

# Introduzione a PHP

## Corso di Laboratorio di Informatica

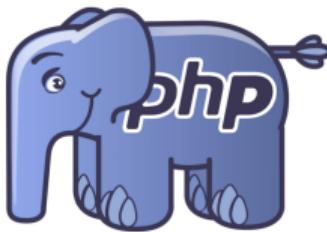
Prof. Dei Rossi, Leonardo Essam



# Introduzione (1)

Il linguaggio PHP, acronimo ricorsivo per "PHP: Hypertext Processor", è un linguaggio di programmazione e *de-facto* è lo standard per la programmazione Web lato server.

È un linguaggio definito di *scripting server-side* e viene usato per generare contenuti Web (HTML) da inviare al browser dell'utente.



## Introduzione (2)

Alcune caratteristiche principali:

- Creato nel 1995, è oggi uno dei linguaggi più usati per creare siti e applicazioni web;
- Open source, gratuito e supportato da una vasta community;
- Si integra facilmente con HTML, database come MySQL e server web come Apache;
- Utilizzato per gestire form, sessioni, login, CMS (come WordPress) e molto altro.

## Requisiti (1)

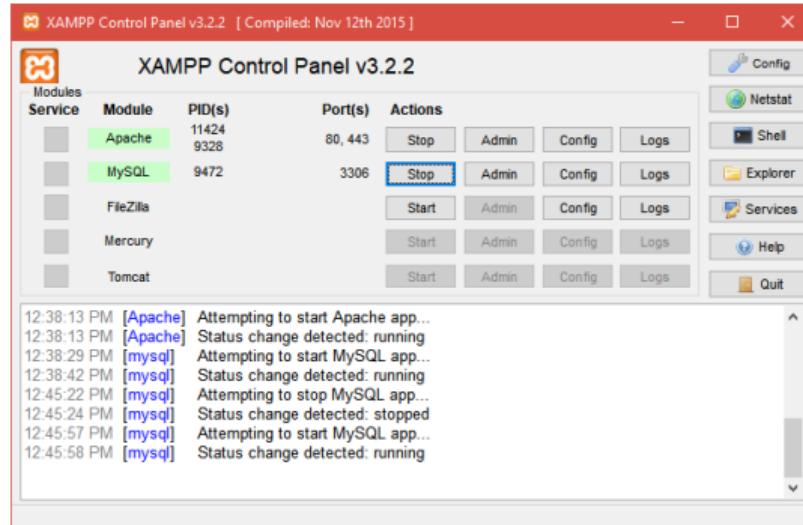
Per usare PHP è necessario un software che fornisca i seguenti componenti:

- Apache (web server);
- MySQL (DBMS);
- PHP (linguaggio).

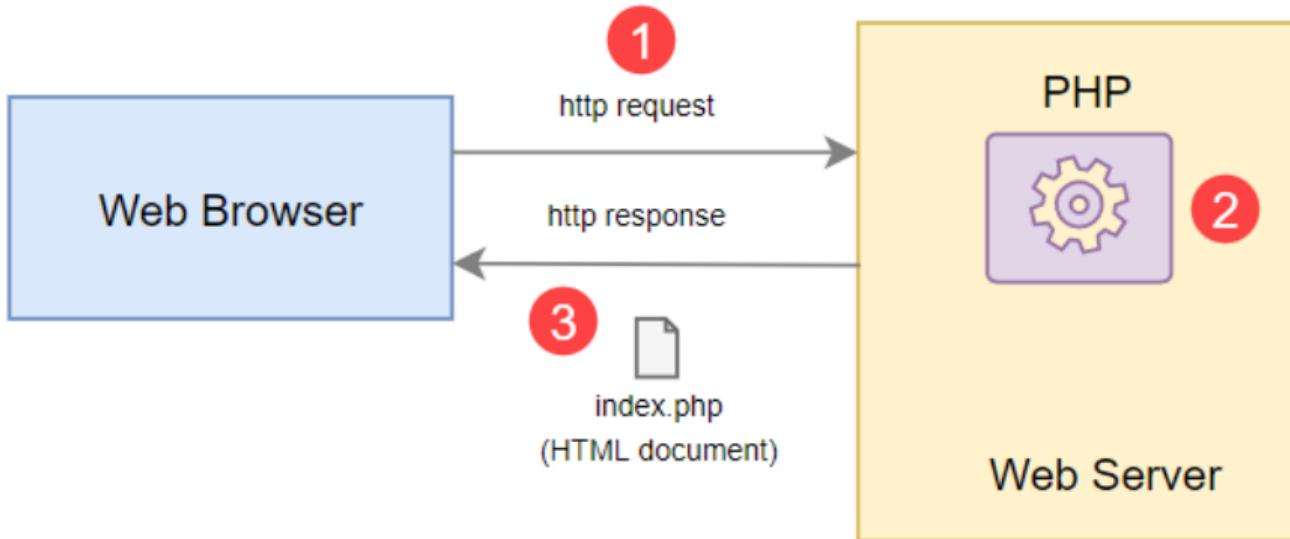
L'unione di questi componenti viene solitamente chiamata **LAMP** (la "L" sta per "Linux").

