

Chapter 1: Introduction to PHP

1.1 What is PHP?

PHP (Hypertext Preprocessor) is a widely-used open-source server-side scripting language designed specifically for web development. It is embedded within HTML and is especially suited for creating dynamic web pages.

1.2 History and Evolution of PHP

1994: Created by Rasmus Lerdorf initially to track visits to his online resume.

1995: PHP Tools were released as Personal Home Page Tools.

1997: PHP/FI 2.0 introduced, adding more functionality.

2000: PHP 4.0 released, improving performance and scalability.

2015: PHP 7.0 was released, making significant strides in performance and language features.

1.3 Setting Up the PHP Environment

To start coding in PHP, you'll need to set up a local environment on your computer. The most common way to do this is by installing a software bundle that includes:

Apache: A popular web server.

MySQL: A database management system.

PHP: The scripting language itself.

Popular bundles include:

XAMPP: Cross-platform (Windows, Mac OS, Linux)

MAMP: For Mac OS and Windows

WAMP: For Windows

Steps to Install XAMPP:

Download XAMPP from Apache Friends.

Run the installer and follow the instructions.

Start the Apache and MySQL modules from the XAMPP control panel.

1.4 PHP Syntax and Basic Commands

Let's start with a simple PHP script:

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

Key points:

PHP code is executed on the server and the result is returned to the browser as plain HTML.

PHP scripts start with <?php and end with ?>.

The echo statement is used to output text to the screen.

1.5 Writing Your First PHP Script

Now let's write a PHP script that takes input from the user and displays a personalized message:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <title>My First PHP Page</title>
```

```
</head>
```

```
<body>
```

```
  <form method="POST">
```

```
    Enter your name: <input type="text" name="name">
```

```
    <input type="submit">
```

```
  </form>
```

```
<?php
```

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {
```

```
  $name = htmlspecialchars($_POST['name']);
```

```
  echo "Hello, " . $name . "! Welcome to PHP.";
```

```
}
```

```
?>
```

```
</body>
```

```
</html>
```

Explanation:

The form uses the POST method to submit data.

The `$_SERVER["REQUEST_METHOD"]` checks if the form has been submitted.

The `htmlspecialchars` function is used to prevent XSS attacks by converting special characters to HTML entities.