

# Bachelor Of Engineering In Information Technology

Semester Six, Third Year(Even semester)

30<sup>th</sup> May 2022

25<sup>th</sup> Offline Lecture

Padre Conceicao College of Engineering

Verna Goa 403722 India

# Web Technology

**RC 2019-20**

**Unit 3**

# Working with Variables and Constants

Chapter 4

# Introduction

- In PHP :Hypertext Preprocessor (PHP), variables and constants are used to store data so that programs can process correctly and efficiently. A variable can be defined as a symbolic name associated with a value, which can be changed during the execution of a program. The value associated with a variable can be numeric or alpha -numeric. Constants are used to store values that can not be changed during execution.

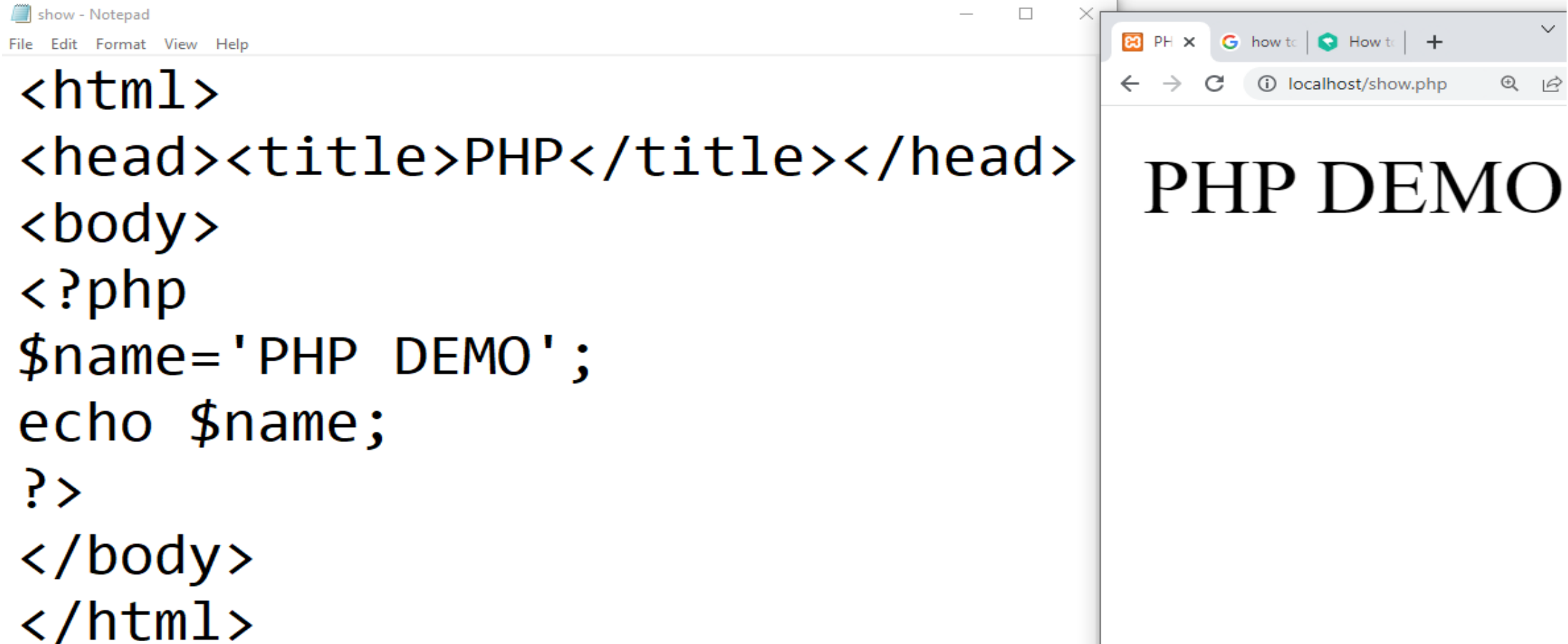
# Using Variables: Naming rules and conventions

1. The naming rules for PHP Variables are as follows:
2. Every variable must be preceded by the \$ sign.Eg \$var is valid,var is invalid
3. A variable must start with a letter or an underscore “\_”.eg\$\_var, \$var are valid.\$1var is invalid
4. A variable name can only contain alpha-numeric characters and underscores(a-z,A-Z,0-9 and \_).Eg\$my\_var is valid.\$23# is invalid
5. A variable name should not contain spaces.If it is more than 1 word ,it should be separated by underscore or capitalization.Eg. \$my\_var,\$myvar are valid. ,\$my var is invalid
6. PHP has no limit on the length of variable name as in the case of other programming languages

# Assigning values to Variables

- In PHP , value is assigned to a variable using the equality sign, which is also known as the assignment operator in PHP.
- This operator assigns the value on the right side of the equation to the variable on the left side

# Assigning values to Variables



The image shows a side-by-side comparison of a PHP script in a text editor and its output in a web browser. On the left, a Notepad window titled 'show - Notepad' contains the following PHP code:

```
<html>
<head><title>PHP</title></head>
<body>
<?php
$name='PHP DEMO';
echo $name;
?>
</body>
</html>
```

On the right, a web browser window shows the output of this script. The address bar indicates the file is located at 'localhost/show.php'. The main content area of the browser displays the text 'PHP DEMO' in a large, black, serif font.

