PhP

Page No.
Date

1 Introduction

- PhP stand for HyPerText PreProcessor
- PhP is a widely-used open source scripting language.
- PhP scripts are executed on the server.
- PhP files have extension .PhP
- PhP files can contain text, HTML, CSS, JS and PhP code.

2 Use of PhP

- PhP can generate dynamic page content.
- PhP can create, open, read, write, delete, and close files on the server.
- PhP can collect form data.
- PhP can send and receive cookies
- PhP can add, delete, modify data in your database.
- PhP can be used to control user access.
- PhP can encrypt data.

3 Syntax

- PhP script start with <?PhP and ends with ?>
- ex

```
<?php
  //Code goes here
?>
```

- PhP statement end with a semicolon(;).
- php keywords are not case-sensitive

4 Comment

- Comments are used to wit a note which is ignore by the browser.
- Single line comment look like this...

```
// single line comment
```

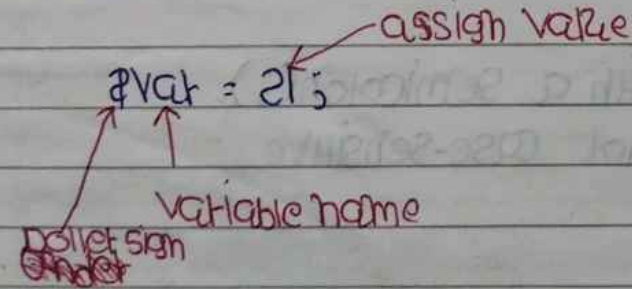
- Multi-line comment look like this....

```
/* Multi-line
comment */
```

5 Variables

1 Rules

- Variable start with the $ sign, followed by the name of the variable.
- Variable name must start with a letter or the underscore character.
- Variable name cannot start with a number.
- Variable name can only contain alpha-numeric characters and underscores (A-z, 0-9 and _)
- Variable names are case-sensitive.
- Syntax



2 data-types

- PhP support following data-type

1 Integer (N (not include dacimal value)
2 Float (include dacimal value)
3 String (sequence of character)
4 Array (multiple value in one value)
5 object
6 Boolean (true and false)
7 NULL (No value assigned)
8 ~~Resource~~

- Note: we can use var_dump(variable name) to find data-type

6 String function

1 Strlen

- This function use for find string length
- ex

```
echo Strlen("Hello world");
```

output
10

2 Str_word_count

- This function use for ~~find~~ count word in string.
- ex

```
echo Str_word_count("Hello world")
```

output
2

3 StrPos

- This function use for find position of word.
- ex

```
echo StrPos("Hello World", "World");
```

Output
7

4 Strtoupper

- This function use for write string in uppercase.
- ex

```
echo Strtoupper("Hello World");
```

Output
HELLO WORLD

5 Strtolower

- This function use for write string in lowercase.
- ex

```
echo Strtolower("Hello World");
```

Output
hello world

6 Str_replace

- This function use for replace word in string.
- ex

```
$x = "Hello World";
echo Str_replace("World", "Dolly", $x);
```

Output
Hello Dolly

7 Concat (.)

- This function use for concatenate string
- ex

```
$x = "Hello";
$y = "World";
$z = $x. $y;
echo $z;
```

Output
Hello World

8 SubStr

- This function use for slicing string.
- ex

```
$a="Hello World!";
echo SubStr($a, 6, 4)
```

Output
Worl

- if we note give character length all character will be print

```
echo SubStr($a, 6)
```

Output
World!

