking Config option in XAMPP Control Panel and then, selecting `php.ini`. This will automatically open the file in Notepad.
- Alternatively, if a user wants to open the file through File Explorer, navigate to the folder where PHP is installed, locate the `php.ini` file, and then, open it using any text editor.

Architecting Web Applications using PHP / Session 10 / 4 of 14

Show slide 4 and explain to students how to send an email using PHP. You must also familiarize them with its prerequisites.

- A user must edit the `php.ini` file to configure anything related to PHP. In order to configure or Send email, user will update the `php.ini` file.
- In Linux, user can quickly identify or search your `php.ini` file using the following command: `locate php.in`.
- The default location is `/etc/php.ini`.
- User can find the same in windows where XAMPP or LAMPP is installed:
- `'C:\xampp\php\php.ini'`

Clarification:

XAMPP (X (for "some OS"), Apache, MySQL, Perl, PHP)

LAMPP (Linux, Apache, MySQL, Perl, PHP)

Following is an example of how to configure SMTP in Windows:

```
[mail function] ; For Win32 only.  
SMTP = smtp.inovastudio.net ; For win32 only  
smtp_port=25  
sendmail_from = webmaster@abc.com  
However, since Papercut will be used here, all the user has to  
do is:  
SMTP=localhost  
smtp_port=25
```

Changing php.ini file to add mail configuration.

1. Open php.ini file.
2. Add email server details to the file or in case you have one can change it (mail server can be own that is local email server or you can use any ESP as a mail server).
3. Check for `SMTP = localhost` and change it to your desired mail server (any ESP or localhost) no changes are required if users are using your own local server.
4. Save/close the php.ini file.
5. The final step, do not forget to restart webserver/php-fpm.

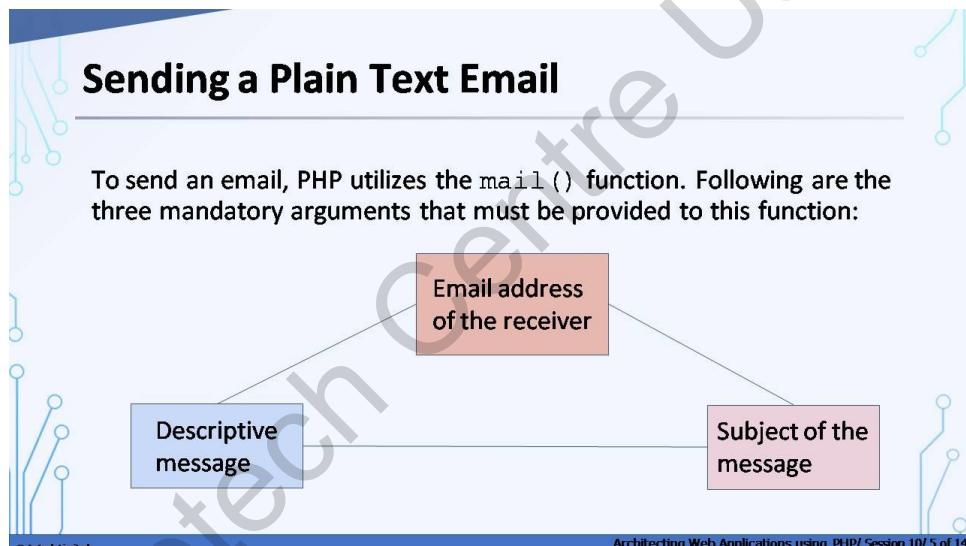
Additional Information

Refer to following links for more information:

<https://www.phpflow.com/php/how-to-send-email-from-localhost-using-php/>

<https://netcorecloud.com/tutorials/send-an-email-via-gmail-smtp-server-using-php/>

Slide 5



Show slide 5 and explain students the three mandatory arguments for an email:

- **subject**
String | Required
The subject line of the email to be sent.
For Example: Regarding leave approval.
- **Message**
Required | String
The content of the email user would like to send. A CRLF (rn) should be used to divide each line of the email, and each line should not exceed 70 characters.
When PHP connects to an SMTP server to send mail on a Windows PC, it removes the full stop located at the beginning of a line. To fix this, use the following code to replace the dot with a double dot:
`<?php $text = str_replace("\n.", "\n..", $text); ?>`

- **to**
String | Required
The email address of the recipient.
For Example, user1@example.com, user2@example.com, user3@example.com, and so on.
- **headers**
String | Optional
This is a non-necessary argument that adds additional headers. Make sure to use a Carriage Return Line Feed (CRLF) to separate each heading (\r\n).
For instance, from, cc, and bcc.
- **Parameters**
String | Optional
This is also an optional argument that specifies whether or not to supply any parameters to the `sendmail()` programme.

The diagram features a central yellow triangle containing three text boxes. The top box states: "Syntax of mail() function is as follows and this includes the optional argument too." The middle box contains the PHP code: "mail(to, subject, message, headers, parameters);". The bottom box says: "When mail() function is called, PHP tries to send out the email." The background of the slide has a decorative pattern of blue lines and circles.

Syntax of mail() Function

Syntax of mail() function is as follows and this includes the optional argument too.

```
mail(to, subject,  
message, headers,  
parameters);
```

When mail() function is called, PHP tries to send out the email.

© Aptech Limited Architecting Web Applications using PHP / Session 10 / 6 of 14

Show slide 6 and explain students the syntax of `mail()` function.

The syntax of `mail()` function is given as:

```
mail(to, subject, message, headers, parameters);
```

It is mandatory to follow this syntax while writing or sending an email.

Following code illustrates how the `mail()` function works:

```
<html>  
<head>  
<title>Sending email using PHP</title>  
</head>  
<body>  
<?php  
$to = "abc@mydomain.com";  
$subject = "This is subject";  
$msg = "This is simple text message in PHP. . .";  
$header = "From:afgh@mydomain.com \r\n";  
$retval = mail ($to, $subject, $msg, $header);  
if( $retval == true )  
{  
echo "Message successfully sent. . .";  
}  
else  
{  
echo "Message could not be sent . . .";  
}  
?>  
</body>  
</html>
```

The code sends an email message from sender idafgh@mydomain.com to abc@mydomain.com. PHP attempts to send an email when the `mail()` function is used. If the email is successfully sent, the method returns true; otherwise, it returns false. A suitable message is displayed to the user based on the return value.

Please refer to following links for more information:

<https://www.php.net/manual/en/function.mail.php>

https://www.w3schools.com/php/func_mail.asp

Slide 7

Sending an HTML Email [1-2]

- In PHP, when a user sends a text message, it is interpreted as simple text by default. Even if the message contains HTML tags, it is treated as text.
- It is important to note that HTML tags are not formatted as per HTML syntax.
- To send such an email, the user has to ensure to input the MIME version, content type, and character set.
- In general, three different types of message formats available are namely, HTML, Plain Text, and Rich Text Formats. However, in PHP, the most common formats are HTML or plain text.

Architecting Web Applications using PHP/ Session 10 / 7 of 14

Show slide 7 and describe to students how to send an HTML email.

Following sends an email from the script with HTML message content using PHP:

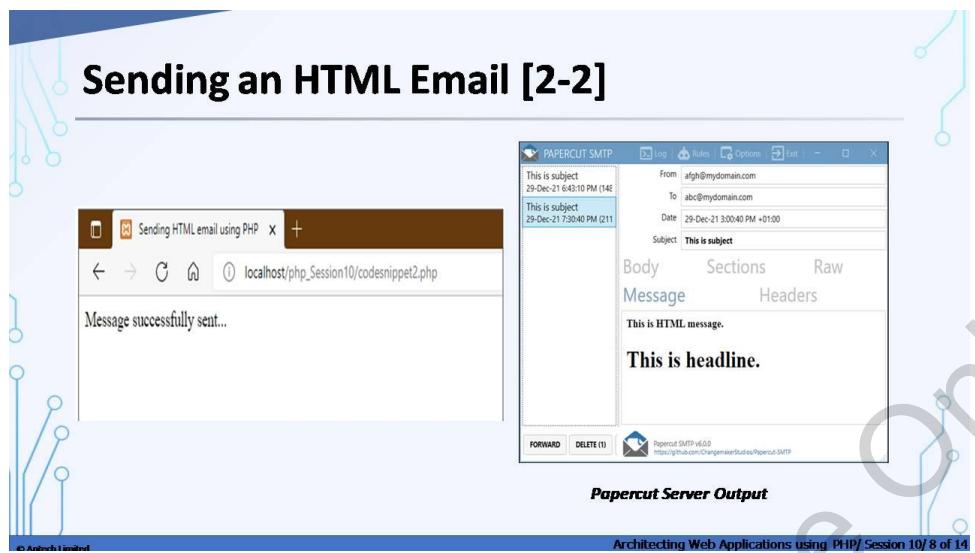
- Use the PHP `mail()` function and provide the required parameters.
to – Recipient email address.
subject – Subject of the email.
message – Message to be sent.
headers – From, Cc, Bcc, Content-type headers.
- The content-type header is mandatory for sending an HTML formatted email.
- The additional headers are used for adding From, Cc, Bcc, etc.
- \$htmlContent variable holds the HTML contents of the email.
- HTML tags are usually shown as basic text using the built-in PHP functions `htmlspecialchars()` and `htmlentities()`.
- HTML, Plain Text, and Rich Text Formats are the three basic sorts of message formats available. Mainly, HTML and plain text are the most popular forms in PHP.

An example of sending an HTML email is shown in given Code Snippet:

```
<html>
<head>
<title>Sending HTML email using PHP</title>
</head>
<body>
<?php
$to = "abc@mydomain.com";
$subject = "This is subject";
$msg = "<b>This is HTML message.</b>";
$msg .= "<h1>This is headline.</h1>";
$header = "From:abc@mydomain.com\r\n";
$header .= "MIME-Version: 1.0\r\n";
$header .= "Content-type: text/html\r\n";
$retval = mail ($to,$subject,$msg,$header);
if( $retval == true )
{
echo "Message successfully sent...";
}
else
{
echo "Message not sent...";
}
?>
</body>
</html>
```

Following code demonstrates what arguments must be supplied in an HTML email. If the senders email is successfully received by the receiver, the code will return the message ‘Message successfully sent...’; otherwise, it will return the message ‘Message not delivered...’.

Slide 8



Show slide 8 and explain how to write HTML code for emailing.

The first image on the slide depicts the result that users will receive when they run the code, it will either be successful or unsuccessful. The second image shows the user's details for sending an email, as well as the Papercut server window, which displays the received email in HTML format.

Sending Attachments with an Email [1-2]

- When a user prefers to send an email that has attachments, it is essential to set the Content-type header to multipart/mixed.
- To begin a boundary, enter two hyphens and after that enter a unique number.
- To generate the unique number, users can utilize the function md5().
- To end a boundary, which also marks the conclusion of the email, use two hyphens.

© Aptech Limited

Architecting Web Applications using PHP / Session 10 / 9 of 14

Show slide 9 and explain to students the process of sending an email with attachments.

To send an email with attachments in PHP, utilize the `mail()` function with certain MIME type headers.

MIME and Content: Type headers are used with the `mail()` function.

- `$to`: The email address of the recipient.
- `$from`: The email address of the sender.
- `$fromName`: is the name of the sender.
- `$subject`: The email's subject.
- `$file`: The relative path of the file that will be attached to the email.
- `$html`: The email's body content is referred to as the content (Text or HTML). The script allows user to send both text and HTML messages together with an attachment file to an email.

Given code demonstrates how to send an email with an attachment:

```
<html>
<head>
<title>Sending attachment using PHP</title>
</head>
<body>
<?php
$to = "Papercut@user.com";
$subject = "This is subject for the mail with attachment";
$msg = "<b>This is HTML message.</b>";
$msg .= "<h1>This is headline.</h1>";
# Open a file and read its contents into a variable.
$file = fopen( "c:\\misc\\test.c", "r" );
if( $file == false )
{
```

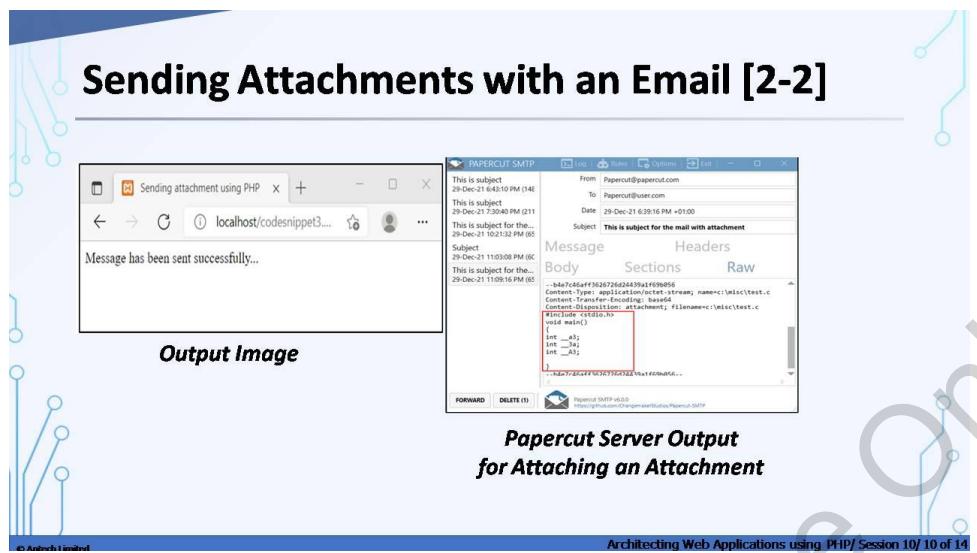
```

echo "Error in opening the file";
exit();
}
$size = filesize("c:\\\\misc\\\\test.c");
$content = fread( $file, $size);
$encoded_content = chunk_split( base64_encode($content));
# Generate a random 32-bit number by seeding time().
$num = md5( time() );
# Define the main headers.
$header = "From:Papercut@papercut.com\r\n";
$header .= "MIME-Version: 1.0\r\n";
$header .= "MIME-Version: 1.0\r\n";
$header .= "Content-type: text/html\r\n";
$header .= "Content-Disposition:attachment; ";
// header
$uid = md5(uniqid(time()));
// message & attachment
//${file_name}="c:\\\\misc\\\\test.c";
//${filename}= "test.c";
$nmessage = "--".$uid."\r\n";
$nmessage .= "Content-type:text/plain; charset=iso-8859-1\r\n";
$nmessage .= "Content-Transfer-Encoding: 7bit\r\n";
//${nmessage} .= $msg."\r\n";
$nmessage .= "--".$uid."\r\n";
$nmessage .= "Content-Type: application/octet-stream;
name=c:\\\\misc\\\\test.c\r\n";
$nmessage .= "Content-Transfer-Encoding: base64\r\n";
$nmessage .= "Content-Disposition: attachment;
filename=c:\\\\misc\\\\test.c\r\n";
$nmessage .= $content."\r\n";
$nmessage .= "--".$uid"--";
$retval = mail ( $to, $subject, $nmessage, $header);
if( $retval == true )
{
echo "Message has been sent successfully...";
}
else
{
echo "Message could not be sent...";
}
?>
</body>
</html>

```

Given code sends the contents of a file test.c as an attachment with the email. In the code user has mentioned all the headers with the unique id inserted with it. A file has been created and in that all the content is placed and for the output user has written two outputs to check whether the message is sent or not.

Slide 10



Using slide 10, explain how to send attachments with an email. The first image on the slide depicts the result that users will receive when they run the code, it will either be successful or unsuccessful.

The second image shows the user's details for sending an attachment with an email, as well as the Papercut server window, which displays the received email with attachment in HTML format.

User Authentication

User authentication is a very crucial process in any Web application to ensure that unauthorized persons do not gain access to the application.

In this process, the login data entered by the user is compared and matched with data present in the database of the server.

Using this security mechanism, it is possible to restrict unauthorized users from accessing Web tools or applications.

Architecting Web Applications using PHP/ Session 10/ 11 of 14

Show slide 11 and explain to students the concept of User Authentication in PHP.

Tell them that,

- The process of validating users via keys, tokens, or other credentials is known as user authentication.
- The authentication process will be successful if the user provides valid credentials.
- Users will be permitted access to the system as authenticated users after successful authentication.
- A login panel is included in this user authentication example to allow users to submit their login credentials. To authenticate, users must input their username and password.
- These login credentials will be sent to a PHP page if user input them. PHP compares the entered information to the database of registered users. It demonstrates how to prepare and run a database query to check and validate the user's data against the database.
- The user will be regarded as authenticated and authorized once a match is found. The authenticated user will be permitted to enter the application after a successful login.

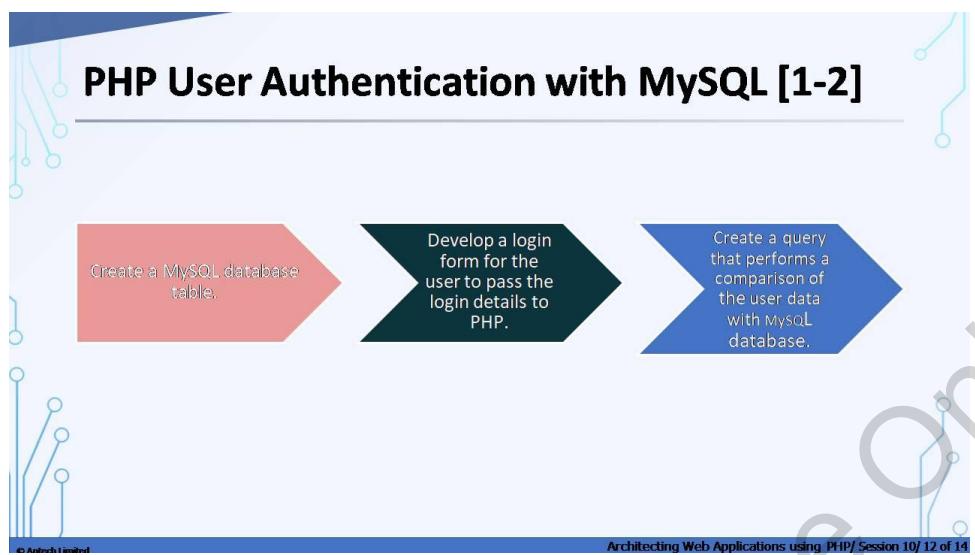
Additional Information

Refer to following links for more information:

<https://phppot.com/php/user-authentication-using-php-and-mysql/>

<https://codeshack.io/secure-login-system-php-mysql/>

<https://www.developerdrive.com/adding-a-simple-authentication-using-php-require-and-includes/>



Show slide 12 and explain to students the concept of PHP User Authorization using MySQL. The user authentication method aids in the verification of users. Validation is accomplished by the use of specific keys, tokens, or credentials. The authentication process is regarded successful if the user enters the allowed credentials else, it is considered failed. The user is granted access to the system if the process is successful.

Following code demonstrates how to display a Login Form to users. They can fill up this form with their authentication information. To render the form, the code largely consists of HTML markup:

```

<html>
<body>
<form name="frmStudent" method="post"
action="authenticate.php">
<div ></div>
<table border="0" cellpadding="10" cellspacing="1" width="500"
align="center" >
<tr >
<td align="center" colspan="2">Enter Login Details</td>
</tr>
<tr >
<td>
<input type="text" name="studentName" placeholder="Student
Name" ></td>
</tr>
<tr >
<td>
<input type="password" name="password" placeholder="Password"
></td>
</tr>
<tr >

```

```

<td align="center" colspan="2"><input type="submit"
name="submit" value="Submit" ></td>
</tr>
</table>
</form>
</body>
</html>

```

A form with two input fields is displayed by the code. The first is for the student's name and the second is for the password. When the user fills out both fields and submits the form, the information is sent to PHP for authentication. This section of code has not yet been written. Data from the MySQL database will be used for authentication. User inputs are compared to database values once the user submits login information in the form.

Following code shows how to generate a query for this comparison operation in a PHP script:

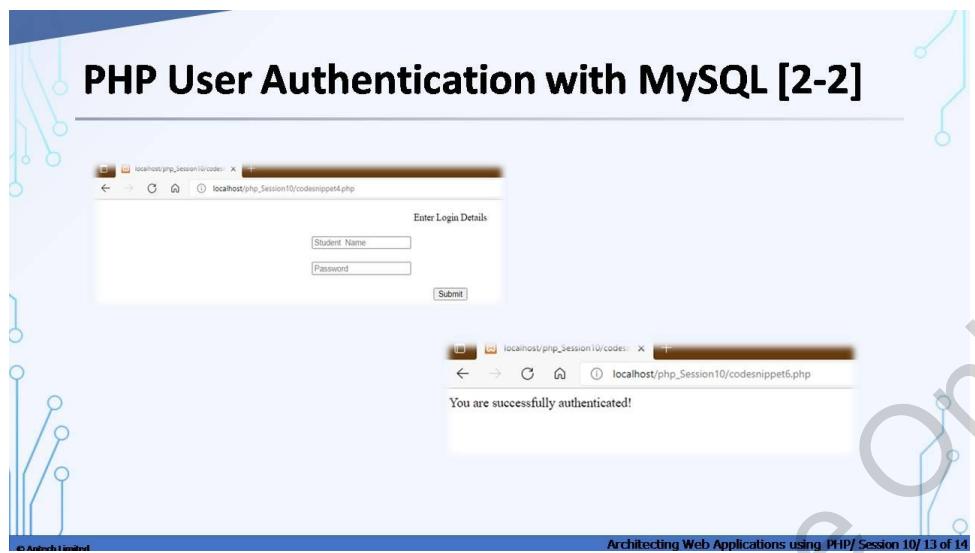
```

<?php
if(count($_POST)>0) {
$conn = mysqli_connect("localhost","root","","login");
$str= "SELECT * FROM student WHERE studentname='".
$_POST["studentName"] . "' and password = '".
$_POST["password"]."'";
$count = mysqli_num_rows(mysqli_query($conn, $str));
echo $count;
if($count==0) {
$message = "Invalid Studentname or Password!";
} else {
$message = "You are successfully authenticated!";
}
echo $message;
}
?>

```

To ensure that a POST action has occurred, the code checks the length of the `$_POST` global array. The user input data is stored in this array and is acquired after the authentication form is submitted. The code then establishes a connection to the MySQL database by giving the necessary credentials, such as the MySQL username and password. Following that, the code creates a query to find the record whose student name and password match the information entered in the form. The user has been authenticated successfully. On the browser screen, an appropriate notice confirming this is displayed.

Slide 13



Show slide 13 and explain how to send attachments with an email.

The first image on the slide depicts the result that users will receive when they run the code, it will either be successful or unsuccessful and showing form on screen.

The second image shows the user's details for confirming that user is successfully authenticated.

In-Class Question: Suppose you receive a form submitted by a post to subscribe to a newsletter. This form has only one field, an input text field named email. How would you validate whether the field is empty?

Print a message 'The email cannot be empty' in this case.

Answer: <?php
if(empty(\$_POST['email']))
{ echo "The email cannot be empty"; }
?>

Summary

- The SMTP server assists in sending emails to the users. Email application can be set using this server to send an email.
- In order to send an email, a user must ensure that the `php.ini` file includes the corresponding configurations for PHP and the details about how the user's system manages mails.
- To send an email, PHP utilizes the `mail()` function. The three arguments that must be provided to this function are the receiver's email address, the subject of the message, and the descriptive message.
- In PHP, when a user sends a text message, it is interpreted as simple text. Even if the message contains HTML tags, it is treated as text.
- To send an HTML message, the user has to ensure to input the MIME version, content type, and character set.
- When a user prefers to send an email that has attachments, it is essential to set the Content-type header to `multipart/mixed`. Then, the user can indicate the text section and attachment section within boundaries.
- User authentication is a process that helps to verify and validate users. Validation is done using certain keys, tokens, or credentials.

Architecting Web Applications using PHP / Session 10 / 14 of 14

Use slide 14 to summarize the session. You will end the session, with a brief summary of what has been taught in the session. Tell the students pointers of the session. This will be a revision of the current session.

10.3 Post Class Activities for Faculty

You should familiarize yourself with the topics of the next session.

Tips:

You can also check the Articles/Blogs/Expert Videos uploaded on the Online Varsity site to gain additional information related to the topics covered in the next session.

Session 11 – File Handling and Exception Handling in PHP

11.1 Pre-Class Activities

Before beginning the session, you should fully familiarize yourself with its topics. Prepare a question or two that will be a key point to relate the current session objectives.

11.2 Teaching Skills

To teach this session, you should be well-versed with the concept of File Handling and Exception Handling in PHP.

You must use the given images and teach the concepts in the theory class. For teaching in the class, you are expected to use slides and LCD projectors.

Tips:

It is recommended that you test the understanding of the students by asking questions in between the class.

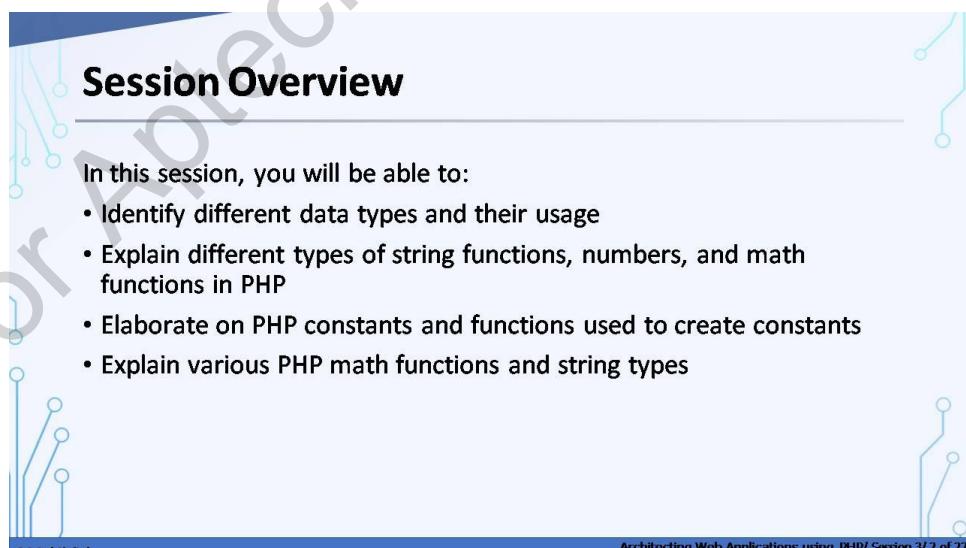
In-Class Activities

Follow the order given here during In-Class activities.

Overview of the Session

Give the students an overview of the current session in the form of session objectives. Read out the objectives on slide 2.

Slide 2



The slide has a light blue background with a decorative border featuring stylized blue circuit board patterns. At the top center, the title "Session Overview" is displayed in bold black font. Below the title, a horizontal line separates it from the main content area. The content area contains the following text and list:

In this session, you will be able to:

- Identify different data types and their usage
- Explain different types of string functions, numbers, and math functions in PHP
- Elaborate on PHP constants and functions used to create constants
- Explain various PHP math functions and string types

At the bottom left, there is a small copyright notice: "© Aptech Limited". At the bottom right, there is a footer bar with the text "Architecting Web Applications using PHP / Session 3 / 2 of 22".

Show slide 2 and provide a brief outline of the current session. Tell students that the session begins with an introduction to file handling and uploading in PHP using the `fwrite()` and `fopen()` methods.

The session also provides an overview of making an upload form and an upload script. Additionally, it covers manipulation of files in PHP. Then, it goes on to explain how to open, read, and close files in PHP using methods like `fgets()`, `fread()`, `fopen()`, `fgetc()`, and `feof()`.

Furthermore, the session covers handling exceptions and errors in PHP. Finally, it covers the `die()` function, the `try...catch...finally` statement, the `try...catch` statement, and the exception object.

Slide 3

File Handling [1-2]

PHP Write File - `fwrite()`

`fwrite()` function helps a user to write some content of string(s) into a file

Syntax:
int `fwrite` (resource `$handle`, string `$string` [, int `$length`])

© Aptech Limited Architecting Web Applications using PHP / Session 11 / 3 of 20

Show slide 3 and explain the concept of file handling in PHP. File management in PHP is comparable to how it is done in other computer languages such as C, C++, and more. There are numerous functions in PHP that are used to work with regular files.

Syntax:

`int fwrite (resource $handle, string $string [, int $length])`
Elaborate on the functions given in the slide.

- **PHP Write File - `fwrite()`** : Using this function, a new file can be created or text can be appended to an existing file. The `fwrite()` function takes two arguments: a file pointer and the text to be written to the file. It can have an optional third argument specifying the length of the text to be written.

For example,

```
<?php  
$file = fopen("test.txt", 'w');  
$text = "Hello PHP\n";  
fwrite($file, $text);  
?>
```

This example demonstrates the usage of `fwrite()` function in a PHP program by using `fopen()` function. The user has created a new text file as test.txt. This file writes 'HELLO PHP' with the help of `fwrite()` function.

***Note** – Explanation continues in the next slide.

File Handling [2-2]

**PHP Delete File -
unlink()**

Users can delete a file using the PHP unlink() function

Syntax:
bool unlink (string \$filename [, resource \$context])

© Aptech Limited Architecting Web Applications using PHP / Session 11 / 4 of 20

Show slide 4 and introduce them to the unlink() function.

- **unlink ()**: The unlink() method is a built-in PHP function used to delete a file. The filename to be removed is sent as a parameter and the method returns true if it succeeds. Otherwise, it returns false if it fails. The unlink() method takes two parameters.

Syntax:

```
unlink( filename, context )
```

Describe the parameters accepted by the function in detail.

