

Unit – 01: Expressions and control statements in PHP

- History and Advantages of PHP,
- Syntax of PHP,
- Variables and constants,
- Data types, Expressions and operators,
- Decision making Control statements - if, if-else, nested if, switch, break and continue statement, Loop control structured

Elective-III

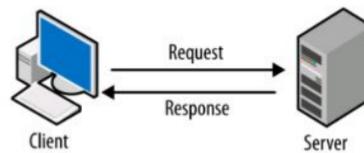
A) Data Sciences: Data Warehousing and Data Mining B) Web Based Application development with PHP

Questions to be discussed:

1. Differentiate between HTML and PHP.
2. What is PHP? Write the advantage of PHP.
3. Explain brief history of PHP.
4. Explain the difference between Client Side Scripting and Server Side Scripting.
5. Differentiate between echo and print statement used in PHP.
6. Define operators. Explain different types of operators used in PHP.
7. Describe the data types used in PHP.
8. Discuss the various looping statements used in PHP with suitable example.
9. Explain “do-while” construct used in PHP. Write a program to print your name 5 times using it.

What is PHP?

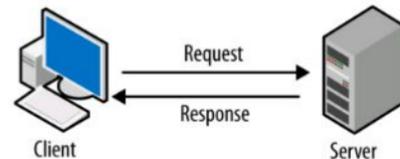
- ❑ PHP stands for Hypertext Pre-Processor.
- ❑ It is widely used, open source, server side scripting language.
- ❑ PHP is a powerful tool used by web developers to build dynamic and interactive websites.
- ❑ The first version of PHP was formulated by Rasmus Lerdorf in 1994.
- ❑ PHP 8.2 is the latest version released in 2022.
- ❑ PHP files save with .php extension.



Advantages of PHP:

- ❖ PHP is open-source and free from cost.
- ❖ It is platform-independent.
- ❖ It is mainly used due to its faster rate of loading over slow internet speed.
- ❖ It helps in reusing an equivalent code and not got have to write lengthy code and sophisticated structure.
- ❖ It helps in managing code easily.
- ❖ It has powerful library support to use various function modules for data representation.
- ❖ PHP's built-in database connection modules help in connecting databases easily.

Difference between HTML and PHP:



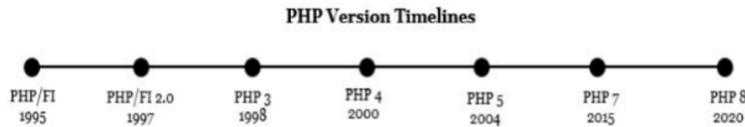
HTML	PHP
HTML is a client-side scripting language.	PHP is a server-side scripting language.
Developed by Tim Berners-Lee.	Developed by Rasmus Lerdorf.
HTML codes are static.	PHP codes are dynamic.
It is used to create static web pages.	It is used to create dynamic web page.
HTML is very easy to learn.	PHP is easy to learn but not as much as HTML.
HTML is used for front-end development	PHP is used for server-side development.
HTML can be used in a PHP file.	PHP can't be used in an HTML file.
HTML files save with .html or .htm extension.	PHP files save with .php extension.

History of PHP:

- Rasmus Lerdorf started developing PHP in 1994.
- A lot of CGI programs were written by him which were used for maintenance of his personal website.
- He modified these programs to make them work with web forms and to interact with databases.
- The implementation of this modification was termed as “Personal Home Page/Forms Interpreter” (PHP/FI)
- The first version of PHP/FI was made available to the world by Lerdorf in June 1995.

PHP 2

- After the first version of PHP/FI, a development team formed which worked for months to finally release official version 2 of PHP/FI in late 1997.
- As PHP was developed organically, there was a lot of inconsistency in function names and parameter order.