9 Escape character

- using Escape character we can print

\n	new line
\t	tab
\'	single qoute
\"	Double qoute
\$	Dolar sign

7 Operators

1 Arithmetic operators

+	Addition	$a + $b
-	Subtraction	$a - $b
*	Multiplication	$a * $b
/	Division	$a / $b
%	Modulus	$a % $b
**	Double Multipication	$a ** $b

2 Comparison operators

==	Equal to	$x == $y
===	Identical	$x === $y //strict equality
!=	Not equal to	$x != $y
!==	Not identical	$x !== $y
<	Less than	$x <> $y
>	Greater than	$x <> $y
<=	Less than or equal to	$x <= $y
>=	Greater than or equal to	$x >= $y

3 Logical operators

$b = True $a = false

&&	Logical AND	$b && $a
||	logical OR	$b || $a
!	Logical NOT	!$a

4 Assignment operators

$a = 10

+=	Addition	$a += 5;
-=	Subtraction	$a -= 5;
*=	Multiplication	$a *= 5;
/=	Division	$a /= 5;
%=	give remnde	$a %= 5;

8 Conditional Statement

1 if Statement
- if Statement use when we have one condition
- Syntax

```
if(condition){
        //Statement-block;
          }
```

2 if else Statement
- if else Statement use when we have one condition and two result.
- Syntax

```
if(condition){
        //Statement-block;
          }
    else{
        //Statement-block;
          }
```

3 if else ladder Statement
- if else ladder statement use when we have more then two or more condition.
- Syntax

```
if(condition){
        //Statement-block;
          }
  else if(condition){
        //Statement-block;
          }
```

```
else{
    //Statement-block;
    }
```

4 Nested if statement

- Nested if statement used when we need condition inner if statement.
- syntax

```
if (condition) {
    if (condition) {
        //Statement-block;
        }
    else {
        //Statement-block;
        }
    }
else{
    if (condition) {
        //Statement-block;
        }
    else{
        //Statement-block;
        }
    }
```

5 Shorthand if

- This statement use when we want to write if statement in one line.

- exc

1 if ($a < 10) $b = "Hello"; // one line if statement

2 $b = $a < 10 ? "Hello"; "Good Bye"; //one line if else statement

6 Switch statement

- Switch statement use when we want to select & break after output.
- Syntax

```
Switch(expression) {
    case Label1:
        // Code block
        break;
    case Label2:
        // Code block
        break;
    case Label3:
        // Code block;
        break
    default:
    // Code block
            }
```

9 Loops

1 While loop

- The While loop executes a block of code as long as the specified condition is true.

- Syntax

```
while(condition){
        // statement-block;
        }
```

2 do while loop

- The do while loop will always execute the block of the code at least once, it will then check the condition, and the loop while the specified condition is true.

- Syntax

```
do{
  //statement-block;
  }while(condition);
```

3 for loop

- The for loop is used when you know how many times the script should run.

- Syntax

```
for(expression1, expression2, expression3){
                    // statement-block;
                    }
```

4 foreach loop

- The foreach loop is loop through the items of an array.
- Syntax

```
foreach($array as $key => $value) {
        //statement-block;
        }
```

10 Web Server

- Web server is computer that runs website.
- It is a computer program that distribute webpages as they are requesting.
- basic objective of the web server is to store process and deliver webpages to the users.
- This inter communication is done using HTTP
- There are most widely used web servers
- 1. IIS
  2. Apache

1 IIS

- IIS stand for internet information services.
- It is a genral purpose web server that runs on the window operating system not in Linux, Mac os and unix
- TheIISS accept and response to the clint computer request and Enable them to share and diever delivery information across the LAN
- we can simpliely say that both works to same accept that a apache web server can be use almost on any operating system such as window, linux, unix and Ma while the IIS available for window.

- The IIS has the security future which make more secure and efficient option then Apache

2 Apache

- Apache is a software that runs on a server.
- It's job is to establish connection between a server and the browser of website visiter while dilevering file between the clinte and server.
- When user want's to load a page there browser sends a request to your server and Apache will return response with all the requested file like like text, image etc
- The server and the clint communicate threw HTTP Protocall
- You can also setup your on server configration threw a file is called .htaccess, Apache configration file.

* Recurise function

```
function Test() {

    // Statement - block

    Test(); ← call function inner function

    }

Test(); ← call function outer function
```

11 Function

- PHP has over 1000 built-in function that can be called directly, from within a script to perform a specific task
- Syntax

```
function function-name(parameter1, parameter2){
                    //statement-block;
                    }
```

- Parameters work as a local variable in statement-block.
- an other type of function is arrow function.
- Syntax

```
$ function-name = fn(Parameters) => statement-block;
```

- we can set default value

```
function function_name($a=10, $b=30){ echo $a+$b; }
```

* anonymous function

```
$variable = function($para) {
  {
    echo "Hello $para";
  };

  $variable("PHP");
```

12 Date function

- The time function allow you to get the date and time from the server where the PHP script runs.