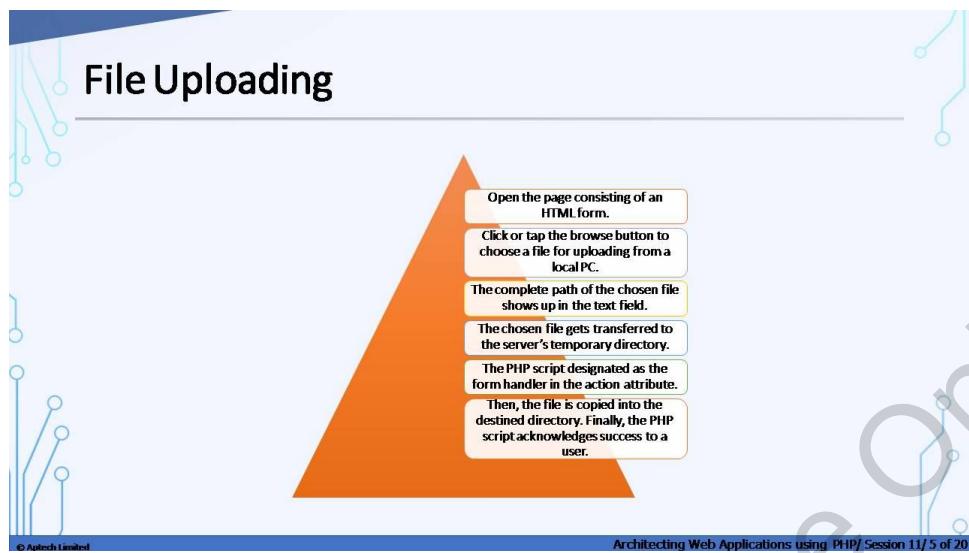
filename: This is a required argument that gives the filename of the file to be removed.

context: This is an optional argument that provides the file handle's context, which can be used to change the stream's nature.

Here is an example:

```
?php  
$file_pointer = fopen('test.txt');  
fwrite($file_pointer, 'Testing');  
fclose($file_pointer);  
unlink('test.txt');  
?>
```

The code opens a test.txt file. The fwrite() function writes 'Testing' into the file. The unlink() function gets the result 1.



Show slide 5 and explain the concept of File Uploading in PHP. To allow the users to upload files to a server, a PHP script can be combined with an HTML form.

Initially, files are uploaded to a temporary directory. Then, a PHP script moves these files to their final destination. The temporary directory used for file uploads is described as `upload_tmp_dir` on the `phpinfo.php` page. The `upload_max_size` defines the maximum allowed size that can be uploaded. These options are specified in the `php.ini` configuration file.

Elaborate on the steps given in the slide.

1. The user navigates to a page with an HTML form, text files, a browse button, and a submit button.
2. The user selects a file to upload from their local computer by clicking the browse button.
3. When the user clicks the submit button, the entire path to the selected file appears in the text field.
4. The specified file is saved to the server's temporary directory.
5. The PHP script supplied as the form handler in the form's action attribute verifies that the file has arrived before copying it to the desired location.
6. The PHP script informs the user that the operation was successful.

When writing files, both the temporary and final locations must have permissions that allow file writing, as is customary. The process will fail if either is set to read-only. A text file, an image file, or any other document can be uploaded.



Upload Form Creation

Using HTML form, PHP code allows users to upload files to the server. In the beginning, files are placed in a temporary directory. Then, a PHP script is used to move the files to a target destination.

The HTML code given in Code creates an uploaded form. The form has the enctype attribute set to multipart/form-data and the method attribute assigned as a post.

© Aptech Limited

Architecting Web Applications using PHP / Session 11 / 6 of 20

Show slide 6 and explain the process of creating an upload form.

An example HTML code to create an uploader form is given. The method attribute on this form is set to post, and the enctype attribute is set to multipart/form-data.

- **Uploadform.html**

```
<form action="uploader.php" method="post" enctype="multipart/form-data">
Select File:
<input type="file" name="TestUpload"/>
<input type="submit" value="Upload Image" name="submit"/>
</form>
```

- **Uploader.php**

```
<?php
$target_path = "e:/";
$target_path = $target_path.basename( $_FILES['TestUpload']['name']);
if(move_uploaded_file($_FILES['TestUpload']['tmp_name'], $target_path)) {
echo "File uploaded successfully!";
}
else{
echo "Sorry, file not uploaded, please try again!";
}
?>
```

The two snippets demonstrate how a form should be created. They also show how an image and file is uploaded.

Upload Script Creation

- `$_FILES['file']['error']` - Error code related to file upload.
- `$_FILES['file']['type']` - Multipurpose Internet Mail Extensions (MIME) type of the uploaded file.
- `$_FILES['file']['name']` - Real name of the uploaded file.
- `$_FILES['file']['size']` - Size of the uploaded file in bytes.
- `$_FILES['file']['tmp_name']` - Uploaded file located in the temporary directory on a Web server.

© Aptech Limited Architecting Web Applications using PHP / Session 11 / 7 of 20

Show slide 7 and explain that `$_FILES` is a globally defined PHP variable. This variable is a double-dimensional array that stores all information about the uploaded file. Hence, PHP would generate five variables if the value assigned to the name attribute of the input while uploading the form was file.

Discuss these variables (also given in the slide).

1. `$_FILES['file']['tmp_name']` - the uploaded file in the Web server's temporary directory.
2. `$_FILES['file']` - The real name of the uploaded file is ['name'].
3. `$_FILES['file']['size']` - the uploaded file's size in bytes.
4. `$_FILES['file']` - The MIME type of the uploaded file is specified by ['type'].
5. `$_FILES['file']` - The error code associated with this file upload is ['error'].

Following is an example of upload script creation:

```
<?php
$target_dir = "uploads/";
$target_file = $target_dir .
basename($_FILES["TestFile"]["name"]);
$uploadOk = 1;
$imageFileType
= strtolower(pathinfo($target_file,PATHINFO_EXTENSION));
// Check if image file is an actual image or fake image
if(isset($_POST["submit"])){
$ccheck = getimagesize($_FILES["TestFile"]["tmp_name"]);
if($ccheck !== false) {
echo "File is an image - " . $ccheck["mime"] . ".";
$uploadOk = 1;
}
else
{
echo "File is not an image.";
$uploadOk = 0;
}
?>
```

Explanation of the PHP script is as follows:

- \$target dir = "uploads/" - defines the destination directory for the file.
- The path to the file to be uploaded is specified by the \$target file.
- \$uploadOk=1 is yet to be utilized (will be used later). The file extension is stored in \$imageFileType (in lower case). Next, it is determined whether the image file is genuine or not.

Slide 8

Manipulating Files [1-3]

- PHP has multiple functions for reading, editing, creating, and uploading files. However, it is important to be cautious while manipulating files. A wrong move can prove to be detrimental to the computer system.
- Following are common errors that occur during file manipulation:
 - Deleting a file's content
 - Filling a hard disk drive with garbage data
 - Editing the wrong file

© Aptech Limited Architecting Web Applications using PHP / Session 11 / 8 of 20

Show slide 8 to the students and inform them that PHP has several functions for creating, reading, uploading, and editing files.

PHP file handling functions are used to manipulate files in various ways. These functions could be used to build features such as custom error logging, store cached files, and so on in applications.

When working with files, user must use extreme caution. If user makes a mistake, you can cause a lot of harm. Editing the wrong file, loading a hard disc with trash material, and deleting the content of a file by accident are all common mistakes.

Inform students that it is important to be careful while manipulating files. A wrong move can cause harm to the computer system.

Manipulating Files [2-3]

Open File - fopen() PHP's fopen() function helps us to open files. Syntax: resource fopen (string \$filename, string \$mode [, bool \$use_include_path = false [, resource \$context]])
Read File - fread() The fread() function in PHP enables users to read the content of an open file. Syntax: string fread(resource \$handle, int \$length)
Close File - fclose() The fclose() function helps a user to close an opened. Syntax: bool fclose (resource \$handle)

© Aptech Limited Architecting Web Applications using PHP / Session 11 / 9 of 20

Show slide 9 and explain various file manipulation functions to the students.

- **fopen () :** It is a PHP function that opens a file. The fopen () function provides much more options as compared to the readfile () function.

Describe the syntax of the fopen () function to the students. The name of the file to be opened is the first parameter of fopen () . The second parameter defines the mode in which the file should be opened. If the fopen () function fails to open the supplied file, the following example displays a message:

```
<?php
$myfile = fopen("test.txt", "r") or die("Unable to open file!");
echo fread($myfile,filesize("test.txt"));
fclose($myfile);
?>
```

If the code is working fine, it will display the content inside the text file. Otherwise, it displays “Unable to open file!” message.

- **fread () :** The fread () function reads from an open file. Explain the given syntax of this function. The first parameter of fread () contains the name of the target file. The second parameter specifies the maximum number of bytes to read.

The following PHP code reads the “test.txt” file to the end:

```
fread($myfile,filesize("test.txt"));
```

- **fclose () :** The fclose () method enables a user to close an open file. Closing all the open files after a user has finished working on them is a good programming practice. Unnecessary open files hog precious system resources on a server.

The function’s parameter consists of the variable that holds the name of the file to be closed.

For example:

```
<?php
$myfile = fopen("test.txt", "r");
fclose($myfile);
?>
```

Manipulating Files [3-3]

Read Single Line - <code>fgets()</code> PHP's <code>fgets()</code> function allows a user to read a single line from an open file.
Check End-Of-File - <code>feof()</code> PHP function <code>feof()</code> helps a user to loop through data of undisclosed length. It examines a file and verifies whether the user has reached the 'End-of-File' (EOF).
Read Single Character - <code>fgetc()</code> PHP's <code>fgetc()</code> function serves as a means to read a single character from the files.

© Aptech Limited Architecting Web Applications using PHP / Session 11 / 10 of 20

Show slide 10 and continue to explain various file manipulation functions to the students.

- `fgets()`: The `fgets()` function is used to read a single line from a file.

Following example outputs the first line of the "test.txt" file:

```
<?php
$myfile = fopen("test.txt", "r") or die("Unable to open
file!");
echo fgets($myfile);
fclose($myfile);
?>
```

The file pointer has changed to the next line after calling the `fgets()` function.

- `feof()`: The `feof()` function determines whether the "end-of-file" (EOF) has been reached. For looping through data of unknown length, the `feof()` function is useful.

Following example reads the "test.txt" file line by line until it reaches the end:

```
<?php
$myfile = fopen("test.txt", "r") or die("Unable to open
file!");
while(!feof($myfile)) {
echo fgets($myfile) . "<br>";
}
fclose($myfile);
?>
```

- `fgetc()`: To read a single character from a file, the `fgetc()` function is used.

Following example reads the "test.txt" file character by character until it reaches the end:

```
<?php
$myfile = fopen("test.txt", "r") or die("Unable to open
file!");
while(!feof($myfile)) {
echo fgetc($myfile);
}
```

```
fclose($myfile);  
?>
```

Additional Information:

Refer to following links for more information:

<https://www.cs.utexas.edu/~mitra/csFall2015/cs329/lectures/phpFile.html>

https://www.w3schools.com/PHP/php_file.asp

<https://www.edureka.co/blog/file-handling-in-php/>

Slide 11



PHP Files

Reading File- fopen() and fread(): Users can read a file's content using a function termed as `fread()`, after it has been opened through the `fopen()` function. The `fread()` function requires two arguments.

Writing to File- fwrite(): Users can use the `fwrite()` function to append text to an existing file or write a new file. This PHP function requires two arguments. One specifies the string of data to be written and the other specifies a file pointer.

Architecting Web Applications using PHP/ Session 11/ 11 of 20

Show slide 11 and tell the students about PHP Files. Inform them that a PHP file is a Web page made up of PHP code. These PHP files can be read, written to, and updated. PHP functions are used to carry out these tasks.

- **File Reading: `fopen()` and `fread()`** After a file has been opened with the `fopen()` method, users can use the `fread()` function to read its contents. Two arguments are required for the `fread()` method - the file pointer and the file length (specified in bytes). The `filesize()` function assists users in determining the length of a file. It returns the file size (expressed in bytes).

Further, elaborate on the steps for reading a file in PHP. These are as follows:

1. Use the `fopen()` function to open a file.
2. Use the `filesize()` method to find out the length of the file.
3. To read the contents of a file, use the `fread()` function.
4. Use the `fclose()` method to close a file.

Writing to a file: `fwrite()` is a command that allows a user to write to a file. The `fwrite()` method can be used to append text to a file or create a new one. Two arguments are required for this PHP function. The first indicates the data string to be written, while the second defines the file pointer.

Using a third optional integer argument, the length of the data to be written can also be specified. If the third parameter is specified, the writing operation will get terminated after the specified length has been reached.

Error and Exception Handling in PHP

The slide features a blue header bar with the title 'Error and Exception Handling in PHP'. Below the title is a large yellow triangle pointing downwards. To the left of the triangle is a callout box containing the text: 'Error handling can be defined as the process of detecting errors in a program and undertaking appropriate measures to get rid of them.' To the right of the triangle is another callout box containing the text: 'die() Function: Users must evaluate any error conditions while writing PHP code. The purpose of the die() function is similar to that of an exit function.' At the bottom of the slide, there is a footer bar with the text 'Architecting Web Applications using PHP | Session 11 / 12 of 20'.

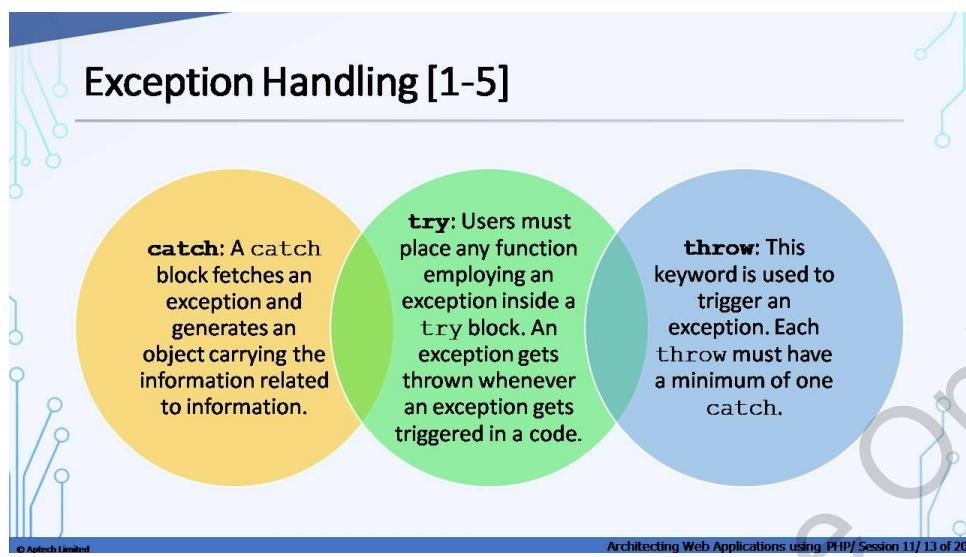
Show slide 12 and explain that error handling is the process of detecting and responding to errors in a program. Ignoring likely errors can cause a lot of unwanted implications. Error handling in PHP is fairly straightforward.

- Using the `die()` function: A user must check for all possible errors and take appropriate action if necessary. The `die()` function can be used to stop the program if a user foresees possible errors.

For example:

```
<?php  
if(!file_exists("/downloads/cmp.txt")) {  
die("File unavailable");  
}else {  
$file = fopen("/downloads/cmp.txt", "r");  
print "Opened your file successfully";  
}  
?>
```

An error message – ‘The file is currently unavailable’ - appears if the code is run without the `/downloads/cmp.txt` file. This enables a user to write code that is both efficient and effective. Furthermore, this feature aids in the display of a far more user-friendly and meaningful message.



Show slide 13 and inform the students that an exception is an object that specifies a PHP script's error or unexpected behavior. Many PHP functions and classes throw exceptions. Exceptions can also be thrown by user-defined functions and classes.

A user can use the try...catch statement to catch exceptions and continue the procedure, avoiding the errors.

Syntax:

```
try {
code that can throw exceptions
}
catch(Exception $e) {
code that runs when an exception is caught
}
```

- A user-defined function or method can throw an exception using the throw statement.
When an exception is thrown, the code that follows it is not run.

For example:

```
<?php
function divide($dividend, $divisor) {
if($divisor == 0) {
throw new Exception("Division by zero");
}
return $dividend / $divisor;
}
try {
echo divide(5, 0);
} catch(Exception $e) {
echo "Unable to divide.";
}
?>
```

Exception Handling [2-5]

In a `catch` block, exceptions can be thrown or re-thrown.

Each `try` block must have a minimum of one matching `catch` block. Users can utilize multiple `catch` blocks for catching multiple classes of exceptions.

An exception can be caught and thrown within PHP. The code that can successfully catch exceptions might be included in a `try` block.

© Aptech Limited Architecting Web Applications using PHP/ Session 11/ 14 of 20

Show slide 14 and elaborate on the given pointers.

In-Class Question: Which PHP version provides Exception handling feature?

Answer: PHP 5 and higher

Additional Information

Refer to following links for more information:

<https://www.geeksforgeeks.org/exception-handling-in-php/>

<https://www.section.io/engineering-education/php-exception-handling/>



Exception Handling [3-5]

- **try...catch Statement**
Users can implement the try...catch statement for catching exceptions and keep the process going.
- **Syntax:**

```
try {  
    code that may cause exceptions  
}  
catch(Exception $e)  
{  
    code that gets executed after catching an exception  
}
```

Architecting Web Applications using PHP/ Session 11/ 15 of 20

Show slide 15 and explains the syntax of try and catch statement. The user can use the try...catch statement to catch exceptions and continue the procedure, avoiding the error.

Syntax:

```
try {  
    code that can throw exceptions  
}  
catch(Exception $e) {  
    code that runs when an exception is caught  
}
```

If there is any exception caught in the program, then a try block can be used to throw an exception. The exception will go to catch block after getting caught.

In-Class Question: What is the intent of try block?

Answer: It introduce block of code in which exception can arise.



Exception Handling [4-5]

- try...catch...finally Statement** Users can also use the `try...catch...finally` statement to catch exceptions.
- The code written in a `finally` block will always get executed irrespective of whether an exception is caught or not. The `catch` block is optional if the `finally` block already exists and catching the exception is not a priority.

Syntax:

```

try {
    code that may cause exceptions
}
catch(Exception $e) {
    code that gets executed after catching an exception
}
finally
{
    code that always gets executed regardless of whether an exception was caught or not
}

```

Architecting Web Applications using PHP / Session 11 / 16 of 20

Show slide 16 and discuss the following pointers in detail:

- User can handle exceptions with the `try...catch` statement. The execution moves to the `catch` block when an exception occurs in the `try` block. Users can put the code that handles the exception in the `catch` block.
- PHP follows the same exception model as other programming languages. Within PHP, an exception can be thrown and caught. To make catching probable exceptions easier, code can be enclosed in a `try` block.
- At least one `catch` or `finally` block is required for each `try`. If an exception is raised with no `catch` block in its current function scope, the exception will ‘bubble up’ the call stack to the calling function until it finds one.
- All of the `final` blocks it encounters will be executed. Unless a global exception handler is present, the program will terminate with a fatal error; if the call stack is unwound to the global scope without meeting a matching `catch` block.
- After or instead of `catch` blocks, a `finally` block can be given. Regardless of whether an exception has been thrown, code in the `finally` block will always be run after the `try` and `catch` blocks and before normal execution resumes.

Syntax:

```

<?php
function inverse($x) {
if (!$x) {
throw new Exception('Division by zero.');
}
return 1/$x;
}
try
{
echo inverse(5) . "\n";
} catch (Exception $e) {
echo 'Caught exception: ', $e->getMessage(), "\n";
} finally {
echo "First finally.\n";
}

```

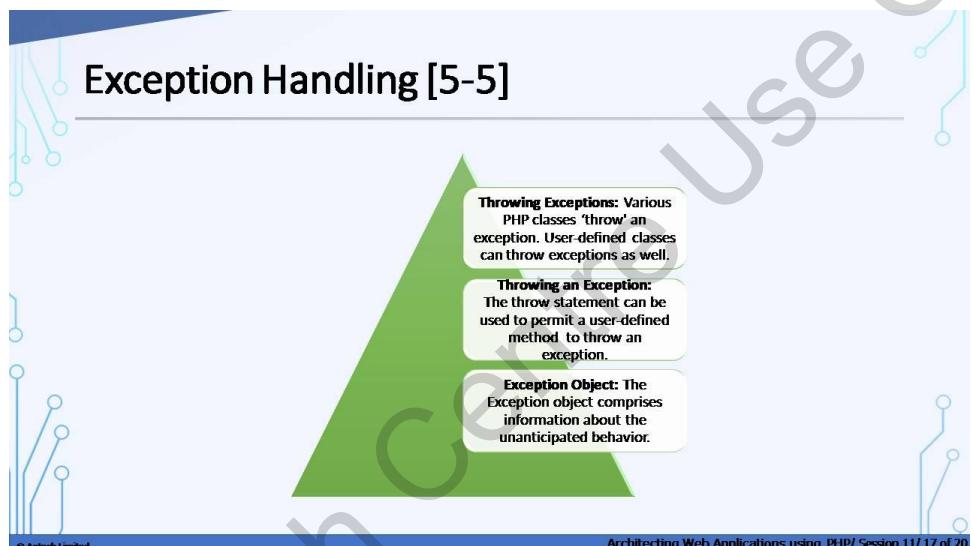
```

}
try {
echo inverse(0) . "\n";
} catch (Exception $e) {
echo 'Caught exception: ', $e->getMessage(), "\n";
} finally {
echo "Second finally.\n";
}
// Continue execution
echo "Hello World\n";
?>

```

Output: 0.2 First finally. Caught exception: Division by zero. Second finally. Hello World

Slide 17



Show slide 17 and elaborate on the following pointers:

- Several PHP classes can 'throw' exceptions.
- Exceptions can also be thrown by user-defined classes.
- Exception throwing: The throw statement can be used to throw an exception from a user-defined method or function. After an exception is thrown, the code that follows the statement will be terminated.

For example,

```

<html>
<body>
<?php
function division($multiplicand, $multiplier) {
if($multiplier == 0) {
throw new Exception("Multiply by Zero");
}
return $multiplicand / $multiplier;
}
echo division(5, 0);
?>
</body>

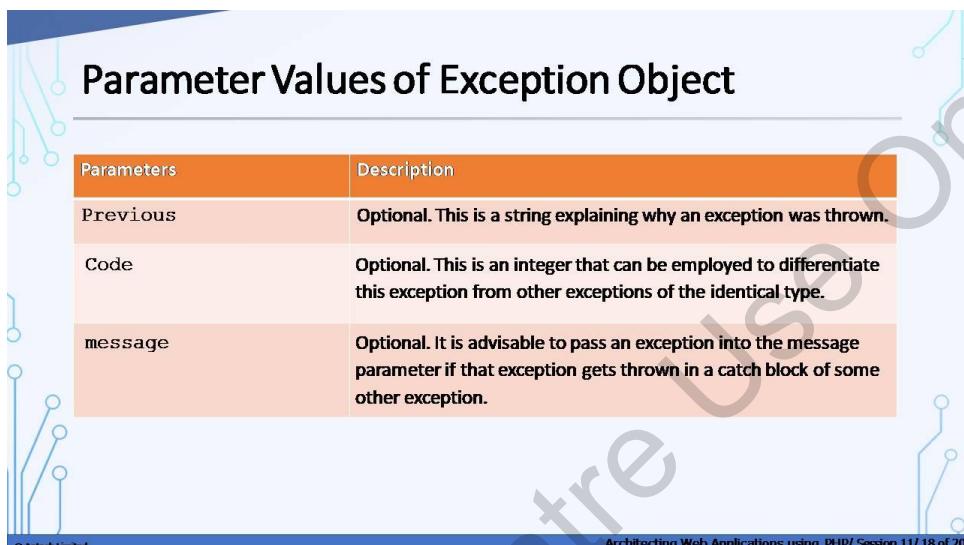
```

```
</html>
```

In this code, there is an attempt to throw an exception without trying to catch it. A throw block is used to throw an exception in this code. There is, however, no catch block in place to catch the exception. As a result, a fatal error occurs.

- Object with an Exception: The `Exception` object contains information about the unexpected behavior or error that a function encountered.

Slide 18



The slide has a decorative background featuring blue and green circuit board patterns on either side of a central white column.

Parameter Values of Exception Object	
Parameters	Description
Previous	Optional. This is a string explaining why an exception was thrown.
Code	Optional. This is an integer that can be employed to differentiate this exception from other exceptions of the identical type.
message	Optional. It is advisable to pass an exception into the message parameter if that exception gets thrown in a catch block of some other exception.

© Aptech Limited Architecting Web Applications using PHP/ Session 11/ 18 of 20

Show slide 18 and elaborate on the Parameters Values of Exception object (given in slide). To create an Exception object and set some of its properties, use the `Exception()` function `Object()`.

Syntax:

```
new Exception(message, code, previous)
```



Methods in Exception Handling

Method	Description
<code>getPrevious()</code>	This method returns a previous exception if the exception gets activated by another one.
<code>getFile()</code>	This method returns the complete path of a file in which the exception gets thrown.
<code>getMessage()</code>	This method returns a string explaining the reason for throwing the exception.
<code>getCode()</code>	This method returns the exception code.
<code>getLine()</code>	This method returns the line number of the code where the exception is thrown.

Architecting Web Applications using PHP/ Session 11/ 19 of 20

Show slide 19 to the students and tell them about the Methods in Exception Handling in detail.

- `getPrevious()` : The `getPrevious()` method returns the previous exception if it was triggered by another one. Otherwise, null is returned.
Syntax: `$exception->getPrevious()`
- `getFile()` : The absolute path to the file where an exception occurred is returned by the `getFile()` function.
Syntax: `$exception->getFile()`
- `getMessage()` : The `getMessage()` method returns a detailed description of the error or behavior that resulted in the exception being thrown.
Syntax: `$exception->getMessage()`
- `getCode()` : The `getCode()` method produces an integer that can be used to pinpoint the source of the error.
Syntax: `$exception->getCode()`
- `getLine()` : The line number of the line of code that threw the exception is returned by the `getLine()` method.
Syntax: `$exception->getLine()`

Additional Information

Refer to following links for more information:

<https://www.tutorialrepublic.com/php-tutorial/php-exception-handling.php>
<https://stackify.com/php-try-catch-php-exception-tutorial/>

Summary

- PHP file handling system enables users to perform tasks including writing, appending, creating, closing, and deleting files.
- PHP has various functions such as `unlink()`, `fwrite()`, `fopen()`, `fclose()`, `fread()`, `fgets()`, `fgetc()`, and so on for file handling.
- It is essential to stay cautious while manipulating files. Any mistakes can lead to errors such as editing a wrong file, deleting a file's content, and filling a hard disk with garbage data.
- Error handling in PHP is a process of identifying errors in a code and tackling those errors attentively.
- Exception handling model in PHP offers better control over errors. The main keywords for exception handling are `try`, `throw`, `catch`, and `finally`.

Use slide 20 to summarize the session. You will end the session with a summary of what has been taught in the session. Tell students the pointers of the session. This will be a revision of the current session.

11.3 Post Class Activities for Faculty

You should familiarize yourself with the topics of the next session.

Tips:

You can also check the Articles/Blogs/Expert Videos uploaded on the Online Varsity site to gain additional information related to the topics covered in the next session. You can also connect to online tutors on the Online Varsity site to ask queries related to the sessions.

Session 12 – Object-Oriented Concepts in PHP

12.1 Pre-Class Activities

Before beginning the session, you should fully familiarize yourself with its topics. Prepare a question or two that will be a key point to relate the current session objectives.

12.1.1 Teaching Skills

To teach this session, you should be well-versed with the concept of internationalization and design patterns in Java. You must also be familiar with the concept of localization.

You must use the given images and teach the concepts in the theory class. For teaching in the class, you are expected to use slides and LCD projectors.

Tips:

It is recommended that you test the understanding of the students by asking questions in between the class.

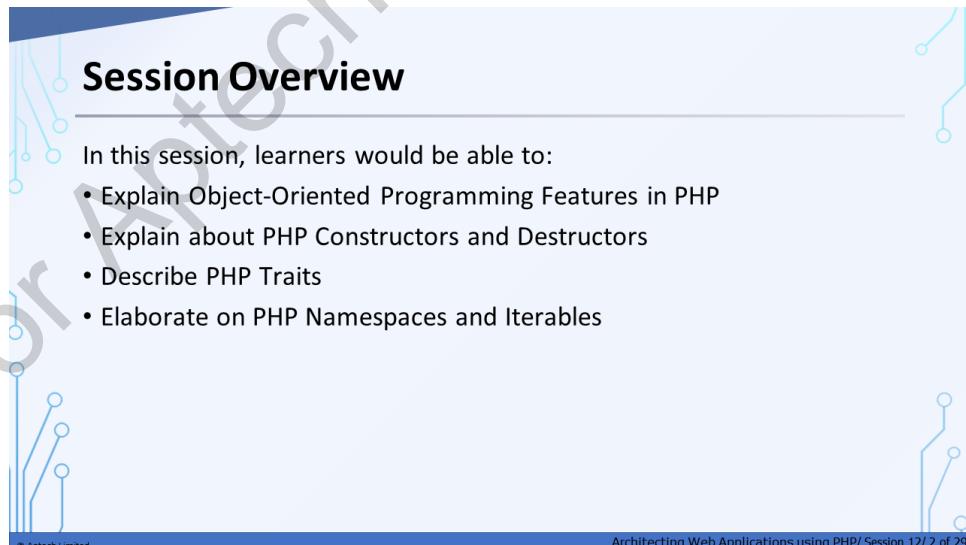
In-Class Activities

Follow the order given here during In-Class activities.

Overview of the Session

Give the students an overview of the current session in the form of session objectives. Read out the objectives on slide 2.

Slide 2



The slide has a light blue background with a decorative border featuring stylized blue lines and circles. At the top center, the title "Session Overview" is displayed in bold black font. Below the title, there is a horizontal line. To the left of the line, there is some small, illegible text. To the right of the line, the footer text "Architecting Web Applications using PHP / Session 12 / 2 of 29" is visible. The main content area contains the following text:
In this session, learners would be able to:

- Explain Object-Oriented Programming Features in PHP
- Explain about PHP Constructors and Destructors
- Describe PHP Traits
- Elaborate on PHP Namespaces and Iterables

Show slide 2 and give the students a brief overview of the current session in the form of session objectives. Inform students that the session begins with an overview of PHP's Object-Oriented Programming features. In addition, the session gives an overview of PHP

Constructors and Destructors. PHP traits, namespaces, and Iterables are also covered in this session.

