Great Place To Work. Certified MAR 2021–FEB 2022 INDIA

UPES UNIVERSITY OF THE FUTURE

UPES

UNIVERSITY OF TOMORROW

nirf RANKING 2021
Ranked amongst **top 100** universities in **India**

NAAC Accredited with Grade A
Accredited **Grade** 'A' by NAAC

QS STARS RATING SYSTEM 2020
QS 5 Star Rating for Academic Development, Employability, Facilities and Program Strength

E-LEAD
Perfect score of **150/150** as a testament to exceptional E-Learning methods

University of the Year (North India) awarded by ASSOCHAM

SAFE GUARD
Certified for **safety and hygiene** by Bureau Veritas

Introduction to PHP

Presented By:

Sagar Kumar, SAPID : 500123720

Mentor :

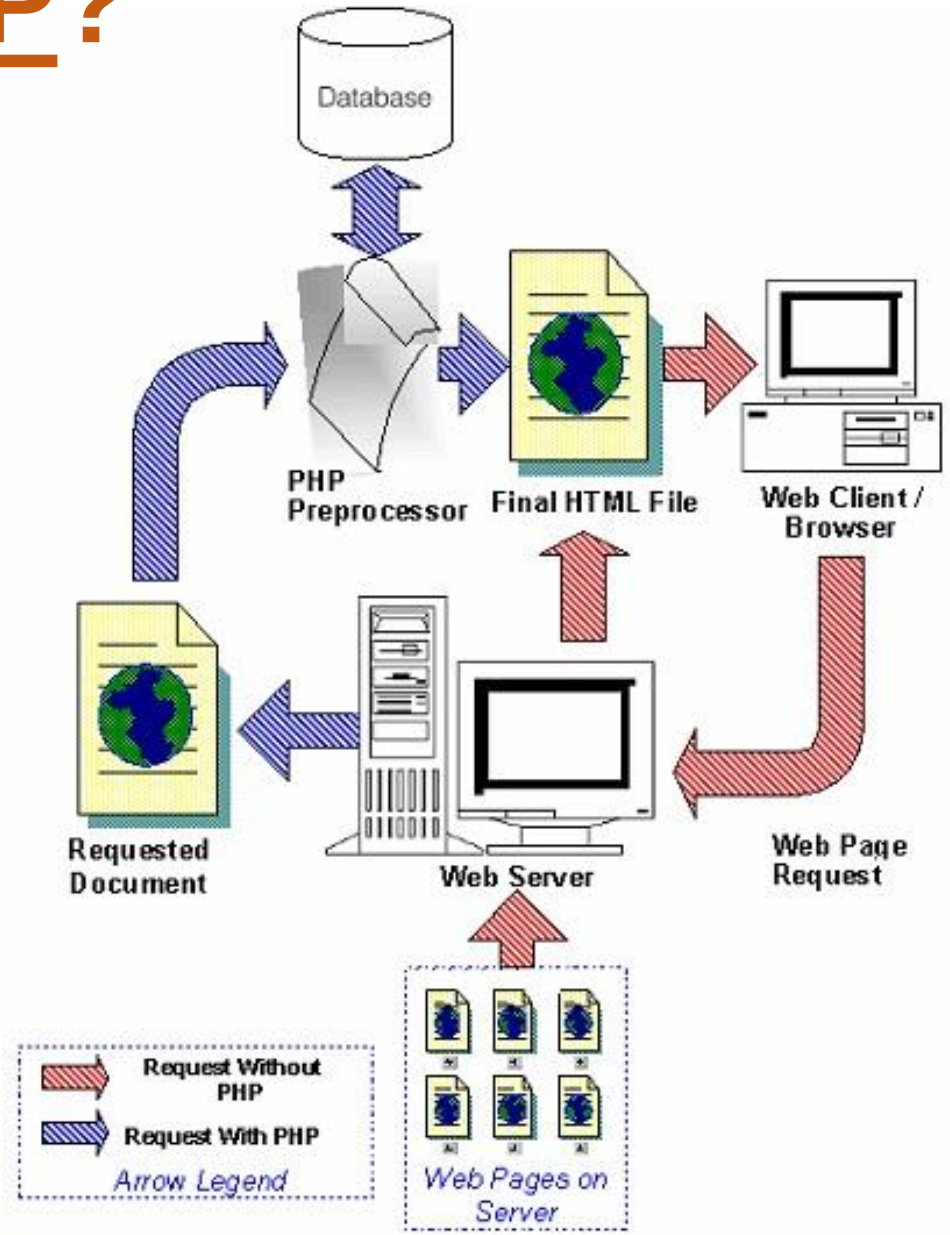
Prof. J. Dhiviya Rose Ma’am

School of Computer Science | UPES

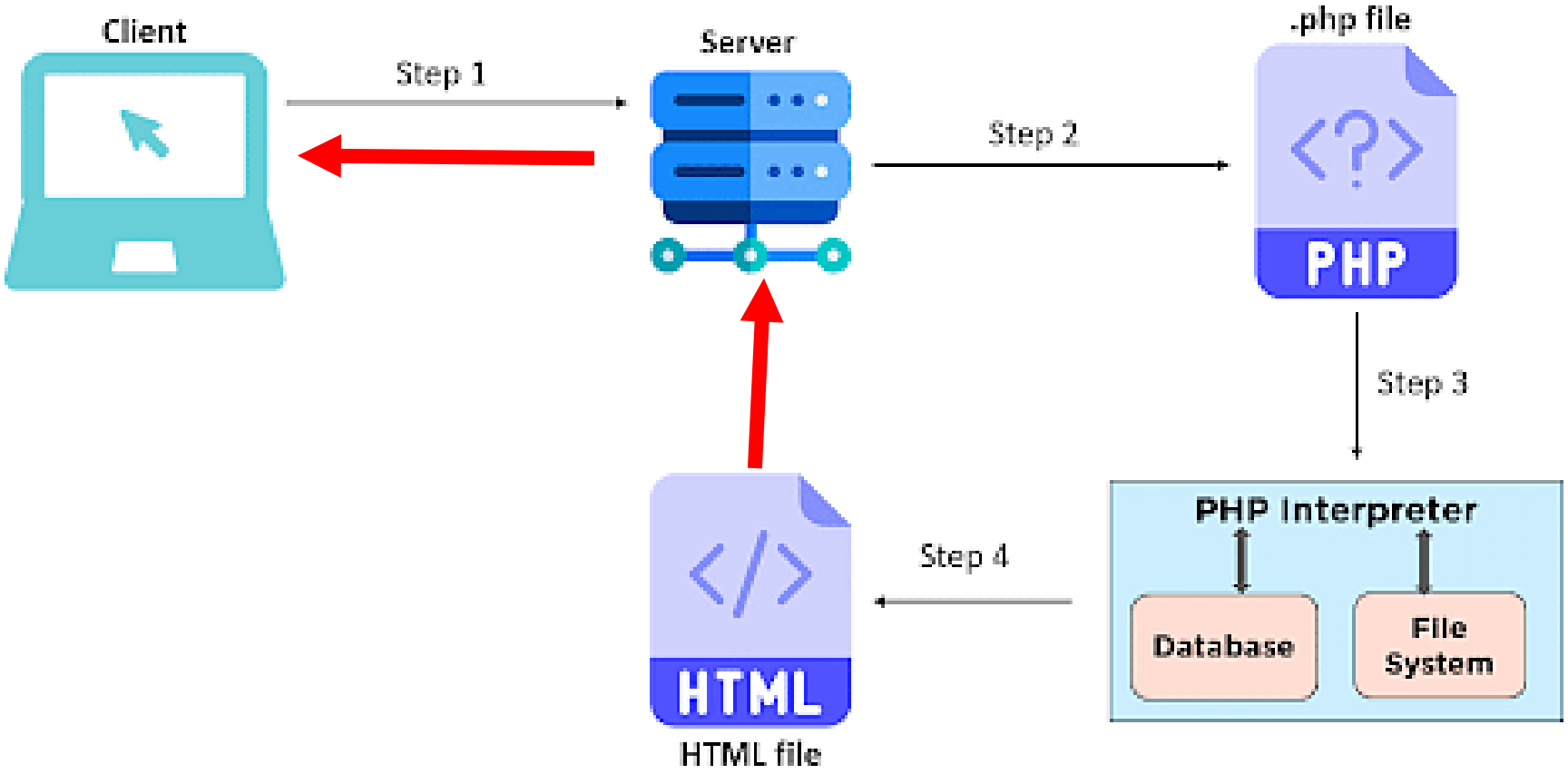
Table of Contents

1. What is PHP?

- **PHP** stands for **Hypertext Preprocessor**.
- A server-side scripting language designed for web development.
- Executes on the server, generating HTML which is then sent to the client.
- In PHP keywords (if , else , while , echo), classes, functions, and user defined functions are not case sensitive.
- However, all variable names are case-sensitive.

