

## PHP

PHP (Hypertext Preprocessor) is an open-source, server-side scripting language primarily used for web development. It allows you to create dynamic and interactive web pages, interface with databases, and manage user sessions.

### Main Features

**Server Side:** PHP runs on the server and generates dynamic HTML that is sent to the user's browser.

**Ease of Use:** It has a simple and intuitive syntax, similar to C and JavaScript, which makes it accessible even to beginners.

**Database Compatibility:** It supports numerous databases, including MySQL, PostgreSQL, SQLite, and many others.

**Cross-Platform:** It runs on various operating systems such as Windows, Linux, and macOS.

**HTML and JavaScript Integration:** PHP can be embedded directly into HTML code and combined with JavaScript to improve interactivity.

**Session and Cookie Management:** It offers native tools for managing user sessions and cookies, essential for advanced web applications.

**Programming Paradigm Support :** It supports procedural, functional, and object-oriented programming (OOP).

**Security :** It includes features to protect applications from common attacks such as SQL Injection and XSS (Cross-Site Scripting), although security also depends on the correct implementation by the developer.

**Large Community and Documentation :** PHP has a large community of developers and official documentation full of examples and guides.

**Supported Frameworks and CMS :** It is the basis of many popular frameworks (Laravel, Symfony, CodeIgniter) and CMS such as WordPress, Joomla, and Drupal.

PHP is still widely used in web development today, due to its flexibility and ease of learning. PHP (Hypertext Preprocessor) is a server-side scripting language widely used for web development. Due to its simplicity and flexibility, it is ideal for creating dynamic and interactive pages. A PHP page is a text file with the .php extension that contains both HTML code and PHP code.

PHP is a case-sensitive language, derived from C, in fact it requires that instructions always end with a semicolon (;). To identify a string you can use either single quotes ( ' ') or double quotes ( " "). Finally, comments, just like in C, can be on one line (//) or on multiple lines (/\*...\*/).

PHP scripts can be placed both in the <HEAD> and in the <BODY> of an HTML document, however they cannot start in the <HEAD> and end in the <BODY>. The file as a whole is executed from top to bottom, and it is possible to switch multiple times from HTML instructions to PHP instructions.

### Variables and operators

Variables in PHP are delimited by the dollar character (\$), followed by an alphabetic character or the underscore character (\_). The characters following the second can contain any sequence of numbers, alphabetic characters and underscores. The language is case-sensitive, therefore \$variable will be different from \$Variable.

The operators are the same as in C, addition (+), subtraction (-), multiplication (\*), division (/) and remainder of integer division (%). Unlike C, however, the result of the operation is always of type float, even if the division occurs between two integers. The fundamental assignment operator is the equal sign (=) and can also be applied in the following way: a+=3; corresponds to a=a+3;. This also applies to all other arithmetic operators, as well as the string concatenation operator.

Variables in PHP are defined with the symbol \$, followed by the name of the variable. Operators are fundamental tools for performing calculations and logical operations. There are different types of operators, including:

- Arithmetic operators (+, -, \*, /, %)
- Comparison operators (==, !=, >, <, >=, <=)
- Logical operators (&&, ||, !)

### Arrays

Arrays in PHP allow you to store multiple values in a single variable. There are three main types of arrays:

- Indexed arrays (elements are associated with a numeric index)
- Associative arrays (elements are associated with textual keys)
- Multidimensional arrays (arrays nested within other arrays)

### The if structure

The if structure allows you to execute blocks of code only if a certain condition is true. It can be combined with else and elseif to handle multiple conditions.

### The while and for structures

These constructs allow you to perform repeated loops.

- while: repeats a block of code as long as a condition is true.
- for: useful when the number of iterations is known in advance.

### User interaction

PHP allows you to collect user input through HTML forms. Data can be sent with the GET or POST methods and managed through the superglobal variables \$\_GET and \$\_POST.