In short, this session describes OOP features in PHP and explores traits and namespaces as well.

12.2 In-Class Explanations

Slide 3

The slide has a blue header bar with the title 'Classes and Objects in PHP [1-3]'. Below the title, there are two main sections: 'Class' and 'Objects'. The 'Class' section contains text about variables, constants, and functions. The 'Objects' section contains text about individual copies or instances of a class. A large watermark 'Aptech Centre Only' is diagonally across the slide.

Class
Variables (also known as properties), constants, and functions (often also known as methods) can all be found in a class and are used to accomplish one or more tasks.

Objects
Individual copies or instances of a class are referred to as objects. A single class can have an unlimited number of objects.

© Aptech Limited Architecting Web Applications using PHP/ Session 12 / 3 of 29

Show slide 3 and explain to students that unlike procedural-oriented programming which gives importance to procedure or functions, Object-Oriented Programming (OOP) gives importance to objects. An object is nothing but data (properties) and functions (methods) that are combined together. OOP is preferred over traditional, structured, or procedural programming languages because of advantages such as reusability. Classes and objects are used to implement OOP principles.

Class

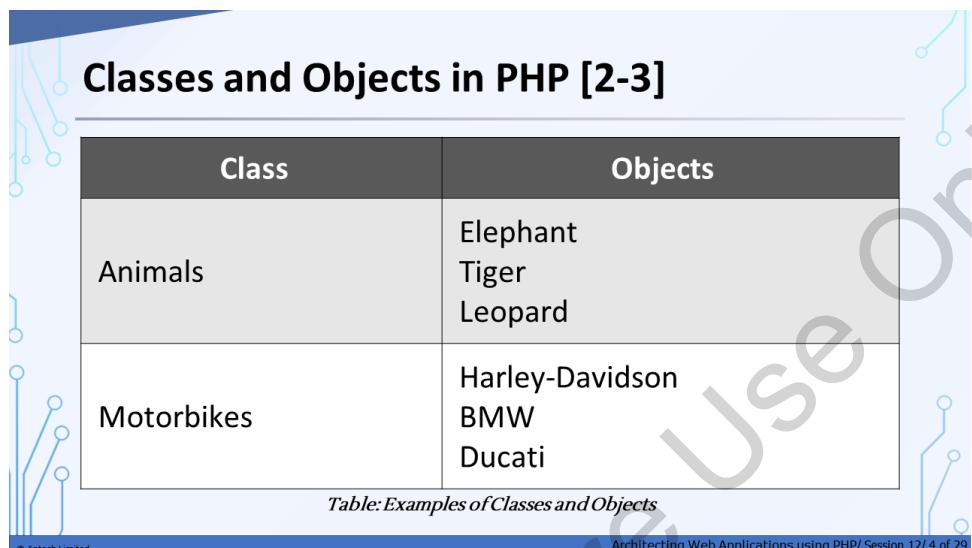
A class can contain variables (termed as properties), constants, and functions (termed as methods) that are designed to perform one or more tasks. In the real world, to build a house, a blueprint is created first. The blueprint contains clear and detailed plans for the house. By using it, any number of similar houses can be built. Similarly, a class defined once can be used to create any number of objects. Hence, a class acts as a blueprint or template for the creation of objects.

Objects

Objects are individual copies or instances of a class. Any number of objects can be created for a single class. Each object that is created will have the same type of properties and behaviors as that of a class. However, each object may have different values in the variables.

For example, once a blueprint is created, any number of copies can be created. Although the blueprint is the same for each house, there can be different paint colors, interior designs, and so on. Here, all these details are the objects.

Slide 4



Classes and Objects in PHP [2-3]

Class	Objects
Animals	Elephant Tiger Leopard
Motorbikes	Harley-Davidson BMW Ducati

Table: Examples of Classes and Objects

© Aptech Limited Architecting Web Applications using PHP / Session 12 / 4 of 29

Show slide 4 and tell students that given table lists some examples of a class and the objects within it.

Additional Information:

Refer to following links for more information:

https://www.w3schools.com/php/php_oop_classes_objects.asp

<https://www.php.net/manual/en/language.oop5.php>

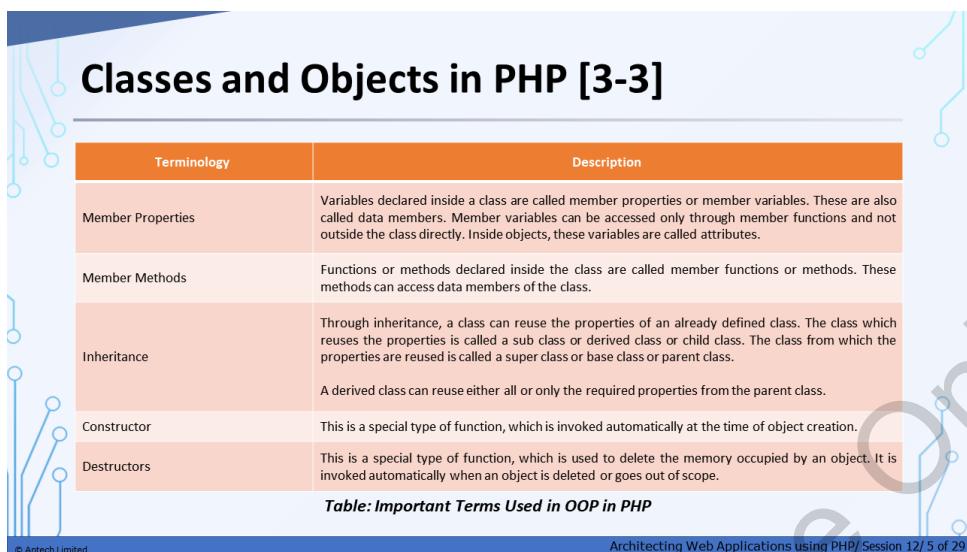
https://www.tutorialspoint.com/php/php_object_oriented.htm

Ask the students following question. Wait for a response before you answer.

In-Class Question: Which keywords can be used to create objects from a class that has been defined?

Answer: New object

Slide 5



Terminology	Description
Member Properties	Variables declared inside a class are called member properties or member variables. These are also called data members. Member variables can be accessed only through member functions and not outside the class directly. Inside objects, these variables are called attributes.
Member Methods	Functions or methods declared inside the class are called member functions or methods. These methods can access data members of the class.
Inheritance	Through inheritance, a class can reuse the properties of an already defined class. The class which reuses the properties is called a sub class or derived class or child class. The class from which the properties are reused is called a super class or base class or parent class. A derived class can reuse either all or only the required properties from the parent class.
Constructor	This is a special type of function, which is invoked automatically at the time of object creation.
Destructors	This is a special type of function, which is used to delete the memory occupied by an object. It is invoked automatically when an object is deleted or goes out of scope.

Table: Important Terms Used in OOP in PHP

© Aptech Limited Architecting Web Applications using PHP/ Session 12/ 5 of 29

Show slide 5 and tell students that given table lists some examples of a class and the objects within it.

Table lists and describes some important class and objected related terms in PHP.

Slide 6

The slide features a blue decorative border with white circles and lines. At the top center, the title 'Defining a Class in PHP' is displayed. Below the title, a 'Code Snippet:' label is followed by a block of PHP code. The code defines a 'City' class with properties '\$name' and '\$country', and methods 'set_name()', 'get_name()', 'set_country()', and 'get_country()'. An object '\$NewYork' is created and assigned to '\$America'. The browser output window shows the results of running the code, displaying 'New York' and 'America'.

Code Snippet:

```
<html>
<body>
<?php
class City {
    // Properties
    public $name;
    public $country;
    // Methods
    function set_name($name) {
        $this->name = $name;
    }
    function set_country($country) {
        $this->country = $country;
    }
    function get_name() {
        return $this->name;
    }
    function get_country() {
        return $this->country;
    }
}
// creating object of class
$NewYork = new City();
$NewYork->set_name('New York');
$America->set_country('America');
echo $NewYork->get_name();
echo "<br>";
echo $America->get_country();
?
</body>
</html>
```

localhost:1231.php

New York
America

Figure: Output for Code Snippet

© Aptech Limited

Architecting Web Applications using PHP/ Session 12 / 6 of 29

Show slide 6 and tell the students that a class is created by using the keyword `class`, followed by the class name and a pair of curly brackets (`{ }`). Properties and methods of the class are given inside the brackets. A class can have any number of properties and methods.

Following is a basic example of a class:

```
<?php
class City {
    // Properties
    public $name;
    public $country;
    // Methods
    function set_name($name) {
        $this->name = $name;
    }
    function get_name() {
        return $this->name;
    }
    function set_country($country) {
        $this->name = $country;
    }
    function get_country() {
        return $this->country;
    }
}
?>
```

Here, a class named `City` is declared. There are two properties defined here namely, `$name` and `$country`. Two methods each are declared for each property to set and retrieve their values.

Code Snippet shows the complete example of creating the class and its objects.

In Code Snippet, a class named `City` is created. Four methods are written inside the class. Two of these methods are given to set the `name` and `country` properties of the class respectively, that is, `set_name()` and `set_country()`. The other two methods are designed to retrieve `name` and `country` values respectively, that is, `get_name()` and `get_country()`. Objects of a class are created using the `new` keyword. `New York` and `America` are instances or objects of class `city`. Class methods are invoked using arrow operator.

Figure shows the output for Code Snippet.

Give following additional information to the students:

Note: Although the PHP OOP terminology states 'methods', the `function` keyword is used to declare class methods.

Objects Creation in PHP

Following are some examples showing the usage of new operator to create the objects:

- \$physics = new Books;
- \$maths = new Books;
- \$chemistry = new Books;

Two of the keywords which deal with object operations are:

PHP – \$this Keyword

PHP – instanceof keyword

© Aptech Limited

Architecting Web Applications using PHP/ Session 12 / 7 of 29

Show slide 7 and tell the students that once a class is defined, any number of objects can be created for the class type. To create objects, new operator should be used.

In the examples mentioned in the slide, the user has created three objects for the class Books and these objects are not dependent on each other. Also, these objects will exist in separate memory locations. Note that here the parentheses after class name is omitted. This is allowed as they are optional.

Additional Information:

Refer to following links for more information:

<https://www.php.net/manual/en/language.types.object.php>

<https://www.geeksforgeeks.org/php-objects/>

<https://appdividend.com/2019/08/30/php-objects-example-object-in-php-tutorial/#:~:text=Creating%20an%20Object%20in%20PHP,built%2Din%20class%20is%20created.>

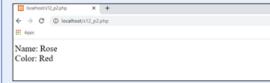
Slide 8

PHP – \$this Keyword [1-2]

Code Snippet:

```
<html>
<body>
</body>
</html>
class Flower {
    // Properties
    public $name;
    public $color;
    // Methods
    function set_name($name) {
        $this->name = $name;
    }
    function get_name() {
        return $this->name;
    }
    function set_color($color) {
        $this->color = $color;
    }
    function get_color() {
        return $this->color;
    }
}
$Rose = new Flower();
$Rose->set_name('Rose');
$Red->set_color('Red');
echo "Name: " . $Rose->get_name();
echo "<br>";
echo "Color: " . $Red->get_color();
?>
</body>
</html>
```

Figure: Output for Code Snippet



Architecting Web Applications using PHP/Session 12/ 8 of 29

Show slide 8 and tell the students that `$this` is a reserved keyword (also referred to as pseudo variable) that is used to access the current object. Pseudo-types are keywords used in the PHP to specify the types or values an argument can have.

`$this` keyword is only available inside a method and refers to the object of the current method. Code Snippet shows use of `$this` keyword inside a class method.

In Code Snippet, `$this` keyword is used. It refers to the current object and is only available inside methods. After creating objects `$Rose` and `$Red` of class `Flower`, method `set_name()` is called using object `$Rose`. Then, it will be passed to `set_name()` method, where value is assigned to `$name` variable as `$Rose` using `$this` keyword.

Next line of the code is used outside the class where value of name is displayed by calling method `get_name()` using same object `$Rose`. This will then call the second method of a class `get_name()` which returns value of name using keyword `$this`.

Figure shows the output for Code Snippet.

PHP – \$this Keyword [2-2]

Following are the two ways of using \$this keyword:

- Outside the class
- Inside the class

© Aptech Limited Architecting Web Applications using PHP/ Session 12/ 9 of 29

Show slide 9 and tell the students that there are two ways of using \$this keyword:

- Outside the class
- Inside the class

These methods are going to be discussed in the upcoming slides.

Additional Information:

Refer to following links for more information:

<https://www.geeksforgeeks.org/this-keyword-in-php/>

<https://www.phptutorial.net/php-oop/php-this/>

Ask the students following question. Wait for a response before you answer.

In-Class Question: What can be written at a start of a class name and can be followed by numerous letters, periods, or underscores?

Answer: Letter

Slide 10

The slide has a blue header bar with decorative circuit board patterns on the sides. The main title is "Using \$this keyword outside the class". Below it, a "Code Snippet:" label is followed by a code block:

```
<html>
<body>
<?php
class House {
    public $color;
}
$white = new House();
$white->color = "WHITE";
echo $white->color;
?>
</body>
</html>
```

To the right, there is a screenshot of a web browser window showing the output: "localhost/s12_p10.php" and "WHITE".

Figure: Output for Code Snippet

At the bottom right, the text "Architecting Web Applications using PHP/ Session 12/ 10 of 29" is visible.

Show slide 10 and tell the students that users can directly change the value of a property from outside the class.

Code Snippet shows declaration of property outside the class.

In Code Snippet, value of property `color` is being assigned outside the class. The `color` property value is assigned as `WHITE` using object `$white` of class `House`. No methods are declared here.

Figure shows the output for Code Snippet.

Additional Information:

Refer to following links for more information:

[https://www.scientecheeasy.com/2020/07/this-keyword-in-java.html#:~:text=The%20keyword%20%E2%80%9Cthis%E2%80%9D%20in%20Java,be%20used%20outside%20the%20class.](https://www.scientecheasy.com/2020/07/this-keyword-in-java.html#:~:text=The%20keyword%20%E2%80%9Cthis%E2%80%9D%20in%20Java,be%20used%20outside%20the%20class.)

<https://tutorials.supunkavinda.blog/php/oop-this>

Slide 11

The slide features a blue header with the title 'Using \$this keyword inside the class'. Below the title is a 'Code Snippet' box containing PHP code. The code defines a class 'House' with a public property '\$color' and a method 'set_color(\$color)'. Inside the method, the value of '\$color' is assigned to '\$this->color'. An object '\$Black' is created from the 'House' class, and its 'set_color("Black")' method is called, followed by an echo statement printing the color. The output window shows the word 'Black'.

Code Snippet:

```
<html>
<body>
<?php
class House {
    public $color;
    function set_color($color) {
        $this->color = $color;
    }
}
$Black = new House();
$Black->set_color("Black");
echo $Black->color;
?>
</body>
</html>
```

Figure: Output for Code Snippet

localhost/s12_p3.php

localhost/s12_p3.php

Black

Architecting Web Applications using PHP/ Session 12/ 11 of 29

Show slide 11 and tell the students that users can define a method in the class and then, call it to change the value of its own property.

Code Snippet shows declaration of a property inside the class.

In Code Snippet, method `set_color()` is declared inside the class `House` and creating the object 'Black' of class `House`. Using that object, user calls `set_color()` function and prints the value of color. Arrow (`->`) operator is used to call a method or value using objects.

Figure shows the output for Code Snippet.

Additional Information:

Refer to following links for more information:

<https://phptutorial.net/php-oop/php-this/>

<https://phptutorial.net/php-oop/php-this/>

Slide 12

The slide features a decorative background with blue and green circuit board patterns. At the top, the title 'PHP – instanceof keyword' is displayed. Below the title, a 'Code Snippet' box contains the following PHP code:

```
<html>
<body>
<?php
class City {
    // Properties
    public $name;
    public $country;
    // Methods
    function set_name($name) {
        $this->name = $name;
    }
    function get_name() {
        return $this->name;
    }
}
$London = new City();
var_dump($London instanceof City);
?>
</body>
</html>
```

To the right of the code snippet is a screenshot of a web browser window titled 'localhost/s12_p11.php'. The browser shows the output of the code: 'bool(true)'. Below the browser screenshot is the caption 'Figure: Output for Code Snippet'.

At the bottom right of the slide, the text 'Architecting Web Applications using PHP/ Session 12/ 12 of 29' is visible.

Show slide 12 and tell the students that PHP uses a keyword `instanceof` to check if an object belongs to a particular class. Code Snippet shows the use of `instanceof` keyword in a script.

In Code Snippet, keyword `instanceof` is used to know whether an object is of specific class type. This can be very useful in large applications that may have several classes and more than one developer working on the scripts.

Here in the code, `var_dump()` function is used to return type of the object. By using a combination of `var_dump()` and `instanceof`, user can check whether `$London` object belongs to `City` class or not. The statement returns `bool` value `true` here because object does belong to `City` class.

Figure shows the output for Code Snippet.

Additional Information:

Refer to following links for more information:

https://www.w3schools.com/php/keyword_instanceof.asp

<https://riptutorial.com/php/example/7595/instanceof--type-operator->

PHP OOP – Constructor and Destructor

Constructors

- Constructors are special methods invoked automatically when an object is created.

Destructors

- Destructors are special member methods invoked automatically when an object is destroyed or an object goes out of scope. Usually, it is invoked at the end of the script.

© Aptech Limited

Architecting Web Applications using PHP/ Session 12/ 13 of 29

Show slide 13 and tell the students that once objects are declared, it is recommended that each member be initialized. Additionally, once the objects go out of scope, memory should be cleared. To perform these operations, special methods called constructors and destructors are used. Their purposes are mentioned in brief in the slide.

In object oriented programming terminology, constructor is a method defined inside a class is called automatically at the time of creation of object. Purpose of a constructor method is to initialize the object. In PHP, a method of special name `__construct` acts as a constructor. Ask the students following question. Wait for a response before you answer.

Destructor is a method automatically as soon as garbage collector finds that a particular object has no more references. In PHP, destructor method is named as `__destruct`. During shutdown sequence too, objects will be destroyed. Destructor method doesn't take any arguments, neither does it return any data type

In-Class Question: Can PHP class have multiple constructors?

Answer: PHP lacks support for declaring multiple constructors of different numbers of parameters for a class unlike languages such as Java. So, if another constructor is declared, PHP will throw a fatal error upon running the code.

Slide 14

The slide features a blue header with the title 'Constructor Method [1-2]'. Below the title is a 'Code Snippet' box containing the following PHP code:

```
<html>
<body>
<?php
class Example {
    public $name;
    public $color;
    function __construct($name) {
        $this->name = $name;
    }
    function get_name() {
        return $this->name;
    }
}
$alpha = new Example("Alpha");
echo $alpha->get_name();
?>
</body>
</html>
```

To the right of the code snippet is a screenshot of a web browser window showing the output. The browser address bar shows 'localhost/s12_p12.php'. The page content displays the word 'Alpha'.

Figure: Output for Code Snippet

Architecting Web Applications using PHP/ Session 12/ 14 of 29

Show slide 14 and tell the students that a constructor is usually declared as public and named as **__construct**. This method should start with two underscores (_).

Following is a basic code showing how to declare a constructor:

```
class Example {
public function __construct() {
// constructor function
echo "Hello World";
}
```

The **__construct()** method will be called automatically when an object is created for the class.

Code Snippet shows the use of the **__construct()** method in a class.

In Code Snippet, using constructors saves time from calling the **set_name()** method thereby, reducing the amount of code. If user creates a **construct()** method, PHP will automatically call this method when an object is created from a class. Here, \$alpha object automatically sets the value Alpha without calling the method **set_name()**. It is only required to call **get_name()** method while displaying the value.

Figure shows the output for Code Snippet.

The slide features a blue decorative border with white circles and lines. The title 'Constructor Method [2-2]' is at the top. A 'Code Snippet' box contains the following PHP code:

```
<html>
<body>
<?php
class City {
    public $name;
    public $country;
    function __construct($name, $country) {
        $this->name = $name;
        $this->country = $country;
    }
    function get_name() {
        return $this->name;
    }
    function get_country() {
        return $this->country;
    }
}
$London = new City("London", "England");
echo $London->get_name();
echo "<br>";
echo $London->get_country();
?>
</body>
</html>
```

To the right, a screenshot of a browser window shows the output: 'localhost/s12_p14.php' with the text 'London' and 'England' displayed.

Figure: Output for Code Snippet

Architecting Web Applications using PHP/ Session 12/ 15 of 29

Show slide 15 and tell the students that Code Snippet shows constructors being used with two parameters.

In Code Snippet, constructor with two properties or parameters is used, that is, name and country. Therefore, values are displayed using object London of class City, by calling the methods `get_name()` and `get_country()`.

Figure shows the output for Code Snippet.

Additional Information:

Refer to following links for more information:

https://www.w3schools.com/php/php_oop_constructor.asp

<https://www.javatpoint.com/php-oops-constructor>

Slide 16

The slide features a blue header with the title 'Destructor Method'. Below the title is a 'Code Snippet' box containing PHP code. The code defines a class 'Flower' with a constructor and a destructor. The constructor initializes properties \$name and \$color. The destructor prints a message using the current object's name. An instance of the class is created with the value 'Pink' for \$name. To the right of the code snippet is a screenshot of a browser window showing the output: 'The flower is Pink.'

Code Snippet:

```
<html>
<body>
<?php
class Flower {
    public $name;
    public $color;
    function __construct($name) {
        $this->name = $name;
    }
    function __destruct() {
        echo "The flower is {$this->name}." ;
    }
}
$Lotus = new Flower("Pink");
?>
</body>
</html>
```

The flower is Pink.

Figure: Output for Code Snippet

Architecting Web Applications using PHP/ Session 12/ 16 of 29

Show slide 16 and tell the students that a destructor is called when an object is to be removed or has gone out of scope. As with constructor, a destructor too should start with two underscores (_). If an object contains __destruct() method, it will be automatically called by PHP at the end of the script.

Code Snippet shows the use of destructor in a class.

In Code Snippet, the constructor is called automatically when the object is created. However, the destructor is called at the end of code when the object goes out of scope. Within the destructor method __destruct(), the value of property name is displayed using this keyword. It refers to the current object (\$Lotus) of class Flower and prints the value as Pink.

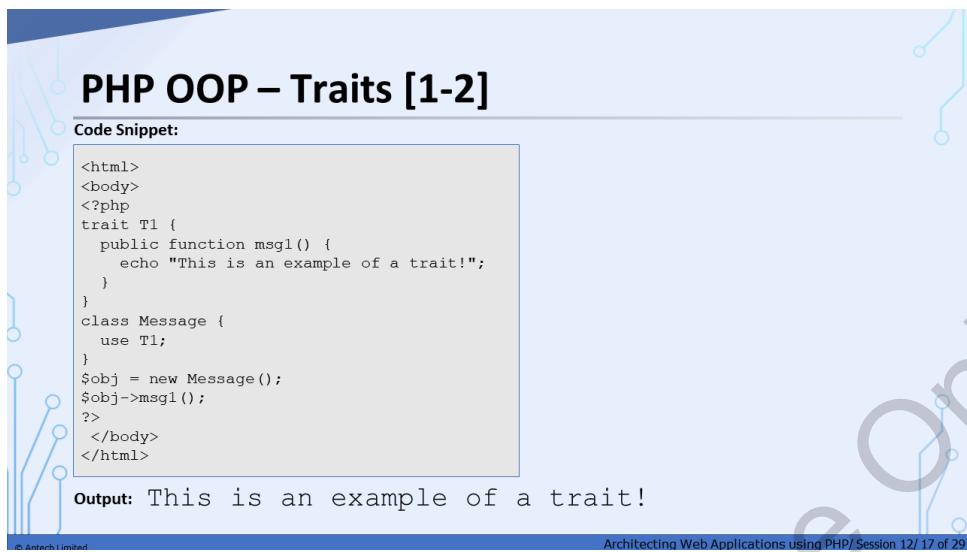
Figure shows the output for Code Snippet.

Additional Information:

Refer to following links for more information:

- https://www.w3schools.com/php/php_oop_destructor.asp
- <https://www.phptutorial.net/php-oop/php-destructor/>
- <https://www.geeksforgeeks.org/php-constructors-and-destructors/>

Slide 17



PHP OOP – Traits [1-2]

Code Snippet:

```
<html>
<body>
<?php
trait T1 {
    public function msg1() {
        echo "This is an example of a trait!";
    }
}
class Message {
    use T1;
}
$obj = new Message();
$obj->msg1();
?>
</body>
</html>
```

Output: This is an example of a trait!

© Aptech Limited

Architecting Web Applications using PHP/ Session 12/ 17 of 29

Show slide 17 and tell the students that a class in PHP supports only single inheritance. A child class can inherit only from one single parent. In large applications, instead of creating everything from scratch, it makes sense to reuse already defined classes or types and build upon their functionality further. One of the biggest advantages OOP offers is that of reusability. To inherit multiple behaviors, PHP uses a concept called traits.

A trait is similar to a class and can contain constants, properties, and methods. It can be used to group functionality in a single form or structure.

Following is the syntax used to declare a trait:

Syntax:

```
<?php
trait TraitName {
    // some code...
}
?>
```

A user cannot instantiate a trait on its own. It has to be used within a class with the help of keyword `use`. Following is the syntax for `use` keyword:

Syntax:

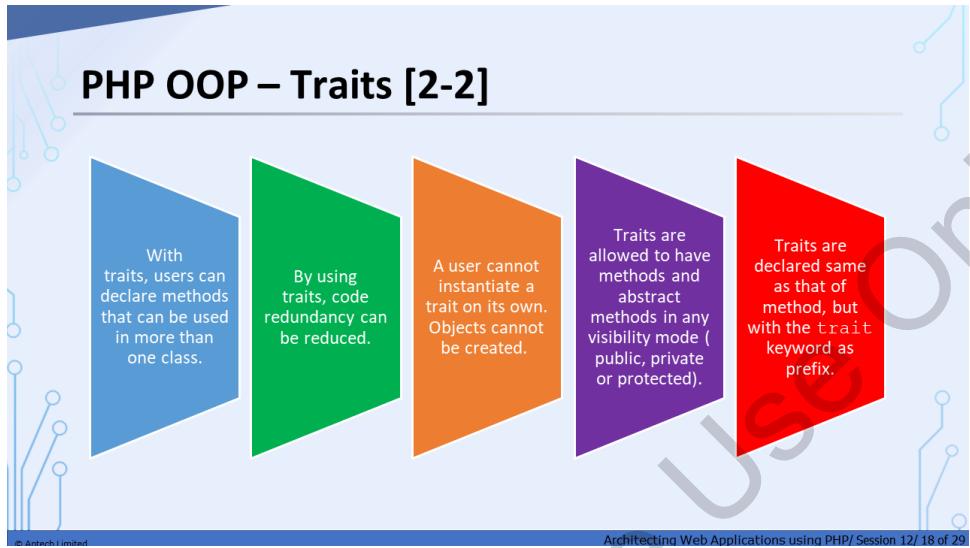
```
<?php
class ClassName {
    use TraitName;
}
?>
```

Code Snippet shows an example of how to use trait keyword in a class.

In Code Snippet, a trait is `T1` declared with keyword `trait`. Its definition contains a method named `msg1()`. This trait can now be inherited by any class with the keyword `use`. This is done in class `Message`. Later, `msg1()` method is called using object `$obj` of class `Message`

without having declared `msg1()` method inside the class `Message`. Thus, `Message` has inherited `msg1()` method from the trait `T1`. Hence, when `msg1()` method is called, it will print the output in the browser.

Slide 18



Show slide 18 and tell the students that a class uses a trait so that the members defined in the trait can be inherited by the class. Code reusability is promoted through usage of traits. Some of the important features of traits are mentioned in the slide.

Additional Information:

Refer to following links for more information:

https://www.w3schools.com/php/php_oop_traits.asp
<https://www.phptutorial.net/php-oop/php-traits/>
<https://www.tutorialspoint.com/what-is-traits-in-php>
<https://www.php.net/manual/en/language.oop5.traits.php>

Ask the students following question. Wait for a response before you answer.

In-Class Question: What is the difference between trait and interface in PHP?

Answer: The main difference between the Traits and Interfaces in PHP is that **the Traits define the actual implementation of each method within each class**, so many classes implement the same interface but having different behavior, while traits are just chunks of code injected in a class in PHP.

Slide 19

PHP – Using Multiple Traits

Code Snippet:

```
<html>
<body>
<?php
trait T1 {
    public function msg1() {
        echo "This is an example of a trait! <br>";
    }
}
trait T2 {
    public function msg2() {
        echo "Code reusability can be achieved through OOP feature
of traits!";
    }
}
class Message1 {
    use T1;
}
class Message2 {
    use T1, T2;
}
$obj1 = new Message1();
$obj1->msg1();
echo "<br>";
$obj2 = new Message2();
$obj2->msg1();
$obj2->msg2();
</body>
</html>
```

The screenshot shows a browser window displaying the output of the PHP code. The output consists of two lines of text: "This is an example of a trait!" followed by "Code reusability can be achieved through OOP feature of traits!".

Figure: Output for Code Snippet

Architecting Web Applications using PHP/ Session 12/ 19 of 29

Show slide 19 and tell the students that Code Snippet shows usage of multiple traits in a class. In Code Snippet, two traits are declared, namely T1 and T2 in two classes, namely Message1 and Message2. The msg1() method is called by using object obj1 of a class Message1. By using object obj2 of a class Message2, both the methods msg1() and msg2() are being called. In Code Snippet, two traits T1 and T2 are used in a class Message2.

Figure shows the output for Code Snippet.

