Module - 3

PHP

→ It is a server scripting language & powerful tool for making dynamic.
→ It starts with <? php Ends with ?>
→ default file extension → .php
→ PHP code is executed on server side & the result is returned to the browser as plain HTML.
→ It is compatible with all servers used today & supports wide range of database
→ PHP is easy to learn runs efficiently on the server side.

PHP Syntax

→ It can be placed anywhere in doc.
→ ip

Eg: <!DOCTYPE html>
<html>
<body>
<? php
echo "My first PHP
?>
</body>
</html>

PHP variable

variable starts $ sign.

```
<?php
$txt="Helloworld"
?>
```

When assigning a text value to variable, put quotes

Rules for Php

- A variable starts with $ sign followed by the name of the vari
- A variable name must starts with lette or an underscore.
- A variable name cannot start with no
- A variable name can have only (A-Z, 0-9,
- variable name are case sensitive ($age and $AGE are two diff variables)

PHP data types

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

Ways to convert data types in PHP.

- Converting data types using Type casting.

Type casting can be achieved by placing the intended types of variable to cast

(array) → array (string) → string : single byte.

(bool) or (boolean) → boolean : TRUE / FALSE

(int) or (integer) → integer . 32 bit

Eg: $number = (float) 16; converts the integer 16 to float 16.0.

(object) → object

(real) or (double) or (float) → float

- Converting data types using type juggling

Type juggling is feature of PHP where variable automatically cast the best fit data type in circumstances while manipulating math op.

Type juggling converts automatically to the upper level data type that supports for calculation

Ex: converts string into integer while adding to the integer

```
<?php
$sum = 15;
$increase = "10";
$sum+ = $increase
$total = 20;
```

PHP operators

PHP operator is a symbol ie used to perform operation on operand.

```
$num = 10 + 20;
```

PHP operators:

- Arithematic
- Assignment
- Bitwise
- Comparison
- Incre/Dec
- logical
- String
- Array
- Type
- Execution
- Error Control

Unary operators: works on single operands e++

Binary " : " " two operands +

Teenary " works on three " : "?:

Flow Control Statement

Control Conditional statements such as if/else switch, allow program to execute diff pieces of code.

Loops such as while & for support the repeated ex

If

The if statement checks the truth of exp: If exp is true evaluates a statment:

```
if (exp)
    statement
```

To specify alternative statement

```
if (exp)
    statement
else stmt
```

Switch: stmt to select one of many blocks of cod to be executed

PHP Loops

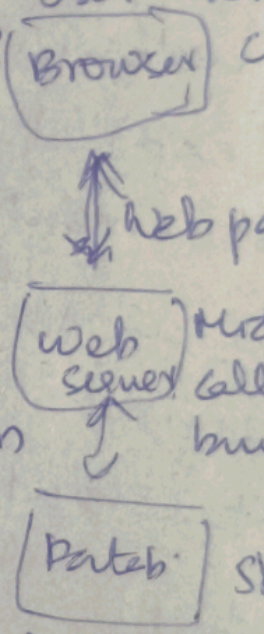
while : loops through a blk of code as long as specified
do while " then repeats the loop as
for : specified no. of times

for each : loops through a block of code for each element

3 tier web based Appli

Top tier also calls User interface

- The bottom tier maintains the applicatn data. this tier stores RDBMS
- the middle tier implements business logic & presentatn logic to control interactn bhwn applicatn clients & its data. It acts an intermediately bhwn data in info tier & applicatn client



Client side Scripting

- with JS can be used to validate user i/p, to interact with browser, to enhance web page.
- It have limitatn, such as browser dependency, the browser / scripting ost must support scripting lang. gt can be viewed by client

Server side Scripting

- It generate custom respons for client
- wider range of programs capabilities than clients
- gt have access to serve side software extend ser functionality.

Super Global Variables : are built in variables that are always available in all scope.
→ Also Always accessible
• $GLOBALS • $_SERVER • $_REQUEST $_POST
• $_GET • $_FILES • $_ENV • $_COOKIE • $_SESSION

ceil(): To round a number UP to nearest integer use of ceil() function
echo (ceil(0.70)); //→ 1

floor(): rounds a no. DOWN to the nearest int
echo (floor(0.8)) → 0

rand(): generates random no.
echo (rand())

Constant: it is a name / variable identifier
it cannot be changed, it is case sensitive
• constant identifiers are always uppercase.
• constant name starts with letter/underscore

Constant	variable
- No need of $	- Need as starting symb
- cannot be changed	- Can be changed
- may be need to define() or other exp → can be defined/accessed anywhere	

Echo

Return no value
Multiple parameters.
faster than print

```
<?php
print Welcome<br><br>
echo (floor(0.01))=0.
?>
```

print

Return 1 value
or can take one argument
slower than

```
<?php
print <b> welcome </b>
?>
```

Palindrome

```
<html> <body>
<?php
$num = 1234567
$x = 0
$n = $num;
while (floor($num)){
$mod = $num %10
$x = $x * 10 + $mod
$num = $num /10;
}
if ($n == $x) {
echo "$n is a Palindrome
}
else
echo "$n is not Palindrome.
} ?> </body> </html.
```

Arrays

→ Storing data in arrays: Arrays are divided into elements.

