INTRODUCTION

PHP (Hypertext Preprocessor) has a history dating back to the mid-1990s. Here's a brief overview of its development and key milestones:

1. **Creation by Rasmus Lerdorf (1994):**

- PHP was originally created by Rasmus Lerdorf in 1994. Initially, it was a set of Common Gateway Interface (CGI) binaries written in the C programming language. The primary purpose was to track visits to Lerdorf's online resume.

2. **PHP/FI (Personal Home Page/Forms Interpreter) (1995):**

- In 1995, PHP evolved into PHP/FI, which introduced more functionality, including form handling and the ability to communicate with databases. This version laid the foundation for PHP's future development.

3. **Zeev Suraski and Andi Gutmans Rewrite (PHP 3 - 1998):**

- Zeev Suraski and Andi Gutmans, two Israeli developers, rewrote the PHP parser in 1997, leading to PHP 3. This version introduced the Zend Engine, which significantly improved performance and added support for dynamic loading of extensions. PHP 3 marked the transition from a set of scripts to a more structured scripting language.

4. **PHP 4 (2000):**

- PHP 4, released in 2000, brought major improvements, including the introduction of Zend Engine 1. It featured better performance, support for object-oriented programming (OOP), and added numerous new features like the Zend Extension Manager.

5. **PHP 5 (2004):**

- PHP 5, released in 2004, introduced the Zend Engine 2, which included significant improvements to performance and introduced new features such as exceptions, interfaces, and the SimpleXML extension. This version marked a shift towards more robust and modern programming paradigms.

6. **PHP 6 (Project Abandoned):**

- PHP 6 was initially planned to include native support for Unicode, among other enhancements. However, due to technical challenges and disagreements within the development community, the PHP 6 project was eventually abandoned in favor of focusing on PHP 5.x releases.

7. **PHP 7 (2015):**

- PHP 7, released in December 2015, brought substantial performance improvements, reduced memory consumption, and introduced new features like the null coalescing operator (`??`), the spaceship operator (`<=>`), scalar type declarations, and return type declarations. PHP 7 marked a significant leap forward in terms of speed and efficiency.

8. **PHP 8 (2020):**

- PHP 8, released in November 2020, introduced numerous enhancements and new features, including the Just-In-Time (JIT) compiler, named arguments, union types, attributes, and improvements to error handling. PHP 8 continued the trend of improving performance and language features.

Throughout its evolution, PHP has become one of the most widely used server-side scripting languages, powering a significant portion of the web. Its open-source nature, large community, and continuous development have contributed to its popularity and adaptability in web development.

What is a PHP?

PHP, which stands for "Hypertext Preprocessor," is a widely used server-side scripting language designed for web development. It is embedded in HTML code and executed on the server, generating dynamic web pages. PHP is an open-source scripting language, and it is particularly well-suited for web development due to its ease of integration with HTML and its ability to interact with databases.

What is a PHP File?

- PHP files can contain text, HTML, CSS, JavaScript, and PHP code
- PHP code is executed on the server, and the result is returned to the browser as plain HTML
- PHP files have extension ".php"

What Can PHP Do?

- PHP can generate dynamic page content
- PHP can create, open, read, write, delete, and close files on the server
- PHP can collect form data
- PHP can send and receive cookies
- PHP can add, delete, modify data in your database

- PHP can be used to control user-access
- PHP can encrypt data

With PHP you are not limited to output HTML. You can output images or PDF files. You can also output any text, such as XHTML and XML.

Why PHP?

- PHP runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.)
- PHP is compatible with almost all servers used today (Apache, IIS, etc.)
- PHP supports a wide range of databases
- PHP is free. Download it from the official PHP resource: www.php.net
- PHP is easy to learn and runs efficiently on the server side

PHP Installation

Installing PHP depends on your operating system. Here are instructions for a few common operating systems:

Windows:

1. **Using XAMPP:**

- Download and install [XAMPP](https://www.apachefriends.org/index.html), which is a package that includes Apache (web server), MySQL (database server), and PHP.

- Follow the installation wizard and choose the components you want to install (Apache, MySQL, PHP).

- Once installed, start the Apache server from the XAMPP Control Panel.

2. **Using WampServer:**

- Download and install [WampServer](https://www.wampserver.com/).

- Follow the installation instructions, and WampServer will install Apache, MySQL, and PHP on your system.

- Start WampServer, and it will automatically start Apache and MySQL.

macOS:

1. **Using Homebrew:**

- Open Terminal.

- Install Homebrew if you haven't already by following the instructions on the [Homebrew website](https://brew.sh/).

- Run the following commands in Terminal:

```bash
brew update
brew install php
```

2. **Using MAMP:**

- Download and install [MAMP](https://www.mamp.info/en/).

- Once installed, start the MAMP application.

- MAMP includes Apache, MySQL, and PHP, and it provides an easy way to manage your local development environment.

Linux (Ubuntu):

1. **Using Apt:**

- Open Terminal.

- Run the following commands:

```bash
sudo apt update
sudo apt install php
```

2. **Using Tasksel:**

- Open Terminal.

- Run the following commands:

```bash
sudo apt update
sudo apt install tasksel
sudo tasksel install lamp-server
```

- This will install Apache, MySQL, and PHP on your system.

After installing PHP, you can test whether it's working by creating a simple PHP file and accessing it through a web browser. For example, create a file named `info.php` with the following content:

```php
<?php
phpinfo();
?>
```

Save the file in the web server's document root (e.g., `htdocs` in XAMPP, `www` in WampServer, or `/var/www/html` in Linux), and then access it in your browser by navigating to `http://localhost/info.php`.

This file will display detailed information about your PHP installation, confirming that PHP is working correctly on your system.