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

PHP – Server Side Scripting

Presented By:

Tejendra Kumar Panda, SAPID : 500125197

Mentor :

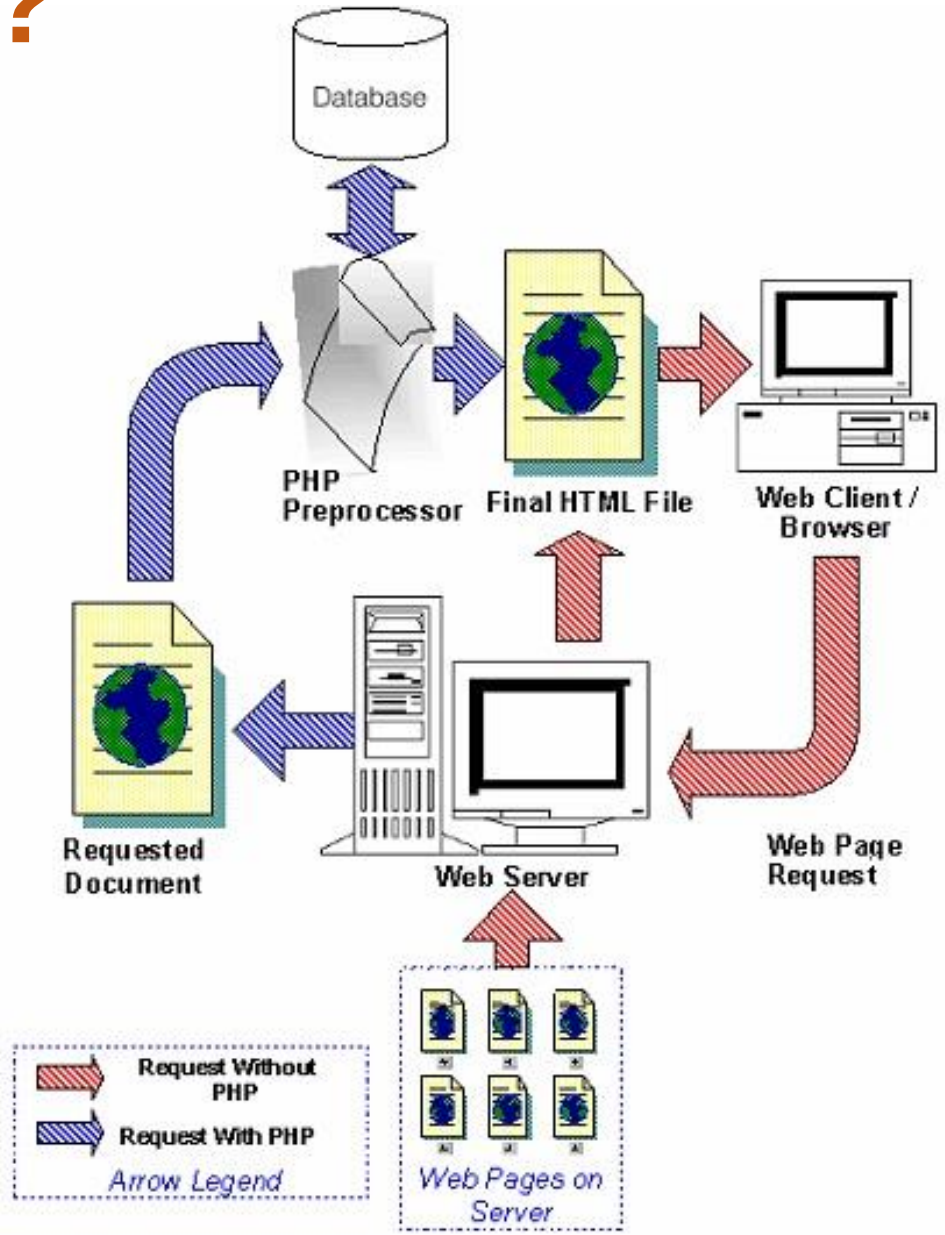
Prof. J. Dhiviya Rose Ma’am

School of Computer Science | UPES

RECOLLECT

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.