# Performing various operations on variables

op - Notepad

File Edit Format View Help

```
<html>
<head><title>PHP operations</title></head>
<body>
<?php
$current_value=10;//assigning value to variable
$value=$current_value;//assigning variable to variable
$next_value=$value+1;//perform calculation
echo "$next_value comes after $current_value";//output:'11 comes after 10'
?>
</body>
</html>
```



# Performing various operations on variables

← → ↻ ⓘ localhost/op.php

11 comes after 10

# Assigning a value to a variable using the Assign by Reference Method

- You can assign values to variables by using the assign by reference method.
- In this method , the new variable points to another variable and changes done in any of the variables affect both the variables.
- In this method, the variable name must be preceded by an ampersand (&) sign before passing its reference to another variable.

# Assigning a value to a variable using the Assign by Reference Method

ref - Notepad

File Edit Format View Help

```
<html>
<head><title>PHP operations</title></head>
<body>
<?php
$name='kogent India';//assign the value 'kogent India' to $name
$alt_name=&$name;    //reference $name via $alt_name
$alt_name="we are $alt_name";//alter $alt_name
echo $alt_name;
echo $name;          //$name is altered too
?>
</body>
</html>
```

# Assigning a value to a variable using the Assign by Reference Method

← → ↻ ⓘ localhost/ref.php

we are kogen India we are kogen India

# Assigning variable name dynamically

- PHP allows you to set the name of the variable dynamically.
- In this case , a value assigned to a variable itself becomes a variable.
- \$name variable is assigned the value , kogent.
- In the next statement, the value assigned to \$name itself becomes a variable (\$kogent) that holds a value, PHP book.
- The value of the \$kogent variable is then printed using the echo statement.

# Assigning variable name dynamically

dy - Notepad

File Edit Format View Help

```
<html>
<head><title>PHP variables</title></head>
<body>
<?php
$name='kogent';//create a new variable
${$name}='PHP book';//its name comes dynamically from the value of $name
echo $name;
?>
</body>
</html>
```

# Assigning variable name dynamically

A simulated web browser address bar with a light gray background. It contains navigation icons on the left: a left arrow, a right arrow, and a circular refresh icon. To the right of these icons is a small circular icon with an 'i' inside, followed by the text 'localhost/dy.php' in a sans-serif font.

← → ↻ ⓘ localhost/dy.php

kogent

# Destroying variables

- PHP allows you to destroy variables or an element from an array when they are not required. You can destroy variables or elements from an array by using the `unset()` function.
- The `$name` variable is assigned a string, we are Kogent India and then the echo statement prints 'Before destroying, we are Kogent India'. In the next statement, the `unset()` function is used to destroy the `$name` variable and now the echo statement shows an error message as the `$name` variable has been destroyed.
- It depends on the PHP error reporting configuration whether the error is printed or not.



# Destroying variables

des - Notepad

File Edit Format View Help

```
<html>
<head><title>destroy variables</title></head>
<body>
<?php
$name='we are kogent India';//assign value to variable
echo "Before destroying,$name";//print variable value
unset($name);//destroy variable
echo "After destroying,$name";//print variable value
?>
</body>
</html>
```

# Destroying variables

← → ↻ ⓘ localhost/des.php

Before destroying, we are kogen India

**Warning: Undefined variable \$name in C:\xampp\htdocs\des.php on line 8**

After destroying,

# Bachelor Of Engineering In Information Technology

Semester Six, Third Year(Even semester)

1<sup>st</sup> June 2022

26<sup>th</sup> Offline Lecture

Padre Conceicao College of Engineering  
Verna Goa 403722 India

# Using Constants

Constants are identifiers that store values, which cannot be changed during the execution of the script. You can store data, such as configuration settings, whose value does not change during the execution of the script. Constants are also used to represent integer values with special meanings in a particular context, such as error codes and flags. Constants are case sensitive. By convention, the constant identifier name is always in uppercase. Constants follow same naming rules as variables; unlike variables the constant names do not start with the \$ sign. A valid constant name starts with a letter or underscore, followed by any number of letters, digits, or underscores. Constants can only hold scalar values, such as numbers, strings and Boolean values. Constants cannot hold arrays or objects. PHP also include a set of pre-defined built-in constants.