- Syntax

```
echo date("code");
```

- Parameter

d - The day of the month.
D - representation of a day.
z - The day of year (0-364)
Y - Four digit year
a - Lowercase (am or pm)
A - Uppercase (AM or PM)
g - 12 hour format of an hour
G - 24 hour format of an hour
i - Minutes
s - seconds
m - numeric representation of a month

13 Associative array

- Associative array are arrays that use named key that you assign to them.
- We can use foreach loop to print Associative array.
- ex

```
$arr = array (
        "Name" => "Mohit",
        "course" => "BCA",
        "Roll No" => "21"
        );
```

14 Multi dimensional array

- A multi dimensional array is an array containing one or more arrays.
- ex
```
$multi = array(
            array(1,2,3),
            array(4,5,6),
            array(7,8,9)
         );
```

15 Connecting database

- There are two ways to connect to a MySQL
  1. Mysqli extension
  2. PDO (Php Data operation)

- Connecting to the Database

```
$servername = "localhost";
$username = "root";
$password = "";
```

- If you working on server it will be provide username and password.

```
$connect = mysqli_connect($servername, $username, $password
                          $database);
```

for more code check github playlist

* Create database

```
$database = "CREATE DATABASE Rathod";
```

syntax — database name

```
mysqli_query($conned, $database);
```

16 Global and Local variable

- Some predefined variable in PhP are always accessible, regardless of scope - and you can access them from any function.
- A variable declare outside a function has a Global scope and can only be accessed outside a function
- inside function variable has called Local va scope and can only be accessed outside a inside a function.
- We must use variable global keyword to declare global variable and bydefault variable is local

- exc

```
$a=20;
function variableC){
    global $a;   ← global variable
    $b=30 ← Local variable
        }
```

17 include and require statement

- The include and require statement takes all the text/code/ markup that exists in the specificed file and copies it into the file that uses the include and require statement

- require will produce a fatal error and stop the script
- include will produce a warning and the script will continue.

* Note string function

1 chr()
- Return a one-character string containing the character specified by ascii.
- syntax

```
$asci = chr(65);
 echo $asci;
```

2 ord()
- Return the ASCII value of the character of string.
- syntax

```
$asci = ord('A');
 echo $asci;
```

3 ltrim()
- strip white space from the beginning of a string.
- syntax

```
$string = " Hello world ";
$trim = ltrim($string);
 echo $trim;
```

4 rtrim()
- strip white space from the end of a string.
- syntax

```
$string = " Hello world ";
$trim = rtrim($string)
 echo $trim;
```

5 trim()
- strip whitespace from the beginning & end of a string

- Syntax
```
$string = " Hello World ";
$trim = trim($string);
echo $trim;
```

6 strcmp()
- Binary safe string comparison
- Syntax
```
$str1 = "Hello"; $str2 = "hello";
$result = strcmp($str1, $str2);
echo $result;
```

7 strcasecmp()
- Binary safe case-insensitive string comparison
- Syntax
```
$result = strcasecmp($str1, $str2);
echo $result;
```

8 strstr()
- Find first occurrence of a string
- Syntax:
```
$str = " Hello World ";
$position = strstr($str, "lo");
echo $position;
```

9 stristr()
- case-insensitive strstr()
- Syntax
```
$position = stristr($str, "lo");
echo $position;
```

10 strrev()
- Return reverse a string.
- syntax

```
$str = "hello world";
$rev = strrev($str);
echo $rev;
```

11 strval()
- Get string value of a variable.
- syntax

```
$num = 124;
$val = strval($num);
echo $val;
```

12 explode()
- Split a string by string.
- syntax

```
$exp = "apple,banana,orange";
$fruit = explode(",", $exp);
```

```
foreach($fruit as $fru){
    echo "- $fru \n";
}
```