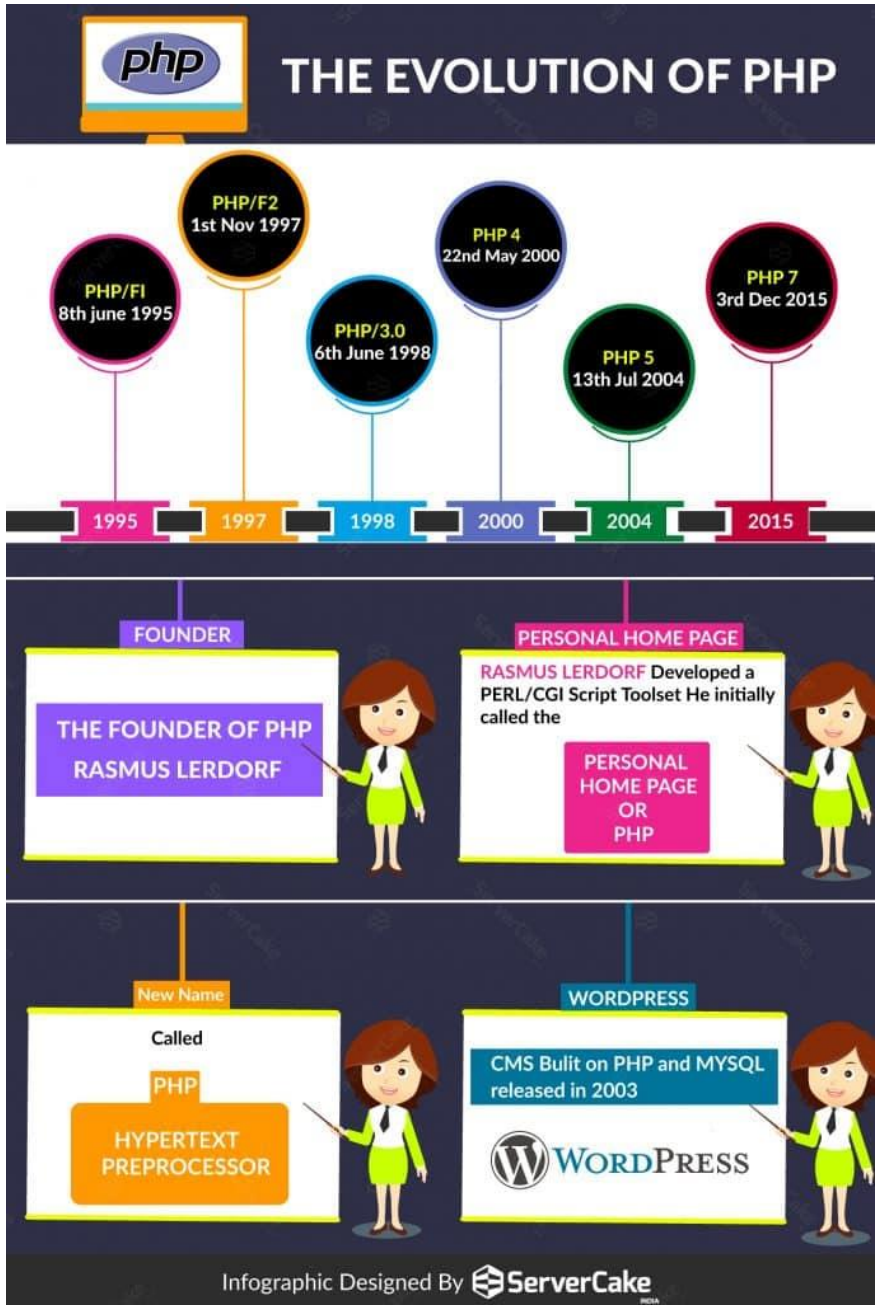


Working of PHP – A server side script's



2. Why Use PHP?

- **Versatility**: Can be used to create dynamic web pages, command-line scripts, and even desktop applications.
- **Integration**: Easily integrates with databases, such as MySQL, and other technologies.
- **Open Source**: PHP is open source and has a vast community for support and development.
- **Platform Independent**: Runs on various platforms like Windows, Linux, macOS, etc.



3. History of PHP

- Created in 1994 by **Rasmus Lerdorf** as a set of Perl scripts.
- Later rewritings and enhancements led to PHP/FI (Personal Home Page/Forms Interpreter).
- **PHP 3** (1997) marked a major rewrite and expansion of the language.
- **PHP 4** (2000) introduced many improvements and better support for web applications.
- **PHP 5** (2004) included a new object model and introduced improvements in performance and stability.