RECOLLECT

Working of PHP
Server side script's

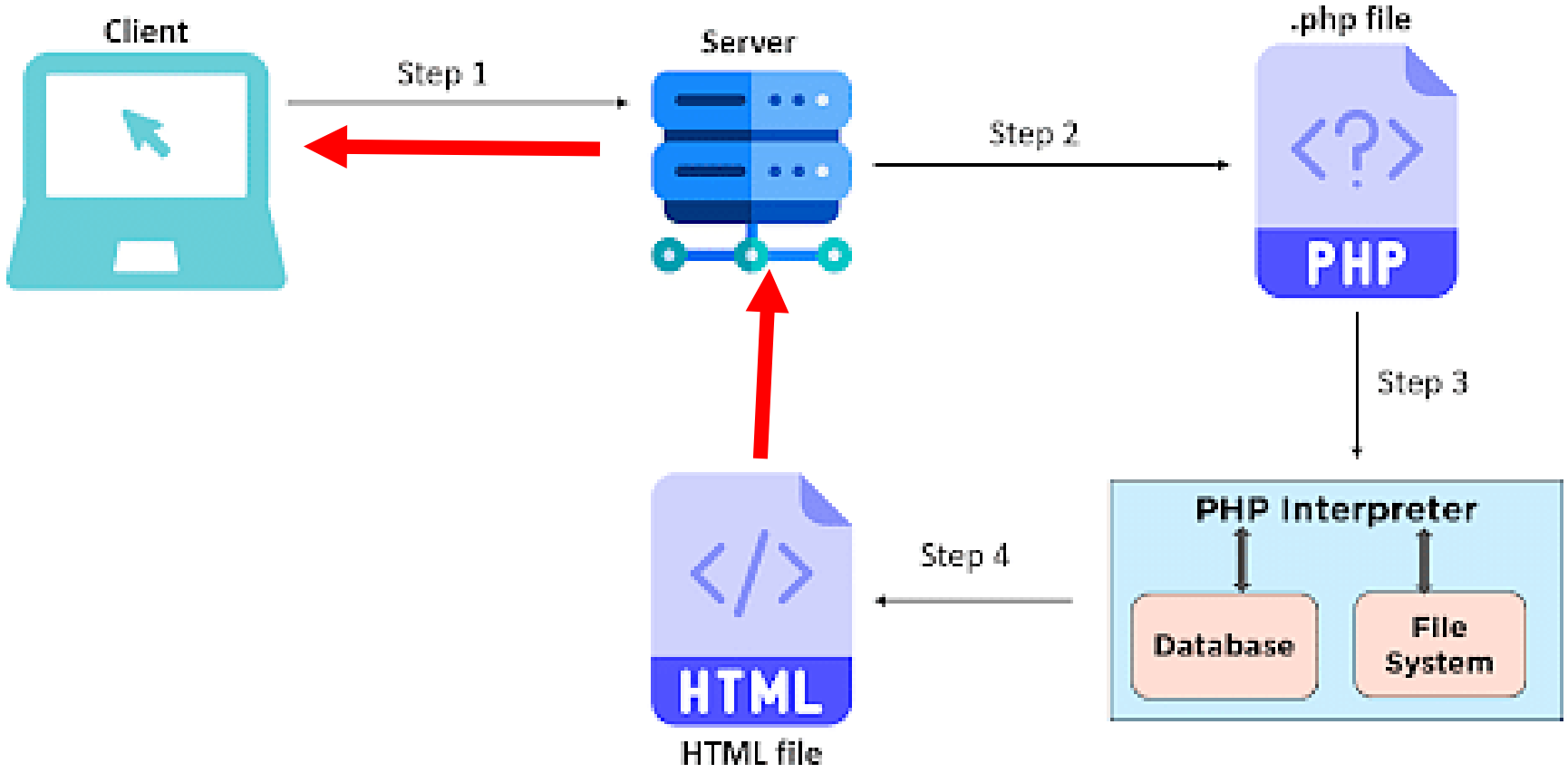


Table of Contents

RECOLLECT

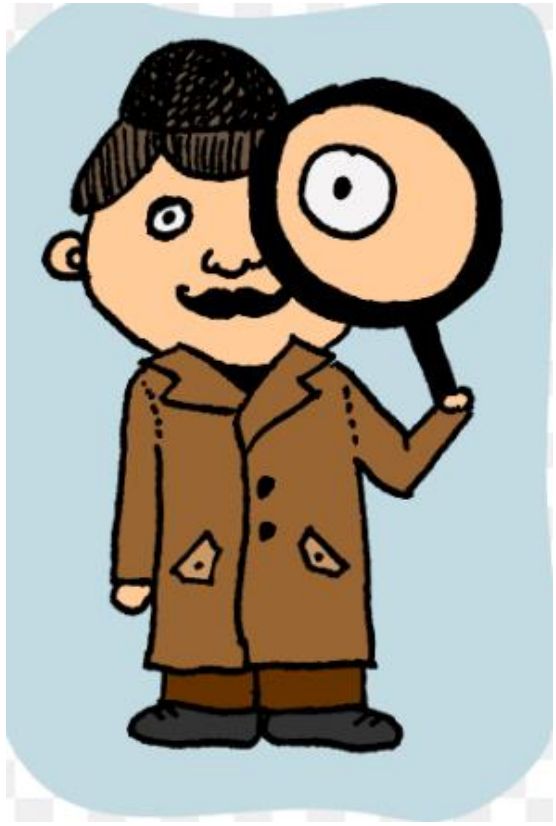
