

## Day 01 - Setting Up the Environment

1. Download and install XAMPP.
2. Verify your PHP installation.
3. Download and install Composer.
4. Install VS Code.
5. Install essential VS Code extensions for PHP.

## Day 02 - Basic PHP Syntax

1. Write your first PHP script ("Hello World").
2. Run your PHP script from the terminal.
3. Explore variable naming conventions in PHP.
4. Understand different variable types in PHP.

## Day 03 - Working with Variables

1. Practice declaring and using variables.
2. Understand variable scope in PHP.
3. Learn about global and local variables.
4. Use superglobals like `$_GET` and `$_POST`.

## Day 04 - Constants and Comments

1. Define and use constants in PHP.
2. Differentiate between `define()` and `const`.
3. Add single-line comments to your code.
4. Practice multi-line comments in PHP.

## Day 05 - Conditional Statements

1. Write basic `if` statements.
2. Use `else` and `elseif` conditions.
3. Understand the ternary operator.

4. Explore the switch statement.

## Day 06 - Loops in PHP

1. Write a `for` loop.
2. Explore the `while` loop.
3. Understand the `do-while` loop.
4. Loop through arrays using `foreach`.

## Day 07 - Introduction to Arrays

1. Create indexed arrays.
2. Access elements in an array.
3. Loop through indexed arrays.
4. Modify array elements.

## Day 08 - Associative Arrays

1. Create associative arrays.
2. Access elements in associative arrays.
3. Loop through associative arrays using `foreach`.
4. Understand the difference between indexed and associative arrays.

## Day 09 - Multidimensional Arrays

1. Declare a multidimensional array.
2. Access elements in a multidimensional array.
3. Loop through a multidimensional array.
4. Work with associative multidimensional arrays.

## Day 10 - Array Functions

1. Use `array_values()` and `array_keys()`.
2. Combine arrays using `array_combine()`.
3. Add elements to arrays with `array_push()`.
4. Remove elements using `array_pop()` and `array_shift()`.

## Day 11 - Advanced Array Operations

1. Slice arrays with `array_slice()` .
2. Replace parts of arrays using `array_splice()` .
3. Search arrays with `in_array()` and `array_search()` .
4. Count array elements using `count()` and `sizeof()` .

## Day 12 - Sorting Arrays

1. Sort arrays in ascending order with `sort()` .
2. Sort arrays in descending order with `rsort()` .
3. Sort arrays by key with `ksort()` and `krsort()` .
4. Use natural order sorting with `natsort()` .

## Day 13 - Introduction to Strings

1. Create and manipulate string variables.
2. Concatenate strings in PHP.
3. Explore string interpolation.
4. Handle special characters in strings.

## Day 14 - Basic String Functions

1. Measure string length with `strlen()` .
2. Find a substring using `strpos()` .
3. Extract a substring with `substr()` .
4. Convert strings to lowercase and uppercase ( `strtolower()` , `strtoupper()` ).

## Day 15 - Advanced String Functions

1. Replace parts of a string with `str_replace()` .
2. Repeat strings with `str_repeat()` .
3. Reverse a string using `strrev()` .
4. Split a string into an array with `explode()` .

## Day 16 - String Manipulation

1. Remove whitespace with `trim()`, `ltrim()`, and `rtrim()`.
2. Compare strings with `strcmp()`.
3. Hash strings with `md5()` and `sha1()`.
4. Format strings using `sprintf()`.

## Day 17 - Regular Expressions with Strings

1. Match patterns in strings using `preg_match()`.
2. Replace patterns with `preg_replace()`.
3. Extract multiple matches with `preg_match_all()`.
4. Use advanced regex patterns for string validation.

## Day 18 - Working with Forms

1. Create an HTML form.
2. Handle form submission with PHP.
3. Validate form inputs.
4. Sanitize user input.

## Day 19 - File Handling Basics

1. Open files using `fopen()`.
2. Read files with `fread()`.
3. Write to files with `fwrite()`.
4. Close files using `fclose()`.

## Day 20 - File Uploads

1. Handle file uploads with `$_FILES`.
2. Validate uploaded files.
3. Move uploaded files using `move_uploaded_file()`.
4. Display uploaded files in the browser.

## Day 21 - Sessions in PHP

1. Start a session in PHP.
2. Store and retrieve session data.
3. Destroy a session.
4. Understand where session data is stored.

## Day 22 - Cookies in PHP

1. Set cookies in PHP.
2. Retrieve cookie values.
3. Delete cookies.
4. Compare sessions and cookies.

## Day 23 - Object-Oriented Programming (OOP) Basics

1. Create a class in PHP.
2. Instantiate objects from a class.
3. Add properties and methods to a class.
4. Use the `__construct` method.

## Day 24 - OOP Inheritance

1. Implement inheritance between classes.
2. Override methods in child classes.
3. Use the `parent` keyword to access parent methods.
4. Understand abstract classes.

## Day 25 - Access Modifiers

1. Use `public` properties and methods.
2. Implement `protected` properties and methods.
3. Understand `private` properties and methods.
4. Practice using access modifiers in OOP.