→ ([]) this is the array format.

Three kind of arrays:

Numeric/indexed - An Array with numeric index. Starting from 0

Associative - An array where each ID key is associated with value.

Multidimensions: An array containing one/more arrays.

Creating numeric Arrays in PHP.

A numeric array stores each array element with a numeric index.

- 2 methods to create numeric array:

1) Assignment operation creates scalar vari assigning value to an element of an array

• If a variable named $list does not exist, the stmt $list[0] = 17 to create array named $list & assign 17 to element at index 0

$list = array (17, ...)

If a scalar variable named $list already exists an assignment $list[0] = 17 will convert scalar to array.

- In $list[] = 42, without specifying numeric key, it uses next available num index.

2) To create array in array constructor index is automatically assigned (index

$cars = array ("Tata", "volvo", "BMW")

$cars[0] = "Tata"

$cars[1] = "volvo"

$cars[2] = "BMW"

Creating Associative Array in PHP.

- Each ID key is associated with value.

→ When storing data about specific named values, a numeric array is not always be

→ With Asso... Array we can use values as keys & assign values to them

$ages=array("Peter"=>32,"Joe"=>34);

```
<?php
$ages['Peter'] = "32";
$ages['Joe'] = "34";
echo "Peter is ". $ages['Peter']. "years old";
?>
```

o/p

Peter is 32 years old.

functions dealing with array

Unset() (1)

A whole array can be deleted with unset() as with scalar variable.

- Individual elements of array can also rem

```
$list=array(2,4,6,8)
unset($list[2])
```

the list have 2,4,8

(2)

- array key funct: takes an array as its paramet & returns an array of keys of the given array. The array values function does for values what array_keys does for keys.

sizeof() ①

The no: of elements in an array can be determined with size of functn.

Eg: $list = array("Bob", "fred")

$len = sizeof($list)

After executing, $len will be 4.

count() ④ → count no: of element off in array

$colors = array("blue", "blk");

count($colors) & sizeof($colors) will be 4

→ Array explode & implode.

Explode: explodes a string into substring & returns them in an array.

→ The delimiters of the substring are defined by first parameter to explode., which is a string, the second parater is string to be connected,

$str = "May god bless you all";

$words = explode(" ", $str).

$words creates an array which contains ("May", "god", "bless", "you", "all")

The implode function does the inverse of explode.

It concatenates the elements of array together

```
$words = array("Are", "you", "coming");
$str = implode(" ", $words)
```

$str has "Are you coming".

Array-push & Array-pop functions to implement a stack in an array.

Array-push functn takes as its first parameter an array. After this first parameter there can be any additional parameters.

- The array-push function returns the new number of elements in the array

```
array-push($existing Array, "element1",
"element 2");
```

Array-pop functn takes single parameter, the name of an array. It removes the last element from the array & returns it. The value NULL is returned if the array is empty

array_unshift(): The funct^n adds one/more elements to beginning of existing array.

$ array_unshift ($existingArray, "elem

To process Array use foreach stat.

The foreach stmt is designed to built loops that process all of the elements of an array. This stmt has 2 forms.

foreach (array as scalar_variable) loop body

for each (array as key => value) loop b.

eg: $list = array (2, 4, 6, 8)

foreach ($list as $temp)

print ("$temp </br>")

This code will produce the values of all element

The second form of foreach provides both key & the value of each element of the array.

Syntax: foreach (array as key => valu

Sorting Arrays.

sort(): - sort arrays in ascending order.

rsort(): " " in descending ord

asort(): Sort associative array in ascending order, according to ~~key~~ value.

ksort(): Sort associative ar " , acc to key.

sort();

```
<?php
$fruit = array("orange", "Grapes", "Apple");
sort($fruits)
foreach ($fruit as $x)
{
  echo $x;
  echo ("<br>");
}
```

rsort

rsort($fruit)

→

←

←

orange
Grape
Apple.

Op
Apple.
Grape
Orgame

asort():

```
<?php
$age = array ("Peter" => "45", "Ben" = "47")
asort ($age).
foreach ($age as $x => $x-value) {.
echo "key". $x. ", value". $x-value;
echo "</br>";
```

o/p

key = ~~Joe~~ Peter value = 25

key - Ben value = 47

→ Array using array_search()

we can remove specific element from an array

Array_using_array_diff()

remove specific element from array

Functions

- A function is a ~~blk~~ block of statement that can be used repeatedly in a program.
- It does not execute automatically when a page loads.

```
<? php
function wantMsg(){
echo "Hello World";
}
wantMsg();
?>
```

o/p:

- The user defined funct^{n} declarat^{n} starts with the word function.
- PHP does not support overloading.
- A function is a self contained block of code that can be called by your script.

general form of PHP funct^{n}.

function name ([parameters]) { }.