1. Variables

- 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 . "!";
?>
```



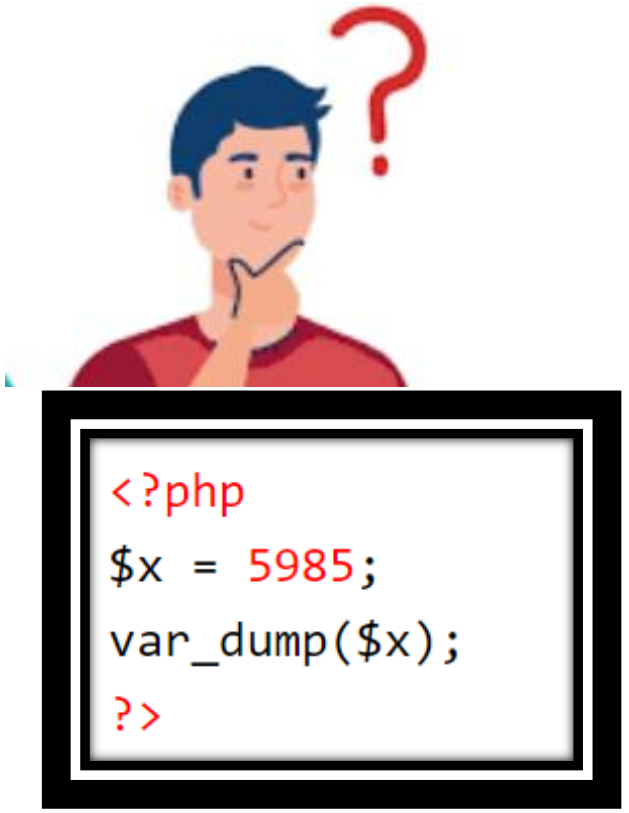
```php
<?php
$x = 5; // global scope