13 implode()
- Join array element with a string.
- syntax

```
$array = array('Hello', 'world');
$imp = implode(" ", $array);
echo $imp;
```

14 str_split()
- Convert a string to an array
- syntax

```
$str = "Hello";
$spl = str_split($str);
echo $spl;
```

15 str_shuffle()
- Randomly shuffles a string
- syntax

```
$shuf = str_shuffle($str);
echo $shuf;
```

16 strcspn()
- Find length of initial segment not matching mask
- ~~syntax~~

17 strspn()
- Find length of initial segment matching mask.
- ~~syntax~~

18 substr_compare()
- Binary safe optionally case insensitive comparison of 2 string from an offset ut to length characters
- ~~syntax~~

19 ucfirst
- Make a string's first character uppercase
- syntax

```
$uc1 = ucfirst($str);
echo $uc1
```

20 ucwords()

- uppercase the first character of each word in a string.
- syntax

```
$ucw = ucwords($str);
echo $ucw;
```

18 readfile function

- This function use for write content of txt file into webpage
- Syntax

```
$a = readfile("myfile.txt");
echo $a
readfile("file.html");
```

19 ~~fopen(), fread(), fclose()~~ function File handling function

1 fopen()

- fopen used to open a file.
- We can use multiple mode in fopen. Some of them given below.

r: Read only
r+: Read and write
a: Write only
a+: Read and write
w+: Read and write

2 fread()

- fread used to reads from an open file.
- Syntax

```
fread(file, length)
```

3 fclose()

- fclose used to closes an open file.
- Syntax

```
fclose(file pointer);
```

4 fgets()
- Get line from file pointer
- Syntax
  fgets(file pointer);

5 fgetc()
- Gets a character from the given file pointer.
- Syntax
  fgetc(file pointer);

6 fwrite()
- Write content in the file.
- Syntax
  fwrite(file pointer, "content of file");

7 file_exists()
- Checks ~~whether~~ whether a file or directory exists
- Syntax
  file_exists(file name);

8 is_readable()
- Tells whether ~~fi~~ the filename is readable.
- Syntax
  is_readable(file name);

9 is_writeble()
- Tells ~~whether~~ the file name is writable
  Syntax
  is_writable(file name)

10 copy()
- Copies file
- Syntax

```
copy(source, new file);
```

11 unlink()
- Deletes a file
- Syntax

```
unlink(file name);
```

12 rename()
- Renames a file or directory
- Syntax

```
rename(old name, new name);
```

* Make array function

1 count()
- Count elements in an array or properties in an object

2 list()
- Assign variables as if they were an array

3 in_array()
- Checks if a value exists in an array

4 current()
- Return current element in an array

5 next()
- Advance the internal array pointer of an array.

6 prev()
- Rewind the internal array pointer

7 end()
- Set the internal pointer to the last element and returns its value.

8 each()
- Return the current key and value pair from an array and advance the array cursor

9 sort()
- Sort an array

10 rsort()
- Sort an array in reverse order

11 asort()
- Sort an array and maintain index association.

12 arsort()
- Sort an array in reverse order and maintain index association.

13 array_merge()
- Merge one or more array

14 array_reverse()
- Return an array with element in reverse order.

15 array_diff()
- Computes the difference of arrays.

16 array_merge_recursive()
- Merge two or more array recursively.

17 array_shift()
- shift an element off the beginning of array

18 array_slice()
- Extract a slice of the array.

19 array_unique()
- Removes duplicate values from an array.

20 array_unshift()
- Prepend one or more elements to the beginning of an array.

21 array_keys()
- Return all the keys of an array.

22 array_key_exists()
- checks if the given key or index exists in the array

23 array_push()
- Push one or more elements onto the end of array

24 array_pop()
- Pop the element off the end of array.

25 array_multisort()
- Sort multiple or multi-dimensional arrays.

20 Cookie

- Cookie is a small piece of data that a website stores on a user's computer or device.
- Syntax

```
setCookie("category", "Books", time()+86400, "/", "example.com",
        true);
```

1 category : the name of the cookie
2 Book : the value assigned to the cookie.
3 time()+86400 : expiration time of the cookie
4 example.com : The domain for which the cookie is available.
5 "/" : The server where the cookie will be available.
6 true : indicates that the cookie should only be transmitted over a secure HTTPS connection.