Additional Information:

Refer to following links for more information:

https://www.w3schools.com/php/php_oop_traits.asp

<https://www.phptutorial.net/php-oop/php-traits/>

<https://www.geeksforgeeks.org/multiple-inheritance-in-php/>

Slides 20 and 21

PHP Namespaces [1-2]

Explanation 1
Assume that a program has a global variable named `$title`, inside a method. This program has another variable with the same name `$title`. If user assigns and changes values of `$title` inside of the method, the global variable is not affected. It will remain unchanged. This is called scope of a variable. Following the same method, two classes can be declared with same name to give it scope.

Explanation 2
Assume that a user is creating an open-source PHP library to send mails and the user shares that with the developer community. The library of the user has a class named `Email`. If a developer who already has a class called `Email` downloads this library, name collision/conflict will occur. The user can now either rename those classes or use Namespacing.

© Aptech Limited Architecting Web Applications using PHP/ Session 12/ 20 of 29

PHP Namespaces [2-2]

Explanation 3
Assume that there are two people with the name 'John'. To separate them, their surname can be used. In the same way, two classes having the same name can be separated by Namespacing.
By using namespaces two different problems are solved:

- They group the classes that work together to perform a task. This allows the user to organize code in a better manner.
- The same name can be used for more than one class, therefore, avoiding dilemma among names.

© Aptech Limited Architecting Web Applications using PHP/ Session 12/ 21 of 29

Using slides 20 and 21, tell the students that PHP allows having multiple classes with the same name by using Namespaces, which is a qualifier. It helps in better organization of code and allows to use same name for more than one class.

Some explanations for it are mentioned in slides 20 and 21.

Further explain:

For example, consider the set of classes describing an HTML table, such as `Table`, `Row`, and `Cell`. Consider having another set of classes to describe furniture, such as `Table`, `Chair`, and `Bed`. By using Namespaces, these classes can be organized into two different groups while also preventing the two different classes with the same name `Table`, from being mixed up.

Additional Information:

Refer to following links for more information:

<https://www.php.net/manual/en/language.namespaces.php>

<https://www.geeksforgeeks.org/php-namespace/>

https://www.w3schools.com/php/php_namespaces.asp

Ask the students following question. Wait for a response before you answer.

In-Class Question: Within a namespace, what is used as a prefix to access the built-in PHP classes?

Answer: Percent

Slide 22

The slide has a blue header bar with the title 'Requirement of Namespaces in PHP and Declaring a Namespace'. Below the title is a section titled 'Code Snippet:' containing the following PHP code:

```
<?php
namespace PHP;
class Table1 {
    public $title = "";
    public $numRows = 0;
    public function msg() {
        echo "<p>Table {$this->title} has {$this->numRows}</p>";
    }
}
$stable1 = new Table1();
$stable1->title = "<br>My table ";
$stable1->numRows = 10;
?>
<!DOCTYPE html>
<html>
<body>
<?php
$stable1->msg();
?>
</body>
</html>
```

To the right of the code is a screenshot of a browser window showing the output:

localhost:12345/12_011.php
localhost:12345/12_011.php
App
Table
My table has 10 rows.

Figure: Output for Code Snippet

Architecting Web Applications using PHP / Session 12 / 22 of 29

Show slide 22 and tell the students that PHP programs cannot have two classes with the same name. For example, if a user has the requirement for two classes to handle blog users and app users, then those classes should have distinct names such as Blog_User and App_User. However, remembering these types of prefixes can be confusing.

Namespaces help the user to simplify and organize the process and give more control over the program.

Declaring a Namespace

Namespaces are declared using the keyword `namespace` at the beginning of a file, that is, as the first statement in the PHP file. Following is the syntax to use while declaring a namespace:

Syntax:

```
<?php
echo "Hello World!";
namespace Html;
```

```
...  
?>
```

Code Snippet shows use of Namespace in a class.

In Code Snippet, a namespace PHP is used and a method is declared inside the namespace. The property `title` is assigned with a dummy value. However, in the later part of the code, the code is assigning same variable `title` value as `My table` contains using object `table1` of class `Table1`.

Figure shows the output for Code Snippet.

Slide 23

The screenshot displays a presentation slide with the title "Using namespaces". On the left, there is a "Code Snippet:" section containing two files: "index.php" and "HTML.php". The "index.php" file includes code that creates a new `Table1` object and prints its `msg()` output. The "HTML.php" file defines a `Table1` class with a `title` property and a `msg()` method, and a `Row` class with a `numCells` property and a `msg()` method. Both classes have a `msg()` method that prints the class name and its properties. On the right, a screenshot of a browser window shows the output of the code execution, displaying the `Table1` object's `msg()` output followed by the `Row` object's `msg()` output.

Show slide 23 and tell the students that any code following a namespace declaration will operate inside the namespace. Therefore, classes belonging to the namespace can be instantiated without any qualifiers. If the user wants to access classes from outside of a namespace, the class should have the namespace attached to it.

Code Snippet shows the use of namespaces in two different PHP codes, first with `index.php` and then, with `HTML.php`.

In this part of Code Snippet, another PHP code is included using keyword `include` as `include "HTML.php";`. The namespace `Html1` and class name `Table1` object `$table` are created. Next, the code is assigning value to a `title` property by using an object. `title` property is declared in class `Table1`. Values are assigned to variables `numRows` and `numCells` as 7 and 3 using objects `$table` and `$row`.

In this part of Code Snippet, namespace `Html1` and classes `Table1` and `Row` are declared. In these classes, you are declaring properties `title` and `numRows`. In the class `Row`, the code is declaring `$numCells`. Next, the code is displaying the value of `title` and `numRows` in a class `Table1` and value of `numCells` in class `Row`.

Figure shows the output for Code Snippet.

Finally, the output is obtained as shown in Figure. `index.php` code has to be run to view the output.

It becomes easier to use `namespace` keyword when many classes from the same namespace are being used at the same time.

Slide 24

The slide features a blue header with the title 'PHP Iterables [1-2]'. Below the title is a section labeled 'Code Snippet:' containing the following PHP code:

```
<html>
<body>
<?php
function displayIterable(iterable
$testIterable) {
    foreach($testIterable as $item) {
        echo $item;
    }
}
$arr1 = ["1", "2", "3"];
displayIterable($arr1);
?>
</body>
</html>
```

To the right of the code is a screenshot of a web browser window showing the output: 'localhost/s12_p25.php' displays the numbers '123'.

Figure: Output for Code Snippet

At the bottom of the slide, there is a footer with the text '© Aptech Limited' and 'Architecting Web Applications using PHP/ Session 12/ 24 of 29'.

Show slide 24 and tell the students that an `iterable` in PHP is a parameter type that can be looped with a `foreach` loop. `iterable` can be used as a data type for function arguments and also as the return values of a function.

Code Snippet shows use of `iterable` in a `foreach` loop.

In Code Snippet, `iterable` keyword is used in function `displayIterable()` to declare `testIterable` of `iterable` type. A `foreach` loop is used to traverse through each value of `testIterable`. Later, an array is defined and passed as argument to function `displayIterable()` call.

Figure shows the output for Code Snippet.

Slide 25

The slide features a blue decorative border with white circles and lines. At the top center is the title **PHP Iterables [2-2]**. Below it is a section titled **Code Snippet:** containing the following PHP code:

```
<html>
<body>
<?php
function getIterable(): iterable {
    return ["x", "y", "z"];
}
$testIterable = getIterable();
foreach($testIterable as $alpha) {
    echo $alpha;
}
?>
</body>
</html>
```

To the right of the code is a screenshot of a web browser window showing the output of the code. The address bar says `localhost/s12_p26.php`. The page content displays the text `xyz`.

Figure: Output for Code Snippet

At the bottom right of the slide is the text `Architecting Web Applications using PHP/ Session 12/ 25 of 29`.

Show slide 25 and tell the students that Code Snippet shows use of `Iterable` keyword with `return` keyword inside the function.

In Code Snippet, `return` keyword is used inside the function `getIterable()` with the keyword `:iterable` to return values 1, 2, and 3. The code is assigning the function `getIterable()` to variable `$testIterable` and using it in `foreach` loop. The value is displayed by `$num` variable.

Figure shows the output for Code Snippet.

Additional Information:

Refer to following links for more information:

<https://www.php.net/manual/en/language.types.iterable.php>

https://www.w3schools.com/php/php_iterables.asp

https://www.tutorialspoint.com/php_iterables

Ask the students following question. Wait for a response before you answer.

In-Class Question: What are Iterables in PHP?

Answer: An iterable is **any value which can be looped through with a foreach() loop**.

The iterable pseudo-type was introduced in PHP 7.1 and it can be used as a data type for function arguments and function return values.



Arrays and Iterators

Arrays

- All arrays are iterables. Therefore, any array can be used as an argument of a method. However, it requires an iterable.

Iterators

- PHP contains an interface called `Iterator`, which can be used as an argument of a function that requires an `iterable`.

Architecting Web Applications using PHP/ Session 12/ 26 of 29

Show slide 26 and explain students following:

Arrays

Iterables are all arrays. As a result, any array may be used as a method argument. It does, however, require the use of an iterable.

Iterators

An iterator contains a list of items and provides functions to loop through. It also contains a pointer to one of the elements in the list. Each item in the list should have a key, through which the item can be located.

Additional Information:

Refer to following links for more information:

https://www.w3schools.com/php/php_iterables.asp#:~:text=All%20arrays%20are%20iterables%2C%20so,function%20that%20requires%20an%20iterable.&text=Any%20object%20that%20implements%20the,methods%20to%20loop%20through%20them.
<https://www.php.net/manual/en/class.arrayiterator.php>

Ask the students following question. Wait for a response before you answer.

In-Class Question: What is used to execute code automatically when a new instance of a class is created?

Answer: Constructor

Iterator Methods [1-2]

An iterator should have following methods:

current ()	• It returns the element that the pointer is currently pointing to. It can be of any data type.
key ()	• It returns the key associated with the current element in the list. The data type of the key can only be an integer, float, Boolean, or string.
next ()	• It moves the pointer to the next element in the list.
rewind ()	• It moves the pointer to the beginning (first element) of the list.
valid ()	• If the internal pointer is operated to point to an invalid element of the list (for example, if next () was called at the end of the list), this function should return false. In any other case, it returns true.

© Aptech Limited Architecting Web Applications using PHP/ Session 12/ 27 of 29

Show slide 27 and explain about the iterator methods using the slide.

Additional Information:

Refer to following links for more information:

<https://stackoverflow.com/questions/9777806/how-should-phps-iterator-methods-validate-current-and-next-behave>

<https://startutorial.com/articles/view/modern-php-developer-iterator>

<https://kompot.petsru.ru/php-manual/spl.iterators.html>

Ask the students following question. Wait for a response before you answer.

In-Class Question: Why do we use iterator?

Answer: The primary purpose of an iterator is **to allow a user to process each element of a container while isolating the user from the internal structure of the container**. This allows the container to store elements in any manner it wishes while allowing the user to treat it as if it were a simple sequence or list.

Iterator Methods [2-2]

Code Snippet:

```

<html>
<body>
<?php
// Create an Iterator
class TestIterator implements Iterator {
    private $alpha = [];
    private $pointer = 0;
    public function __construct($alpha) {
        $this->alpha = array_values($alpha);
    }
    public function current() {
        return $this->alpha[$this->pointer];
    }
    public function key() {
        return $this->pointer;
    }
    public function next() {
        $this->pointer++;
    }
    public function rewind() {
        $this->pointer = 0;
    }
    public function valid() {
        return $this->pointer < count($this->alpha);
    }
}

// A function that uses iterables
function printIterable(iterable $testIterable) {
    foreach($testIterable as $alpha) {
        echo $alpha;
    }
}

// Use the iterator as an iterable
$iterator = new TestIterator(["m", "n", "o"]);
printIterable($iterator);
?>
</body>
</html>

```



Figure: Output for Code Snippet

© Aptech Limited

Architecting Web Applications using PHP/ Session 12/ 28 of 29

Show slide 28 and tell the students that Code Snippet shows implementation of Iterator in a class and pointers inside the function.

In Code Snippet, `iterable` is implemented to a class `TestIterator`. Two variables `$num` and `$pointer` are declared inside the class and assigning pointer equal to 0. Then, the `constructor` function is declared inside which values of `num` are assigned using `this` keyword.

The functions `current()`, `key()`, `next()`, `rewind()`, and `valid()` are defined. Each function carries out a particular task. Currently, the pointer value is set to 0, which will point to number 1 of function argument. `key()` function will point to pointer, `next()` function will point to next element or number 2, `rewind()` function will again point to pointer as 0, and `valid()` will count numbers present by using `count()` function. Outside the class, the code is creating the object of class `Testitable` and calling the function `printIterable` by passing object as argument. `printIterable()` function is using `iterable` type variable, which is used in `foreach` loop to print numbers.

Figure shows the output for Code Snippet.



Summary

- Object-Oriented Programming (OOP) is a programming paradigm that is mainly based on the use of classes and objects.
- A class is a data type that can contain constants, variables (properties'), and functions (methods).
- An object is an instance of a class and is created using new operator.
- A class is created by using the keyword class, followed by the class name, and a pair of curly brackets ({}).
- \$this is a reserved keyword used to access the current object.
- PHP makes use of instanceof keyword to check if an object belongs to a particular class.
- Constructors and destructors are special member functions.
- A constructor is invoked automatically when an object is created.
- A destructor is invoked automatically when an object is destroyed or goes out of scope.
- In PHP, the traits concept is used to inherit multiple behaviors.
- Namespaces are qualifiers that help to organize code better and use the same name for multiple classes.

Architecting Web Applications using PHP/ Session 12/ 29 of 29

Use slide 29 to summarize the session. You will end the session, with a brief summary of what has been taught in the session. Tell the students pointers of the session. This will be a revision of the current session.

12.3 Post Class Activities for Faculty

You should familiarize yourself with the topics of the next session.

Tips:

You can also check the Articles/Blogs/Expert Videos uploaded on the Online Varsity site to gain additional information related to the topics covered in the next session.

Session 13 – Methods in PHP and Other OOP Concepts

13.1 Pre-Class Activities

Before beginning the session, you should fully familiarize yourself with its topics. Prepare a question or two that will be a key point to relate the current session objectives.

13.1.1 Teaching Skills

To teach this session, you should be well-verses with the concept of internationalization and design patterns in Java. You must also be familiar with the concept of localization.

You must use the given images and teach the concepts in the theory class. For teaching in the class, you are expected to use slides and LCD projectors.

Tips:

It is recommended that you test the understanding of the students by asking questions in between the class.

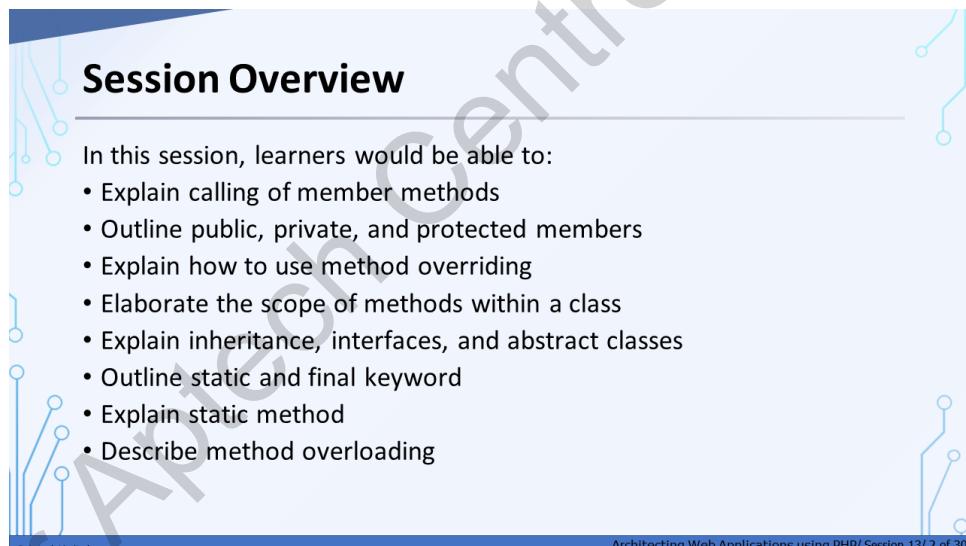
In-Class Activities

Follow the order given here during In-Class activities.

Overview of the Session

Give the students an overview of the current session in the form of session objectives. Read out the objectives on slide 2.

Slide 2



The slide has a blue header bar with the text "Session Overview". The main content area has a light blue background with a decorative border featuring blue circuit board patterns. The text in the content area reads: "In this session, learners would be able to:" followed by a bulleted list of nine items. At the bottom of the slide, there is a small footer bar with the text "Aptech Limited" on the left and "Architecting Web Applications using PHP / Session 13 / 2 of 30" on the right.

In this session, learners would be able to:

- Explain calling of member methods
- Outline public, private, and protected members
- Explain how to use method overriding
- Elaborate the scope of methods within a class
- Explain inheritance, interfaces, and abstract classes
- Outline static and final keyword
- Explain static method
- Describe method overloading

Show slide 2 and give the students a brief overview of the current session in the form of session objectives. Inform students that the session begins with detailed overview of calling of member methods.

In PHP OOP terminology, class member functions are called 'methods'. The function keyword is used to declare class methods. The session explains how to call member methods. The session further describes the concept of inheritance and explores public, private, and protected members in classes. Next, one will learn about method scope, interfaces, and abstract classes. The session further explains about static and final keywords. This session will also elaborate on method overloading.

13.2 In-Class Explanations

Slide 3

The slide has a blue header bar with the title 'Calling Member Methods [1-2]'. Below the title is a section labeled 'Example:' containing a snippet of PHP code:

```
...  
$lion->setTitle(" Alex in the Madagascar ");  
$tiger->setTitle(" Joshua in the central zoo ");  
$zebra->setTitle(" Marty in the central zoo ");  
$lion->setQuantity(150);  
$tiger->setQuantity(100);  
$zebra->setQuantity(550);
```

Below the code is a table titled 'Table: List of PHP Access Modifiers' with the following data:

Modifier	Methods	Variables	Classes
public	Applicable	Applicable	Not Applicable
private	Applicable	Applicable	Not Applicable
protected	Applicable	Applicable	Not Applicable
abstract	Applicable	Not Applicable	Applicable
final	Applicable	Not Applicable	Applicable

At the bottom left is the copyright notice '© Aptech Limited' and at the bottom right is the page information 'Architecting Web Applications using PHP / Session 13 / 3 of 30'.

Show slide 3 and tell the students that OOP is a paradigm based on the concept of objects that help in building complex and reusable applications using data and code. The key object-oriented concepts in PHP are class, object, inheritance, polymorphism, overloading, data abstraction, constructors, and destructors. An object is created in PHP using the new keyword.

Once the objects are created, the variables and member methods of the class can be accessed using the operator `->`. Member functions are accessed to get information on the object properties.

The example on the slide shows how object properties are accessed through the member methods where member methods are called to set the title and quantity of three animal-based books. In this example, using the `setTitle()` method of class `Animal`, the title variable can be set. Using the `setQuantity()` method of class `Animal`, the quantity can be set.

OOP has certain keywords called access modifiers that decide the accessibility for various class methods, variables, and other member methods. Some PHP keywords represent the access modifiers and are used to set access rights with their members and PHP classes. PHP access modifiers can be classified based on application with class, variables, and methods. Table lists different access modifiers and their applicability for methods, variables, and classes.

Slide 4

Calling Member Methods [2-2]

Code Snippet:

```
<?php
class AnimalBooks {
    // Properties
    public $title;
    public $quantity;
    // Methods
    function setTitle($title) {
        $this->title = $title;
    }
    function getTitle() {
        return $this->title;
    }
    function setQuantity($quantity) {
        $this->quantity = $quantity;
    }
    function getQuantity() {
        return $this->quantity;
    }
}
$lion = new AnimalBooks();
$tiger = new AnimalBooks();
$zebra = new AnimalBooks();
$lion->setTitle("Alex in the Madagascar");
$tiger->setTitle("Joshua in the central zoo");
$zebra->setTitle("Marty in the central zoo");
$lion->setQuantity(150);
$tiger->setQuantity(100);
$zebra->setQuantity(550);
$lion->getTitle();
$tiger->getTitle();
$zebra->getTitle();
```

```
$lion->getQuantity();
$tiger->getQuantity();
$zebra->getQuantity();
echo " Title: " . $lion->getTitle();
echo " Quantity: " . $lion->getQuantity();
echo "<br>";
echo " Title: " . $tiger->getTitle();
echo " Quantity: " . $tiger->getQuantity();
echo "<br>";
echo " Title: " . $zebra->getTitle();
echo " Quantity: " . $zebra->getQuantity();
echo "<br>";
```

Figure: Output for Code Snippet

Architecting Web Applications using PHP / Session 13 / 4 of 30

Show slide 4 and tell students that Code Snippet shows how the member methods are called to get values. The code in the example shown previously has been used here.

In Code Snippet, access modifiers are used. Here, when defining the class members, `public` access modifier is used so that a user can access at class level. Member variables `title` and `quantity` of class `Animal` are declared with `public`. To change the behavior of a class, access modifiers can be assigned to the class itself.

Figure shows the output for Code Snippet.

Additional Information:

Refer to following links for more information:

<https://stackoverflow.com/questions/7628921/calling-member-function-from-other-member-function-in-php>

https://www.tutorialspoint.com/php/php_object_oriented.htm

Inheritance [1-2]

Following are the things to remember while using inheritance:

- The child class has access to only the non-private methods and properties on the parent class.
- The methods of the child class are not available to the parent class.
- The methods defined in the parent class can be overridden by the child class with its own implementation.

© Aptech Limited Architecting Web Applications Using PHP / Session 13 / 5 of 30

Show slide 5 and explain that Inheritance is an important principle in OOP, as it enables a class to use methods and properties of other classes. In programming, there is always a requirement to create a new class with functionalities of other or existing classes. In such situations, you can either copy or inherit all the methods and properties of the existing class into another class.

Inheritance is beneficial for reusability when creating multiple similar classes. Instead of creating classes from scratch, user can build upon existing classes and extend their functionality. The inherited class is called the Parent class (super or base class) and the inheriting class is called Child Class (Sub or derived class). The common properties and methods in the parent class can be inherited into the child class, based on access modifiers.

Additional Information:

Refer to following links for more information:

https://www.w3schools.com/php/php_oop_inheritance.asp
<https://www.phptutorial.net/php-oop/php-inheritance/>

Ask the students following question. Wait for a response before you answer.

In-Class Question: Which pseudo-variable is not available inside methods that are declared as static?

Answer: \$this

Slide 6

The slide features a decorative background with blue circuit board patterns on the left and right sides. In the center, there's a screenshot of a code editor and a browser window.

Example:

```
...  
class Bank {  
    // parent class code  
}  
class Transaction extends Bank {  
    // child class code  
}  
...
```

Code Snippet:

```
<?php  
// parent class  
class BankAccount {  
    // public property type  
    public $type;  
    // public function credit  
    public function credit() {  
        echo $this->type. " CREDIT transaction in  
progress...chr/">;  
    }  
    // public function debit  
    public function debit() {  
        echo $this->type. " DEBIT transaction in  
progress...chr/">;  
    }  
} // child class  
class Savings extends BankAccount {  
    // No code in child class  
}  
// child class  
class Currentacct extends BankAccount {  
    // No code in child class  
}  
// Instantiate the derived classes to create objects  
$savings = new Savings();  
$savings->type = "Salary Savings Account";  
  
$currentacct = new Currentacct();  
$currentacct->type = "Trading Current Account";  
  
// call class methods  
$savings->credit();  
$currentacct->debit();  
?>
```

Figure: Output for Code Snippet

Output in browser: Salary Savings CREDIT transaction in progress... Traders Current DEBIT transaction in progress...

© Aptech Limited Architecting Web Applications Using PHP/ Session 13 / 6 of 30

Show slide 6 and tell students that in PHP, the user can use `extends` keyword to specify name of the parent class while defining the child class. Following is the syntax used to inherit a class:

Syntax:

```
class <Child> extends class <Parent>
```

where, Parent is an existing class.

The example on the slide shows `extends` keyword being used to inherit a class.

In this example, `extends` keyword is used to specify name of the parent class while defining the child class. Here, `Bank` class is the parent or base class and `Transaction` class is the child or derived class. The `Bank` class is inherited by the `Transaction` class. Thus, any object of `Transaction` class has all the properties and methods of `Bank` class as well as any members defined within `Transaction`. This is the power and advantage of inheritance.

Code Snippet shows an example with class `BankAccount` containing basic functions `debit()` and `credit()`. Child classes `Currentacct` and `Savings` are created extending the parent class `BankAccount`.

In Code Snippet, the child classes `Savings` and `Currentacct` inherit from the `BankAccount` class. This allows them to access public properties and methods of parent class.

Figure shows the output for Code Snippet.

Slide 7

Code Snippet:

```
<?php  
// parent class  
class Television {  
    // public property name  
    public $name;  
  
    // public function transmit_sound  
    public function transmit_sound() {  
        echo $this->name. " - does transmit sound  
successfully...<br/>";  
    }  
  
    // public function transmit_video  
    public function transmit_video() {  
        echo $this->name. " - does transmit video  
successfully...<br/>";  
    }  
}  
  
// child class  
class Ledtv extends Television {  
    // specific category of Television  
    // and methods  
    public function control_feature() {  
        echo "Manufacturer Name: " .  
$this->name.  
"  
    }  
    echo " Our best feature: The  
picture quality of  
an LED display is far better than an  
LCD<br/>";  
}  
  
$ledtv = new Ledtv();  
$ledtv->name = "Samsung";  
// calling parent class method  
$ledtv->transmit_sound();  
// calling child class method  
$ledtv->control_feature();  
echo "<br/>";  
  
$lcdtv = new Lcdtv();  
$lcdtv->name = "Sony";  
// calling parent class method  
$lcdtv->transmit_sound();  
// calling child class method  
$lcdtv->control_feature();  
>>
```

Figure: Output for Code Snippet

Samsung – does transmit sound successfully
Variant Name Samsung
Our best feature: The picture quality of an LED display is far better than an LCD

Architecting Web Applications using PHP/ Session 13/ 7 of 30

Show slide 7 and explain that a child class can access the non-private members of the parent class. A child class can have its own member properties and methods.

Code Snippet shows an example of a child class defining and accessing its own methods and properties.

In Code Snippet, class `Television` is the parent or base class. There are two sub classes `Ledtv` and `Lcdtv` that inherit from `Television` class.

Figure shows the output for Code Snippet.

The properties and methods defined in the parent class can be used by the child class without defining them repeatedly. Additionally, the child class can have its own set of methods and properties similar to any other class. This is possible even though it inherits the properties of the parent class.

After understanding what inheritance is, one can gain clarity on the working of access modifiers such as `public`, `private`, and `protected`.

Additional Information:

Refer to following links for more information:

- <https://www.simplilearn.com/tutorials/php-tutorial/inheritance-in-php>
- <https://tutorials.supunkavinda.blog/php/oop-inheritance>
- <https://www.educative.io/edpresso/what-is-class-inheritance-in-php>



Public Members

By defining the classes and their members as public, they can be accessed from any of the following:

- Outside the scope of a class
- Within the scope of a class
- Another class within the declared class

© Aptech Limited Architecting Web Applications using PHP / Session 13 / 8 of 30

Show slide 8 and tell that the classes and their members are treated public, unless specified. For better understanding, it is a good practice to specify the classes and their members as public.

Unless users specify otherwise, properties and methods of a class are public. That is to say, they may be accessed in three possible situations:

- From outside the class in which it is declared
- From within the class in which it is declared
- From within another class that implements the class in which it is declared

Use on slide text for further explanation.

Additional Information:

Refer to following links for more information:

https://www.w3schools.com/PHP/php_oop_access_modifiers.asp
<https://www.studytonight.com/php/php-access-modifiers>

Ask the students following question. Wait for a response before you answer.

In-Class Question: Can you name the method types that a subclass inherits from the parent class, when the class is extended?

Answer: public and protected methods

Slide 9

Private Members

Code Snippet:

```
<?php
// Define the base class
class Footwear {
    private $price = "We have a fixed
    price of 3000";
    private function show()
    {
        echo "This is a private method
        of a Footwear(base class)";
    }
}

// Define the derived classes
class Sneakers extends Footwear {
    function printPrice()
    {
        echo $this->price;
    }
}

// Create the object of the derived
class
$obj= new Sneakers;

// The given line is trying to call a
private method. Hence, trying to access
private function show() throws error

$obj->show();

// The given line also throws error
since $price is private member of
parent class Footwear
$obj->printPrice();

?>
```

© Aptech Limited

Architecting Web Applications using PHP/ Session 13/ 9 of 30

Show slide 9 and tell students that a class can be defined as private or protected to limit the accessibility of the class members. Private class members can be accessed from within the class. It means a private member can neither be accessed from outside the class in which it is declared nor the inherited class.

Code Snippet demonstrates private members.

In this code, base class Footwear and child class Sneakers are created. In the base class, \$price and show() are private to the base class. \$obj->show() and \$obj->printPrice() are ways of accessing private variables.

Additional Information:

Refer to following links for more information:

<https://stackoverflow.com/questions/4361553/what-is-the-difference-between-public-private-and-protected>

<https://www.geeksforgeeks.org/what-is-the-difference-between-public-private-and-protected-in-php/>

<https://www.educba.com/private-in-php/>

Slide 10

Protected Members

Code Snippet:

```
<?php
// Define the base class
class Footwear {
    protected $price1 = 5000;
    protected $price2 = 15000;
    function total() {
        echo $sum = $this->price1 + $this->price2;
        echo "<br>";
    }
}
// Define the derived classes
class Sneakers extends Footwear {
    function printBill() {
        $tax = 100;
        //Calculating the total + tax
        echo $sub = $this->price1 + $this->price2
        + $tax;
        echo "<br>";
    }
}
```

```
$obj= new Sneakers;
$obj->total();
//This code accesses the function
printBill which has calculation with
protected members of base class namely
$price1 and $price2
$obj->printBill();
?>
```

© Aptech Limited

Architecting Web Applications using PHP/ Session 13/ 10 of 30

Show slide 10 and tell students that a protected member is similar to a private member. Protected members can be accessed from the classes that are derived from a parent class. They can also be accessed in the subclasses in which they are declared. The class member is made protected by adding the `protected` keyword and is not available outside the two kinds of classes.