# Syntax to create a constant

```
define("CONSTANT_NAME",constant_value);
```

**CONSTANT\_NAME** refers to the name of the constant  
**constant value** refers to the value that the constant holds.

# Creating and displaying Constants

co - Notepad

File Edit Format View Help

```
<html>
<head><title>constant</title></head>
<body>
<?php
define("HELLO","Hello");
define("KOGENT","Kogent");
echo HELLO;
echo KOGENT;
?>
</body>
</html>
```

constant

localhost/co.php

HelloKogent

# Exploring Data Types in PHP

- Data Type describes categories of values that a programming language can use. A data type refers to the type of data that a variable can hold. PHP includes 8 data types, which are as follows.

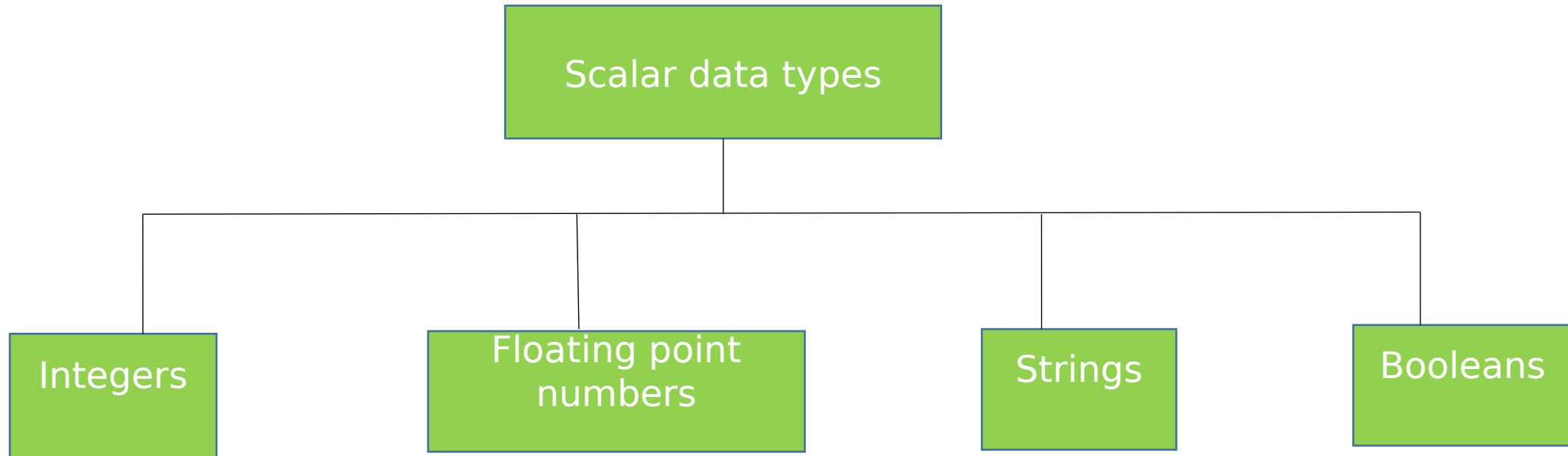
1. Integers
2. Floating point numbers
3. Strings
4. Booleans
5. Arrays
6. Objects
7. Resources
8. Null

# Exploring Data Types in PHP

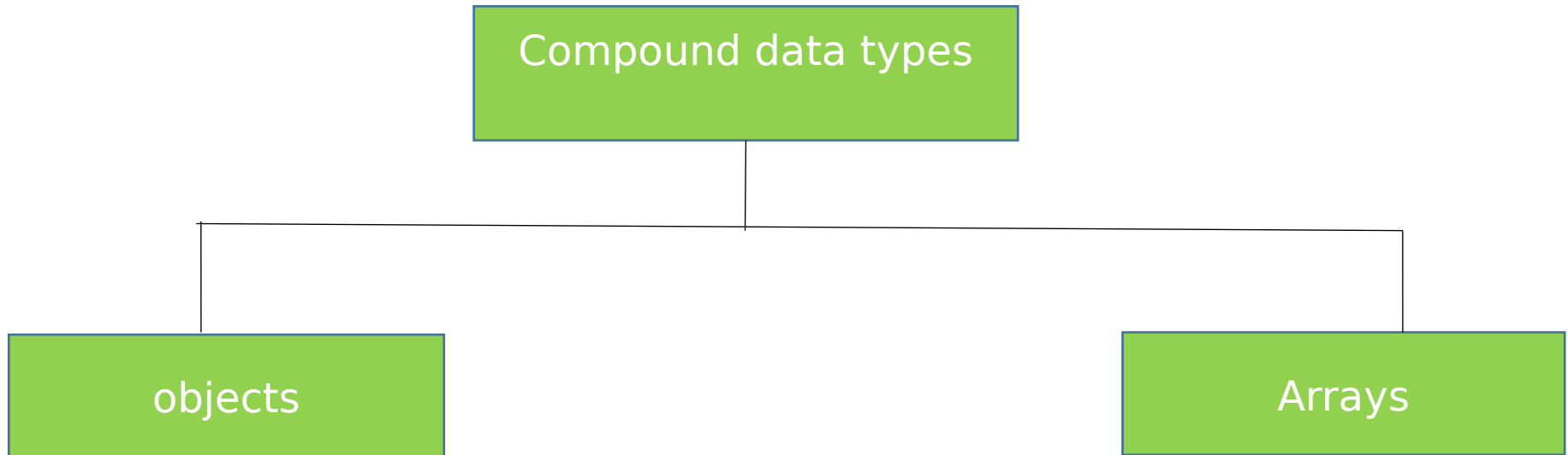
- In PHP ,it is not necessary to specify the data type of a variable before using it. PHP automatically decides the data type according to the value assigned to the variable. PHP data types can be broadly divided into 2 categories.
  1. Scalar or primitive data types
  2. Compound data types