- **PHP 7** (2015) brought significant performance improvements and new features.

4. Basic Syntax & Output Statements

- PHP code is embedded within HTML.
- Start PHP code with “<?php” and end with “?>”.
- Example:

```
<?php
 echo "Hello, World!";
?>
```

`echo` and `print` are more or less the same. 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 (although such usage is rare) while `print` can take one argument. `echo` is marginally faster than `print`.

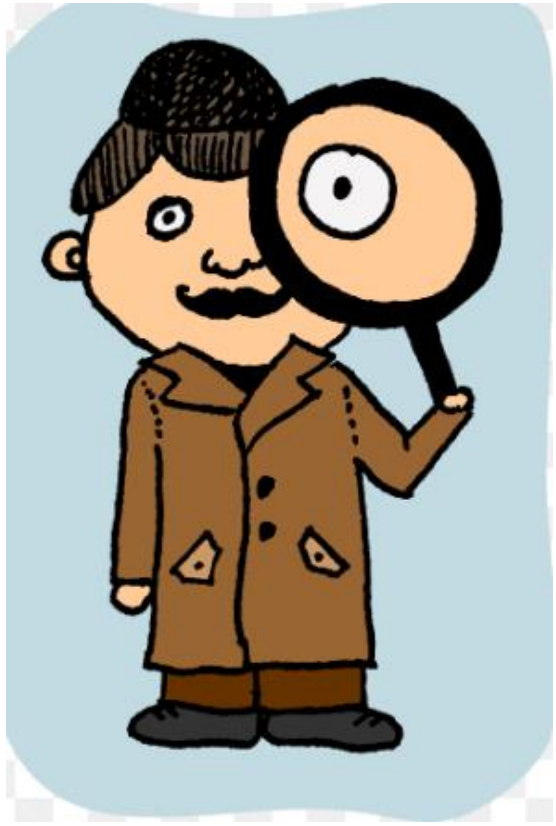
5. Variables

- Variables start with the '$' symbol followed by the variable name.
- Variable names are case-sensitive and must start with a letter or underscore.
- Example:

  $name = "John";

```php
<?php
$txt = "W3Schools.com";
echo "I love $txt!";
?>
```

```php
<?php
$txt = "W3Schools.com";
echo "I love " . $txt . "!";
?>
```