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>";
?>
```

- Where & How to write a PHP script
- Output Statement in PHP
- Create variables using $varname and use of var_dump()
- Create functions & call it
- How PHP scripts are executed ,What is a localhost

2. 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:

3. Operators

PHP supports various operators, including:

- Arithmetic operators
- Comparison operators
- Logical operators
- Assignment operators
- Concatenation operator
- Increment/Decrement operators

(i) PHP Arithmetic Operators

The PHP arithmetic operators are used with numeric values to perform common arithmetical operations, such as addition, subtraction, multiplication etc.

Operator	Name	Example	Result
+	Addition	$x + $y	Sum of $x and $y
-	Subtraction	$x - $y	Difference of $x and $y
*	Multiplication	$x * $y	Product of $x and $y
/	Division	$x / $y	Quotient of $x and $y
%	Modulus	$x % $y	Remainder of $x divided by $y
**	Exponentiation	$x ** $y	Result of raising $x to the $y'th power

(ii) PHP Assignment Operators

The PHP assignment operators are used with numeric values to write a value to a variable.
The basic assignment operator in PHP is "=". It means that the left operand gets set to the value of the assignment expression on the right.

Assignment	Same as...	Description
x = y	x = y	The left operand gets set to the value of the expression on the right
x += y	x = x + y	Addition
x -= y	x = x - y	Subtraction
x *= y	x = x * y	Multiplication
x /= y	x = x / y	Division
x %= y	x = x % y	Modulus

(iii) PHP Comparison Operators

The PHP comparison operators are used to compare two values (number or string):

Operator	Name	Example	Result
==	Equal	$x == $y	Returns true if $x is equal to $y