# Scalar or primitive data types



# Compound data types



- Integers-Represents a whole number with no fractional components, such as -21,0,67.
- The permissible range of the integer data type varies. The range is decided by the operating system, generally a range of 2,147,483,648 to +2,147,483,647 is used. Integers can be written in decimal (1,786,-32,+1063856),octal(01,0123,+05,-0123456),or hexadecimal (0x1,0xff,0x1a3,+0x6,-0x1a234d) format.
- Floating point numbers – Represents real numbers that include decimal place . PHP includes two types of floating point numbers. The first is a simple numeric literal with a decimal point. The second is a floating-point number written in scientific notation. Scientific notation is in the form of [number]E[exponent].For example,1.23,0.003,-2.13,0.214E2,-3.14E3.

- Strings – Represents text literals of arbitrary length. In PHP , strings are enclosed within single quotes or double quotes. Strings inside double quotes are parsed ,while strings inside single quotes are not. It means that if variables or special characters are enclosed in double-quotes with strings, then the values of variables are printed with the specified string. When variable names and special characters are enclosed in single quotes, then the output is printed in the same way as you typed them.

# single quotes and double quotes

qu - Notepad

File Edit Format View Help

```
<html>
<head><title>quotes</title></head>
<body>
<?php
$name="Kogent India";//value assigned to $name
echo "we are $name";//prints 'we are kogent India'
echo 'we are $name';//prints 'we are $name'
?>
</body>
</html>
```

quotes

localhost:9090/qu.php

we are Kogent India we are \$name

# Booleans

- Represents a true or false value. All conditions return a true/false Boolean value based on the condition being tested. Some of the statements always return a false value, which are as follows:
  1. The keyword literal false
  2. The integer 0
  3. The floating point number 0.0
  4. The empty string ("")
  5. The string "0" (zero)
  6. An object with no values or methods
  7. The null value.

# Array

- Represents a variable that stores a collection of related data elements. Each individual element of an array can be accessed by referring to its index position. The position is either specified numerically or alphabetically.

# Object

- Allows you to store data as well as information to process that data. The data elements stored within an object are referred to as its properties or attributes of the object. To declare objects, first you must declare a class of object. Then you need to instantiate the object. Objects also allow you to create your own data types. You can define the data type in the object class and then use the data type in instances of that class.



Resource - Represents a special data type, which stores references to functions and resources external to PHP. The most common example of the resource data type is a database call.

NULL - Represents a special data type that can have only 1 value, null. Null is not only a data type, but also a keyword literal. A variable of the null data type is a variable that has no value assigned to it. When no value is assigned to a variable, it is automatically assigned a value, null.

# Determining Variable data Type

- PHP automatically determines the data type of variables from the value that the variable. If the value of a variable changes in the program, PHP automatically sets the appropriate new data type. The `gettype()` function is used to determine the datatype of a particular variable. The `$we_are` variable is assigned the value, Kogent.
- Therefore, PHP sets the datatype of the `$we_are` as string. The value of `$we_are` is changed to 99.8 which makes it a floating point variable. Next, the variable is destroyed using the `unset()` function which makes it a NULL type variable.