Code Snippet shows how a class member is declared protected by using the `protected` keyword.

In this code, `$price1` and `$price2` are variables that are declared with `protected` access specifier in base class `Footwear`. Hence, they can be accessed only within the same class or within classes that are inheriting the base class. They cannot be accessed outside the derived class or the base class.

Give following additional information to the students:

Note: Private and protected access specifiers are often used to implement abstraction and encapsulation principles in OOP. For example, when defining built-in libraries, developers may want to give users some access but not full access. They may want to 'protect' some of the functionality from being modified. In such cases, these specifiers are helpful.

Method Overriding

Example:

```

...
function getQuantity() {
    echo $this->quantity. "<br/>";
    return $this->quantity;
}
function getTitle() {
    echo $this->title. "<br/>";
    return $this->title;
}

```

Code Snippet:

```

<?php
// parent class
class Car {
    // public property name
    public $title;
    // public method
    public function drive() {
        echo "Driving the Car<br/>";
    }
}
// child class
class BMW extends Car {
    public function drive() {
        echo "Driving the BMW<br/>";
    }
}
// child class
class HondaCity extends Car {
    public function drive() {
        echo "Driving the Honda City<br/>";
    }
}
$bmw = new BMW();
$bmw->title = "BMW 3 Series (G20/G21)";
// calling child class method
$bmw->drive();
$hmvcity = new HondaCity();
$hmvcity->title = "Honda City V MT";
// calling child class method
$hmvcity->drive();
?>

```

Figure: Output for Code Snippet

Driving the BMW
Driving the Honda City

Architecting Web Applications using PHP/ Session 13/ 11 of 30

Show slide 11 and tell students that method overriding is an important OOP concept in PHP, where both the parent and child classes have the same method name. The main objective is to change the behavior of the parent class method.

Method or function overriding is adopted when the class wants to use the parent class method differently. The method defined in the parent class is overridden with a different definition.

The example on the slide shows overriding `getQuantity()` and `getTitle()` methods to return some values.

In this example, methods `getQuantity()` and `getTitle()` are overridden in the extended class, even though they are already declared in the parent class.

Code Snippet shows how methods can be overridden.

In Code Snippet, the parent class named `Car` has two child classes `BMW` and `Honda City` extending the parent class. In the parent class, the method `drive()` is overridden in the child classes and has different definitions to it.

Figure shows the output for Code Snippet.

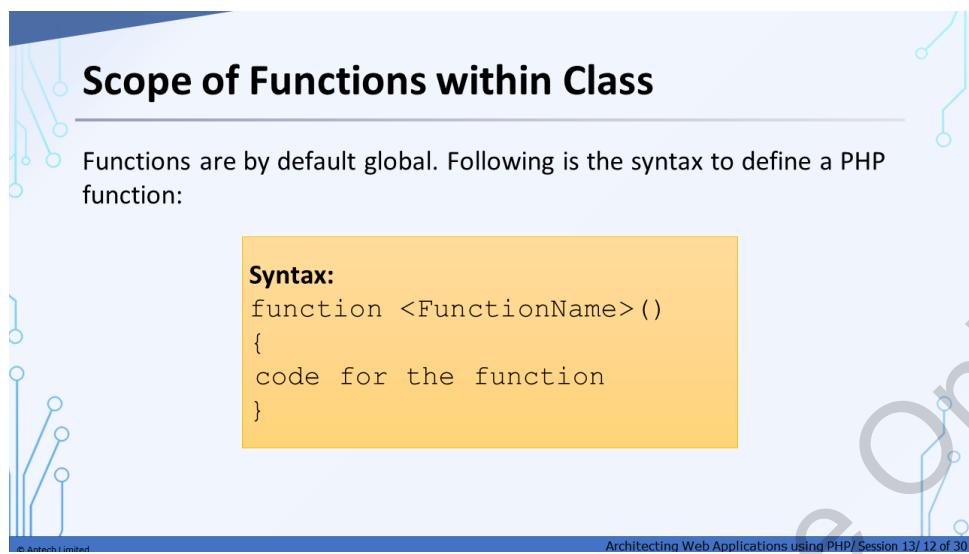
Additional Information:

Refer to following links for more information:

<https://www.phptutorial.net/php-oop/php-override-method/>

<https://www.geeksforgeeks.org/function-overloading-and-overriding-in-php/>

<https://www.tutorialspoint.com/function-overloading-and-overriding-in-php>



Scope of Functions within Class

Functions are by default global. Following is the syntax to define a PHP function:

Syntax:

```
function <FunctionName>()
{
    code for the function
}
```

© Aptech Limited Architecting Web Applications using PHP/ Session 13/ 12 of 30

Show slide 12 and tell students that parameters and return statements can be included in a PHP function. Functions have global scope and can be initiated from anywhere within the program. They can even be called from an instance of a class.
Explain the syntax to define a PHP function from the slide.

Additional Information:

Refer to following links for more information:

<https://phppot.com/php/variable-scope-in-php/>

<https://www.i-programmer.info/programming/php/1130-php-inner-functions-and-closure.html>

Inner Functions

Example:

```
function Function1()
{
    echo("Function1");
    function InnerFunction2()
    {
        echo("Inner Function 2");
    }
}
```

Example:

```
function Function1()
{
    echo("My Function");
    InnerFunction2();
    if(!function_exists("InnerFunction2")){
        function InnerFunction2()
        {
            echo("InnerFunction2 ");
        }
    }
}
```

Example:

```
function Function1()
{
    echo("Function1");
    if(!function_exists("InnerFunction2")){
        function InnerFunction2()
        {
            echo("Inner Function 2");
        }
    }
}
```

© Aptech Limited

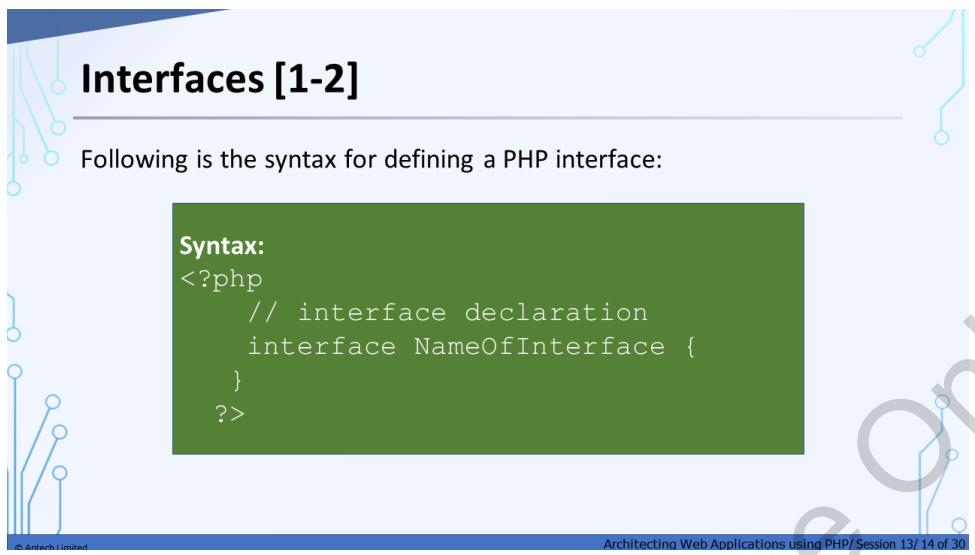
Architecting Web Applications using PHP/ Session 13/ 13 of 30

Show slide 13 and tell students that Inner functions, also known as nested functions, are global functions that behave the same way in which they are declared outside the containing function. For example, the variable defined within the function is local to that function. However, as per the global function rule, the function defined within another function is a local variable.

First example shows how two functions can be defined one within the other. In this example, `InnerFunction2()` will not execute if the outer function `Function1()` is not executed first. This code works only once after calling `Function1`. Calling it again will generate an error.

To create a function more than once; one can use the idea of conditional declaration. This is demonstrated in the second example on the slide. In this example, the `if` condition verifies whether the function name is declared global or not. If it does, then no error occurs. It is always a safe practice to enclose the inner function declarations within `if` statements.

The inner function can be called within the enclosed function only after it is defined. This is demonstrated in the third example on the slide. In this example, `InnerFunction2()` is a function declared and also called inside `Function1()` function.



Interfaces [1-2]

Following is the syntax for defining a PHP interface:

Syntax:

```
<?php
    // interface declaration
    interface NameOfInterface {
    }
?>
```

© Aptech Limited Architecting Web Applications using PHP/ Session 13/ 14 of 30

Show slide 14 and tell students that Interfaces help in defining a blueprint for classes. They contain only public methods and no abstract methods. Interfaces also do not have variables. The classes that inherit the interface must define the methods declared inside the interface. It is possible for interfaces to have constants. Interface constants work exactly similar to class constants. Constants are identifiers whose value cannot change during the execution of the script. If the methods declared in the interface are not implemented, an error will occur. Interfaces are defined in the same way as a class. However, this is done using the `interface` keyword instead of the `class` keyword and without any of the methods having their contents defined.

Additional Information:

Refer to following links for more information:

https://www.w3schools.com/php/php_oop_interfaces.asp
<https://www.geeksforgeeks.org/php-interface/>
<https://www.javatpoint.com/php-oops-interface>

Ask the students following question. Wait for a response before you answer.

In-Class Question: Which keyword is used to define child classes?

Answer: extends

Interfaces [2-2]

Example:

```
// interface declaration
interface ClothingApp {
    // methods declaration
    public function login($phone,
    $password);
    public function register($phone, $password,
    $username);
    public function logout();
}
```

Example:

```
// class declaration
class BestClothing implements ClothingApp
{
    // methods definition
    public function login($mobile,
    $password) {
        echo "Login the Customer with phone
        number: " .
        $mobile;
    }
    public function register($mobile,
    $password,
    $username) {
        echo "Customer registered: Phone
        Number=". $mobile. " and Username=". $username;
    }
    public function logout() {
        echo "Customer logged out!";
    }
}
```

© Aptech Limited

Architecting Web Applications using PHP/ Session 13/ 15 of 30

Show slide 15 and tell students that first example shows how to create an interface with a few methods declared in it. In this example, an interface with name `ClothingApp` is created and has three methods declared in it, that contain different parameters. They are `login()`, `register()`, and `logout()`. These methods do not have any definitions, that is, they do not have any statements and are only declared.

Second example shows the code used to create a class that implements the `ClothingApp` interface. In this example, the interface `ClothingApp` is implemented inside the class `BestClothing` and methods declared in the interface are defined with functionality inside the class.

Implementing Multiple Interfaces

Example:

```
// interface declaration
interface Reward {
    // methods declaration
    public function earnReward($post);
}
```

Example:

```
// class declaration
class BestClothing implements ClothingApp,
Reward {
    // methods definition
    public function login($mobile,
$password) {
        echo "Login with phone number: " .
$mobile;
    }
    public function register($mobile,
$password, $username) {
        echo "Customer registered: Phone
Number
= ".$mobile." and Username=".$username;
    }
    public function logout() {
        echo "Customer logged out!";
    }
    public function earnReward ($post) {
        echo $post." rewards earned!";
    }
}
```

© Aptech Limited

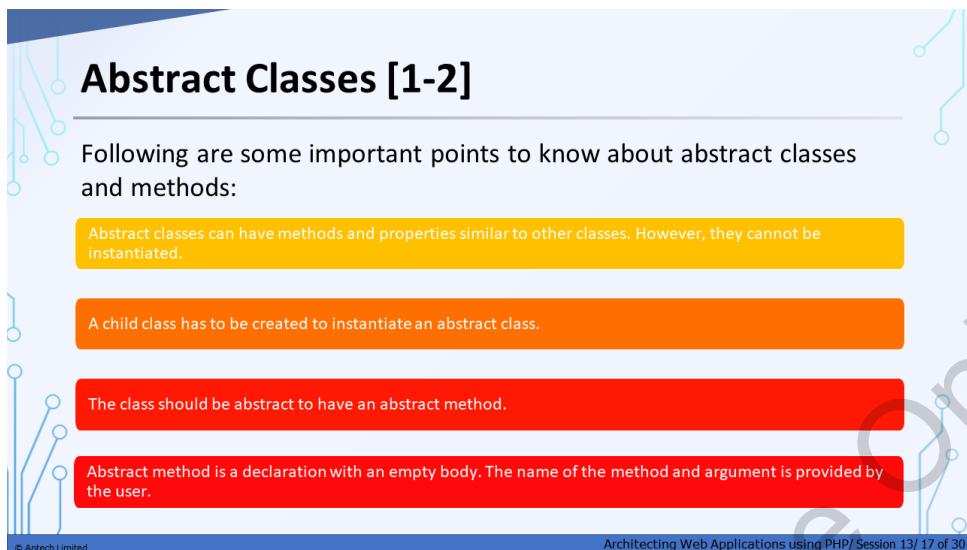
Architecting Web Applications using PHP/ Session 13/ 16 of 30

Show slide 16 and tell students PHP allows a class to implement multiple interfaces. To implement multiple interfaces, the class will have to define methods declared in the interfaces that are implemented by the class. First example shows creating another interface. Here, using the interface keyword, the user can declare interface Reward and a public method earnReward if defined inside the interface.

Second code shows an example of how to inherit from the additional interface in the class BestClothing. In this example, the class BestClothing implements two different interfaces ClothingApp and Reward. Implementation of methods declared inside both interfaces should be done as given here.

Give following additional information to the students:

Note: To execute the code successfully, the code for the interfaces and class must be placed in the same program. Also, note that here, the full functionality for the logic is not yet implemented since, this is just given to illustrate multiple interface inheritance.



Abstract Classes [1-2]

Following are some important points to know about abstract classes and methods:

- Abstract classes can have methods and properties similar to other classes. However, they cannot be instantiated.
- A child class has to be created to instantiate an abstract class.
- The class should be abstract to have an abstract method.
- Abstract method is a declaration with an empty body. The name of the method and argument is provided by the user.

© Aptech Limited Architecting Web Applications using PHP/ Session 13/ 17 of 30

Show slide 17 and tell that while implementing inheritance, it can be seen that one can create object of parent class as well as child class. However, what if the user wants to restrict direct use of the parent class? That is, in some cases, one might want to define a base class that declares the structure of a given entity without giving a complete implementation of every method. Such a base class serves as a generalized form that will be inherited by all of its subclasses. The methods of the base class serve as a contract or a standard that the subclass can implement in its own way.

An abstract class serves as a framework that provides certain behavior for other classes. The subclass provides the requirement-specific behavior of the existing framework. Abstract classes cannot be instantiated and they must be subclassed to use the class members. The subclass provides implementations for the abstract methods in its parent class.

PHP provides the `abstract` keyword to accomplish this task.

Additional Information:

Refer to following links for more information:

Ask the students following question. Wait for a response before you answer.

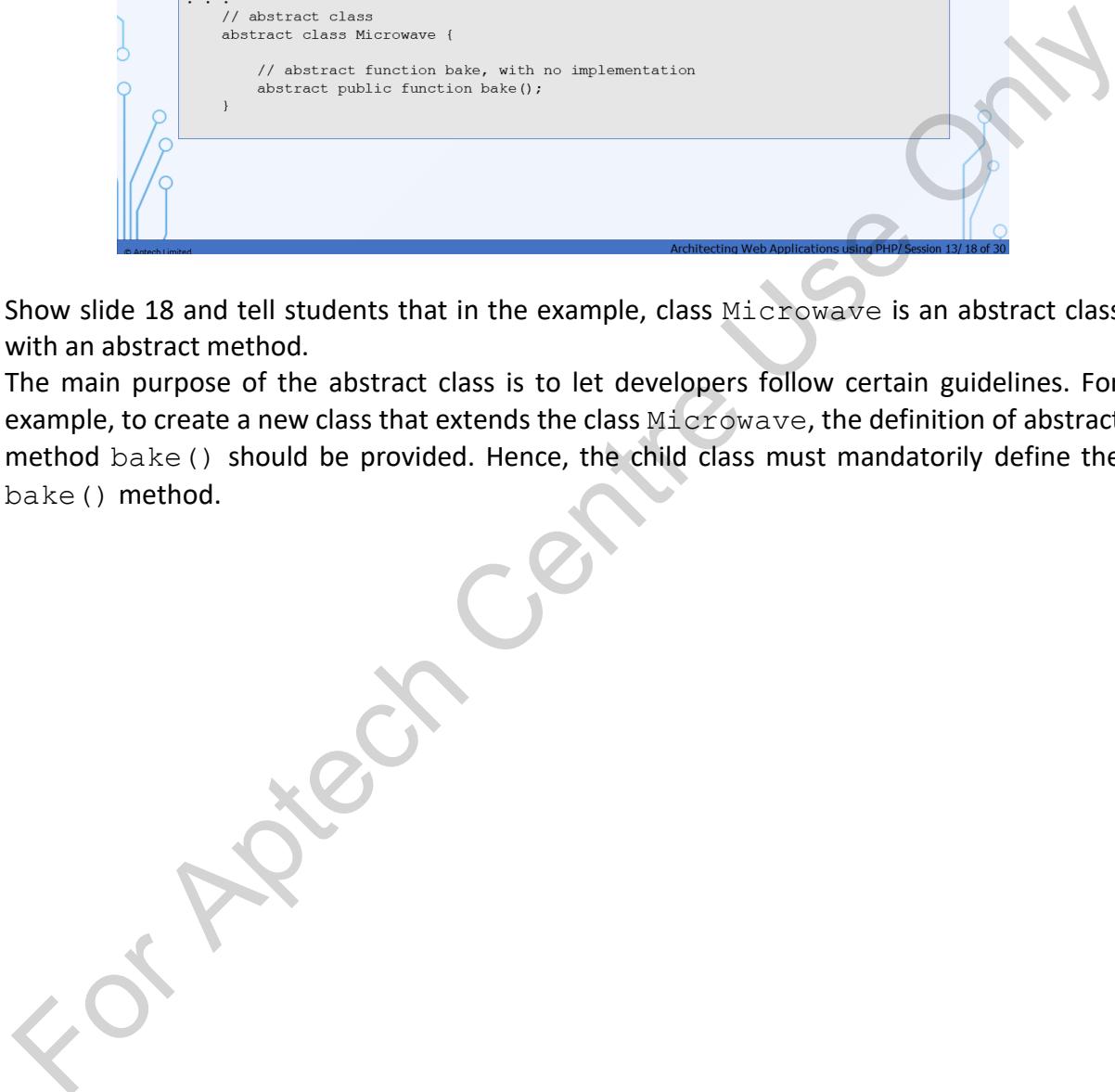
https://www.w3schools.com/php/php_oop_classes_abstract.asp

<https://www.php.net/manual/en/language.oop5.abstract.php>

<https://www.geeksforgeeks.org/abstract-classes-in-php/>

In-Class Question: Which keyword is used to create objects from the class, once the class is defined?

Answer: `new`



Abstract Classes [2-2]

Following is an example to show definition of an abstract class:

Example:

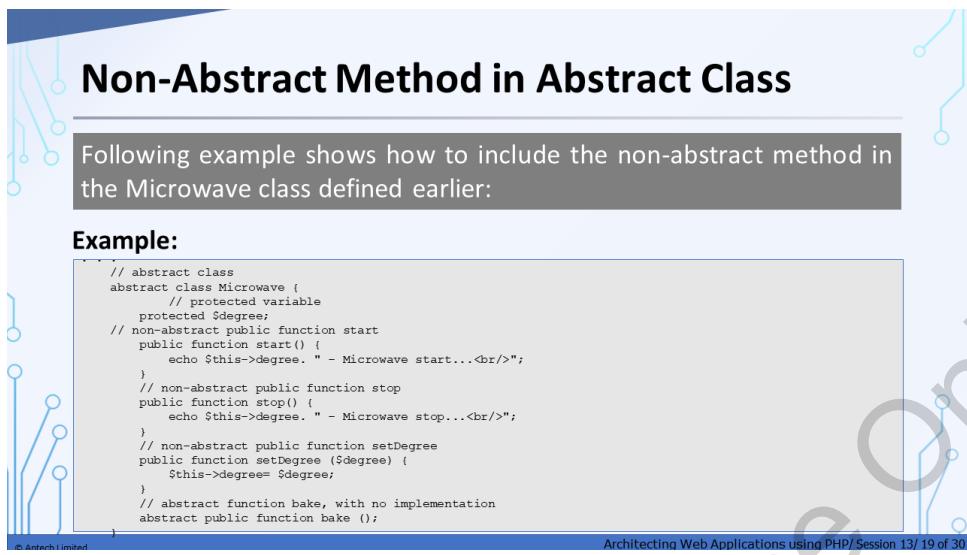
```
// abstract class
abstract class Microwave {
    // abstract function bake, with no implementation
    abstract public function bake();
}
```

© Aptech Limited

Architecting Web Applications using PHP/ Session 13/ 18 of 30

Show slide 18 and tell students that in the example, class `Microwave` is an abstract class with an abstract method.

The main purpose of the abstract class is to let developers follow certain guidelines. For example, to create a new class that extends the class `Microwave`, the definition of abstract method `bake()` should be provided. Hence, the child class must mandatorily define the `bake()` method.



Non-Abstract Method in Abstract Class

Following example shows how to include the non-abstract method in the Microwave class defined earlier:

Example:

```
// abstract class
abstract class Microwave {
    // protected variable
    protected $degree;
    // non-abstract public function start
    public function start() {
        echo $this->degree. " - Microwave start...<br/>";
    }
    // non-abstract public function stop
    public function stop() {
        echo $this->degree. " - Microwave stop...<br/>";
    }
    // non-abstract public function setDegree
    public function setDegree ($degree) {
        $this->degree= $degree;
    }
    // abstract function bake, with no implementation
    abstract public function bake ();
}
```

© Aptech Limited

Architecting Web Applications using PHP/ Session 13/ 19 of 30

Show slide 19 and tell students that an abstract method can have both abstract and non-abstract methods. A class with an abstract method must be declared abstract, which can be accessed by the child class without overriding.

In the example, three non-abstract methods namely start(), stop(), and setDegree() are added to abstract Microwave class.

Additional Information:

Refer to following links for more information:

<https://phppenthusiast.com/object-oriented-php-tutorials/abstract-classes-and-methods>

<https://tutorials.supunkavinda.blog/php/oop-abstract-classes-methods>

Inheriting Abstract Classes [1-2]

Example:

```
// child class
class Onida extends Microwave {
    public function bake() {
        echo "This version baking
temperature is - 180
to 300 degrees celsius";
    }
}
```

Example:

```
// child class
class Samsung extends Microwave {
    public function bake() {
        echo " This version baking temperature
is -
200 to 400 degrees celsius ";
    }
}
```

Example:

```
$samsung = new Samsung ();
$samsung ->setDegree("300");
$samsung ->bake();
```

© Aptech Limited Architecting Web Applications using PHP/ Session 13/ 20 of 30

Show slide 20 and tell students that similar to other classes, abstract classes can be inherited too. The child class should define the abstract method defined as abstract class. On the contrary, the child class should be defined as abstract; it does not define the abstract method.

First example shows how to create two child classes that inherit the class Car with definition for the abstract method bake(). In this example, Onida is the child class which extends the parent class Microwave. Now, the child class Onida is responsible for defining the code of the bake() method.

The abstract class can have as many as child classes. Second example shows this. In this example, class Samsung is the child class extending the parent class Microwave.

An abstract class can have objects indirectly – via child classes. Once a child class is defined, an object for it can be created. Third example shows this. The example shows object creation in PHP. Here, the object \$samsung is of class Samsung.

Slide 21

The slide features a blue header with the title 'Inheriting Abstract Classes [2-2]'. Below the title is a 'Code Snippet' section containing two blocks of PHP code. The first block defines an abstract class 'Microwave' with methods 'start()', 'stop()', and 'setDegree()'. The second block defines a child class 'Samsung' extending 'Microwave' and implementing its 'bake()' method. A screenshot of a browser window shows the output of the code execution.

```
<?php
// abstract class
abstract class Microwave {
    // protected variable
    protected $degree;
    // non-abstract public function start
    public function start() {
        echo $this->degree. " - Microwave
start...<br/>";
    }
    // non-abstract public function stop
    public function stop() {
        echo $this->degree. " - Microwave
stop...<br/>";
    }
    // non-abstract public function setDegree
    public function setDegree($degree) {
        $this->degree = $degree;
    }
    // abstract function bake
    abstract public function bake();
}

// child class
class Samsung extends Microwave {
    public function bake() {
        echo "This version baking
temperature is - 200 to
400 degrees celsius ";
    }
}
$Samsung = new Samsung();
$Samsung->setDegree("400");
$Samsung->bake();?
```

This version baking temperature is - 200 to 400 degrees celsius

Figure: Output for Code Snippet

Architecting Web Applications using PHP/ Session 13/ 21 of 30

Show slide 21 and tell students that Code Snippet shows the complete code for the examples discussed in slide 20. In Code Snippet, if the user tries to create an object of class microwave, an error will occur, as the class microwave is declared as abstract. However, a new class Samsung can extend that abstract class Microwave and implement the abstract method declared in parent class. Figure shows the output for Code Snippet.

Additional Information:

Refer to following links for more information:

<https://www.javatpoint.com/php-oops-abstract-class>
<https://www.php.net/manual/en/language.oop5.abstract.php>

Slide 22

Interface vs Abstract Class

Following are some differences between an interface and an abstract class:

- While an interface cannot have concrete methods and have only abstract methods, an abstract class can have both.
- In an interface, the user can only declare the methods as public. However, in an abstract class, the user can declare methods as public, private, or protected.
- In a class, though multiple interfaces can be implemented, only one abstract class can be implemented.

© Aptech Limited Architecting Web Applications using PHP/ Session 13/ 22 of 30

Using slide 22, explain students the differences between an interface and an abstract class.

Additional Information:

Refer to following links for more information:

<https://www.c-sharpcorner.com/article/what-is-the-difference-between-abstract-class-and-interface-in-php/>

<https://codeinphp.github.io/post/abstract-class-vs-interface/>

<https://betterprogramming.pub/php-a-practical-demonstration-of-interfaces-vs-abstract-classes-56d9838cd5b7>

<https://ashallendesign.co.uk/blog/interfaces-vs-abstract-classes-in-php>

Ask the students following question. Wait for a response before you answer.

In-Class Question: Which is better to use abstract class or interface?

Answer: Abstract classes should be **used primarily for objects that are closely related**, whereas interfaces are best suited for providing a common functionality to unrelated classes. Interfaces are a good choice when users think that the API will not change for a while.

Slide 23

The slide has a blue header bar with the title 'final Keyword'. Below it is a section titled 'Code Snippet:' containing the following PHP code:

```
<?php
class Bill {
    public function check() {
        echo "Bill::check() called<br>";
    }
    final public function applytax() {
        echo "Bill::applytax() called<br>";
    }
}
class RentBill extends Bill {
    public function applytax() {
        echo "ChildClass::applytax() called<br>";
    }
}
$rentBill = new RentBill();
$rentBill->applytax();
?>
```

To the right of the code snippet is a screenshot of a browser window showing the error output:

Fatal error: Cannot override final method Bill::applytax() in C:\xampp\htdocs\updated6.php on line 15

Below the slide content is a footer bar with the text 'Architecting Web Applications using PHP/ Session 13/ 23 of 30'.

Figure: Output for Code Snippet

Show slide 23 and tell students that the keyword `final` helps declare a method as final. It prevents a class from being inherited or overridden. This keyword when used with a method prevents a child class from overriding a method.

Code Snippet shows an example where a final method cannot be overridden. In Code Snippet, an error occurs because overriding of final method is not permitted in `RentBill`. Figure shows the output for Code Snippet.

Additional Information:

Refer to following links for more information:

- <https://www.geeksforgeeks.org/final-keyword-in-php/>
- <https://www.php.net/manual/en/language.oop5.final.php>
- https://www.w3schools.com/php/keyword_final.asp
- <https://www.javatpoint.com/php-oops-final-keyword>

Static Members [1-2]

Following is the syntax to create a static method:

Syntax:
<?php
class ClassName {
 public static function <staticMethod>() {
 <code>
 }
}>

To access a static method use the class name, a double colon (::), and the method name. Following is the syntax:

Syntax:
ClassName::staticMethod();

© Aptech Limited

Architecting Web Applications using PHP/ Session 13/ 24 of 30

Show slide 24 and tell students that in PHP, static members are created using static keyword. Static members including methods can be accessed without instantiating the class. All such static members are bound to the class and not individual objects.

The static keyword can be used to declare properties and methods of a class as static. Static methods can be accessed without instantiating the class. However, a static member cannot be accessed using an instantiated class object. The scope resolution operator (::) is used to access static properties. However, the same cannot be done using the object operator (->).

A class can be referenced using a variable. However, the value of the variable should not be a keyword such as self, parent, static, and so on.

Static Members [2-2]

Example:

```

    . .
    class Test {
        public static $my_static =
    "test";
        public function staticValue() {
            return self::$my_static;
        }
    }
    print Test::$my_static. "<br>";
    $test = new Test();
    print $test->staticValue() . "<br>";
?>

```

Code Snippet:

```

<!DOCTYPE html>
<html>
<body>
<?php
class greeting {
    public static function welcome() {
        echo "Warm wishes for the season..";
    }
}
// Call static method
greeting::welcome();
?>
</body>
</html>

```

Figure: Output for Code Snippet

© Aptech Limited

Architecting Web Applications using PHP/ Session 13/ 25 of 30

Show slide 25 and tell students that the example on the slide shows how to use the static keyword. In this example, to access `my_static` and `staticvalue` methods, there is no requirement to instantiate new class of `Test`. Instead, they can be accessed as shown here. Code Snippet shows an example where the static members are accessed. In Code Snippet, a static method `welcome()` is declared. Then, the static method is called by using the class name `greeting`, a double colon (::), and the method name `welcome()` (without creating an instance of the class first).

Figure shows the output for Code Snippet.