===	Identical	$x === $y	Returns true if $x is equal to $y, and they are of the same type
!=	Not equal	$x != $y	Returns true if $x is not equal to $y
<>	Not equal	$x <> $y	Returns true if $x is not equal to $y
!==	Not identical	$x !== $y	Returns true if $x is not equal to $y, or they are not of the same type
>	Greater than	$x > $y	Returns true if $x is greater than $y
<	Less than	$x < $y	Returns true if $x is less than $y
>=	Greater than or equal to	$x >= $y	Returns true if $x is greater than or equal to $y
<=	Less than or equal to	$x <= $y	Returns true if $x is less than or equal to $y
<=>	Spaceship	$x <=> $y	Returns an integer less than, equal to, or greater than zero, depending on if $x is less than, equal to, or greater than $y. Introduced in PHP 7.

(iv) PHP Increment / Decrement Operators

The PHP increment operators are used to increment a variable's value.
The PHP decrement operators are used to decrement a variable's value.

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

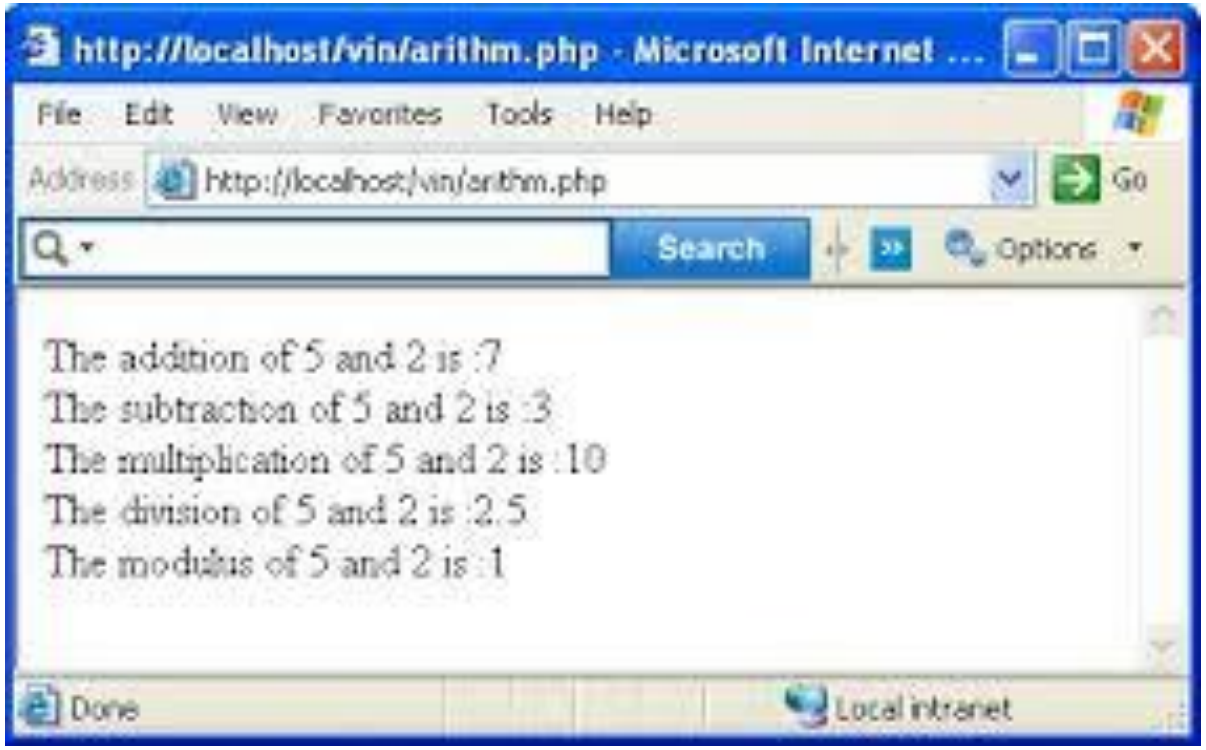
(v) PHP Logical Operators

The PHP logical operators are used to combine conditional statements.

Operator	Name	Example	Result
and	And	$x and $y	True if both $x and $y are true
or	Or	$x or $y	True if either $x or $y is true
xor	Xor	$x xor $y	True if either $x or $y is true, but not both
&&	And	$x && $y	True if both $x and $y are true
||	Or	$x || $y	True if either $x or $y is true
!	Not	!$x	True if $x is not true

For more operators Ref: https://www.w3schools.com/php/php_operators.asp

Activity on Operators



Use two initialized variables
$no1 =26
$no2 =8

4. Control Structures

If-Else

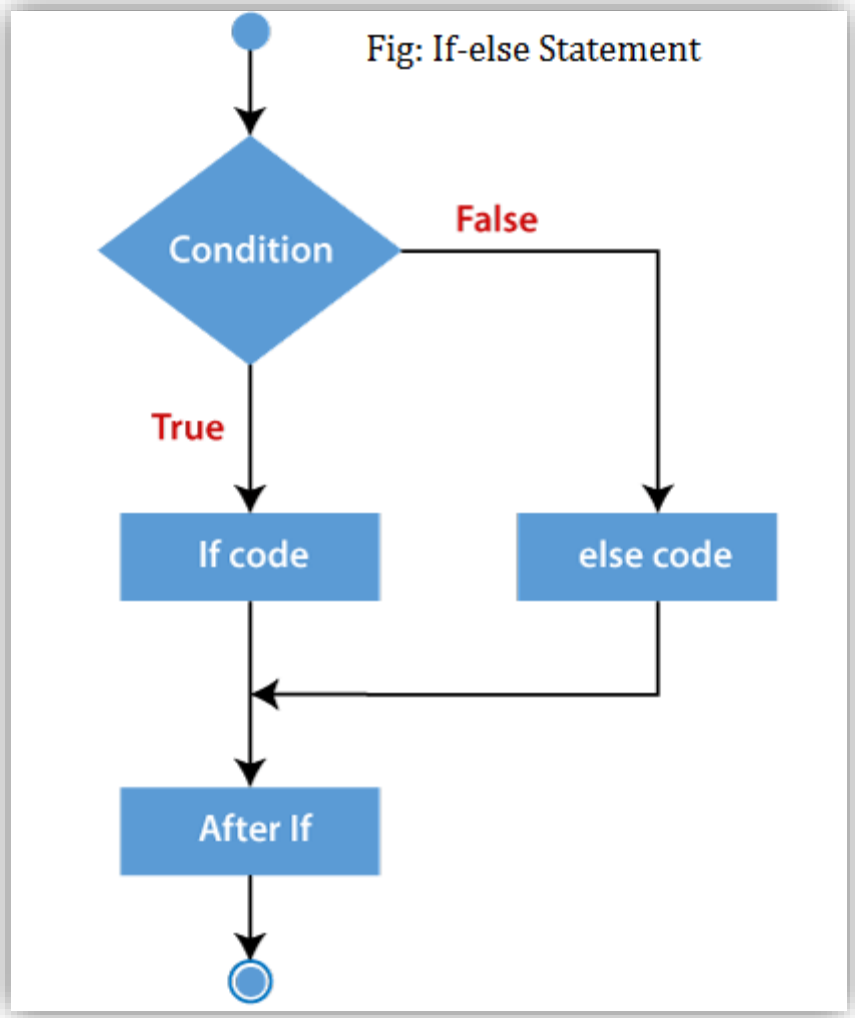
- If-else statements are used for decision-making.
- Syntax:

```
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
$num=12;
if($num<=100)
{
echo "$num is less than 100";
}
Else
{
echo "$num is greater than 100";
}
?>
```

Here is the Flowchart of If...else statement



Else-if

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

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

Example Of Else-if

```
<?php
    $marks=69;
    if($marks<=100 && $marks>=90{
        echo “A+”;
    }
   else if($marks <90 && $marks >=85){
   echo “A”;
   }
   else if($marks <85 && $marks>=65){
   echo “B”;
   }
   else {
        echo "Invalid input";
    }
?>
```

Here is the Flowchart of If-else-if statement

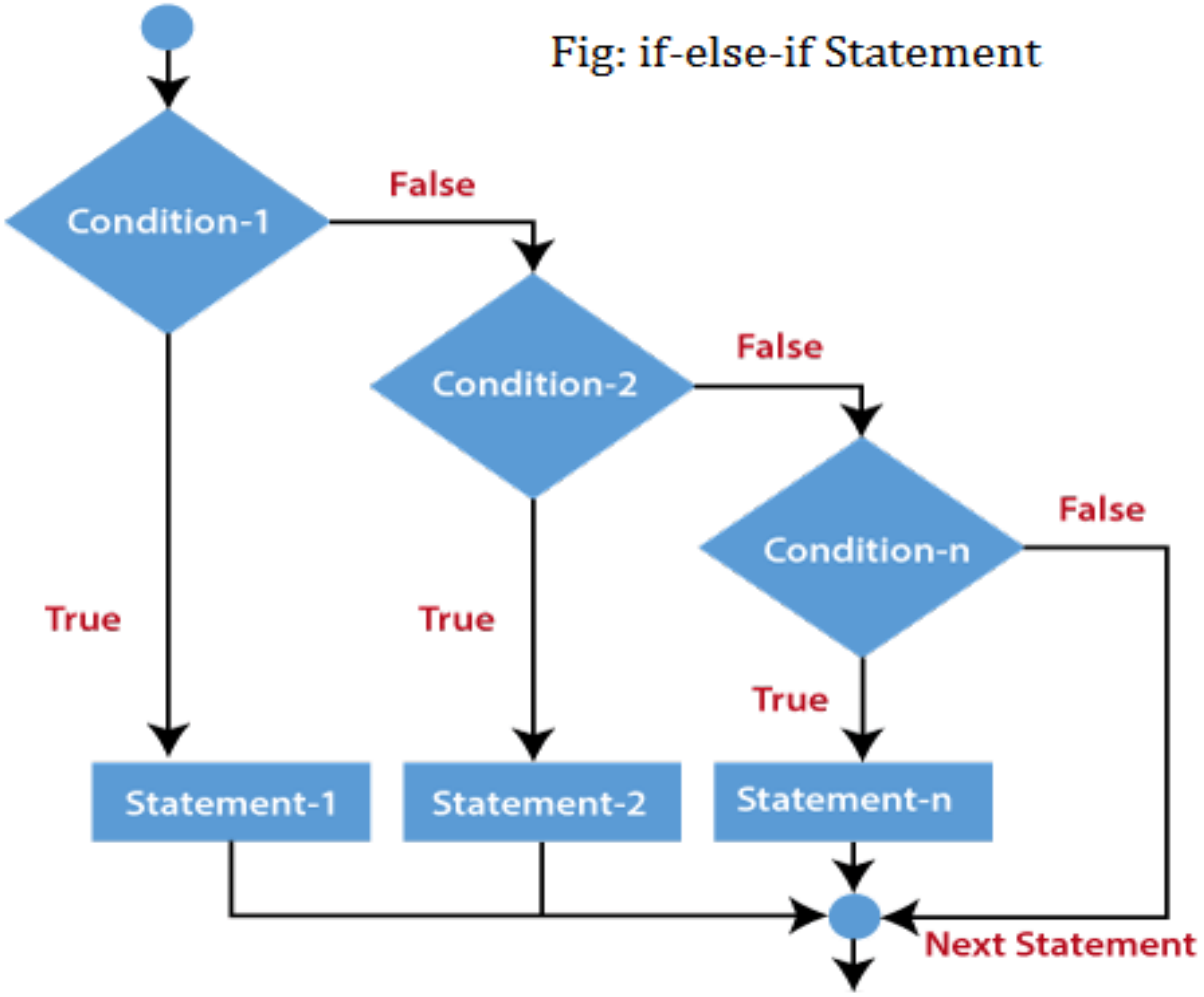


Fig: if-else-if Statement

Nested-If

The nested if statement contains the if block inside another if block. The inner if statement executes only when specified condition in outer if statement is **true.**

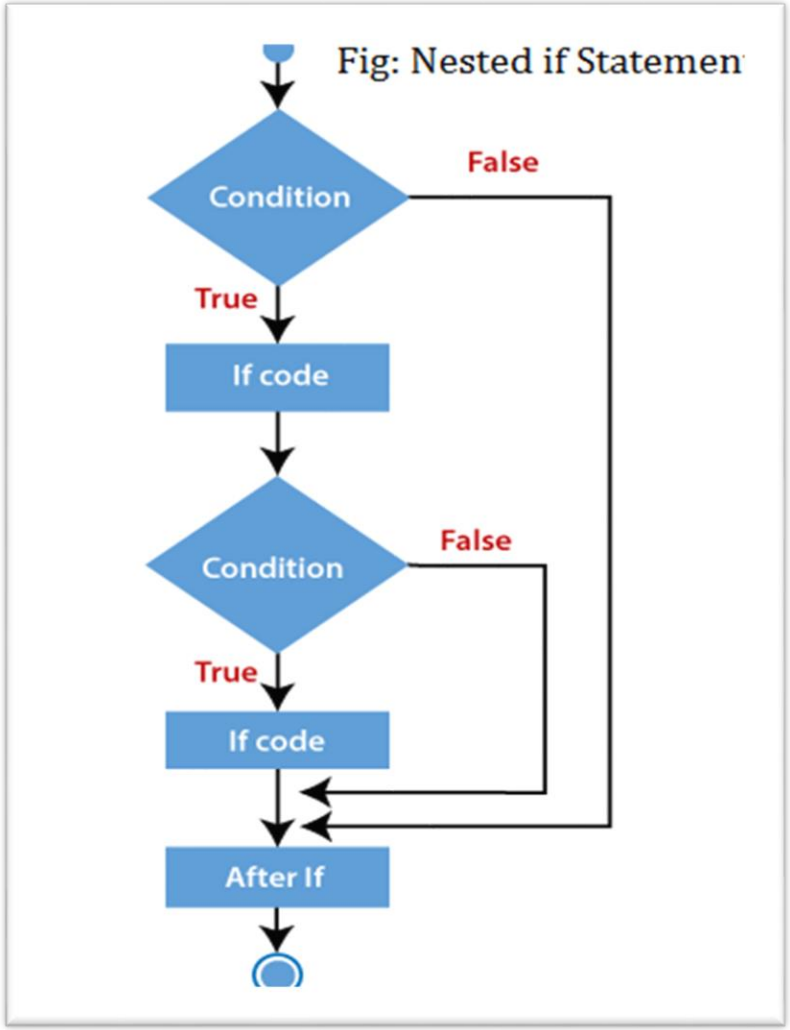
Syntax:

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

Example Of Nested if