# Determining Variable data Type using gettype()

var - Notepad

File Edit Format View Help

```
<html>
<head><title>Variable data type</title></head>
<body>
<?php
$we_are='kogent';//define string variable
echo gettype($we_are);//output: 'string'
$we_are=99.8; //assign new integer value to variable
echo gettype($we_are);//output: 'double'
unset($we_are);//destroy variable
echo gettype($we_are); //output: 'NULL'
?>
</body>
</html>
```

Variable data type

localhost:9090/var.php

stringdouble

Notice: Undefined variable: we\_are in C:\xampp\htdocs\var.php on line 10  
NULL

# Table 4.1:List of specialized functions

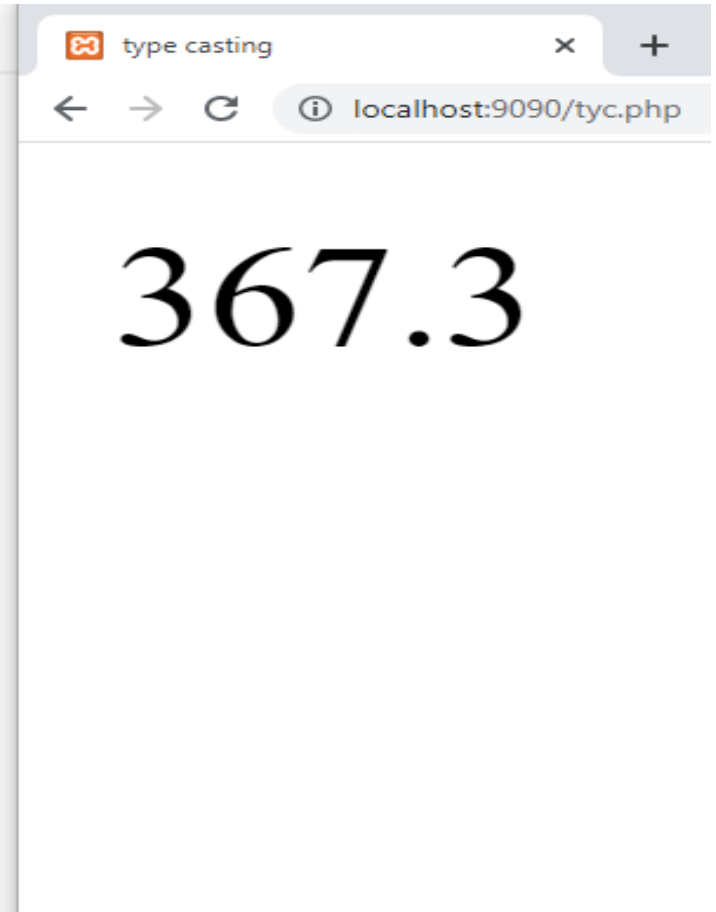
Sr.No	Functions	Purpose
1	is_bool()	Tests if a variable holds a Boolean value
2	is_numeric() )	Tests if a variable holds a numeric value
3	is_int()	Tests if a variable holds an integer
4	is_float()	Tests if a variable holds a floating-point value
5	is_string()	Tests if a variable holds a string value
6	is_null()	Tests if a variable holds a NULL value
7	is_array()	Tests if a variable is an array
8	is_object()	Tests if a variable is an object

# Using Type Casting

- In PHP ,you can convert the data type of a variable to another data type. For example, if a variable is of type integer and you want to store string value in it then you can convert the integer data type to the string data type. This form of converting data type of a variable to another data type is known as type casting. PHP does not require explicit type definition in variable declaration. An example of automatic type conversion of PHP is the addition (+) operator. If either operand is a float, then both operands are evaluated as floats, and the result will be a float. Otherwise, the operands are interpreted as integers, and the result is also an integer. Note that this does not change the data types of the

# Demonstration of Type Casting

```
tyc - Notepad
File Edit Format View Help
<html>
<head><title>type casting</title></head>
<body>
<?php
$x="3";//$x is string (ASCII 48)
echo $x;//output is 3
$x+=3;//$x is now an integer (3)
echo $x; //output is 6
$x=$x+ 1.3;//$x is now a float (3.3)
echo $x;//output is 7.3
?>
</body>
</html>
```



- The \$x variable is assigned the value 3, as string.
- In the next statement, the value of \$x is incremented by 3 and the data type of the variable changes to integer implicitly.
- In the next statement, \$x is incremented by 1.3 and the data type of \$x is now changed to float.
- You can explicitly type cast a variable by enclosing the name of the desired type in parentheses before the variable which is to be cast.

# Syntax for explicit type casting

`variable = (target_type) variable`

`variable` refers to a variable whose data type you want to convert  
`target_type` refers to the target data type to which you want to convert the data type of a variable

Example

`$x=(boolean)$x;      //converts data type of $x to boolean`

`$x=(float)$x;      //converts data type of $x to float`

`$x=(binary)$x;      //can also be used in place of  
binary.converts data type of $x to binary`



# settype() function

- Another way to type cast a variable is to use the settype() function. The syntax to use the function is as follows:

```
settype( variable_name , "target_type ");
```

`variable_name` refers to a variable whose data type you want to convert

`target_type` refers to the target data type to which you want to convert the data type of a variable.

