PHP Notes — (Simple & Practical)

Intro:
PHP ek server-side programming language hai jise mostly web development ke liye use kiya jata hai. Ye HTML ke saath easily milkar dynamic webpages banata hai — matlab aise pages jo user ke input ya database ke hisaab se badalte hain.

---

1) Setup / Shuruaat

- Aapko chahiye: **XAMPP** ya **WAMP** (Windows) / **MAMP** (Mac) / Linux par LAMP stack.
- Install karne ke baad htdocs (XAMPP) folder me apni project folder banao.
- File ka extension .php hona chahiye.
- Browser me URL: http://localhost/yourfolder/file.php

Example: hello.php

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

---

2) PHP Syntax Basics

- PHP code <?php ... ?> tags ke andar likhte hain.
- Statements semicolon ; se end karte hain.

Example:

```
<?php
  // single line comment
  echo "Line 1"; // prints Line 1
  /* multi-line comment
    ye comment 2 lines tak ja sakta hai */
?>
```

---

3) Variables (Variables kya hote hain)

- Variable hamesha $ se start hota hai: $name, $age.
- Type automatic hota hai (dynamic typing).

Example:

```
<?php
  $name = "Aman";
  $age = 20;
  echo "$name ki age $age hai."; // Aman ki age 20 hai.
?>
```

---

4) Data Types (Common)

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

Example:

```
<?php
  $s = "Hello"; // string
  $i = 10; // integer
  $f = 3.14; // float
  $b = true; // boolean
  $n = null; // null
?>
```

---

5) Operators

- Arithmetic: + - * / %
- Assignment: =, +=, -=
- Comparison: ==, ===, !=, !==, >, <
- Logical: &&, ||, !

Example:

```
<?php
 $a = 5;
 $b = 2;
 echo $a + $b; // 7
 var_dump($a == '5'); // true (value same)
 var_dump($a === '5'); // false (type different)
?>
```

---

6) Control Structures

if / else

```
<?php
 $marks = 70;
 if($marks >= 75) {
  echo "A+";
 } elseif($marks >= 60) {
  echo "B";
 } else {
  echo "C";
 }
?>
```

switch

```
<?php
 $day = 3;
 switch($day) {
  case 1: echo "Monday"; break;
  case 2: echo "Tuesday"; break;
  default: echo "Other day";
 }
?>
```

loops (for, while, foreach)

```
<?php
 for($i=1; $i<=5; $i++) echo $i; // 1 2 3 4 5

 $i = 1;
 while($i<=3) { echo $i; $i++; }

 $arr = ["apple","banana"];
 foreach($arr as $fruit) echo $fruit; // apple banana
?>
```

---

7) Functions (Apni functions banana)

- function keyword se.

Example:

```
<?php
 function add($x, $y) {
  return $x + $y;
 }
 echo add(3,4); // 7
?>
```

- Default parameters, variable-length args (...$nums) bhi possible hain.

---

8) Arrays

PHP Notes — (Simple & Practical)

Indexed array

```
<?php
 $a = [1,2,3];
 echo $a[0]; // 1
?>
```

Associative array (key => value)

```
<?php
 $person = ["name"=>"Rita","age"=>22];
 echo $person['name']; // Rita
?>
```

Useful functions

- count(), array_push(), array_pop(), foreach loop

---

9) Strings

- Concatenate: . operator.
- Useful functions: strlen(), strpos(), substr(), str_replace().

Example:

```
<?php
 $g = "Hello";
 $h = $g . " World"; // Hello World
 echo strlen($h); // 11
?>
```

---

10) Superglobals (Bohot important)

- $_GET, $_POST, $_REQUEST, $_SESSION, $_COOKIE, $_SERVER, $_FILES.

Form Example (GET):

- form.html

```
<form action="process.php" method="get">
  <input name="name" />
  <button>Send</button>
</form>
```

- process.php

```
<?php
  echo "Hello " . htmlspecialchars($_GET['name']);
?>
```

Note: htmlspecialchars() se XSS attack kam hota hai.

---

11) Form Handling (POST)

- Method POST bhejna secure mana jata hai (sensitive data ke liye).

Example:

```
<form action="login.php" method="post">
  <input name="email" />
  <input type="password" name="password" />
  <button>Login</button>
</form>
```

login.php

```
<?php
  $email = $_POST['email'];
  $pass = $_POST['password'];
  // sanitize/validate zaroor karein
?>
```

---

12) Sessions & Cookies

Sessions

- Server-side storage, user ko identify karne ke liye.
- Start with session_start().

Example:

```
<?php
  session_start();
  $_SESSION['user'] = 'Aman';
  echo $_SESSION['user'];
?>
```

Cookies

- Client-side. setcookie() se.

```
<?php
  setcookie('theme','dark', time()+3600); // 1 hour
?>
```

---

13) File Handling

- fopen, fwrite, fread, fclose, file_get_contents, file_put_contents.