```
1. <?php
2.              $age = 23;
3.    $nationality = "Indian";
4.     //applying conditions on nationalit
   y and age
5.    if ($nationality == "Indian")
6.    {
7.        if ($age >= 18) {
8.            echo "Eligible to give vote";
9.        }
10.        else {
11.            echo "Not eligible to give vo
   te";
12.       }
13.   }
14.?>
```

Here is the Flowchart of Nested-if statement



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
<?php
    $ch = "Bsc";
    switch ($ch)
    {
        case "BCA":
            echo "BCA is 3 years course";

            break;
        case "Bsc":
            echo "Bsc is 3 years course";
            break;
        default:
            echo "Wrong Choice";
            break;
    }
?>
```

Ref:https://www.javatpoint.com/php-switch

ACTIVITY TIME

Make an Electricity Bill Calculator in PHP using the **conditional statements** with the following conditions specified below.

- For the first 50 units – Rs. 3.50/unit
- For next 100 units – Rs. 4.00/unit
- For next 100 units – Rs. 5.20/unit
- For units above 250 – Rs. 6.50/unit

Note: Use a PHP variable that stores the no_of_units fixed and not a dynamic input from the form and take a snapshot by changing the values to test for all 4 ranges.

For Loop

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

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

- Example:

```
for($n=1;$n<=10;$n++){
echo "$n<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:

```
$n=1;
while($n<=10){
echo "$n<br/>";
$n++;
```

Do-While Loop

The main difference between both loops is that while loop checks the condition at the beginning, whereas do-while loop checks the condition at the end of the loop. It executes the code at least one time always because the condition is checked after executing the code.

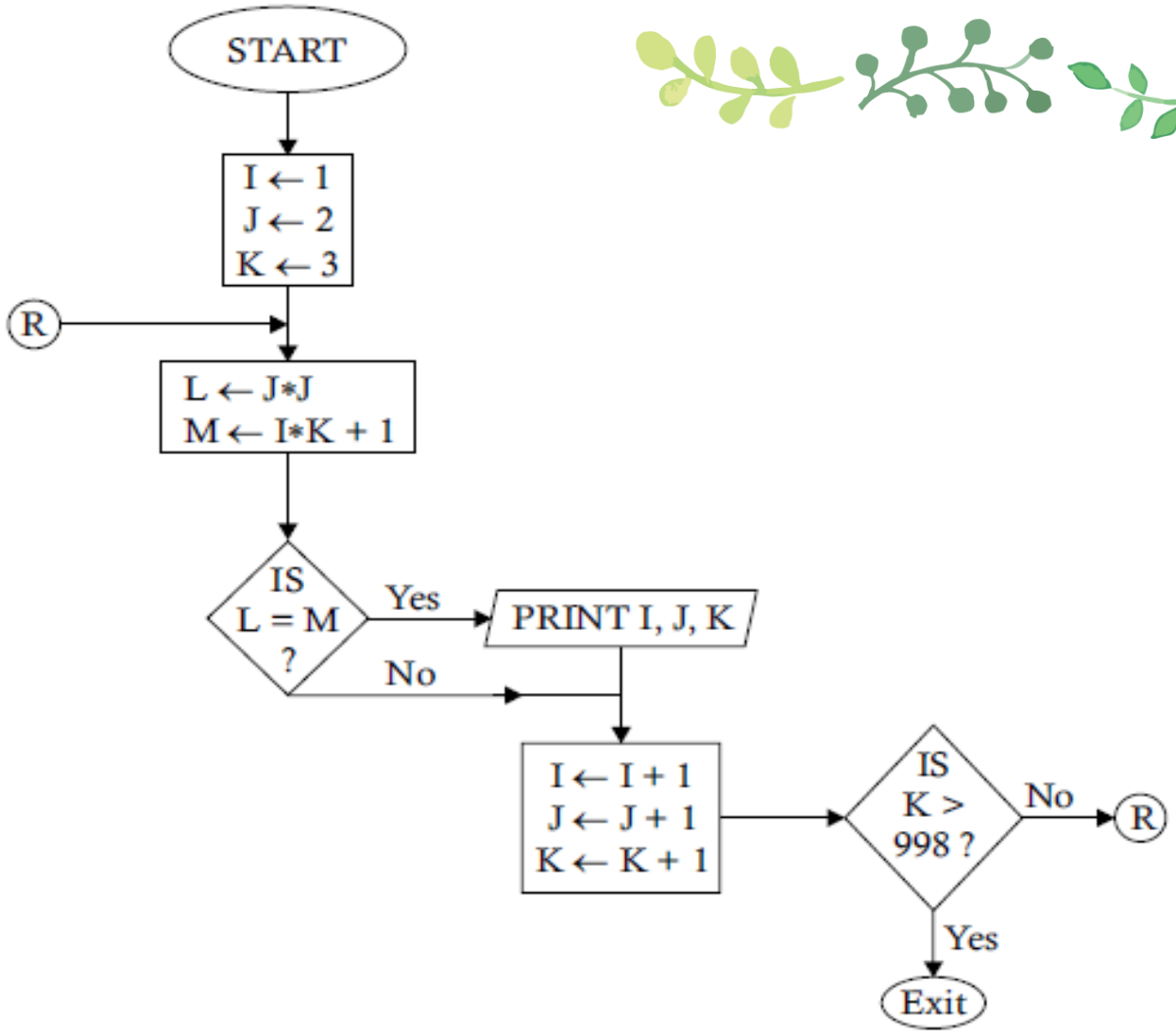
Syntax:

```
do{
//code to be executed
}while(condition);
```

Example:

```
$n=1;
do{
echo "$n<br/>";
$n++;
}while($n<=10);
```

ACTIVITY TIME



Implement It!

More Activity ..

Working on simple PHP programmings by using conditional and looping statements

Write A Program in PHP to check :

- If the entered number is a Armstrong number or not.

  Ex: 371=3^3+7^3+1^3=371

- If the given number is a Palindrome number or not

  Ex: 1,2,3,4,5,6,7,8,9,11,22,33,44,… all are palindrome numbers.

- Print Fibonacci series

THANK YOU