21 session

- session refers to the process of maintaining stateful information about a user across multiple page requests with in a single browsing session.

* some property of session

1 session_start()
- used to start a session.

2 unset()
- used to remove specific data stored in the session

3 session_destroy()
- used to destroy all data associated with the current session.

22 Math function

1 abs()
- Absolute value
- syntax

```
abs(value);
```

2 ceil()
- Returns the next highest integer value by rounding up value if necessary.
- syntax

```
ceil(value);
```

3 floor()
- Round fractions down
- syntax

```
floor(value);
```

4 round()
- Returns the rounded value of val to specified precision
- syntax

```
round(value);
```

5 min()
- Find lowest value.
- syntax

```
min(1,2,3,4,5);
```

6 max()
- Find highest value.

- Syntax
  max(1,2,3,5,4);

7 pow()
- Exponential expression.
- Syntax
  pow(2,2);

8 sqrt()
- Return square root.
- Syntax
  sqrt(9);

9 rand()
- Generate a random integer.
- Syntax
  rand(5,15); //generate number between 5 to 15.

10

23 Get and Post method

1 Get method

- Get method used to retrieve information from the user.
- get method appends data to the URL and it visible to all
- it is less secure as data is exposed in the URL.
- We can make get method using following syntax

```
<form action="default.php" method="get"> ..... </form>
```

- We can store data using below syntax

```
$var = $_GET['username'];
```

2 Post method

- Post method to send data to the create/update a resource
- It is more secure then get method because data is not exposed in the URL.
- We can make post method using following syntax

```
<form action="location.php" method="Post"> ..... </form>
```

- We can store data using below syntax

```
$var = $_POST['username'];
```

MYSQL

1 SQL DML Statement

1 insert Statement

~~$insert = `Insert INTO`~~

$insert = "INSERT INTO Table-name(column1, column2, column3)
VALUES('value1', 'value2', 'value3');";

- ex.

$insert = "INSERT INTO `calege`(`Email`, `Password`) VALUES
('Rathod50442@gmail.com', 'rahul@123')";

2 update Statement

$Update = "UPDATE Table-name SET `column` = value
WHERE `column` = "value";

- exc

$Update = "UPDATE `employees` SET `salary` = 5000 WHERE
Name = 'Pratik';

3 Select Statement

$Select = "SELECT * FROM `table-name` WHERE `column` =
'value';

- exc

$Select = "SELECT * FROM `employee` WHERE `department` =
'sales';

4 Delete qs statement

$Delete = "DELETE FROM `table-name` WHERE `column` = value ";

- ex

$Delete = "DELETE FROM `employees` WHERE `department` = 'HR' ";

5 Create database

$create = "CREATE DATABASE `database-name` ";

- ex

$create = "CREATE DATABASE `Rathod` ";

2 Data-type

1 Numeric Type

- INT: integer type can store whole number.
- FLOAT: Floating-point type, used for approximate numeric values with fractional parts
- Double: Double-precision floating-point-type, used for larger floating-point number.
- Decimal: Fixed point type used for exact numeric values with decimal points.

2 Date and Time Type

- Date: Date value in 'YYYY-MM-DD' format
- Time: Time value in 'HH-MM-SS' format
- DateTime: Date and time value in "YYYY-MM-DD-HH-MM-SS" format
- Timestamp: typically stores the current data and time.

3 String Type

- Char: Fixed-length type up to 255 characters.
- varChar: Variable-length string type upto 65,535 characters
- TEXT: Variable-length string for large text data up to 65,535 characters.

3 Mysqli function

1 mysqli_connect: connect to a MYSQL database server. It takes parameters for hostname, username, user password, and optionally port and socket.

- ex.

```
$conn=mysqli_connect("localhost","username","password",
"database");
```

2 mysqli_select_db: Select mysqli database to work with it.

- ex

```
mysqli_select_db($conn,"database-name");
```

3 mysqli_query: Performs query on the database.