Example:

```
settype($x, "array");           //converts data type of $x to array
settype( $x , "bool");           //converts data type of $x to
boolean
```

# Exploring Operators in PHP

- Operators in PHP are used to perform various operations such as assign,multiplication,addition,subtraction, and concatenation on variables and values. Operators work with operands, which specify the variables and values that are to be used in a particular operation.
- Different PHP Operators are as follows:
  1. Assignment Operators
  2. Arithmetic Operators
  3. String Operators
  4. Comparison Operators
  5. Logical Operators
  6. Increment/Decrement Operators
  7. Arithmetic Assignment Operators

# Assignment Operators

- Assignment Operator is used to assign values to variables or assign one variable to another as a value. Such an Assignment of value is done with the equal ( = ) operator.

- For Example

```
$my_var = 9;
```

```
$another_var = $my_var ;
```

Both the `$my_var` and `$another_var` variables contain the value 9.

Assignment Operators can also be used in conjunction with Arithmetic Operators

# Arithmetic Operators

Sr.No	Example	Name	Result
1	$-\$x$	Negation	Displays opposite of $\$x$
2	$\$x + \$y$	Addition	Displays addition of $\$x$ and $\$y$
3	$\$x - \$y$	Subtraction	Displays difference of $\$x$ and $\$y$
4	$\$x * \$y$	Multiplication	Displays product of $\$x$ and $\$y$
5	$\$x / \$y$	Division	Displays quotient after dividing $\$x$ by $\$y$
6	$\$x \% \$y$	Modulus	Displays remainder after dividing $\$x$ by $\$y$

# String Operators

- String concatenation Operator is used to combine values to create a string. The string concatenation Operator is represented by a period (.) and can be used to build a string from other strings, variables containing non-strings (such as numbers) and even constants.

# Using the String concatenation Operator

```
<?php  
echo 'we are'. 'TE IT students';  
?>
```

the String concatenation Operator is used to concatenate two strings. The second string is appended to the end of the first string, which gives the output.we are TE IT students

concat - Notepad

File Edit Format View Help

```
<html>
<head><title>string Concatenation</title></head>
<body>
<?php
echo 'we are'. 'TE IT students';
?>
</body>
</html>
```

string Concatenation

localhost:9090/concat.php

we areTE IT  
students

# Using the String concatenation Operator with Variables

```
<?php  
$myclass='TE IT';  
echo 'my favorite class is'. $myclass;  
?>
```

You can also use the String concatenation Operator to concatenate the values of 2 Variables. The value of the `$myclass` variable is appended at the end of the string, my favourite class is, by using the concatenation Operator . The echo statement now prints the string, my favourite class is TE IT



conCatVar - Notepad

File Edit Format View Help

```
<html>
<head><title>string Concatenation with variable</title></head>
<body>
<?php
$myclass='TE IT';
echo 'my favorite class is'. $myclass;
?>
</body>
</html>
```

string Concatenation with variabl x



localhost:9090/conCatVar.php



my favorite class  
isTE IT

# Comparison Operators

- The Comparison Operators are used to compare one value with another and return either a true or false depending on the status of the match. For example, you can use a comparison operator to check if a variable value matches a particular number or whether one string is identical to another or not. Comparison Operators are used with two operands, one to the left and one to the right of the operator.

# List of Comparison Operators

Sr.No	Operator	Type	Description
1	==	Equal to	Returns true if first operand is equal to second
2	!=	Not Equal to	Returns true if first operand is not equal to second
3	<>	Not Equal to	Returns true if first operand is not equal to second
4	===	Identical to	Returns true if first operand is equal to second in both value and type
5	!==	Not Identical to	Returns true if first operand is not equal to second in both value and type
6	<	Less than	Returns true if first operand is less than the second operand
7	>	Greater than	Returns true if first operand is greater than the second operand
8	<=	Less than or Equal to	Returns true if first operand is less than or equal to the second operand

# Logical Operators

- Logical Operators are also known as Boolean Operators because they evaluate parts of an expression and return

Sr.No	Operator	Type	Description
1	and	AND	Performs a logical AND operation
2	or	OR	Performs a logical OR operation
3	xor	XOR	Performs a logical XOR operation
4	&&	AND	Performs a logical AND operation
5		OR	Performs a logical OR operation

# Increment/Decrement Operators

Sr.no	Example	Name	Effect
1	\$x++	Post Increment	Returns the value of \$x and then increments its value by one
2	++\$x	Pre Increment	increments the value of \$x by one and then returns the value of \$x
3	\$x--	Post Decrement	Returns the value of \$x and then decrements its value by one
4	--\$x	Pre Decrement	Decrement the value of \$x by one and then returns the value of \$x