Example:

```
<?php
  $text = "Hello file\n";
  file_put_contents('notes.txt', $text, FILE_APPEND);
  echo file_get_contents('notes.txt');
?>
```

---

14) Working with MySQL (mysqli & PDO)

- **mysqli** (procedural / OOP) aur **PDO** (recommended — prepared statements + multiple DB support).

Simple mysqli example:

```
<?php
 $con = new mysqli('localhost','root','','testdb');
 $res = $con->query("SELECT * FROM users");
 while($row = $res->fetch_assoc()){
  echo $row['name'];
 }
?>
```

PDO + Prepared statement example (safe):

```
<?php
 $pdo = new PDO('mysql:host=localhost;dbname=testdb','root','');
 $stmt = $pdo->prepare('SELECT * FROM users WHERE email = ?');
 $stmt->execute([$_POST['email']]);
 $user = $stmt->fetch();
?>
```

Never directly insert $_POST data into SQL — use prepared statements to prevent SQL injection.

---

15) OOP in PHP (Basic)

- Class, object, constructor, visibility (public, private, protected).

Example:

```
<?php
 class Car {
  public $color;
  function __construct($c) { $this->color = $c; }
  function honk() { return "Beep!"; }
 }
 $c = new Car('red');
 echo $c->color; // red
?>
```

- Inheritance, interfaces, traits bhi available.

---

16) Error Handling

- error_reporting() aur try { } catch (Exception $e) {}

Example:

```
<?php
 try {
  throw new Exception('Kuch galat hua');
 } catch(Exception $e) {
  echo $e->getMessage();
 }
?>
```

---

17) Security Best Practices (Must read)

1. **Use prepared statements** for DB queries (PDO/mysqli prepare).
2. **Validate & sanitize** user input. filter_var() bahut helpful.
3. **Escape output** with htmlspecialchars() to prevent XSS.
4. **Password hashing**: password_hash() aur password_verify() use karein (never store plain passwords).
5. **Protect file uploads**: check MIME type, file extension, save outside webroot or give random name.
6. **Use HTTPS** on production.

PHP Notes — (Simple & Practical)

Password Example:

```
<?php
  $hash = password_hash('secret123', PASSWORD_DEFAULT);
  if(password_verify('secret123', $hash)) echo 'OK';
?>
```

---

18) File Upload Example (Basic safe-ish)

```
<form action="upload.php" method="post" enctype="multipart/form-data">
  <input type="file" name="file" />
  <button>Upload</button>
</form>
```

upload.php

```
<?php
  if(isset($_FILES['file'])){
    $f = $_FILES['file'];
    // validate type & size
    $ext = pathinfo($f['name'], PATHINFO_EXTENSION);
    $allowed = ['jpg','png','pdf'];
    if(in_array(strtolower($ext), $allowed) && $f['size'] < 2*1024*1024){
      $new = uniqid().'.'.$ext;
      move_uploaded_file($f['tmp_name'], __DIR__.'/uploads/'.$new);
      echo 'Uploaded';
    } else {
      echo 'Invalid file';
    }
  }
```

```
?>
```

---

19) Composer (Dependency Manager)

- composer se packages install karte hain.
- Example: composer require phpmailer/phpmailer
- Autoload karne ke liye: require 'vendor/autoload.php';

---

20) REST APIs in PHP (Simple idea)

- Receive JSON: file_get_contents('php://input') aur json_decode().
- Set headers: header('Content-Type: application/json');

Example:

```
<?php
  header('Content-Type: application/json');
  $data = json_decode(file_get_contents('php://input'), true);
  echo json_encode(['ok'=>true, 'you_sent'=>$data]);
?>
```

---

21) Popular Frameworks (Short note)

- Laravel (sabse zyada popular), Symfony, CodeIgniter, CakePHP.
- Frameworks structure aur security features dete hain — bade projects ke liye use karo.

---

22) Deployment (Basic)

- Local se production server par push karna: FTP / SFTP / Git + CI.
- Make sure display_errors off in production. Use proper DB credentials and environment variables (.env).

PHP Notes — (Simple & Practical)

23) Useful Built-in Functions (Quick list)

- isset(), empty(), trim(), explode(), implode(), array_map(), in_array(), headers_sent().

---

24) Cheatsheet — Common Tasks

- Redirect: header('Location: page.php'); exit;
- JSON response: echo json_encode($arr);
- Start session: session_start();

---

25) Small Project Ideas to Practice

1. Simple login/register with MySQL (use password_hash).
2. CRUD app (notes / tasks) using PDO.
3. File upload gallery with thumbnails.
4. Small REST API for tasks (use JSON).

---

26) Last Tips (Teacher ki awaaz)

- Har cheez step-by-step practice karo.
- Errors se na daro — error messages padho aur samjho.
- Security pe dhyan do — aaj ke projects me bahut important hai.
- Documentation padhne ki aadat daalo: https://www.php.net (agar zaroorat lage).