PHP 3

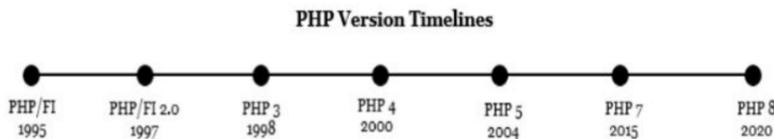
- The parser was rewritten in 1997 by Zeev Suraski and Andi Gutmans which led to the formation of PHP3.
- After changing its name to PHP which meant Hypertext Preprocessor, its public testing was started.
- PHP3 was officially launched in June 1998.

PHP 4

- Zend Engine 1.0 powered PHP 4 was released in May 2000.
- PHP 4 reached its version 4.4.9 in August 2008.

PHP 5

- Powered by Zend Engine 2, PHP 5 was launched in July 2004.
- It had features like PDO Extension, improved support for OOPS and many other performance improvements.



PHP 6 (Never Released)

- Due to the lack of support for native Unicode, PHP received mixed reviews.
- A project was initiated in 2005 to introduce native Unicode support through PHP.
- This project was headed by Andrei Zmievski. As it would require major changes, it was decided to launch this update as a new version naming it PHP 6.0.
- But the shortage of developers who understood the necessary changes resulted in the delay of the project.
- Considering this delay, PHP 5.3 was released in 2009 which included many features from the PHP 6 project.
- The project development was stopped in March 2010.

PHP 7

Another major version of PHP was developed in 2014 and 2015.

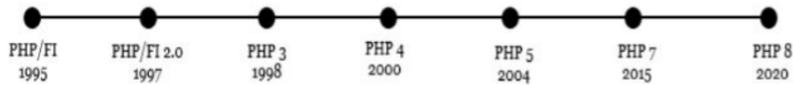
This version was termed as PHP 7 which caused some confusion as PHP 6 was never released.

PHP 7 was also known as PHP next generation(PHPNG).



PHP 8.0:

- On the 25th year of PHP history, PHP 8.0 was developed in Nov. 2020.
- PHP 8.1, released in 2021.
- PHP 8.2 is the latest PHP version.
- PHP 8.3 is the active development branch of PHP versions. It's the current target for new [RFCs](#), and is scheduled to be released towards the end of 2023.

PHP Version Timelines**Syntax of PHP:**

- A PHP script starts with `<?php` and ends with `?>`:

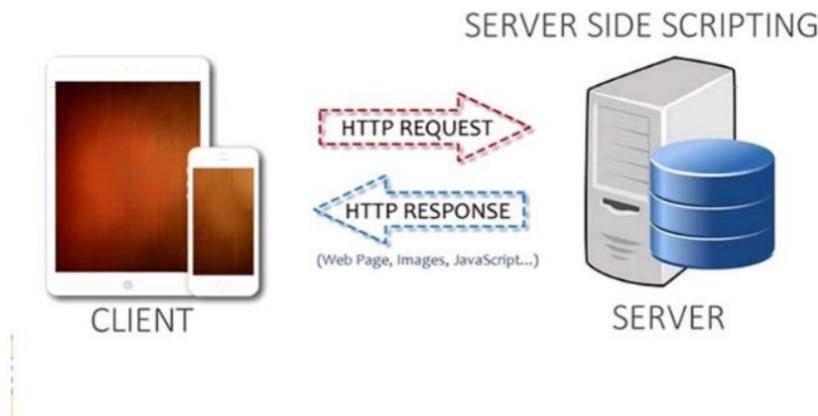
echo and print Statements:

- ❑ We frequently use the echo statement to display the output.
- ❑ There are two basic ways to get the output in PHP:
 1. echo
 2. print

Difference between echo and print:

echo	print
echo is a statement, which is used to display the output.	print is also a statement, used as an alternative to echo at many times to display the output.
echo can be used with or without parentheses.	print can be used with or without parentheses.
echo does not return any value.	print always returns an integer value, which is 1.
We can pass multiple strings separated by comma (,) in echo.	Using print, we cannot pass multiple arguments.
echo is faster than print statement.	print is slower than echo statement.