# Demonstration of Increment/Decrement Operators

```
<?php
$x=19;    //define variable
$y=$x++;  //post increment
Echo $y;  //output:19
$y=++$x;  //pre increment
Echo $y;  //output:21
$y=$x--;  //post decrement
Echo $y;  //output:21
$y=- -$x; //pre decrement
Echo $y;  //output:19
?>
```

incDec - Notepad

File Edit Format View Help

```
<html>
<head><title>increment and decrement operator</title></head>
<body>
<?php
$x=19;           //define variable
$y=$x++;         //post increment
echo $y;         //output:19
$y=++$x;         //pre increment
echo $y;         //output:21
$y=$x--;         //post decrement
echo $y;         //output:21
$y=--$x;         //pre decrement
echo $y;         //output:19
?>
</body>
</html>
```

increment and decrement operat x +

localhost:9090/incDec.php

19212119

- You can use Increment/Decrement Operators with variables. The \$x variable is assigned the value 19, and then \$x is assigned to the \$y variable. Therefore, now \$y has value 19 and then the value of \$x is incremented by 1. Now, the value of \$x is 20. The echo statement prints the value of \$y as 19. This is known as post increment. In pre increment, the value of \$x is first incremented by 1 and then its incremented value 21 is assigned to \$y. Therefore, now both \$x and \$y contains value, 21. The echo statement now prints the value of \$y as 21. Similarly, the post and pre decrement operation are performed in the next statements.



# Arithmetic Assignment Operators

- Arithmetic Assignment Operators are a combination of arithmetic and assignment operators. They first perform the basic arithmetic operations on variables and then assign the resultant value to a variable itself.

Sr.No	Operator	Description
1	+ =	Adds the value and assigns it to a variable
2	- =	Subtracts the value and assigns it to a variable
3	* =	Multiplies the value and assigns it to a variable
4	/ =	Divides the value and assigns it to a variable
5	% =	Divides the value and assigns the modulus to a variable
6	. =	Concatenates and assigns the value (for strings only) to a variable

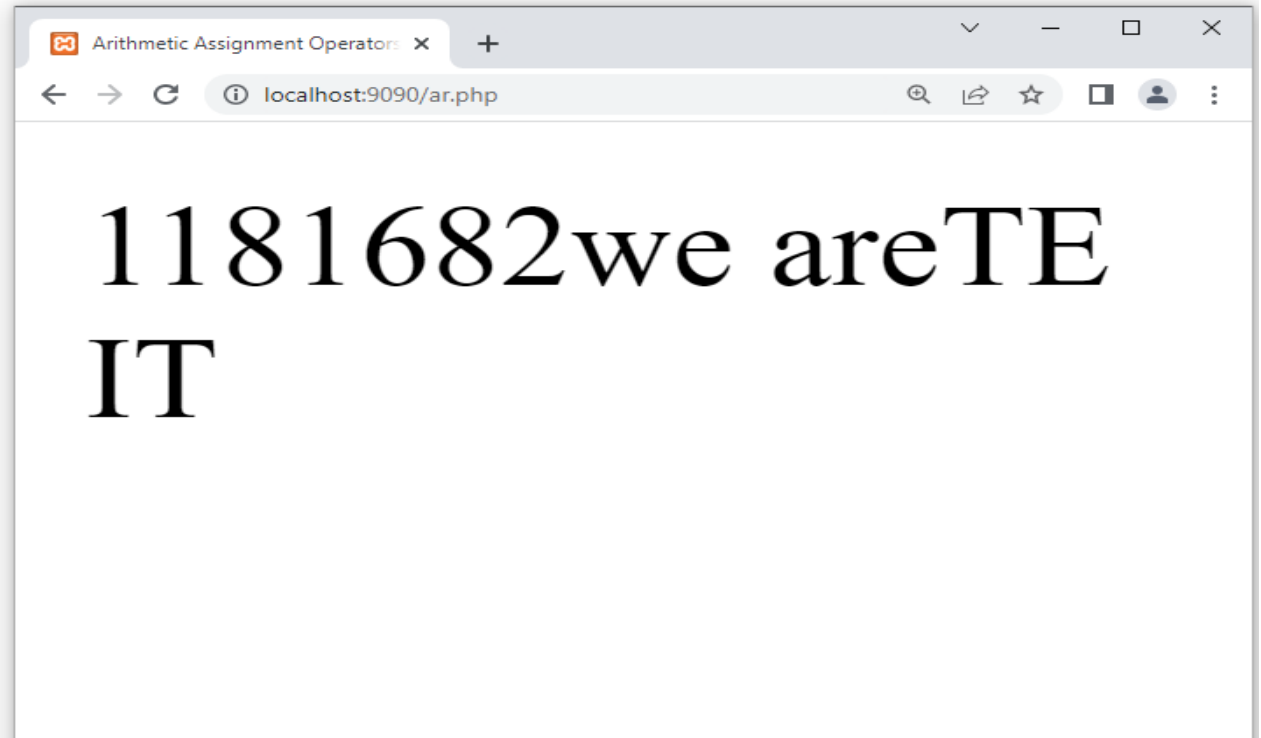
# Demonstration of Arithmetic Assignment Operators

```
<?php
4x=9;
$x+=2; //similar to $x=$x+2
Echo $x; //displays 11
$x-=3; //similar to $x=$x-2
Echo $x; //displays 8
$x*=2; //similar to $x=$x*2
Echo $x; //displays 16
$x/=2; //similar to $x=$x/2
Echo $x; //displays 8
$x%=3; //similar to $x=$x%3
Echo $x; //displays 2
$x='we are';
$x.='TE IT';
ECHO $X; //Displays 'we are TE IT'
?>
```