## Requisiti (2)

Per Windows, un software versatile che fornisce lo stack di cui sopra è XAMPP. È pre-installato nelle postazioni in laboratorio, a casa si può installare dal [\(link\)](#).



# Come funziona PHP? (1)



# Come funziona PHP? (2)

Panoramica sul funzionamento di PHP:

1. **Richiesta dal browser**

L'utente inserisce un URL o invia un form → il browser invia una richiesta HTTP al server.

2. **Il server individua il file .php**

Se la pagina richiesta contiene codice PHP, il server passa il file all'interprete PHP.

# Come funziona PHP? (3)

## 3. Esecuzione lato server

L'interprete PHP esegue lo script:

- Elabora i dati (input);
- Interagisce con il database (es. MySQL);
- Esegue logica, controlli e calcoli.

## 4. Output HTML generato

Il risultato dell'esecuzione è trasformato in codice HTML puro.

## 5. Risposta al browser

Solo l'HTML finale viene inviato al client → il codice PHP non è visibile all'utente.

## La sintassi di PHP (1)

Un file che contiene codice PHP è un file che termina con l'estensione .php, il codice PHP effettivo viene contenuto all'interno di un carattere di apertura e di chiusura:

```
<?php  
// Inserire qui il codice PHP...  
?>
```

## La sintassi di PHP (2)

Essendo PHP un linguaggio *Web-oriented*, è logico aspettarsi che PHP venga usato all'interno di delle pagine Web HTML.

Il vantaggio di avere un tag di apertura (`<?php`) e di chiusura (`?>`) sta nel fatto che è possibile inserire una porzione di codice PHP all'interno di una pagina HTML già strutturata.

**Attenzione però che l'estensione della pagina deve essere .php, altrimenti la porzione di codice PHP verrà ignorata dall'interprete!**

## La sintassi di PHP (3)

Un esempio di codice è:

```
<body>
    <h1>Ciao, mondo!</h1>

    <p>
        La data di oggi è <?php echo(date("d/m/Y")); ?>.
    </p>
</body>
```

## La sintassi di PHP (4)

L'output sarà:

**Ciao, mondo!**

La data di oggi è 02/11/2025

## Le variabili in PHP (1)

In PHP, una variabile inizia con il simbolo del dollaro \$, seguito dal nome della variabile:

```
<?php  
$a = 10;  
$b = "Ciao, mondo!";  
?>
```

## Le variabili in PHP (2)

Alcune regole sulle variabili:

- Il nome inizia sempre con il \$;
- Il nome deve iniziare sempre con un carattere o un underscore \_;
- Il nome non può iniziare con un numero;
- Il nome è case-sensitive ( $\$A \neq \$a$ ).

# La sagra dei tipi di dato (1)

**Come ampiamente visto e discusso in JavaScript, anche in PHP è la sagra dei tipi di dato.**

Detta più formalmente, PHP fa parte della famiglia dei linguaggi debolmente tipizzati. Segue che il tipo di dato della variabile non è pre-fissato all'atto della dichiarazione ma può cambiare al variare del valore (es. in caso di ri-assegnazione).

## La sagra dei tipi di dato (2)

Riprendendo l'esempio precedente:

```
<?php
$a = 10;
$b = "Ciao, mondo!";

echo(gettype($a)); // integer
echo(gettype($b)); // string

?>
```

