



White-
Competence

PHP Programming from Scratch

A Complete Beginner's Training Guide

Table of Contents

1. Introduction to PHP
2. Setting up the Environment
3. PHP Syntax and Variables
4. Data Types & Operators
5. Control Structures
6. Functions in PHP
7. Arrays & Superglobals
8. Working with Forms
9. Sessions and Cookies
10. File Handling
11. Working with MySQL
12. Error Handling & Debugging
13. Security Essentials
14. Object-Oriented PHP
15. PHP and Web Development
16. Building a Simple Project
17. FAQ – Frequently Asked Questions
18. Appendices

Introduction to PHP

- PHP stands for Hypertext Preprocessor.
- It is a widely-used open source server-side scripting language.
- PHP scripts are executed on the server and return HTML to the client.

Setting up the Environment

- Install XAMPP/WAMP/MAMP for local development.
- Check PHP version using: `php -v` in terminal.
- Use built-in PHP server: `php -S localhost:8000`

PHP Syntax and Variables

- PHP code is written inside `<?php ... ?>` tags.
- Variables start with \$ sign, e.g., `$name = 'John';`
- Statements end with a semicolon (`;`).

Data Types & Operators

- PHP supports strings, integers, floats, booleans, arrays, objects.
- Operators: arithmetic (+, -, *), comparison (==, !=), logical (&&, ||).

Control Structures

- If-else and switch statements control logic.
- Loops: for, while, do-while, foreach.

Functions in PHP

- Functions group reusable code.
- Syntax: `function greet($name) { return 'Hello ' . $name; }`
- Built-in functions: `strlen()`, `count()`, `date()`.

Arrays & Superglobals

- Indexed and associative arrays.
- Superglobals: `$_GET`, `$_POST`, `$_SESSION`, `$_COOKIE`, `$_SERVER`.

Working with Forms

- Form data is submitted via GET or POST method.
- Access using `$_GET['field']` or `$_POST['field']`.
- Validate and sanitize input.

Sessions and Cookies

- Sessions store user info on the server side.
- Cookies store small data on client side.
- Use `session_start()` to begin sessions.

File Handling

- Open files with `fopen()`, read with `fread()`, write with `fwrite()`.
- Upload files via HTML forms and `$_FILES` superglobal.

Working with MySQL

- Connect using `mysqli_connect()` or PDO.
- CRUD: Create, Read, Update, Delete records.
- Always use prepared statements to prevent SQL Injection.

Error Handling & Debugging

- Use `error_reporting(E_ALL);` to display errors.
- `try...catch` for exception handling.
- Debug with `var_dump()` and `print_r()`.

Security Essentials

- Escape user input before using in SQL (prepared statements).
- Use `password_hash()` and `password_verify()` for passwords.
- Protect against XSS by using `htmlspecialchars()`.

Object-Oriented PHP

- Classes and objects structure PHP code.
- Use inheritance, encapsulation, and polymorphism.
- Example:

```
class Car { public $color; function __construct($c) { $this->color=$c; } }
```


PHP and Web Development

- Integrate PHP with HTML, CSS, and JavaScript.
- Build dynamic pages with PHP templates.
- Use AJAX for async interactions.

Building a Simple Project

- Mini Project: Login System with MySQL.
- Steps: Create user table, build form, validate login, manage sessions.
- Extend project into a simple blog or CMS.

FAQ – Frequently Asked Questions

- Q: Can PHP be used without HTML? A: Yes, via CLI.
- Q: Is PHP outdated? A: No, PHP 8+ is modern and fast.
- Q: What is the difference between mysqli and PDO? A: PDO supports multiple databases.

Appendices

- PHP Syntax Cheat Sheet.
- Best Practices for Secure PHP Development.