Method Overloading

Following example shows method overloading:

Example:

```
...
class Model {
    public function __call($name, $arguments) {
        echo "This is object method '$name' "
            . implode(',', ', ', $arguments). "<br>";
    }
    public static function __callStatic($name, $arguments) {
        echo "This is static method '$name' "
            . implode(',', ', ', $arguments). "<br>";
    }
}
// Create new object
$obj = new Model;
$obj->runModel("Object context");
Model::runModel("Static context");
```

© Aptech Limited

Architecting Web Applications using PHP/ Session 13/ 26 of 30

Show slide 26 and tell students that the process of overloading the properties and methods with different functionality is known as method/property overloading respectively.

In PHP, method overloading allows you to create numerous methods with the same name but different functioning. Using method overloading, the user can create dynamic methods, which are not declared inside the scope of the class. Method overloading is meant to create methods dynamically. The overloaded methods are defined as public. The entities (methods and properties) can be accessed, once the object for a class is created.

In the example, `Model` class is defined in which magic functions have been created. These functions are `__call()` and `__callStatic()`. `__call()` is triggered while invoking overloaded methods in the object context. `__callStatic()` is triggered while invoking overloaded methods in static context. One can access the functions through the new object, using respective arguments to call in object context and static context.

Additional Information:

Refer to following links for more information:

<https://www.geeksforgeeks.org/overloading-in-php/#:~:text=Method%20Overloading%3A%20It%20is%20a,both%20object%20and%20static%20context.>

<https://www.php.net/manual/en/language.oop5.overloading.php>

<https://www.tutorialspoint.com/what-is-method-overloading-in-php>

Property Overloading [1-2]

Following are some functions used in property overloading:

| | |
|-----------|---|
| __set() | • To initialize overloaded properties |
| __get() | • To read data from inaccessible properties |
| __isset() | • To check the overloaded properties |
| __unset() | • To be invoked when used for overloaded properties |

© Aptech Limited Architecting Web Applications using PHP/ Session 13/ 27 of 30

Show slide 27 and explain that property overloading in PHP allows creation of context-based dynamic properties. It does not require a separate line of code to create properties. An overloaded property is often neither associated with a class instance nor is it declared within the scope of the class.

- `__set()` is run when writing data to inaccessible (protected or private) or non-existing properties.
- `__get()` is utilized for reading data from inaccessible (protected or private) or non-existing properties.
- `__isset()` is triggered by calling `isSet()` or `empty()` on inaccessible (protected or private) or non-existing properties.
- `__unset()` is invoked when `unset()` is used on inaccessible (protected or private) or non-existing properties.

Additional Information:

Refer to following links for more information:

- <https://www.geeksforgeeks.org/overloading-in-php/#:~:text=Property%20Overloading%3A%20PHP%20property%20overloading,is%20considered%20as%20overloaded%20property.>
- <https://stackoverflow.com/questions/8711987/overloading-properties-and-methods-in-php-whats-the-reason>
- https://docstore.mik.ua/oreilly/webprog/pcook/ch07_09.htm
- <https://medium.com/oceanize-geeks/phps-overloading-example-a432dd130169>
- <https://www.javatpoint.com/php-oops-overloading>

Ask the students following question. Wait for a response before you answer.

In-Class Question: What is \$Self in PHP?

Answer: self is used to access static or class variables or methods and this is used to access non-static or object variables or methods. So use self when there is a requirement to access something which belongs to a class and use \$this when there is a requirement to access a property belonging to the object of the class.

Slide 28

The slide has a decorative background with blue and white circuit board patterns on the left and right sides. The title 'Property Overloading [2-2]' is centered at the top in a large, bold, black font. Below the title is a yellow horizontal bar containing the text 'Following example shows property overloading:'. The main content area is divided into two sections: 'Example:' on the left and code snippets on the right. The 'Example:' section contains a brief description of the code's purpose. The code snippets show the implementation of __set, __get, __isset, __unset, and __getHidden methods within a Model class, along with a demonstration of how these methods interact with an object instance \$obj.

```
class Model {
    private $data = array();
    public $declared = 1;
    // Overloading used when accessed outside the class
    private $hidden = 2;
    // Function definition
    public function __set($name, $value) {
        echo "Set members '$name' to '$value'<br>";
        $this->data[$name] = $value;
    }
    // Function definition
    public function __get($name) {
        echo "Get members '$name':";
        if (array_key_exists($name, $this->data)) {
            return $this->data[$name];
        }
        $trace = debug_backtrace();
        return null;
    }
    // Function definition
    public function __isset($name) {
        echo "Is '$name' trying to set?<br>";
        return isset($this->data[$name]);
    }
}

// Definition of __unset function
public function __unset($name) {
    echo "Trying to Unset '$name'<br>";
    unset($this->data[$name]);
}
// getHidden function definition
public function getHidden() {
    return $this->hidden;
}

// Create an object
$obj = new Model();
// Set value 1 to the object variable
$obj->a = 1;
echo $obj->a . "<br>";
// Use isset function to check 'a' is set or not var_dump(isset($obj->a));
// Is 'a' set?
unset($obj->a);
var_dump(isset($obj->a));
echo $obj->declared . "<br>";
echo "Private properties are allowed to access inside the class ";
echo $obj->getHidden() . "<br>";
echo "Private properties are not allowed to access outside of class<br>";
echo $obj->hidden . "<br>";

Architecting Web Applications using PHP / Session 13 / 28 of 30
```

Show slide 28 and tell students that in the example, class Model is created in which an array has been created to locate overloading data. A variable declared has been created, along with one private variable hidden. The functions __set, __get, __isset, __unset, and get hidden functions are defined in Model class. An object \$obj is created, instantiating the Model class. Then, the code sets value of 1 to the object variable. The iset function is used to check a is set or not. var_dump () function is used to dump information about one or more variables.

Private properties are allowed to access inside the class when declared is used. However, private properties are not allowed to access outside of class when hidden in used.

Slide 29

Magic Methods

- `__set()` is triggered when overloaded properties are initialized.
- `__get()` is triggered when overloaded properties are used with PHP print statements.
- `__isset()` is a magic method that is invoked when overloaded properties are checked with the `isSet()` function.
- `__unset()` is a function that is invoked on using PHP `unset()` for overloaded properties.

© Aptech Limited Architecting Web Applications using PHP / Session 13 / 29 of 30

Show slide 29 and tell that In PHP, method overloading also creates magic methods. Magic methods are triggered in object context and used to create dynamic entities. Some magic methods used in PHP are mentioned in the slide.

Additional Information:

Refer to following links for more information:

<https://www.php.net/manual/en/language.oop5.magic.php>
<https://www.phptutorial.net/php-oop/php-magic-methods/>
https://www.tutorialspoint.com/php_magic_methods

Ask the students following question. Wait for a response before you answer.

In-Class Question: Which magic method is used to implement overloading PHP?

Answer: In PHP, function overloading is done with the help of magic function `__call()`. This function takes function name and arguments.



Summary

- Variables and member methods of a class can be accessed through an object of the class by using the `→` operator.
- Access modifiers are keywords that decide the accessibility for various class methods, variables, and other member methods.
- Inheritance allows users to reduce code duplication by creating a new class with properties and methods inherited from other classes. The child class inherits the properties and methods of the parent class.
- An interface can be considered a blueprint of a class and allows the user to specify what all methods and properties a class should implement.
- Abstract class is a class that defines a structure for other classes to extend. An abstract class cannot be instantiated, has at least one abstract method, and can contain method definitions.
- Overloading is an important feature in PHP that allows users to dynamically create and use methods and properties.

Architecting Web Applications using PHP/ Session 13/ 30 of 30

Use slide 30 to summarize the session. You will end the session, with a brief summary of what has been taught in the session. Tell the students pointers of the session. This will be a revision of the current session.

13.3 Post Class Activities for Faculty

You should familiarize yourself with the topics of the next session.

Tips:

You can also check the Articles/Blogs/Expert Videos uploaded on the Online Varsity site to gain additional information related to the topics covered in the next session.

Session 14 – PHP Web Concepts

14.1 Pre-Class Activities

Before beginning the session, you should fully familiarize yourself with its topics. Prepare a question or two that will be a key point to relate the current session objectives.

14.1.1 Teaching Skills

To teach this session, you should be well-versed with Web concepts of PHP. You must use the given images and teach the concepts in the theory class. For teaching in the class, you are expected to use slides and LCD projectors.

Tips:

It is recommended that you test the understanding of the students by asking questions in between the class.

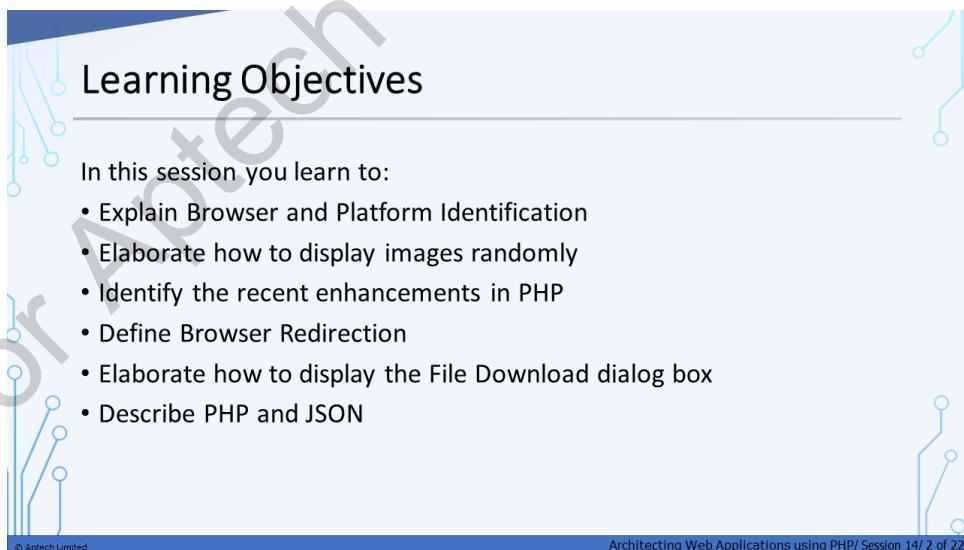
In-Class Activities

Follow the order given here during In-Class activities.

Overview of the Session

Give the students an overview of the current session in the form of session objectives. Read out the objectives on slide 2.

Slide 2



The slide has a light blue background with a decorative border of blue and white circuit board patterns. At the top left, the text "Learning Objectives" is displayed. Below it, a horizontal line separates the title from the content. The content area contains the following text and list:

In this session you learn to:

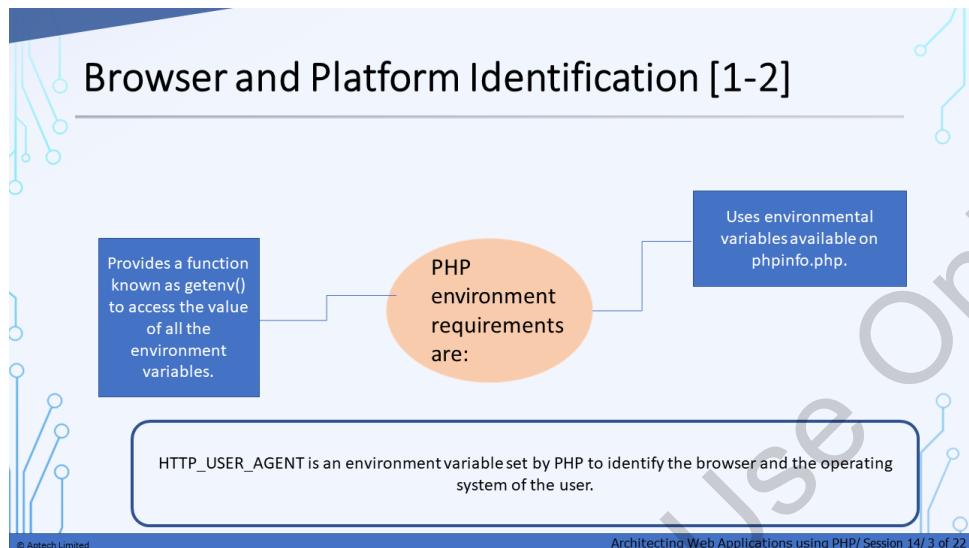
- Explain Browser and Platform Identification
- Elaborate how to display images randomly
- Identify the recent enhancements in PHP
- Define Browser Redirection
- Elaborate how to display the File Download dialog box
- Describe PHP and JSON

At the bottom left, there is a small copyright notice: "© Aptech Limited". At the bottom right, the slide footer reads: "Architecting Web Applications using PHP / Session 14 / 2 of 22".

Show Slide 2 and give overview of Browser and Platform Identification and explain how to display images randomly. Tell the recent enhancements in PHP and say about Browser Redirection. Explain how to display the File Download dialog box and describe PHP and JSON.

14.2 In-Class Explanations

Slide 3



Show Slide 3 and explain to the students about Browser and Platform Identification. Say that PHP uses environmental variables available on `phpinfo.php` and it also provides a function known as `getenv()` to access the value of all the environment variables. Then, say about the `HTTP_USER_AGENT`, environment variable.

Additional Information:

Refer to following links for more information:

- <https://www.php.net/manual/en/tutorial.useful.php>
- https://www.w3schools.com/php/func_misc_get_browser.asp

Browser and Platform Identification [2-2]

Code snippet:

```
<html>
<body>
<?php
$viewer = getenv( "HTTP_USER_AGENT" );
$browser = "An unidentified browser";
if( preg_match( "/MSIE/i", "$viewer" ) )
{
$browser = "Internet Explorer";
}
else if( preg_match( "/Netscape/i", "$viewer" ) )
{
$browser = "Netscape";
}
else if( preg_match( "/Mozilla/i", "$viewer" ) )
{
$browser = "Mozilla";
}
$platform = "An unidentified OS!";
if( preg_match( "/Windows/i", "$viewer" ) )
{
$platform = "Windows!";
}
else if( preg_match( "/Linux/i", "$viewer" ) )
{
$platform = "Linux!";
}
echo("You are using $browser on $platform");
?>
</body>
</html>
```

© Aptech Limited Architecting Web Applications using PHP / Session 14 / 4 of 22

Using the code snippet on Slide 4, explain how to identify the browser and operating system of the user. Say that the output for code snippet displays the browser as Google Chrome and operating system as Windows. Output on other systems might vary based on the browser.

Ask the following question to the students. Wait for the response before you answer.

In-Class Question: State the use of the `preg_match()` function in the code snippet.

Answer: The `preg_match()` function helps to identify the browser which can be either Internet Explorer, Mozilla, or Netscape or any other browser such as Google Chrome. Further, it includes the code to check the operating system of the user, which can be either Linux or Windows.

Displaying Images Randomly

rand () function

- Used to generate a random number within a range.

srand () function

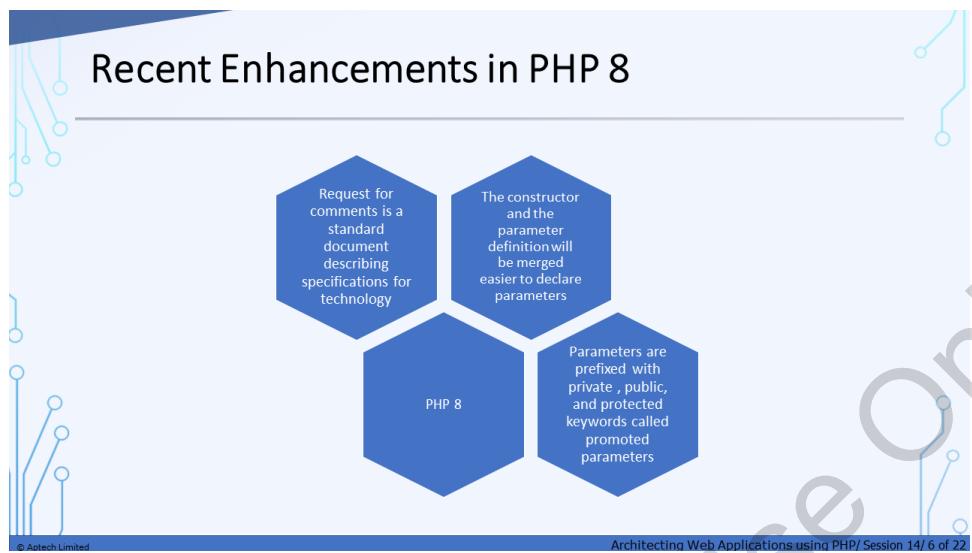
- Used to generate a random number that sets the seed number as its argument.

Show Slide 5 and explain to the students about the PHP functions: `rand()` function and `srand()` function. Tell the uses of these functions. Explain that `rand()` function is used to generate a random number within a range and `srand()` function is used to generate a random number that sets the seed number as its argument.

Additional Information:

Refer to following links for more information:

- https://www.w3schools.com/php/func_math_rand.asp
- https://www.w3schools.com/php/func_math_srand.asp



Show Slide 6 and explain about the recent enhancements in PHP 8. Tell that as per PHP 8 Request for comments, the constructor and the parameter definition will be merged to make it easier to declare parameters and Request for Comments is a standard document describing specifications for a technology.

Say that a Request for Comments (RFC) is a document describing specifications for a technology and become a standards document. Moreover, only parameters prefixed with private, public, and protected keywords, which are called promoted parameters can be defined and merged with the constructor as per the RFC.

Arrays Starting With a Negative Index

- With PHP 8, arrays beginning with negative index change their behavior.
- The remaining indices will start from 0.

Example:

```
$a = array_fill(-6, 5, true);  
var_dump($a);
```

© Aptech Limited Architecting Web Applications using PHP / Session 14 / 7 of 22

Show Slide 7 and explain about the arrays beginning with a negative index change their behavior with PHP 8. Tell them with examples. Say that in the second index will change as start index + 1, irrespective of start index value.

Additional Information:

Refer to following links for more information:

- https://wiki.php.net/rfc/negative_array_index
- <https://linuxprograms.wordpress.com/2011/10/20/php-arrays-negative-index/>
- <https://www.geeksforgeeks.org/move-negative-numbers-beginning-positive-end-constant-extra-space/>

Slide 8



Nullsafe Operator

- PHP 8 introduces nullsafe operator `$->`.
- If an operator in a chain evaluates to `null`, then the execution of the chain stops and results in `null`.

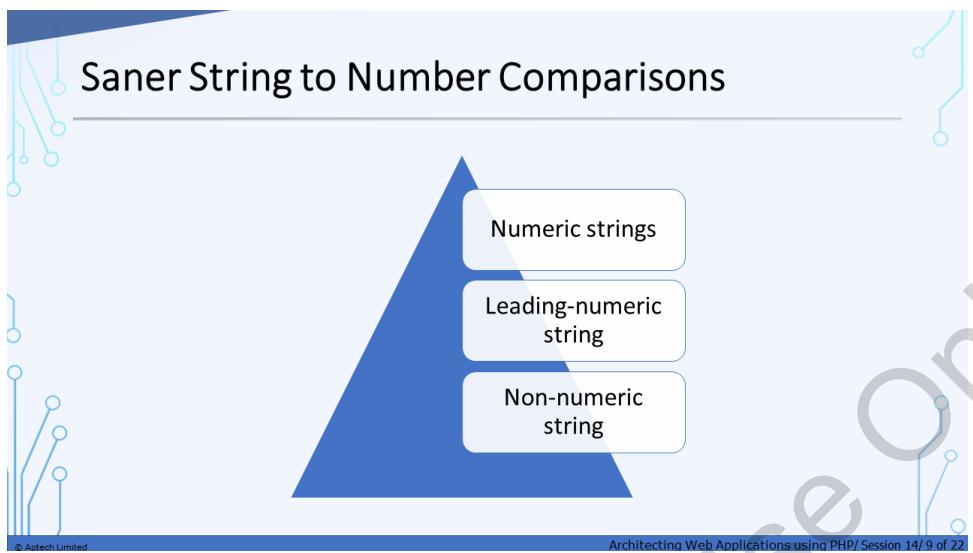
© Aptech Limited Architecting Web Applications using PHP/ Session 14/ 8 of 22

Show Slide 8 and explain about the null safe operator which is a new syntax in PHP 8. Explain that the null safe operator returns its first operator if it exists and is not null, otherwise it returns its second operator.

Additional Information:

Refer to following links for more information:

- <https://php.watch/versions/8.0/null-safe-operator>
- <https://www.tutorialspoint.com/nullsafe-operator-in-php-8>
- https://wiki.php.net/rfc/nullsafe_operator



Show Slide 9 and explain the students that in PHP string can be categorized in three ways – Numeric strings that contains only numbers. Leading – numeric string starts with a numeric strings followed with non – numeric characters and the third on Non – numeric string which cannot be numeric and also a non-leading numeric string. Tell the students that PHP 8 combines various numeric string modes into a single concept. Any type of string is considered non-numeric and will throw `TypeError`.



Trailing Comma in Parameter List

- Trailing commas are commas that are appended to the list of items during various circumstances.
- Trailing commas were introduced in list syntax, in PHP 7.2 and PHP 7.3.
- The trailing commas in parameter lists with methods, functions, and closures is introduced in PHP 8 version.

© Aptech Limited

Architecting Web Applications using PHP/ Session 14/ 10 of 22

Show Slide 10 and explain about the trailing comma in parameter list in PHP 8 version. Say that it was in list syntax in PHP 7.2 and PHP 7.3 in function calls whereas in PHP 8 it is used in parameter lists with methods, functions, and closures.

Ask the following question to the students. Wait for the response before you answer.

In-Class Question: What is trailing Comma in PHP?

Answer: Trailing commas are commas that are appended to the list of items during various circumstances.

Consistent Type Errors for Internal Functions

Behavior of internal and user-defined functions when illegal Parameter type is passed:

To retrieve the name of a class, use the syntax:
`Foo\Bar::class`

The diagram consists of two blue rounded rectangles. The top one is labeled "Internal functions" and the bottom one is labeled "User-defined functions". A large blue arrow points from left to right, with the text "Shows warning and returns null" positioned above the arrow next to the internal functions box, and "Shows TypeError" positioned below the arrow next to the user-defined functions box.

© Aptech Limited Architecting Web Applications using PHP / Session 14 / 11 of 22

Show Slide 11 and explain about the consistent type of Errors for Internal functions. Say that whenever an illegal parameter type is passed the internal functions throw a warning and return null. Say that with PHP 8, the same syntax is extended to the objects as well. Therefore, it becomes much easier to retrieve the name of the class for a given object.



Browser Redirection [1-2]

- One of the functionalities that can be done smoothly in PHP.
- PHP uses `header()` function.
- The `header()` function of PHP helps to redirect the browser.
- By providing the raw HTTP headers to it.
- It helps to redirect to a different location.



© Aptech Limited Architecting Web Applications using PHP/ Session 14/ 12 of 22

Show Slide 12 and discuss about Browser Redirection. Say that it is one of the functionalities that can be done smoothly in PHP and it uses `header()` function which helps to redirect the browser.

Ask the following question to the students. Wait for the response before you answer.

In-Class Question: What is browser redirection in PHP?

Answer: PHP redirect mechanism is used to navigate the user from one page to another without clicking any hyperlinks.

Additional Information:

Refer to following link for more information:

<http://www.slideshare.net>php-for>

Slide 13

Browser Redirection [2-2]

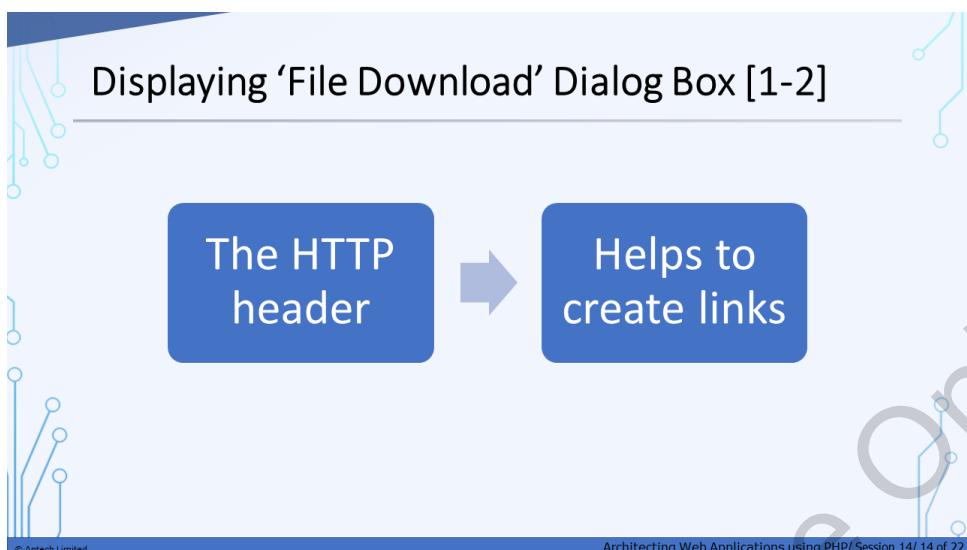
Code snippet:

```
<?php  
error_reporting(0);  
if($_POST["location"]){  
$location = $_POST["location"];  
header("Location:$location");  
exit();  
}  
>  
<html>  
<body>  
<p>Select a Website to visit :</p>  
<form action=<?php $PHP_SELF ?>"  
method="POST">  
  
<select name="location">  
<option value="http://w3c.org">  
World Wide Web Consortium  
</option>  
<option value="http://www.google.com">  
Google Search Page  
<option value="http://www.sample.com">  
Php tutorial page  
</option>  
</select>  
<input type="submit" value="Submit"/>  
</form>  
</body>  
</html>
```

© Aptech Limited Architecting Web Applications using PHP / Session 14 / 13 of 22

Say that Browser Redirection is one of the functionalities that can be done smoothly in PHP. PHP uses `header()` function to accomplish this. Show Slide 13 and explain the code snippet that shows the source code used to redirect the browser request to a different Web page.

Say that when the code is executed in the browser, user can choose the site he/she wants to visit. When user selects a particular option and clicks submit button, he/she will be redirected to that specific Website.



Using Slide 14 explain the students about the other functionality, the HTTP header which helps to create links. Here, Content-Type must be sent as text/html\n\n and the HTTP header will not be the actual header.

Additional Information:

Refer to following links for more information:

- <https://www.ostraining.com/blog/coding/download-php/>
- <https://educatech.in/how-can-you-display-a-file-download-dialog-box-using-php/>
- https://www.tutorialspoint.com/how_to_force_file_download_with_php

Slide 15



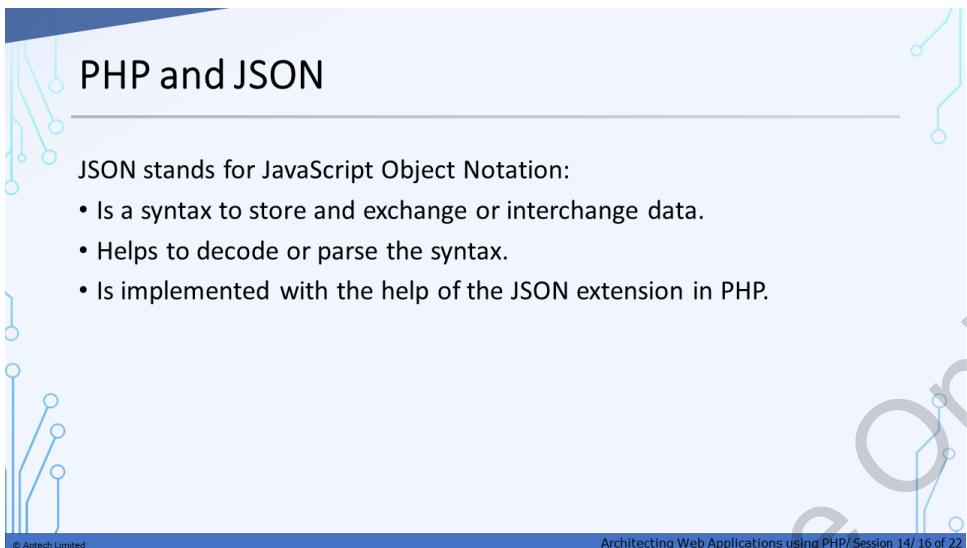
Displaying 'File Download' Dialog Box [2-2]

Code snippet:

```
#!/usr/bin/perl
# HTTP Header
print "Content-Type:application/octet-stream; name=\"FileName\"\r\n";
print "Content-Disposition: attachment; filename=\"FileName\"\r\n\r\n";
# Actual File Content
open( FILE, "<FileName" );
while(read(FILE, $buffer, 100) ){
print("$buffer");
}
```

© Aptech Limited Architecting Web Applications using PHP / Session 14 / 15 of 22

Using Slide 15 explain the syntax for making a FileName file downloadable from a link. Say that the content-type given in the syntax is application/octet-stream. The original filename will be included along with it.



PHP and JSON

JSON stands for JavaScript Object Notation:

- Is a syntax to store and exchange or interchange data.
- Helps to decode or parse the syntax.
- Is implemented with the help of the JSON extension in PHP.

© Aptech Limited Architecting Web Applications using PHP/ Session 14/ 16 of 22

Show Slide 16 and tell the students about PHP and JSON. Say that JSON stands for JavaScript Object Notation. Explain that it is a syntax to store and exchange or interchange data.

Ask the following question to the students. Wait for the response before you answer.