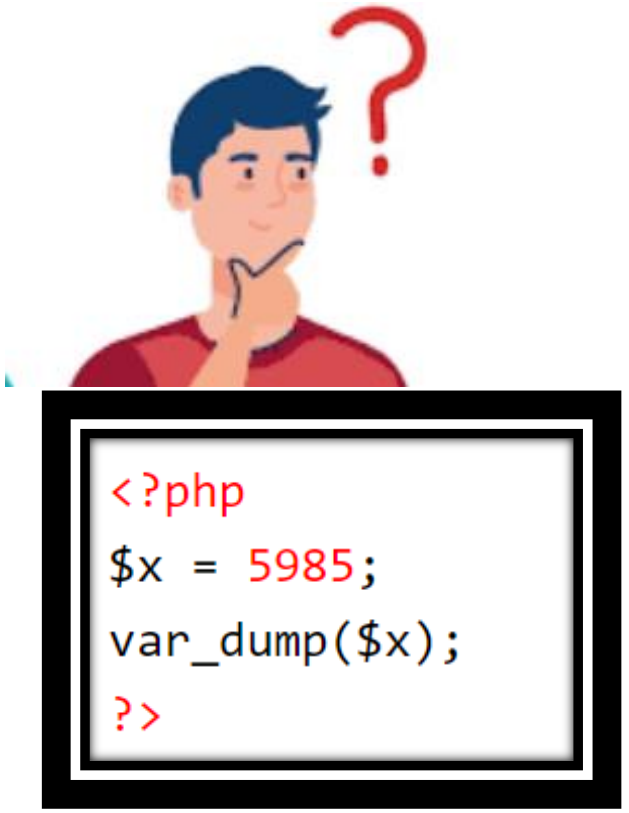


```php
<?php
$x = 5; // global scope

function myTest() {
  // using x inside this function will generate an error
  echo "<p>Variable x inside function is: $x</p>";
}
myTest();

echo "<p>Variable x outside function is: $x</p>";
?>
```

6. Data Types

PHP supports various data types, including:

- Integer
- Float
- String
- Boolean
- Array
- Object
- NULL



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

7. Operators

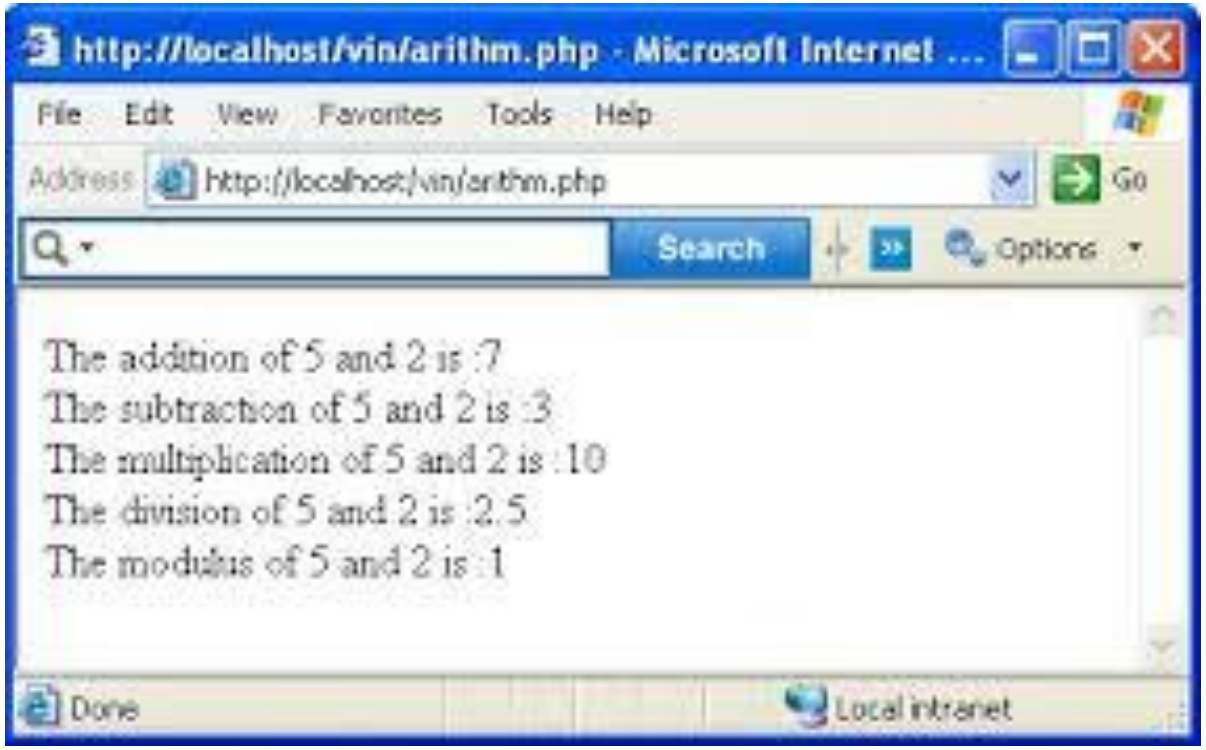
PHP supports various operators, including:

- Arithmetic operators: ‘ + ’ , ‘ – ‘ , ‘ * ‘ , ‘ / ‘ , ‘%’
- Comparison operators: ‘ == ‘ , ‘ != ‘ , ‘ > ‘ , ‘ < ‘ , ‘ >= ‘ , ‘ <= ‘
- Logical operators: ‘ && ‘ , ‘ || ‘ , ‘ ! ‘
- Assignment operators: ‘ = ‘ , ‘ += ‘ , ‘ -= ‘ , ‘ *= ‘ , ‘ /= ‘
- Concatenation operator: ‘ . ‘
- Increment/Decrement operators: ‘ ++ ‘ , ‘ – ‘

Operator	Name	Description
++$x	Pre-increment	Increments $x by one, then returns $x
$x++	Post-increment	Returns $x, then increments $x by one
--$x	Pre-decrement	Decrements $x by one, then returns $x
$x--	Post-decrement	Returns $x, then decrements $x by one

WORKOUT **Create a variable and try to print the various options**

ACTIVITY on Operators



Use two initialized variables
$no1 =5
$no2 =2

8. Control Structures - If-Else

- If-else statements are used for decision-making.
- Syntax:

```
if (condition) {
    // code to be executed if the condition is true
} else {
    // code to be executed if the condition is false
}
```

Example Of If-Else

```
<?php
$age = 20;
if ($age > 18) {
    echo "You are an adult.";
} else {
    echo "You are a minor.";
}
?>
```

Elseif

- **Elseif** allows for multiple conditions.
- Syntax:

```
if (condition1) {
    // code to be executed if condition1 is true
} elseif (condition2) {
    // code to be executed if condition2 is true
} else {
    // code to be executed if all conditions are false
}
```

Example Of Elseif

```
<?php
$grade = 75;
if ($grade >= 90) {
    echo "A";
} elseif ($grade >= 80) {
    echo "B";
} elseif ($grade >= 70) {
    echo "C";
} else {
    echo "F";
}
?>
```

Switch

- **Switch** is another way to handle multiple conditions.
- Syntax:

```
switch (expression) {
    case value1:
        // code to be executed if expression equals value1
        break;
    case value2:
        // code to be executed if expression equals value2
        break;
    default:
        // code to be executed if expression doesn't match any case
}
```

Example of Switch

```
<?php
$day = "Monday";
switch ($day) {
    case "Monday":
        echo "Start of the week";
        break;
    case "Friday":
        echo "TGIF!";
        break;
    default:
        echo "Another day";
}
?>
```

For Loop

- **For** loops are used for iteration.
- Syntax:

```
for (initialization; condition; increment) {
    // code to be executed in each iteration
}
```

- Example:

```
for ($i = 1; $i <= 5; $i++) {
    echo "Number: $i <br>";
}
```

While Loop

- **While** loops are used for iterative execution based on a condition.
- Syntax:

```
while (condition) {
    // code to be executed while the condition is true
}
```

- Example:

```
$num = 1;
while ($num <= 5) {
    echo "Number: $num <br>";
    $num++;
}
```

THANK YOU