## Day 26 - Working with MySQL

1. Connect to MySQL using MySQLi.
2. Execute basic SQL queries.
3. Fetch data from the database.
4. Close the database connection.

## Day 27 - Prepared Statements

1. Use prepared statements with MySQLi.
2. Bind parameters to prepared statements.
3. Execute and fetch results from prepared statements.
4. Handle errors in prepared statements.

## Day 28 - Working with JSON

1. Encode PHP arrays to JSON.
2. Decode JSON to PHP arrays.
3. Handle JSON data in APIs.
4. Work with JSON and forms.

## Day 29 - Error Handling Basics

1. Understand syntax errors.
2. Handle fatal errors in PHP.
3. Manage warnings and notices.
4. Use custom error handling with `trigger_error()`.

## Day 30 - Debugging in PHP

1. Use `var_dump()` and `print_r()` for debugging.
2. Set up XDebug in VS Code.
3. Set breakpoints and step through code.
4. Watch variables in XDebug.

## Day 31 - Regular Expressions

1. Use `preg_match()` to validate input.
2. Search for patterns with `preg_match_all()`.
3. Replace text using `preg_replace()`.
4. Practice using regex in form validation.

## Day 32 - File System Operations

1. Create and delete directories in PHP.
2. Read directory contents with `scandir()`.
3. Copy and move files in PHP.
4. Handle file paths using `realpath()`.

## Day 33 - Advanced Error Handling

1. Use try-catch blocks for exceptions.
2. Create custom exception classes.
3. Log exceptions to a file.
4. Display user-friendly error messages.

## Day 34 - Functions with Parameters and Return Types

1. Define functions with parameters.
2. Use functions with default parameter values.
3. Return values from functions.
4. Understand type hinting in function arguments.

## Day 35 - Advanced Array Functions

1. Use `array_map()` to apply a function to array elements.
2. Filter arrays with `array_filter()`.
3. Merge arrays using `array_merge()`.
4. Replace array values using `array_replace()`.

## Day 36 - Working with Sessions

1. Start and manage sessions in PHP.
2. Store and retrieve session data.
3. Implement session security.
4. Destroy sessions properly.

## Day 37 - Object-Oriented Programming: Static Properties and Methods

1. Define static properties in a class.
2. Access static properties without instantiation.
3. Create static methods in a class.
4. Call static methods without creating an instance.

## Day 38 - Advanced Object-Oriented Programming

1. Use traits for code reuse in classes.
2. Understand interfaces and polymorphism.
3. Implement design patterns (Singleton, Factory).
4. Explore dependency injection in PHP.

## Day 39 - File Handling: Reading and Writing

1. Read entire files using `file_get_contents()`.
2. Write to files using `file_put_contents()`.
3. Handle file locking for safe writing.
4. Work with large files efficiently.

## Day 40 - Security Best Practices

1. Prevent SQL injection with prepared statements.
2. Sanitize user input with `htmlspecialchars()`.
3. Protect against XSS attacks.
4. Use HTTPS for secure connections.



## Day 41 - Working with Databases

1. Normalize database structure for efficiency.
2. Implement foreign keys and relationships.
3. Use transactions in MySQL.
4. Optimize database queries.

## Day 42 - API Integration

1. Send HTTP requests using `file_get_contents()` and `cURL`.
2. Handle API responses in JSON.
3. Authenticate API requests with tokens.
4. Use `cURL` for advanced API requests.

## Day 43 - Handling Large Datasets

1. Paginate database results in PHP.
2. Optimize SQL queries for large data sets.
3. Implement lazy loading for efficiency.
4. Use `LIMIT` and `OFFSET` for data pagination.

## Day 44 - Advanced MySQLi Techniques

1. Perform complex queries with joins.
2. Handle subqueries in MySQL.
3. Use indexing to improve query performance.
4. Implement database backup and restore procedures.

## Day 45 - Error Logging and Monitoring

1. Set up error logging in PHP.
2. Monitor error logs for issues.
3. Handle and log exceptions.
4. Implement a custom error handler.

## Day 46 - JSON Web Tokens (JWT)

1. Understand the structure of JWT.
2. Create and sign JWTs in PHP.
3. Validate and decode JWTs.
4. Implement JWT-based authentication.

## Day 47 - Testing in PHP

1. Write unit tests with PHPUnit.
2. Test object-oriented code.
3. Use assertions in PHPUnit.
4. Test API endpoints with PHPUnit.

## Day 48 - Advanced File Upload Security

1. Validate file types before uploading.
2. Restrict file upload sizes.
3. Scan uploaded files for viruses.
4. Store uploaded files securely.

## Day 49 - Building REST APIs

1. Set up a basic REST API in PHP.
2. Handle GET requests.
3. Handle POST requests.
4. Secure API endpoints.

## Day 50 - Sessions and Cookies Deep Dive

1. Explore session configuration options.
2. Implement session locking for concurrency.
3. Secure cookies with `SameSite` and `Secure` attributes.
4. Handle cross-site cookie security.