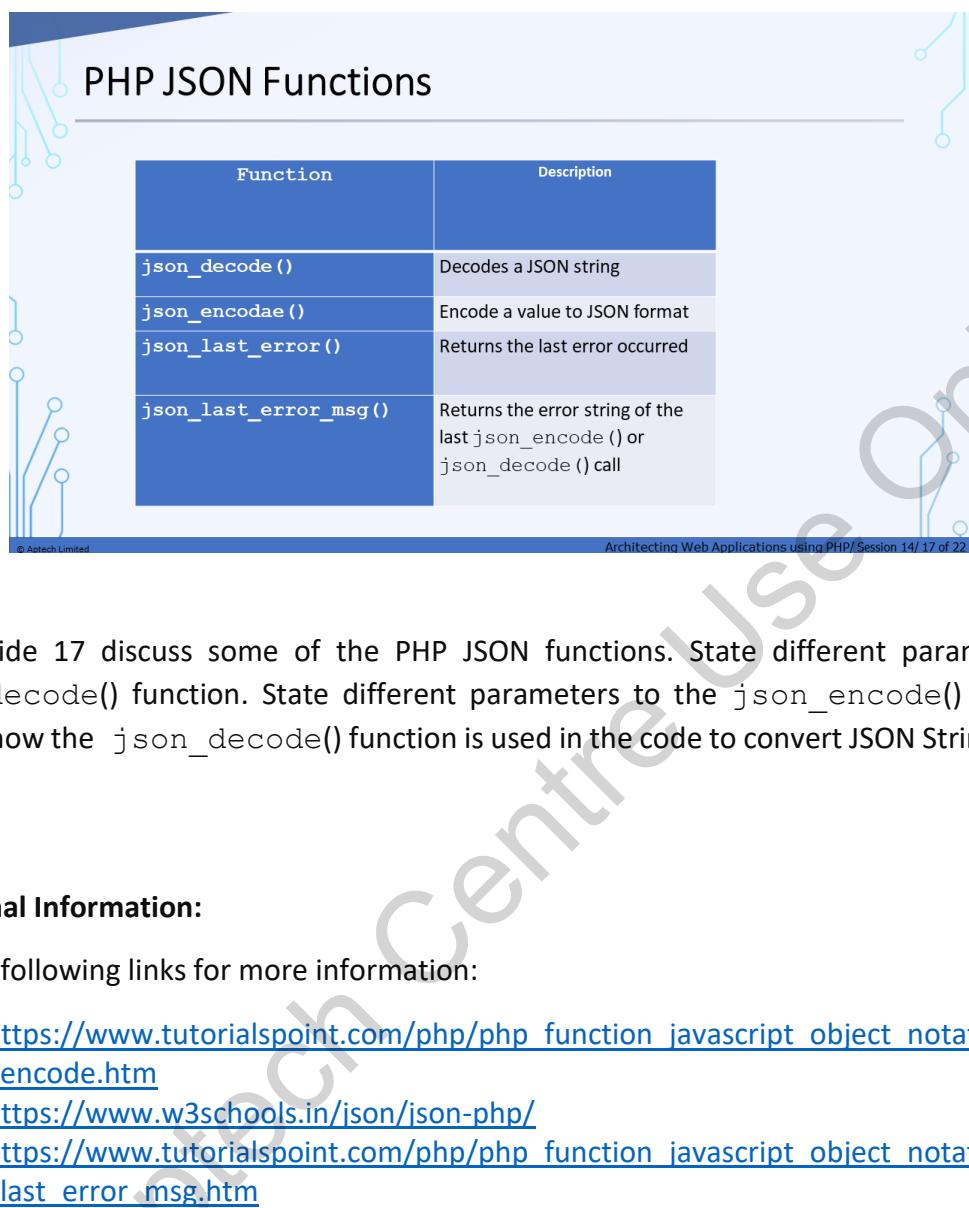
In-Class Question: What is the JSON in PHP?

Answer: JSON stands for JavaScript Object Notation and is a syntax for storing and exchanging data.

Additional Information:

Refer to following link for more information:

- <http://www.tutorialspoint.com>



PHP JSON Functions

Function	Description
<code>json_decode()</code>	Decodes a JSON string
<code>json_encode()</code>	Encode a value to JSON format
<code>json_last_error()</code>	Returns the last error occurred
<code>json_last_error_msg()</code>	Returns the error string of the last <code>json_encode()</code> or <code>json_decode()</code> call

© Aptech Limited Architecting Web Applications using PHP / Session 14 / 17 of 22

Using Slide 17 discuss some of the PHP JSON functions. State different parameters in `json_decode()` function. State different parameters to the `json_encode()` function. Explain how the `json_decode()` function is used in the code to convert JSON String to PHP variable.

Additional Information:

Refer to following links for more information:

- https://www.tutorialspoint.com/php/php_function_javascript_object_notation_json_encode.htm
- <https://www.w3schools.in/json/json-php/>
- https://www.tutorialspoint.com/php/php_function_javascript_object_notation_json_last_error_msg.htm

PHP – Accessing Decoded Values [1-2]

```
<?php
$jsonobj = '{"John":29,"Nick":28,"Sean":31}';
$obj = json_decode($jsonobj);
echo $obj->John . " ";
echo $obj->Nick . " ";
echo $obj->Sean. " ";
?>
```

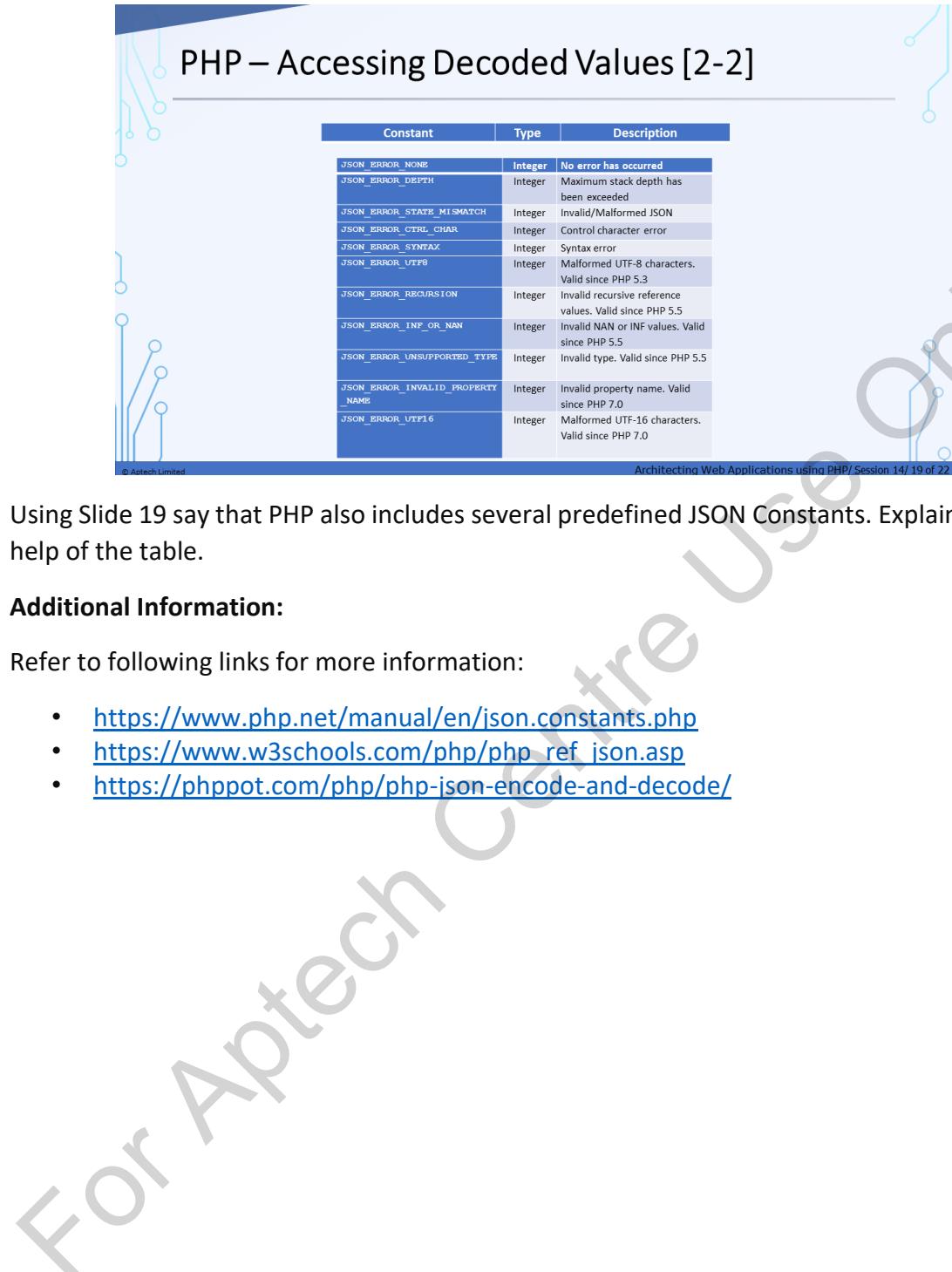
© Aptech Limited Architecting Web Applications using PHP/ Session 14/ 18 of 22

Show Slide 18 and say about the decoded values. Explain that it can be accessed either from an object or from an associative array. Show the code snippet and explain.

Additional Information:

Refer to following links for more information:

- <https://www.php.net/manual/en/function.json-decode.php>
- <https://code.tutsplus.com/tutorials/using-php-urlencode-and-urldecode--cms-37725>
- <https://guides.codepath.com/websecurity/PHP-Encoding-for-URLs>



PHP – Accessing Decoded Values [2-2]

Constant	Type	Description
<code>JSON_ERROR_NONE</code>	Integer	No error has occurred
<code>JSON_ERROR_DEPTH</code>	Integer	Maximum stack depth has been exceeded
<code>JSON_ERROR_STATE_MISMATCH</code>	Integer	Invalid/Malformed JSON
<code>JSON_ERROR_CTRL_CHAR</code>	Integer	Control character error
<code>JSON_ERROR_SYNTAX</code>	Integer	Syntax error
<code>JSON_ERROR_UTF8</code>	Integer	Malformed UTF-8 characters. Valid since PHP 5.3
<code>JSON_ERROR_RECURSION</code>	Integer	Invalid recursive reference values. Valid since PHP 5.5
<code>JSON_ERROR_INF_OR_NAN</code>	Integer	Invalid NAN or INF values. Valid since PHP 5.5
<code>JSON_ERROR_UNSUPPORTED_TYPE</code>	Integer	Invalid type. Valid since PHP 5.5
<code>JSON_ERROR_INVALID_PROPERTY_NAME</code>	Integer	Invalid property name. Valid since PHP 7.0
<code>JSON_ERROR_UTF16</code>	Integer	Malformed UTF-16 characters. Valid since PHP 7.0

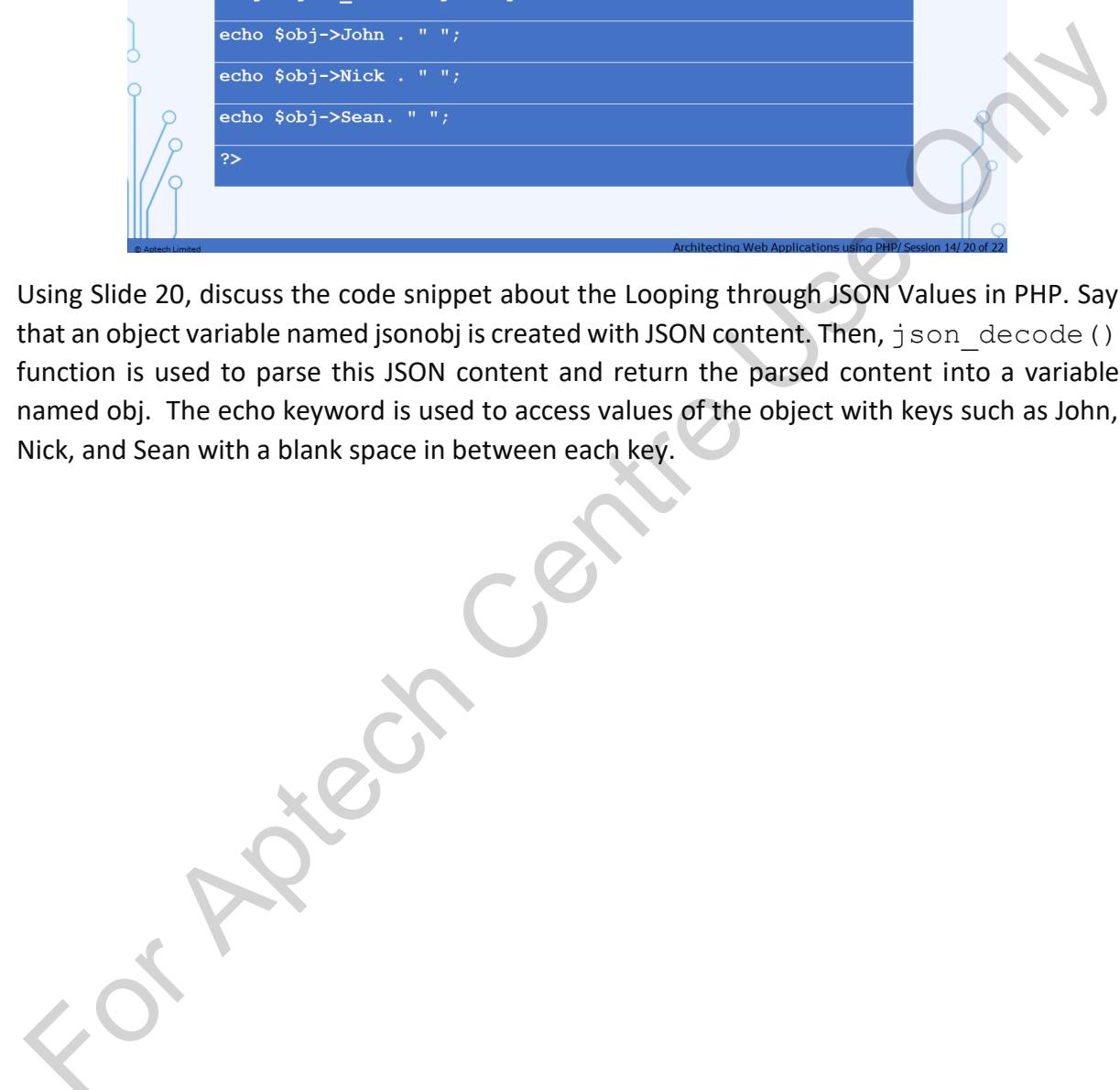
© Aptech Limited Architecting Web Applications using PHP/ Session 14/ 10 of 22

Using Slide 19 say that PHP also includes several predefined JSON Constants. Explain with the help of the table.

Additional Information:

Refer to following links for more information:

- <https://www.php.net/manual/en/json.constants.php>
- https://www.w3schools.com/php/php_ref_json.asp
- <https://phppot.com/php/php-json-encode-and-decode/>

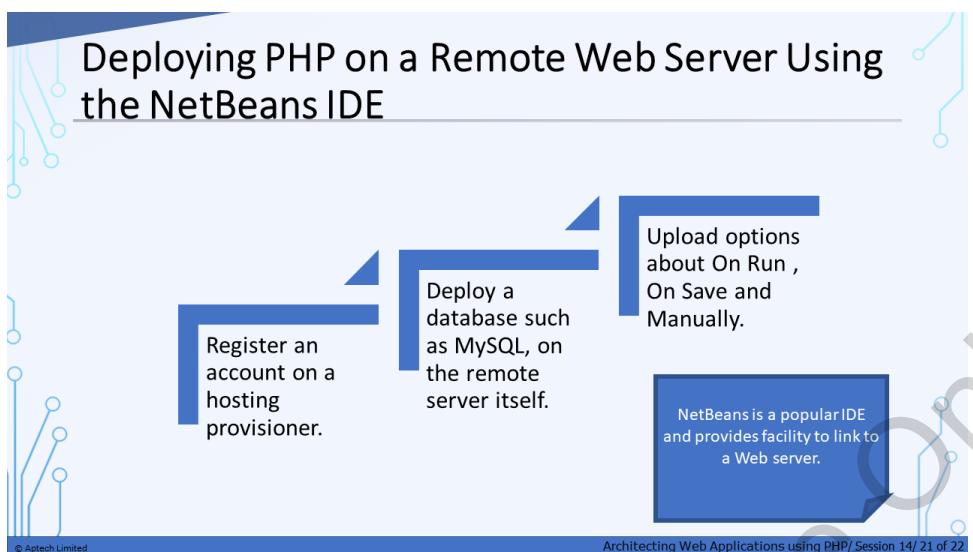


PHP - Looping Through JSON Values

```
<?php
$jsonobj = '{"John":29,"Nick":28,"Sean":31}';
$obj = json_decode($jsonobj);
echo $obj->John . " ";
echo $obj->Nick . " ";
echo $obj->Sean. " ";
?>
```

© Aptech Limited Architecting Web Applications using PHP/ Session 14/ 20 of 22

Using Slide 20, discuss the code snippet about the Looping through JSON Values in PHP. Say that an object variable named jsonobj is created with JSON content. Then, `json_decode()` function is used to parse this JSON content and return the parsed content into a variable named obj. The echo keyword is used to access values of the object with keys such as John, Nick, and Sean with a blank space in between each key.



Using Slide 21 explain the students about deploying PHP on a remote Web server using the NetBeans IDE. Tell the students about NetBeans. Say that it is a popular IDE which provides facility to link to a Web server.

Explain the available upload options about On Run, On Save, and Manually.

- On Run – The source files generally get uploaded to the server when the project is run (or executed).
- On Save – Modifications such as create, rename, edit, or delete are immediately passed on to the remote server.
- Manually – For a manual upload, the function in IDE has to be used.

Additional Information:

Refer to following links for more information:

- <https://netbeans.apache.org/kb/docs/php/remote-hosting-and-ftp-account.html>
- <https://www.adoclib.com/blog/how-to-set-up-a-new-php-project-in-netbeans-ide-version-8-2-or-11-3-using-wampserver.html>
- <http://free-tutorials.org/remote-web-server-php-using-the-netbeans-ide/>

Slide 22



Summary

- PHP provides a function `getenv()` to access the value of all the environment variables.
- PHP `rand()` and `srand()` functions are used to generate random numbers.
- Browser redirection can be achieved using the `header()` function.
- JSON is a popular data exchange format for the Web.
- `json_encode()` is used to encode JSON and `json_decode()` is used to decode JSON in PHP.
- JSON decoded values can be accessed either from an object or from an associative array.
- PHP `foreach()` loop is used to loop through the values of a JSON array or a list.
- Codes can now be methodical, definitive, and dependable with the new enhancements in PHP 8.
- User has to register an account on a hosting provider and deploy a database such as MySQL on the remote server.

Architecting Web Applications using PHP/ Session 14/ 22 of 22

Use slide 22 to summarize the session. You will end the session with a summary of what has been taught in the session. State the most important pointers of the session. This will be a revision for the students.

14.3 Post Class Activities for Faculty

You should familiarize yourself with the topics of the next session.

Tips:

You can also check the Articles/Blogs/Expert Videos uploaded on the Online Varsity site to gain additional information related to the topics covered in the next session.

Session 15 – PHP-AJAX

15.1 Pre-Class Activities

Before beginning the session, you should fully familiarize yourself with its topics. Prepare a question or two that will be a key point to relate the current session objectives.

15.1.1 Teaching Skills

To teach this session, you should be well-versed with the concept of internationalization and design patterns in Java. You must also be familiar with the concept of localization.

You must use the given images and teach the concepts in the theory class. For teaching in the class, you are expected to use slides and LCD projectors.

Tips:

It is recommended that you test the understanding of the students by asking questions in between the class.

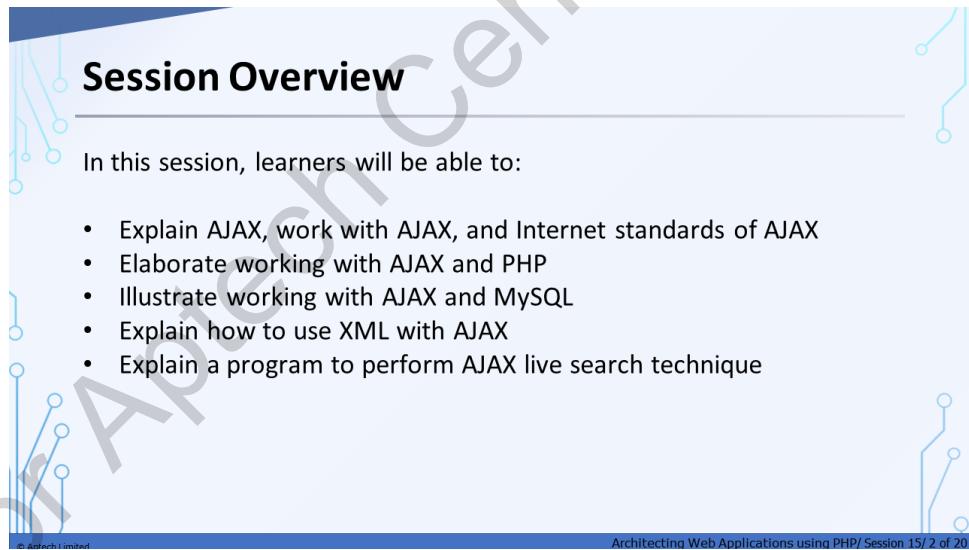
In-Class Activities

Follow the order given here during In-Class activities.

Overview of the Session

Give the students an overview of the current session in the form of session objectives. Read out the objectives on slide 2.

Slide 2



The slide has a light blue background with a decorative border of blue lines and circles. At the top center, the title "Session Overview" is displayed in bold black font. Below the title, a horizontal line separates it from the main content area. The content area contains the following text:
In this session, learners will be able to:

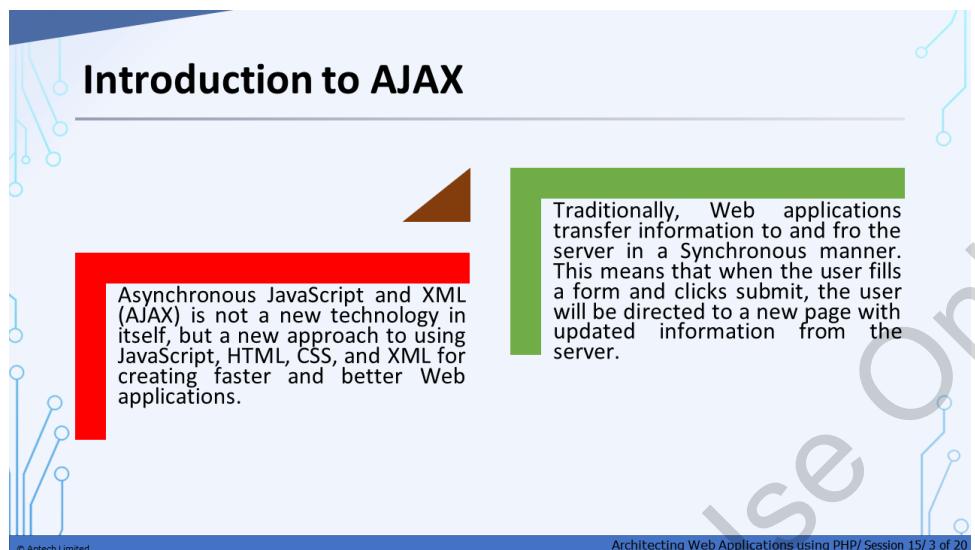
- Explain AJAX, work with AJAX, and Internet standards of AJAX
- Elaborate working with AJAX and PHP
- Illustrate working with AJAX and MySQL
- Explain how to use XML with AJAX
- Explain a program to perform AJAX live search technique

At the bottom left, there is a small copyright notice: "© Aptech Limited". At the bottom right, the slide footer reads: "Architecting Web Applications using PHP / Session 15 / 2 of 20".

Show slide 2 and give a brief overview of the current session in the form of session objectives. Inform students that the session begins by defining what AJAX is. The session then, provides an overview of working with AJAX and Internet standards of AJAX and explains how to work with AJAX and PHP. Further, it covers details on how to work with AJAX and MySQL. It also tells how to use XML with AJAX. Finally, it explains a program to perform AJAX live search technique.

15.2 In-Class Explanations

Slide 3



The slide has a blue header bar at the top. Below it is a title section with the text "Introduction to AJAX". A red callout box contains the text: "Asynchronous JavaScript and XML (AJAX) is not a new technology in itself, but a new approach to using JavaScript, HTML, CSS, and XML for creating faster and better Web applications." To the right of this is a green box containing text about traditional synchronous web application behavior. At the bottom, there is a footer bar with the text "© Aptech Limited" on the left and "Architecting Web Applications using PHP / Session 15 / 3 of 20" on the right.

Show slide 3 and tell students that with AJAX, Web applications make fast incremental updates to the user interface without reloading the entire page. In simple terms, using AJAX, one can make fast and dynamic Web Pages.

Applications or Web pages developed using traditional non-AJAX techniques reload the whole Web page even for changing few parts of the content. This can lead to server overhead and reduced performance.

With AJAX, the server is communicated behind the curtain. This means that the user would not notice the communication of the Web page to the server. This makes the application more interactive to the user's actions.

Give following additional information to the students:

Note: Google Search, Reddit handling of votes, Twitter updation of trending posts, and Facebook tabs are some examples of Web pages using AJAX.

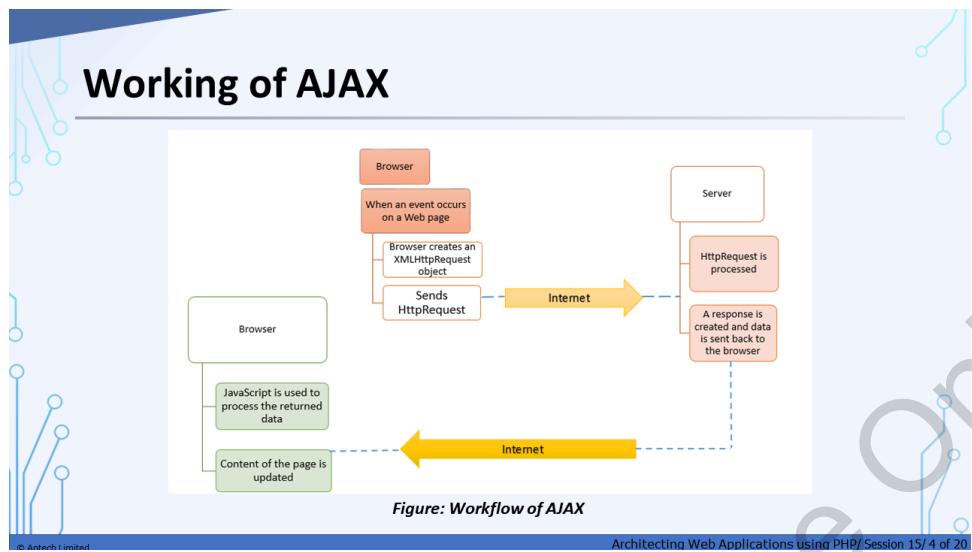
Additional Information:

Refer to following links for more information:

https://www.w3schools.com/php/php_ajax_intro.asp#:~:text=AJAX%20%3D%20Asynchronous%20JavaScript%20and%20XML,without%20reloading%20the%20whole%20page.

<https://www.tutorialspoint.com/ajax/index.htm>

<https://www.geeksforgeeks.org/ajax-introduction/>



Show slide 4 and explain to students that whenever an event occurs in a browser, an `HttpRequest` object is created and transferred to the server over the Internet. The server then, processes this request and sends an `HttpResponse` back to the browser. Browser uses JavaScript to execute this data and update the page contents.

Figure explains the workflow of AJAX.

Additional Information:

Refer to following links for more information:

<https://www.javatpoint.com/how-ajax-works>

<https://www.w3resource.com/ajax/working-with-PHP-and-MySQL.php>

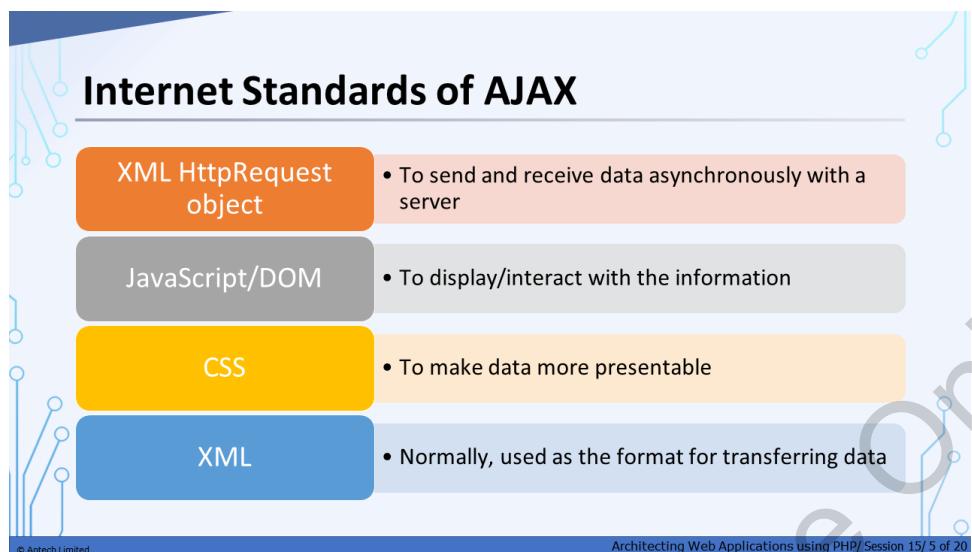
https://developer.mozilla.org/en-US/docs/Web/Guide/AJAX/Getting_Started

Ask the students following question. Wait for a response before you answer.

In-Class Question: AJAX is based on which two languages?

Answer: JavaScript and XML

Slide 5



Show slide 5 and AJAX is mainly used in Web development. It is based on Internet standards. It is a combination of XMLHttpRequest object, JavaScript/DOM, CSS, and XML. AJAX is independent of Web server software.

Additional Information:

Refer to following links for more information:

<https://vinothkumar1986.wordpress.com/2010/12/09/ajax-is-based-on-internet-standards/>
<https://www.koombea.com/blog/ajax-in-php/>
<https://codingwithfun.com/p/introduction-to-php-ajax/>

Ask the students following question. Wait for a response before you answer.

In-Class Question: When does a browser create an XMLHttpRequest?

