



Introduzione a PHP prt 2

Programmazione

PHP (Hypertext Preprocessor) è un linguaggio di scripting open source progettato principalmente per lo sviluppo web lato server.

Nasce con l'obiettivo di rendere dinamiche le pagine HTML, permettendo di generare contenuti personalizzati in base alle richieste dell'utente, ai dati provenienti da database o allo stato dell'applicazione.

Il codice PHP viene eseguito sul server e il risultato finale inviato al browser sotto forma di HTML, CSS e JavaScript, rendendo il linguaggio trasparente all'utente finale.

Grazie alla sua sintassi semplice, alla forte integrazione con i database (come MySQL) e all'ampia diffusione in CMS e framework web, PHP è ancora oggi una delle tecnologie più utilizzate per la creazione di siti e applicazioni web dinamiche.



Ecco 4 caratteristiche fondamentali che definiscono PHP, spiegate in modo tecnico ma chiaro:

- **Linguaggio server-side**

PHP viene eseguito interamente sul server: il browser riceve solo l'output finale (HTML/CSS/JS). Questo lo rende adatto alla gestione di logica applicativa, autenticazione utenti, accesso ai database e controllo dei dati sensibili.

- **Integrazione nativa con HTML**

PHP può essere inserito direttamente all'interno del codice HTML, senza bisogno di compilazione. Questa caratteristica lo rende immediato per lo sviluppo web e facile da adottare anche per chi proviene da HTML e CSS.



Ecco 4 caratteristiche fondamentali che definiscono PHP, spiegate in modo tecnico ma chiaro:

- **Forte supporto ai database**

PHP offre API native per l'accesso ai database relazionali (in particolare MySQL/MariaDB, ma anche PostgreSQL, SQLite, ecc.). È ampiamente utilizzato per applicazioni data-driven come CMS, e-commerce e gestionali web.

- **Multipiattaforma e open source**

PHP è open source, gratuito e compatibile con i principali sistemi operativi (Windows, Linux, macOS) e web server (Apache, Nginx). Questo lo rende estremamente diffuso e supportato da una vasta community e da numerosi framework.



- **Generazione dinamica di contenuti HTML**

Uno degli utilizzi fondamentali di PHP è la generazione dinamica di pagine HTML.

A differenza delle pagine statiche, il contenuto può cambiare in base a variabili, condizioni o dati provenienti dal server. PHP viene eseguito lato server e produce HTML che il browser renderizza normalmente.

```
1.<?php  
2.$nome = "Mirko";  
3.echo "<h1>Ciao $nome!</h1>";  
4.?>
```

Il codice PHP viene interpretato dal server: la variabile \$nome viene sostituita nel testo e il browser riceve solo l'HTML finale (<h1>Ciao Mirko!</h1>). L'utente non vede il codice PHP.



- **Gestione delle condizioni e del flusso logico**

PHP permette di controllare il flusso dell'applicazione tramite strutture condizionali.

Questo è essenziale per personalizzare il comportamento di una pagina (es. accessi, permessi, stati applicativi).

```
1.<?php  
2.$eta = 20;  
3.  
4.if ($eta >= 18) {  
5.    echo "Accesso consentito";  
6.} else {  
7.    echo "Accesso negato";  
8.}  
9.?>
```



Il server valuta la condizione e decide quale output generare. Questo approccio è alla base di login, autorizzazioni e contenuti riservati.

- **Raccolta e gestione dei dati da form HTML**

PHP è ampiamente usato per gestire input dell'utente, come form HTML.

I dati inviati tramite GET o POST vengono intercettati dal server e possono essere validati, elaborati o salvati.

```
1.<?php  
2.if (isset($_POST["username"])) {  
3. echo "Username inserito: " . $_POST["username"];  
4.  
5.?>
```

```
1.<form method="post">  
2. <input type="text" name="username">  
3. <input type="submit">  
4.</form>
```

 **Il super-array `$_POST` contiene i dati inviati dal form. PHP consente di controllare l'esistenza dei dati e usarli in modo sicuro lato server.**

- **Interazione con database (caso d'uso data-driven)**

Uno dei punti di forza di PHP è l'accesso ai database, fondamentale per applicazioni dinamiche come CMS, e-commerce e dashboard, PHP può leggere, scrivere e aggiornare dati persistenti.

```
1.<?php  
2.$conn = new mysqli("localhost", "root", "", "scuola");  
3.  
4.$result = $conn->query("SELECT nome FROM studenti");  
5.  
6.while ($row = $result->fetch_assoc()) {  
7.    echo $row["nome"] . "<br>";  
8.}  
9.?>
```



Il codice si connette al database, esegue una query SQL e cicla i risultati.
Ogni riga diventa contenuto dinamico della pagina web.

- **Gestione dello stato con le sessioni**

Il web è stateless, ma PHP permette di mantenere informazioni tra più richieste tramite le sessioni, fondamentali per login, carrelli e preferenze utente.

```
1.<?php  
2.session_start();  
3.$_SESSION["utente"] = "admin";  
4.echo "Sessione attiva";  
5.?>  
6.
```

La sessione salva dati lato server associati all'utente.

Finché la sessione è valida, PHP può recuperare queste informazioni in qualunque pagina.



- **Inclusione e modularizzazione del codice**

PHP favorisce la modularità del codice, consentendo di suddividere l'applicazione in file riutilizzabili.

Questo migliora manutenzione, leggibilità e scalabilità.

1.// header.php

2.<h1>Benvenuto nel sito</h1>

1.<?php

2.include "header.php";

3.echo "Contenuto principale";

4.?>

Il file viene incluso ed eseguito come se fosse parte dello stesso script.

Questo approccio è alla base di template, layout e architetture MVC.





Buon Davante a tutti