→ The no: of actual parameters in a call to a function does not need to match the no: of formal parameters

Actual parameters → paramt^{rs} in the call to fun
formal " → " that are listed in funct^{n} def
→ function comes into 2 flavors; those built in t
the lang and those you define yourself.
→ PHP has hundreds of built in functions.

$text = strtoupper ("Hello World");

returns a string value, so it requires a pre
of variable to accept new string

$new_string = strtoupper ("Hello world");
echo $new string;

O/p HELLO WORLD

Scope refers to a region of a problem where a variable / function is accessible.
→ If a local variable defined within functn is same as functn, there is no interference btwn two due to concept of scope.

eg def my_functn():
    print("Display my_function")
def another_functions():
    my_function = ("u u")
    print(my_function).

In this the function & local variable have same name & it does not interfere.

- PHP - Return Declaratn

→ To declare a type for functn to return, add colon (:) and type right before opening curly ({}) bracket when declaring the function

<?php 'declare (strict_types=1)
function add (float $a, float $b):
float { return $a + $b; }
echo "Sum of 1.2 + 5.2 is " add (1.2, 5.2);
?>

function → pass by value.
→ Default parameter passing

• The values of actual parameter (100, 87) are copied into separate memory locatn for corresponding formal parameter ($first, $second).

o/p: 100

```
<?php
declare(strict_types=1);
function max_abs($first, $second)
int {
  $first = abs($first);
  $second = abs($second);
if ($first >= $second)
  return $first;
else
  return $second;
}
echo 'Max 100, 87 is.
max(100, 87)
?>
```

function : Pass by reference

It can be done in 2 ways:
→ One way is to add an & to beginning of the name of the formal parameter that you want to be passed
→ Another is to add & to actual parameter in function call

```
<? php
function add_five(& $value)
{ $value += 5; }
$num = 2;
add_five($num);
echo "Num = $num";
?>
```

o/p: 102

objects

- An object is a sort of theoritical box of things variables, function
- The input mechanism are methods, methods have properties
- An object exist as a data structure & definition of the structure called class
  → In class, define set of characteristics

obj: creating object

```
class myclass {
  // code
}
```

create an instance of class.

```
$ object1 = new class ();
```

- The variable declared inside an object are called properties → values, array / even other obj

```
<? php
class myCar {
```

public $color = "blue"
public $make = "jeep";

keyword
→ public, private, protected.

Object Methods:

→ Methods add functionality to object
- '→' op is used to call object method
and access public properties

```
<? php
class myClass {
    public functn sayHello(){
    echo "Hello"
}}
$object1 = new myClass()
$object1→sayHello()
?>
```

Public method called sayHello

Object of my myClass creates & instance of a myclass & assign to var object1.

→ Method call
↳ this to call sayHello() method which belong to obj 1. This execute the code within sayHello()

If declare an public name = "jumbo"
the echo part "Helo! my name is". $this→name

$o/p → Helo! my name is Jumbo

$this is used to refer to currently instantiated object

→ Any time an obj refers to itself, you use $this

Constructor

→ Spcl funct^n inside class that automatically runs when you create a "new" object from the class. To initialize the object's property / set up initial state

__construct().

Object Inheritance -

Class inherits funct^n ly from parents

```
<?php
class myclass{
public$name;
function__construct($n){
$this→name=$n;}
Public function sayHello(){
echo "Hello Myname ".$this→name;
}
}
```

Method overloading

A child can overide methods to parent class

```
class childclass extends myClass {
    public function sayHello()
    {
        echo "<br/> i wont tell
        my name " $this->name;
        echo "<br/> </br">
    }
}
$object2 = new myClass ("Benson")
    -> sayHello()
$obj1 = new childclass ("Baby")
$obj1 -> sayHello1();
?>
```

```
} = "
echo '<br> I wont
tell my name';
echo <br> overriding
<br
}
$obj1 - new child class
    ('Baby');
obj1 -> sayHello();
?>
```

o/p Hello! My name is Benson.
I wont tell ~~my~~ my name
B Baby.

I wont tell my na
overriding.

String Comparison.

strcmp → compare 2 string.
Relational op (==, !=, <=, >=) can also be used to compare.

String Processing with Regular E.

- Text manipulation is done with regular expressions - a series of character that serves as a pattern matching templates in string, text files & database.

→ An RE is sequence of character that forms a search pattern.

→ A pattern is sequence of characters to be searched for a character string.

patterns are enclosed in / characters

/def/ → pattern def

∴ Eg: When search redefine, the pattern match 3rd 4th ...

PHP has 2 diff kinds of string pattern matching used in RE

- One based on POSIX RE & one is Perl Regular Expres.

POSIX regular expression are compiled to PHP.

preg-match (regex, str).

Returns - Boolean value.