- ex

```
$result='INSERT INTO 'Rathod'('email','pass') VALUES
('Rathodsoo442@gmail.com',1234)';
mysqli_query($conn,$result);
```

4 mysqli_affected_rows: Return the number of rows affected by the last insert, update, delete query

- ex

```
$mysqli_affected_rows($conn);
```

5 mysqli_num_rows: Returns the number of rows in a result set.

- ex

```
mysqli_num_rows($conn);
```

6 mysqli_fetch_array: Fetches a row from the result as an array with both numeric and associative keys.

- ex

```
mysqli_fetch_array($result);
```

7 mysqli_fetch_assoc: Fetches a row from the result set as an associative array.

- ex

```
mysqli_fetch_assoc($result);
```

8 mysqli_fetch_object: Fetchs a row from the result set as an object

- ex

```
mysqli_fetch_object($result);
```

9 mysqli_autocommit: Turns on or off auto-commit mode for MYSQL transactions.

- ex

```
mysqli_autocommit($conn, false);
```

10 mysqli_commit: Commits the current transaction

- Ex

```
mysqli_commit($conn);
```

24 AJAX

- AJAX stand for Asynchronous Javascript and XML.
- AJAX is a web development technique which helps in increasing the speed of webpage

* Element of AJAX

1 Javascript
2 DOM
3 XML
4 CSS

* Component of AJAX

1 XMLHttpRequest()

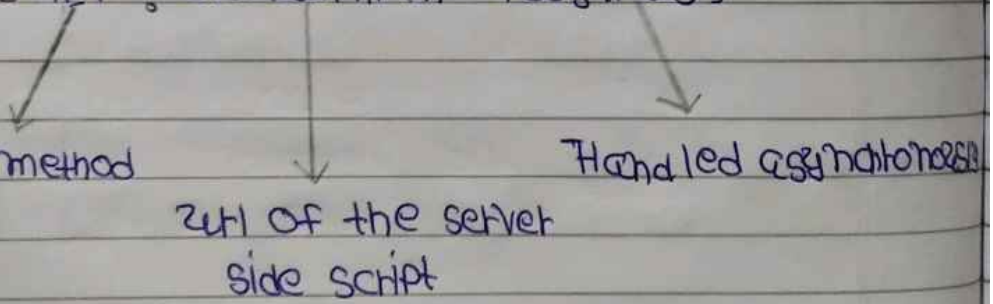
- this object use for asynchronously exchanging XML data between the client and server.
- Syntax

```
let xht = new XMLHttpRequest();
```

2 open()

- using open() method we open a connecting with the server.
- Syntax

```
xht.open("GET", "display.php?n=" + tab, true);
```



2 send()

- using send method we send the request to the server
- syntax

```
Xhr.send("form");
```

↑ parameter

- if we open with post method we can send parameters

* XMLHttpRequest object properties.

1 onreadystatechange

- The onreadystatechange property stores the function that will process the response from a server
- we can also use .onload() function.
- syntax

```
Xhr.onreadystatechange = function(){
    //statement - block
}
```

2 readyState

- readystate is the status of our server's response is stored.

- possible status.

1 0 - Request has been created but not initialized.

2 1 - Represents a "setup" state.

2 - Represents a "send" state.

3 - Represents a "receiving" state.

4 - Represents a "loaded" state.

- Syntax

```
if(this.status) xhr.onreadystatechange = function() {
        console.log('ready state', xhr.readyState);
}
```

3 Status

- The status Property represents the HTTP status code.

- Status code
  - 300 - Successful Redirection
  - 200 - Success
  - 40x - Server ERROR
  - 50x - Internal server Error.

- Syntax

```
xhr.onreadystatechange = function() {
    if (xhr.status = 200) {
        //statement block;
    }
```

4 responseText:

- The responseText property contains the text of the HTTP response received by th client.
- Syntax

```
xhr.responseText
Console.log(xhr.responseText);
```

5 responseXML

- The responseXML property represents the XML response when the complete HTTP response has been received.
- Syntax

```
Consele.log(xhr.responseXML);
```