

# PHP

## Prerequisites

You should not be a complete beginner for this course. If you cannot [pass this test](#), you must do [Intro to Programming](#) first.

## After this course you should be able to

- Use PHP to transform static HTML pages into dynamic web sites
- Use email from your web server
- Creating feedback forms
- Create and populate your own MySQL database tables, and work with data stored in files
- Perform sophisticated MySQL queries with joins, and refining your results
- Uploading images
- Protect your data from SQL injection attacks
- Secure your pages with HTTP Authentication
- Build a basic functional PHP website with management and user functionality that could be useful to a small business with basic requirements

## Further Training

The next course in the series is [Advanced PHP](#) .

## Course Material

Included.

## Course Contents

### Day 1

- Add life to your static pages
- PHP brings web pages to life
- Forms are made of HTML
- HTML acts on the CLIENT

- PHP acts on the SERVER
- Use PHP to access the form data
- The server turns PHP into HTML
- A few PHP rules to code by
- Variables are for storing script data
- \$\_POST transports form data to your script
- Creating the email message body with PHP
- Newlines need doublequoted strings
- Variables store the email pieces and parts
- Sending an email message with PHP
- Connecting to MySQL
- MySQL excels at storing data
- Create a MySQL database and table
- The INSERT statement in action
- Use SELECT to get table data
- Connect to your database from PHP
- Insert data with a PHP script
- Use PHP functions to talk to the database
- Get connected with `mysqli_connect()`
- Build the INSERT query in PHP
- Query the MySQL database with PHP
- Close your connection with `mysqli_close()`
- \$\_POST provides the form data
- Create and populate a database
- It all starts with a table
- Make contact with the MySQL server
- Create a database for emails
- Create a table inside the database
- We need to define our data
- MySQL data types
- Create your table with a query
- USE the database before you use it
- DESCRIBE reveals the structure of tables
- Create the Add Email script
- The Send Email script
- `mysqli_fetch_array()` fetches query results
- Looping through data with while

- Removing data with DELETE
- Use WHERE to DELETE specific data
- Minimize the risk of accidental deletions

## Day 2

- Realistic and practical application
- Demand good form data
- The logic behind Send Email validation
- Your code can make decisions with IF
- Testing for truth
- IF checks for more than just equality
- The logic behind Send Email validation
- PHP functions for verifying variables
- Test multiple conditions with AND and OR
- Form users need feedback
- Ease in and out of PHP as needed
- Use a flag to avoid duplicate duplicate code
- Code the HTML form only once
- A form that references itself
- Point the form action at the script
- Check to see if the form has been submitted
- Table rows should be uniquely identifiable
- Primary keys enforce uniqueness
- From checkboxes to customer IDs
- Loop through an array with foreach

## Day 3

- Working with data stored in files
- The application needs to store images
- Planning for image file uploads
- The high score database must be ALTERed
- How do we get an image from the user?
- Insert the image filename into the database
- Find out the name of the uploaded file
- Where did the uploaded file go?
- Create a home for uploaded image files
- Shared data has to be shared

- Shared script data is required
- Think of `require_once` as “insert”
- Order Timing is everything with high scores
- Format the top score with HTML and CSS
- Only small images allowed
- File validation makes the app more robust
- Plan for an Admin page
- Generate score removal links on the Admin page
- Scripts can communicate with each other
- Of GETs and POSTs
- GET, POST, and high score removal
- Isolate the high score for deletion
- Control how much you delete with LIMIT

## Day 4

- Securing your application
- Protecting the Admin page
- HTTP authentication requires headers
- Header Exposed
- Take control of headers with PHP
- Authenticating with headers
- Create an Authorize script
- Subtraction by addition
- Security requires humans
- Plan for moderation
- Make room for approvals with ALTER
- Unapproved scores aren't worthy
- The million-point hack
- Tricking MySQL with comments
- The Add Score form was SQL injected
- Protect your data from SQL injections
- A safer INSERT (with parameters)
- Form validation can never be too smart

## Day 5

- Using Variables in PHP
- Understanding Data Types

- Operators and Expressions
- if, else Statement
- Testing One Expression Many Times with the switch Statement
- Compact Coding with the Ternary Operator
- Doing Repetitive Tasks with Looping
- Creating and Accessing Strings
- Searching Strings
- Replacing Text within Strings
- Dealing with Upper- and Lowercase
- Formatting Strings
- The Anatomy of an Array
- Creating Arrays
- Accessing Array Elements
- Looping Through Arrays with foreach
- Working with Multidimensional Arrays
- Manipulating Arrays

### **Practical project for competency certificate :**

*Create a website with 2 tables: Product table (id, date, description, price, picture) Members table (id, name, surname, email-address, description) and pages for:*

*Customers:* to join the site to view the product list (will be expanded in the Advanced Course to add a shopping cart)

Pages and tables should be designed to be secure from SQL injections by hackers

*Management:* display the product list and add / remove and change products moderate new customer applications, delete customers the management pages must be protected with http authentication send an email newsletter to members Ensure that you use self-referencing forms that remember fields

### **Duration and pricing**

- [In pricing group A](#)

## **Certificates**

- Read about our certificates

## **Bookings**

You can download the course registration form on our home page or by clicking [here](#)

## **Brochure**

You may download a pdf copy of this page by clicking on the pdf icon at the top of the page.

## **Questions**

Please [email us](#)

## **Schedule**

On the calendar below. If your browser doesn't display the calendar below, please click on [this link](#) or try using [Google Chrome](#), alternatively please enquire via our [Contact Us](#) page.