ar - Notepad

File Edit Format View Help

```
<html>
<head><title>Arithmetic Assignment Operators</title></head>
<body>
<?php
$x=9;
$x+=2;           //similar to $x=$x+2
echo $x;         //displays 11
$x-=3;           //similar to $x=$x-2
echo $x;         //displays 8
$x*=2;           //similar to $x=$x*2
echo $x;         //displays 16
$x/=2;           //similar to $x=$x/2
echo $x;         //displays 8
$x%=3;           //similar to $x=$x%3
echo $x;         //displays 2
$x='we are';
$x.='TE IT';
echo $x;         //Displays 'we are TE IT'
?>
</body>
</html>
```



- The \$x variable is assigned a value 9, and in the next statement, its value is incremented by 2 and assigned to itself. Therefore, now its value is 11, which is printed using the echo statement. Similarly , other assignment operators are used in next statements.

# Operator Precedence

- Operator Precedence refers to the strength with which the two operators are bound together. For example, in the expression  $1 + 5 * 3$ , the answer is 16 and not 18 because the multiplication ( $*$ ) operator has a higher precedence than the addition ( $+$ ) operator.

```
<?php
```

```
echo 5*2+1;    //displays 11
```

```
echo 5+2*1;    //displays 7
```

```
echo 5*(2+1);  //displays 15
```

```
?>
```

pre - Notepad

File Edit Format View Help

```
<html>
<head><title>Operator Precedence</title></head>
<body>
<?php
echo 5*2+1;           //displays 11
echo 5+2*1;           //displays 7
echo 5*(2+1);         //displays 15
?>
</body>
</html>
```



Operator Precedence



localhost:9090/pre.php

11715

- The first echo statement prints 11 while the next echo statement prints 7. This is because the precedence of the  $*$  operator is higher than the  $+$  operator; therefore, in the first statement the expression is evaluated as  $(5*2)+1$  and in the second case it is evaluated as  $5+(2*1)$ . Parenthesis is used to force precedence which is seen in the third echo statement where  $(2+1)$  is evaluated first and its result is then multiplied with 5 and the output is 15. If the operators have same precedence then left to right associativity decides the order of precedence.

# Demonstration of Operator precedence

```
<?php
```

```
Echo 5*2+1;    //displays 11
```

```
Echo 5+2*1;    //displays 7
```

```
Echo 5*(2+1);  //displays 15
```

```
?>
```



# Demonstration of left to right associativity

```
<?php
```

```
Echo 3*3/2;    //evaluates (3*3)/2,displays 4.5
```

```
Echo 3/3*2;    //evaluates (3/3)*2,displays 2
```

```
?>
```

- Both (\*) and (/) operator has the same precedence hence the expression are evaluated on the basis of left to right precedence.

aso - Notepad

File Edit Format View Help

```
<html>
<head><title>left to right associativity</title></head>
<body>
<?php
echo 3*3/2;      //evaluates (3*3)/2,displays 4.5
echo 3/3*2;      //evaluates (3/3)*2,displays 2
?>
</body>
</html>
```

4.52

# Operator Precedence

Sr.No	Associativity	Operators	Additional Information
1	Right	!	Logical
2	Left	* / %	Arithmetic
3	Left	+ - .	Arithmetic and string
4	Left	&	Bitwise and references
5	Left		Bitwise
6	Left	&&	Logical
7	Left		Logical
8	Right	= += -= *= /= .= %= &=   = ^= <<= >>=	Assignment
9	Left	And	Logical
10	Left	Xor	Logical
11	Left	Or	Logical
12	Left	,	Many uses

# Assignment 3

- Q2 ) Demonstrate type casting in PHP with a suitable program. Display the output(5 marks)
- *Assignment Announced to students : AA :01<sup>st</sup> June 2022*
- *Assignment to be Submitted by students : AS: 06<sup>th</sup> June 2022*