Difference between Client Side Scripting and Server Side Scripting:



Client-side scripting:

- A client-side script is a small program that is inserted into a web page.
- It is processed within the client browser instead of the web server.
- A language in which a client side script or program is written using syntax is called client side scripting language or client side programming.

Example: JavaScript, VBScript, jQuery etc.



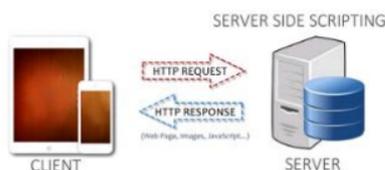
Server-side scripting:

- It is a program that is executed on the server side when the user requests information.
- The script do not download at the client side.
- The language in which server side script or program is written using syntax is called server side scripting language or server side programming language.
- These kinds of scripts are always executed before a web page is loaded on the browser.

Example: PHP, Python, Ruby, ASP.NET, Perl etc.

Difference between Client Side Scripting and Server Side Scripting:

Client-side scripting	Server-side scripting
Web browsers execute client-side scripting.	Web servers are used to execute server-side scripting.
They are used to create static pages.	They are used to create dynamic pages.
Source code is visible to the user.	Source code is not visible to the user.
Its main function is to provide the requested output to the end user.	Its primary function is to manipulate and provide access to the respective database as per the request.
It runs on the user's computer.	It runs on the webserver.
It does not provide security for data.	It provides more security for data.
HTML, CSS, and javascript are used.	PHP, Python, Java, Ruby are used.



Syntax of PHP:

- A PHP script starts with `<?php` and ends with `?>`
- A PHP script can be placed anywhere in the document.
- A PHP script is executed on the server, and the result is sent back to the browser.
- PHP statements end with semicolon.

PHP Variables:

- A variable is the name of the memory location that holds data.
- Variables are also known as "containers" for storing information.
- In PHP, a variable is declared using a **\$ sign** followed by the variable name.

Rules for declaring PHP variable:

- A variable must start with a dollar (\$) sign, followed by the variable name.
- It can only contain alpha-numeric character and underscore (A-z, 0-9, _).
- A variable name must start with a letter or underscore (_) character.
- Variable name cannot start with a number or special symbols.
- A PHP variable name cannot contain spaces.
- Variables are case-sensitive.

PHP Constants:

- PHP constants are name or identifier that can't be changed during the execution of the script.
- Constants are similar to the variable except once they defined, they can never be undefined or changed.
- They remain constant across the entire program.
- PHP constants follow the same variable rules, that means it can be start with a letter or underscore only.
- Use the `define()` function to create a constant.

Syntax:

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

name: It specifies the constant name.

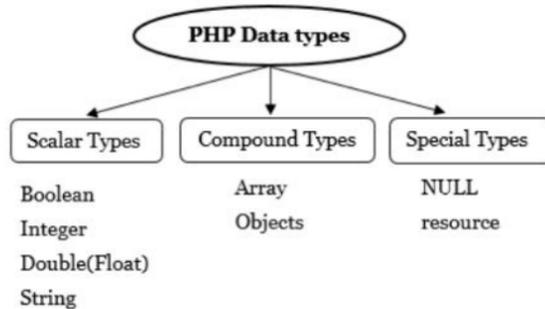
value: It specifies the constant value.

case-insensitive: Specifies whether a constant is case-insensitive.

Default value is false. It means it is case sensitive by default.

PHP Data Types:

- Data Types define the type of data a variable can store.
- PHP data types are used to hold different types of data or values.
- PHP supports 8 primitive data types that can be categorized further in 3 types:
 1. Scalar Types (predefined)
 2. Compound Types (user-defined)
 3. Special Types

**PHP Boolean**

- Booleans are the simplest data type works like switch.
- It holds only two values: **TRUE (1)** or **FALSE (0)**.
- It is often used with conditional statements.
- If the condition is correct, it returns TRUE otherwise FALSE.

**PHP Integer**

- Integer means numeric data with a negative or positive sign.
- It holds only whole numbers, i.e., numbers without fractional part or decimal points.