Answer: When an event occurs on Web page

Slide 6

AJAX and PHP [1-2]

Example for AJAX PHP

```
CREATE TABLE customer (
    name varchar(50) NOT NULL,
    gender varchar(1) NOT NULL,
    bill int(10) NOT NULL,
    year int(10) NOT NULL,
    PRIMARY KEY (name)
)
```

The user must also insert customer data into the table as shown here:

```
INSERT INTO 'customer' VALUES ('Paul', 'm', 10000, 1998);
INSERT INTO 'customer' VALUES ('Tyron', 'm', 5000, 1992);
INSERT INTO 'customer' VALUES ('Hans', 'm', 25000, 1994);
INSERT INTO 'customer' VALUES ('Erin', 'f', 12000, 1997);
INSERT INTO 'customer' VALUES ('Vinn', 'f', 2000, 2000);
```

© Aptech Limited Architecting Web Applications using PHP / Session 15 / 6 of 20

Show slide 6 and tell the students that AJAX integrated with PHP and MySQL establishes smooth and robust communication with the database and the server. It helps in improving performance of the application. It becomes easy to access information from the database using AJAX and PHP. Consider an example to understand this. To use AJAX and PHP in the example with a database, one must build some MySQL queries. In this example, the user creates a table called `customer` and sends a query via AJAX. The results are then displayed in the browser Web page using '`customerdetail.html`'.

As a prerequisite to execute the example properly, a MySQL table should be created as given in the example on the slide. The user must also insert customer data into the table as shown in the second code snippet example.

Additional Information:

Refer to following links for more information:

https://www.w3schools.com/php/php_ajax_php.asp
https://www.tutorialspoint.com/php/php_and_ajax.htm

Slide 7

AJAX and PHP [2-2]

Code Snippet:

```
<html>
<body>
<script language="javascript" type="text/javascript">
//Checking Browser Compatibility
function runAjax(){
var ajaxHttpRequest; // Key variable that is necessary for AJAX
try{
// Opera 8.0+, Firefox, Safari
ajaxHttpRequest = new XMLHttpRequest();
}
catch (e){
// Internet Explorer Browsers
try{
ajaxHttpRequest = new ActiveXObject("Microsoft.XMLHTTP");
}
catch (e){
// Something went wrong
alert("Your browser is not working!");
return false;
}
}
//Setting up ajax to update the page on receiving the query results
ajaxHttpRequest.onreadystatechange = function(){
if(ajaxHttpRequest.readyState == 4){
var displayResponse = document.getElementById('result');
displayResponse.innerHTML = ajaxHttpRequest.responseText;
}
}
//Setting up ajax to update the page on receiving the query results
ajaxHttpRequest.onreadystatechange = function(){
if(ajaxHttpRequest.readyState == 4){
var displayResponse = document.getElementById('result');
displayResponse.innerHTML = ajaxHttpRequest.responseText;
}
}

//Setting up the input data to server-side
var bill = document.getElementById('bill').value;
var year = document.getElementById('year').value;
var gender = document.getElementById('gender').value;
var queryString = "bill=" + bill ;
queryString += "year=" + year + "gender=" + gender;
queryString += "&method=GET" , "customerdetail.php" +
queryString, true);
ajaxHttpRequest.send(null);
}
</script>
<form name="customerForm">
Total bill of the customer: <br/><br/><input type="number"
id="bill" /> <br />
Year the customer joined: <input type="number" id="year" />
<br />
Gender: <select id="gender">
<option value="m">m</option>
<option value="f">f</option>
</select>
<input type="button" onclick="runAjax()" value="Submit"/>
</form>
<div id="result">Your result will be shown here</div>
</body>
</html>
```

© Aptech Limited Architecting Web Applications using PHP/ Session 15/ 7 of 20

Show slide 7 and tell the students that the given Code Snippet is a client-side code which displays a Web Page for a dynamic user interaction.

Note: The query variables are passed according to HTTP standards.

In Code Snippet 1, the user enters a bill amount, a year a customer joined, and a gender. Upon clicking 'Submit', the Web page displays data of all customers whose total bill is more than or equal to the input bill, year in which they joined is less than or equal to the input year, and gender equal to given gender (m/f).

The query is sent by PHP and AJAX updates the Web page with the results without reloading the entire Web page. This action is done at the server side.

Slide 8

Server-Side PHP [1-2]

Code Snippet:

```
<?php
    //Passing the credentials to access the database
    $db_host = "localhost";
    $db_user = "root";
    $db_pass = "";
    $db_name = "test01";
    //Establishing connection
    $con = mysql_connect($db_host, $db_user, $db_pass, $db_name);
    $bill = $_GET['bill'];
    $year = $_GET['year'];
    $gender = $_GET['gender'];
    //Constructing the query with all conditions
    $query = "SELECT * FROM customer WHERE gender = '$gender'";
    if(is_numeric($bill))
        $query .= " AND bill >= $bill";
    if(is_numeric($year))
        $query .= " AND year <= $year";
    //Running the query
    $qry_result = mysqli_query($con,$query) or die(mysql_error());
    //Table structure
    $display_tb = "<table>";
    $display_tb .= "<tr>";
    $display_tb .= "<th>Name</th>";
    $display_tb .= "<th>Total Bill</th>";
    $display_tb .= "<th>Year Joined</th>";
    $display_tb .= "<th>Gender</th>";
    $display_tb .= "<tr>";
    //Inserting data for each person
    while($row = mysqli_fetch_array($qry_result)){
        $display_tb .= "<tr>";
        $display_tb .= "<td>$row[name]</td>";
        $display_tb .= "<td>$row[bill]</td>";
        $display_tb .= "<td>$row[year]</td>";
        $display_tb .= "<td>$row[gender]</td>";
        $display_tb .= "</tr>";
    }
    echo "Displaying the results for " . $query . "<br />";
    $display_tb .= "</table>";
    echo $display_tb;
?>
```

© Aptech Limited

Architecting Web Applications using PHP/ Session 15/ 8 of 20

Show slide 8 and tell the students that once client-side scripting is done, server-side script should be written. This server-side code will retrieve bill, year, and gender from the database based on given inputs and return it to the client. Code Snippet is saved in the file named as customerdetail.php. Assume that a database named test01 has been created.

In Code Snippet, a connection is made with MySQL database to fetch the results of the query. Data supplied in the form is retrieved using `$_GET`. Each of these data parameters are then, substituted in the SELECT query so that the relevant matching data can be retrieved.

Additional Information:

Refer to following links for more information:

<https://www.thoughtco.com/server-side-scripting-2694142>

<https://stackoverflow.com/questions/31237984/php-is-a-scripting-language-or-server-side-language>

https://en.wikipedia.org/wiki/Server-side_scripting

Slide 9

The figure consists of two screenshots of a web browser window. The top screenshot shows a form with fields for 'Total bill of the customer' (10000), 'Year the customer joined' (2000), and 'Gender' (m). Below the form is a placeholder 'Your result will be displayed here'. The bottom screenshot shows the results of the query: 'Displaying the results for SELECT * FROM customer WHERE gender = m AND bill >= 10000 AND year <= 2000'. It lists two rows of data:

Name	Total Bill	Year Joined	Gender
Hans	25000	1994	m
Paul	10000	1998	m

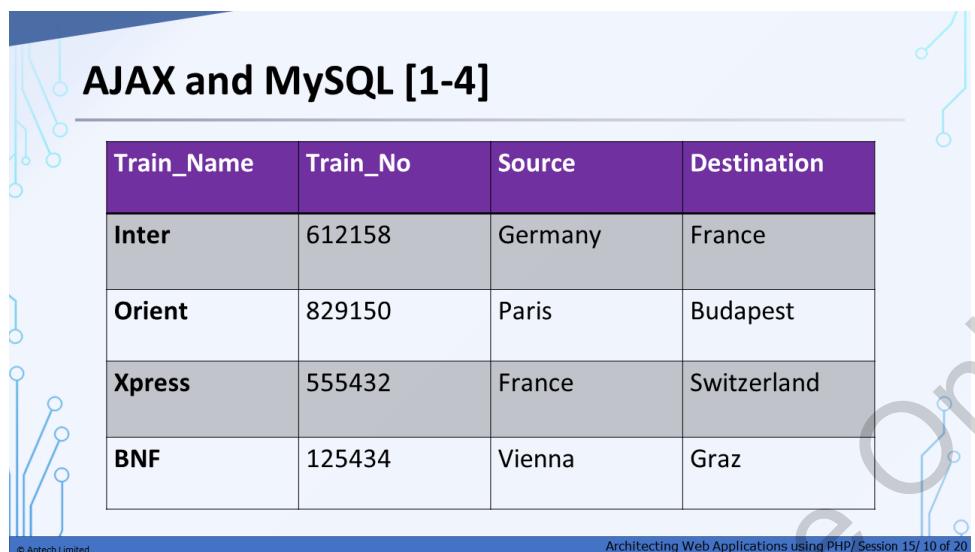
Figure: Web Page For User Inputs

Figure: Output of Server Side Script Using AJAX

© Aptech Limited Architecting Web Applications using PHP/ Session 15/ 9 of 20

Show slide 9 and tell the students that first and second figures on the slide show the output of Code Snippets on slides 7 and 8 respectively.

Once the user enters the required data in the input box and clicks 'Submit', the results are displayed as shown in second Figure. Here, the entire page will not be reloaded. Only the results section will be refreshed with the retrieved data.



AJAX and MySQL [1-4]

Train_Name	Train_No	Source	Destination
Inter	612158	Germany	France
Orient	829150	Paris	Budapest
Xpress	555432	France	Switzerland
BNF	125434	Vienna	Graz

© Aptech Limited Architecting Web Applications using PHP/ Session 15/ 10 of 20

Show slide 10 and tell the students that AJAX can also be used for communicating with the database interactively. One can create a simple live database search functionality by utilizing AJAX and PHP techniques, where the search results will be displayed as the characters are typed in the search input box.

Consider an example where user wants to display train information based on a given train name or train number. This input is to be accepted via a drop-down list. The data will be retrieved from a database using AJAX. The values for the drop-down are hardcoded in this example, but in practical scenarios, they may be populated from the database too.

The MySQL database table used in the on-slide example contains data as shown.

Additional Information:

Refer to following links for more information:

https://www.w3schools.com/php/php_ajax_database.asp
https://www.tutorialspoint.com/ajax/ajax_database.htm

Ask the students following question. Wait for a response before you answer.

In-Class Question: Which property returns the response data as a string?

Answer: responseText

Slide 11

The slide features a decorative background with blue circuit board patterns and a large watermark reading 'For Academic Use Only' diagonally across the center.

AJAX and MySQL [2-4]

Code Snippet: (displayTrain.html)

```
<html>
<head>
<script>
//Function to use Ajax to transfer the data
function displayTrain(train_data) {
var ajaxHttpRequest = new XMLHttpRequest();
if (train_data == "") {
document.getElementById("result").innerHTML = "";
return;
}
ajaxHttpRequest.onreadystatechange = function(){
if(ajaxHttpRequest.readyState == 4){
var displayResponse = document.getElementById('result');
displayResponse.innerHTML = ajaxHttpRequest.responseText;
}
}
//The variable is passed to PHP to retrieve the result
ajaxHttpRequest.open("GET", "displayTrain.php?q=" + train_data, true);
ajaxHttpRequest.send(null);
}
</script>
</head>
<body>
```

```
<form>
<select name="train_name" onchange =
"displayTrain(this.value)">
<option value="">Select the train name:</option>
<option value="Inter">Inter</option>
<option value="Orient">Orient</option>
<option value="Xpress">Xpress</option>
<option value="BNF">BNF</option>
</select>
&ampnbsp&ampnbsp&ampnbsp
<select name="train_number" onchange =
"displayTrain (this.value)">
<option value="">Select the train number:</option>
<option value="125434">125434</option>
<option value="555432">555432</option>
<option value="612158">612158</option>
<option value="829150">829150</option>
</select>
<br>
<div id="result"><b>Train's information...</b></div>
</body>
</html>
```

© Aptech Limited

Architecting Web Applications using PHP / Session 15 / 11 of 20

Show slide 11 and tell the students that in the example mentioned in the slide 10, whenever a user clicks any of the two drop-downs and chooses either the train number or the train name from the list, a function named 'displayTrain()' is invoked. The `onchange()` event triggers this function. For this, the HTML code shown in Code Snippet is used.

In Code Snippet, the user first checks for any empty input through the condition `train_data= ""`. If the condition is true, the function returns an empty string. Whereas, if the condition is false and there is any selected data from either drop-down, an XMLHttpRequest object is created. Once the server response is ready, it sends the request to the server-side file. The data is passed as a parameter 'q' in the URL.

Corresponding PHP code will be stored in a file named 'displayTrain.php'. This code executes a query from the database, and the result is returned to the HTML table. Code Snippet in slide 12 shows this code, which will be saved as `displayTrain.php`. It returns the data of the input train or number entered by the user.

Slide 12

AJAX and MySQL [3-4]

Code Snippet: (displayTrain.php)

```
<html>
<head>
<style>
table {
    width: 100%;
    border-collapse: collapse;
}

table, td, th {
    border: 1px solid black;
    padding: 5px;
}

th {text-align: left;}
</style>
</head>
<body>

<?php
$g = $_GET['q'];
$c = mysqli_connect('localhost', 'root', '', 'test01');
if((int)$g > 0){
$sql="SELECT * FROM train WHERE Train_No= '".$g."'";
}else{
$sql="SELECT * FROM train WHERE Train_Name= '".$g."'";
}
$result = mysqli_query($c, $sql);

echo "<table>
<tr>
<th>Train Number</th>
<th>Train Name</th>
<th>Source</th>
<th>Destination</th>
</tr>";
while($row = mysqli_fetch_array($result)) {
    echo "<tr>";
    echo "<td>" . $row['Train_No'] . "</td>";
    echo "<td>" . $row['Train_name'] . "</td>";
    echo "<td>" . $row['Source'] . "</td>";
    echo "<td>" . $row['Destination'] . "</td>";
    echo "</tr>";
}
echo "</table>";
mysqli_close($c);
?>
</body>
</html>
```

© Aptech Limited

Architecting Web Applications using PHP/ Session 15/ 12 of 20

Show slide 12 and tell the students that once JavaScript sends the query to the PHP file, following actions will happen:

- A new connection is established by PHP to the MySQL database server.
- The corresponding train with all the details is found from database.
- An HTML table is created, filled with found data, and is sent back to the 'result' placeholder.

Code for these actions is implemented in Code Snippet.

Slide 13

The slide features a blue header bar with the title "AJAX and MySQL [4-4]" in white. Below the header are two screenshots of a web application. The top screenshot shows a dropdown menu with options: "Select the train name:" followed by "Inter", "Orient", "Xpress", and "BNF". The bottom screenshot shows a table with one row of data: "Train Number" (829150), "Train Name" (Orient), "Source" (Paris), and "Destination" (Budapest). A watermark reading "For Aptech Centre Use Only" is diagonally across the slide.

Figure: Web Page for Selecting Trains

Figure: Details of the Train Selected is Displayed

© Aptech Limited

Architecting Web Applications using PHP/ Session 15/ 13 of 20

Show slide 13 and tell the students that first and second figures show the output for Code Snippets from slides 11 and 12 respectively.

AJAX and XML [1-2]

There are two major differences between HTML and XML:

- HTML defines a certain set of tags to be used compulsorily whereas XML does not.
- XML is strict about document structure, how it is stored, and transferred, but HTML does not follow this strictly.

© Aptech Limited Architecting Web Applications using PHP/ Session 15/ 14 of 20

Show slide 14 and tell that XML is a markup language similar to HTML. XML document contains plain text along with tags enclosed within < and >. XML is commonly used as a data format. XML and AJAX is used for sending a request and receiving a response to deliver data on the Web pages.

XML gives its users a lot more freedom than HTML. HTML has a defined set of tags for each function. For example, <a> tag surrounds a link, the
 breaks a line, and so on.

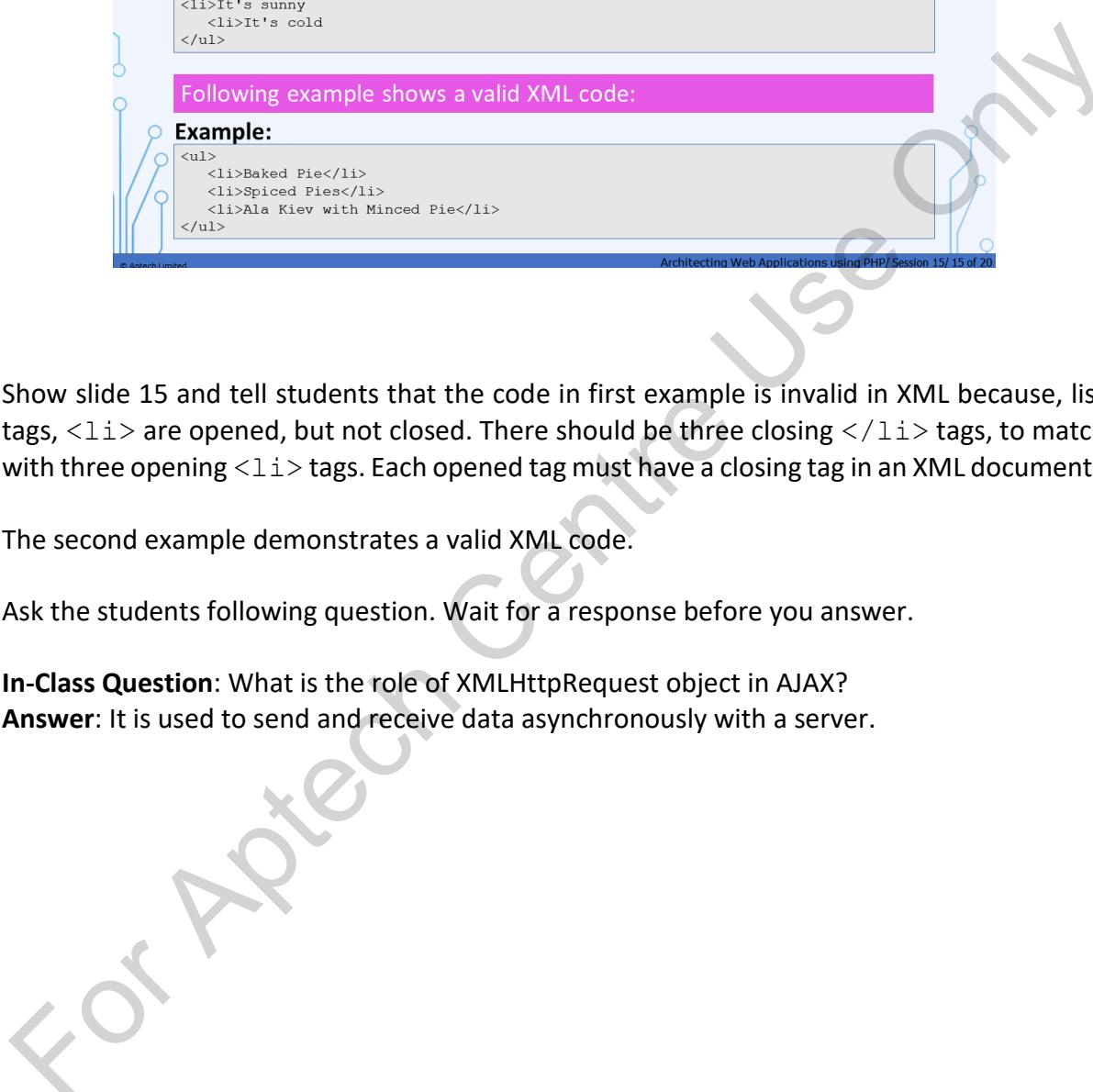
Conversely, XML documents can use any tags according to user's choice such as putting <rating></rating> tags around a movie rating, <height></height> tags around a person's height, and so on. Thus, XML gives users an option to define user's own tags.

HTML allows the user to play with the tags and ignores some opening and closing tags, unlike XML.

Additional Information:

Refer to following links for more information:

https://www.w3schools.com/php/php_ajax_xml.asp
<http://www.javascriptkit.com/dhtmltutors/ajaxgetpost3.shtml>



AJAX and XML [2-2]

Following example shows an invalid XML code:

Example:

```
<ul>
<li>It's raining
<li>It's sunny
    <li>It's cold
</ul>
```

Following example shows a valid XML code:

Example:

```
<ul>
    <li>Baked Pie</li>
    <li>Spiced Pies</li>
    <li>Ala Kiev with Minced Pie</li>
</ul>
```

© Aptech Limited

Architecting Web Applications using PHP/ Session 15/ 15 of 20

Show slide 15 and tell students that the code in first example is invalid in XML because, list tags, `` are opened, but not closed. There should be three closing `` tags, to match with three opening `` tags. Each opened tag must have a closing tag in an XML document.

The second example demonstrates a valid XML code.

Ask the students following question. Wait for a response before you answer.

In-Class Question: What is the role of XMLHttpRequest object in AJAX?

Answer: It is used to send and receive data asynchronously with a server.

Slide 16

A live search box is a search input box that displays the search results as and when the user starts typing words.

It uses a reliable database to retrieve information faster.

As an example, a live search box has been created that will search the employees table and show the results asynchronously.

Example:

```
CREATE TABLE emp (
    Emp_No INT(15) NOT NULL PRIMARY KEY,
    Name VARCHAR(50) NOT NULL
);
```

© Aptech Limited Architecting Web Applications using PHP/ Session 15/ 16 of 20

Show slide 16 and tell students that for the example mentioned in the slide, a table `emp` is created which stores the details of the employees such as `Emp_No` and Employee name. First, the user creates the table structure as shown in the example.

Once the table is created, data must be inserted so that the results are fetched accordingly.

After the data is inserted, a Web page must be created for user interaction. Thus, as and when the user types to search for an employee, the employee details will be displayed. It is quite similar to autocomplete text boxes or typeahead.

Additional Information:

Refer to following links for more information:

https://www.w3schools.com/php/php_ajax_livesearch.asp
<https://www.cloudways.com/blog/live-search-php-mysql-ajax/>
<https://ajaxlivesearch.com/>

Ask the students following question. Wait for a response before you answer.

In-Class Question: What are the advantages of AJAX?

Answer: Some advantages of AJAX are mentioned as follows:

- Better user interaction
- Faster data retrieval
- Bandwidth utilization

AJAX Live Search [2-4]

Code Snippet: (employeeSearch.php)

```

<html>
  <head>
    <h1> PHP Live MySQL Database Search </h1>
    <style>
      /* Formatting search box */
      .search-box{
        width: 500px;
        position: relative;
        display: inline-block;
        font-size: 14px;
      }
      .search-box input[type="text"]){
        height: 32px;
        padding: 5px 10px;
        border: 1px solid #cccccc;
        font-size: 14px;
      }
    </style>
    <script>
      function displayEmployee(){
        var ajaxHttpRequest = new XMLHttpRequest();
        var name = document.getElementById('name').value;
        if(name == ""){
          document.getElementById("result").innerHTML = "";
          return;
        }
        ajaxHttpRequest.onreadystatechange = function(){
          if(ajaxHttpRequest.readyState == 4){
            var displayResponse = document.getElementById('result');
            displayResponse.innerHTML = ajaxHttpRequest.responseText;
          }
        }
        var query = "?name=" + name ;
        ajaxHttpRequest.open("GET", "getEmployee.php" + query, true);
        ajaxHttpRequest.send(null);
      }
    </script>
  </head>
  <body> <div class="search-box">
    <input id="name" type="text" autocomplete="off"
    placeholder="Search employee" onkeyup="displayEmployee()" />
    <br>
    <div id="result">Results will be displayed here...</div>
  </div>
</body>
</html>

```

© Aptech Limited Architecting Web Applications using PHP/ Session 15/ 17 of 20

Show slide 17 and tell the students that Code Snippet shows a file `employeeSearch.php` written in PHP.

The keyup or input change event will occur whenever the search input value is changed. This will generate an AJAX request which is sent to the `getEmployee.php` file. It fetches records that match the data in the employee table. The records are then inserted inside a `<div>` tag which are ready to be displayed to the user.

Slide 18

The slide features a blue header bar with the title "AJAX Live Search [3-4]" in white. Below the header is a code block with a light blue background and a thin black border. The code is PHP, connecting to a MySQL database named "test01" and performing a SELECT query on the "emp" table where the "Name" column matches the value passed in the GET parameter "name". It then loops through the results and outputs them as HTML. If no results are found, it displays a message. The footer of the slide contains the copyright notice "© Aptech Limited" and the page information "Architecting Web Applications using PHP/ Session 15/ 18 of 20".

```
<?php
$db_host = "localhost";
$db_user = "root";
$db_pass = "";
$db_name = "test01";

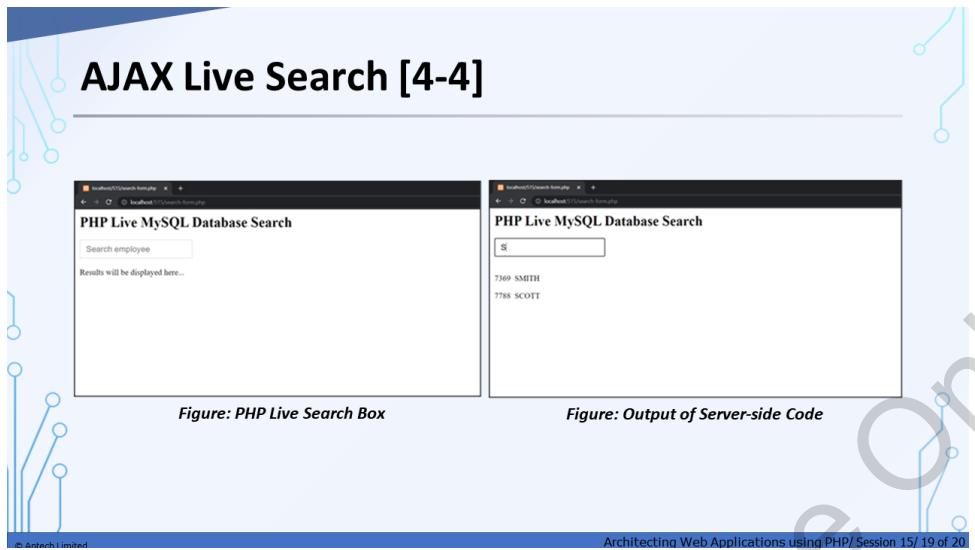
//Establishing the connection
$con = mysqli_connect($db_host, $db_user, $db_pass, $db_name) or die(mysql_error());
$name = $_GET['name'];
$name = mysqli_real_escape_string($con,$name);
$sql = "SELECT * FROM emp WHERE Name LIKE '$name%'";
//Constructing the query with condition
$result = mysqli_query($con,$sql);
//Retrieving the result from the database
if(mysqli_num_rows($result) > 0){
    while($row = mysqli_fetch_array($result)){
        echo "<p>" . $row["Emp_No"] . "&ampnbsp " . $row["Name"] . "</p>";
    }
} else {
    echo "<p>No matches found</p>";
}
mysqli_close($con);
?>
```

Show slide 18 and tell the students that once a query is sent by AJAX, PHP searches the database and sends the results to the browser. Code Snippet shows the PHP code for `getEmployee.php`.

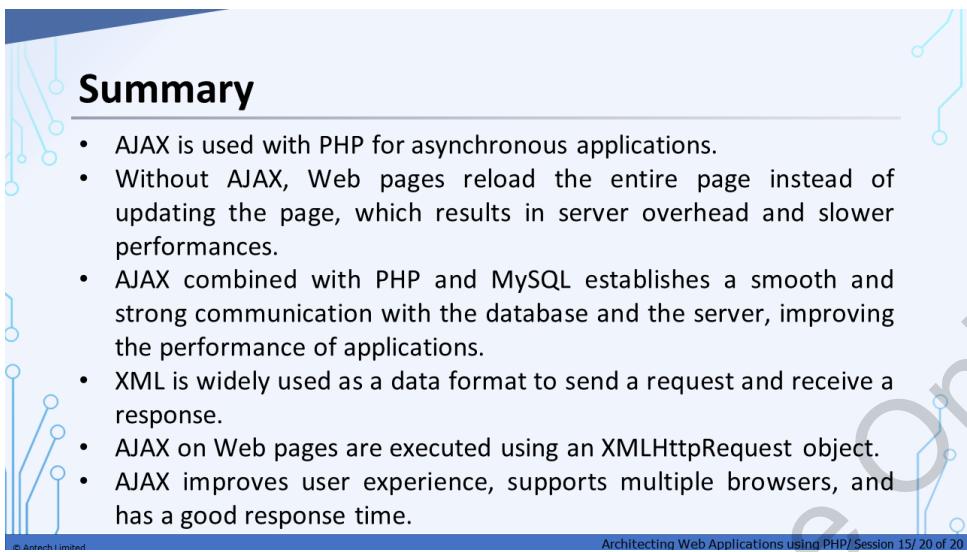
The SQL `SELECT` and `LIKE` operators are used to match records and fetch the `Emp_No` and `Name` as results. If the entered value is not matching with any record then 'No matches found' result is displayed.

One can make use of PHP `mysqli_real_escape_string()` method to truncate special characters present in the input. This will avoid any additional spaces and special characters and make it a valid SQL string.

Slide 19



Show slide 19 and tell the students that first and second figures show the output for Code Snippets from slides 17 and 18 respectively.



Summary

- AJAX is used with PHP for asynchronous applications.
- Without AJAX, Web pages reload the entire page instead of updating the page, which results in server overhead and slower performances.
- AJAX combined with PHP and MySQL establishes a smooth and strong communication with the database and the server, improving the performance of applications.
- XML is widely used as a data format to send a request and receive a response.
- AJAX on Web pages are executed using an XMLHttpRequest object.
- AJAX improves user experience, supports multiple browsers, and has a good response time.

Architecting Web Applications using PHP/ Session 15/ 20 of 20

Use slide 20 to summarize the session. You will end the session with a summary of what has been taught in the session. Tell students the pointers of the session. This will be a revision of the current session.

15.3 Post Class Activities for Faculty

Tips:

You can check the Articles/Blogs/Expert Videos uploaded on the Online Varsity site to gain additional information related to the topics covered in the session.

As this is the last session, you can briefly summarize all course topics covered.

End of the Document

For Aptech Centre Use Only