PHP Float

- A floating-point number is a number with a decimal point.
- It can hold numbers with a fractional or decimal point, including a negative or positive sign.

PHP String

- A string is a non-numeric data type.
- It holds letters or any alphabets, numbers, and even special characters.
- String values must be enclosed either within **single quotes** or in **double quotes**.

PHP Array

- An array is a compound data type.
- It can store multiple values of same data type in a single variable.

PHP object

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

PHP Null

- Null is a special data type that has only one value: **NULL**.
- There is a convention of writing it in capital letters as it is case sensitive.
- The special type of data type NULL defined a variable with no value.

PHP Resource

- Resources are not the exact data type in PHP.
- Basically, these are used to store some function calls or references to external PHP resources.

PHP Operators:

- An operator is a symbol that tell to the compiler which operations to be performed between the operands.
- Basically, it is used to perform operations on operands.
- In simple words, operators are used to perform operations on variables or values.
- An operator takes some values (the operands) and does something.
- There are different types of operators in PHP:
 1. Arithmetic operators
 2. Logical operators
 3. Conditional(Ternary) operators
 4. Assignment operators
 5. Comparison(Relational) operators
 6. Increment/Decrement operators
 7. String operators
 8. Array operators

Set up PHP on your own PC

1. PHP
2. Database(MySQL)
3. Server(Apache)

Third Party Software

- **WAMP** Windows, Apache, MySQL, PHP
- **LAMP** Linux, Apache, MySQL, PHP
- **MAMP** MacOS, Apache, MySQL, PHP
- **Xampp** Cross-Platform, Apache, MySQL, PHP-Perl

Decision Making control statements:

- Like other language PHP also allows us to perform actions based on some type of conditions.
- It's just like a two- way path.
- If you want something then go this way or else turn that way.
- To use this feature, PHP provides us with four conditional statements:
 1. **if** statement
 2. **if...else** statement
 3. **if...elseif...else** statement
 4. **switch** statement

if Statement:

if...else Statement:

- We understood that if a condition will hold TRUE or FALSE.
- If a condition is TRUE then if block gets executed, otherwise else block gets executed.

Syntax:**if...elseif...else Statement:**

- This allows us to use multiple if...else statements.
- We use this when there are multiple conditions of TRUE cases.
- It is also called nested if-else statement.

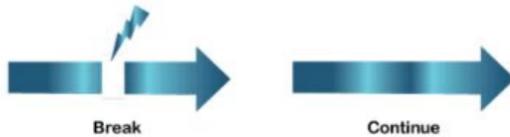
Syntax:**switch Statement:**

- The “switch” performs in various cases i.e., it has various cases to which it matches the condition and appropriately executes a particular case block.
- It first evaluates an expression and then compares with the values of each case.
- If a case matches then the same case is executed.
- To use switch, we need to get familiar with two different keywords namely, **break** and **default**.
 1. The **break** statement is used to stop the automatic control flow into the next cases and exit from the switch case.
 2. The **default** statement contains the code that would execute if none of the cases match.

Syntax:

Break and continue statements:

- Just like other programming languages, [PHP](#) also have two keywords **break** and **continue**.
- **It is used** to control the loop.
- These statements are known as jumping statement or flow of transfer in the program.



Break:

- The break statement is used inside the loop (for, foreach, while and do-while) and switch case statement.
- As the break statement encountered inside a loop, it immediately terminated the loop statement and transferred the control to the next statement, followed by a loop to resume the execution.
- It is also used in the nested loop statement, where it breaks the inner loop first and then proceed to break the outer loop.
- It is also used in the switch case statement to terminate a particular case's execution in the program.

Continue:

- The [continue statement](#) is used in the middle of [for loop](#), [while loop](#), [do-while loop](#), and the [foreach loop](#).
- As the continue statement is encountered in a loop, it skips the current iteration of the loop and transfers the control to the beginning of the loop for further execution of the loop.
- For example, in some situations where we want to skip the program's current statement and then starts the next statement, we use the continue statement.