

Primi passi con PHP

1

In questo capitolo

- Introduzione alla programmazione (o coding)
 - Introduzione al linguaggio PHP
 - Chi usa PHP e per che cosa
 - L'ecosistema PHP
 - L'ambiente di sviluppo XAMPP
 - Editor di testo
 - Che cosa hai imparato
-

In questo capitolo, dopo aver brevemente introdotto alcuni concetti fondamentali come informatica, computer, algoritmo e programma, presentiamo il linguaggio PHP. Vediamo quali sono l'ambito d'uso, la diffusione e gli strumenti per iniziare a programmare con il linguaggio.

Introduzione alla programmazione (o coding)

Partiamo da alcuni concetti base. L'*informatica* è la scienza che si occupa del trattamento dell'informazione, ovvero l'acquisizione, la memorizzazione, l'elaborazione e la trasmissione dei dati. I *computer* sono macchine programmabili che acquisiscono dati in ingresso (*input*) e li elaborano in base alle istruzioni contenute in un *programma* informatico per produrre dati in uscita (*output*) (Figura 1.1).



Figura 1.1 Schema base di funzionamento di un programma in un computer.

Algoritmi, linguaggi, programmi

Un *programma* è una sequenza di istruzioni scritte in un apposito *linguaggio* che dice all'esecutore passo passo che cosa fare, proprio come una ricetta di cucina contiene le istruzioni per la preparazione di una pietanza e uno spartito contiene tutte le informazioni per eseguire un brano musicale.

Programmare (in inglese, *to program* o *to code*) significa scrivere programmi che risolvono determinati problemi:

1. si parte da un problema, cioè una situazione o un quesito che richiede una soluzione, e si raccolgono gli elementi noti di rilievo;
2. si escogita un *algoritmo*, cioè un metodo per trasformare i dati noti nei giusti risultati;
3. si scrive l'algoritmo in un linguaggio di programmazione e si verifica che il programma faccia quanto previsto (Figura 1.2).



Figura 1.2 Il programma implementa un algoritmo per risolvere un problema.

Una definizione di algoritmo

La parola *algoritmo* deriva dal nome del matematico persiano medievale *al-Khwarazmi* e indica un procedimento definito da una sequenza finita di passi eseguibili per raggiungere un determinato obiettivo o ottenere un certo risultato.

Introduzione al linguaggio PHP

PHP è un linguaggio di programmazione utilizzato soprattutto per la creazione di applicazioni web e si caratterizza per essere:

- *di alto livello e interpretato*: il programmatore non deve occuparsi degli aspetti di “basso livello” legati all’hardware (per esempio, come pilotare, o comandare, una specifica stampante oppure come accendere i pixel del display); il programma *interprete* di PHP si occupa di tradurre le righe scritte in PHP in istruzioni direttamente comprensibili dal computer (cioè in *linguaggio macchina*);
- *multiplatforma*: i programmi scritti in PHP sono *portabili*, cioè possono essere eseguiti su differenti sistemi operativi, tra i quali Windows, macOS e GNU/Linux;
- *gratuito*: chiunque lo può scaricare da Internet, installare e utilizzare per scrivere programmi;
- *open source*: puoi vedere il codice sorgente, segnalare problemi agli sviluppatori e partecipare allo sviluppo del codice;
- *semplice e conciso*: la sintassi del linguaggio è abbastanza facile da imparare e la creazione di pagine web è piuttosto rapida;
- *general purpose*: PHP non è legato allo sviluppo di specifiche applicazioni (per esempio, procedure finanziarie), ma si presta alla realizzazione di programmi di varia natura anche grazie a una vasta libreria di funzioni;
- *HTML-embedded*: si integra con facilità nelle pagine HTML.

Un po' di storia

Nel 1994, il programmatore danese-canadese Rasmus Lerdorf inizia a sviluppare un sistema per realizzare semplici pagine dinamiche che lo aiuti nella creazione della sua home page (da qui il nome iniziale di PHP come *Personal Home Page*). Lerdorf mette insieme i linguaggi Perl e C e qualche idea innovativa e aggiunge, passo dopo passo, ciò che sembra più opportuno in quel momento, senza una vera intenzione di scrivere un linguaggio di programmazione, cosa che dichiara di non saper fare! Nel 1997 si uniscono al progetto gli israeliani Zeev Suraski e Andi Gutmans: grazie al loro apporto e alla nascente e già molto attiva comunità *open source*, vede la luce la versione 3 di PHP con l’interprete *Zend*. L’acronimo

PHP cambia e diventa *ricorsivo* assumendo il significato di “PHP: Hypertext Pre-processor”. Negli anni successivi le nuove versioni via via rilasciate integrano nel linguaggio *feature* fino ad allora mancanti come, per esempio, la programmazione orientata agli oggetti. In particolare, l’ultima versione al momento della stesura di questo libro, la 7, risponde ad alcune critiche sulla sintassi del linguaggio e lo rende ancora più ricco e completo.

NOTA

I programmi affrontati nel testo fanno riferimento a PHP 7 con una buona retrocompatibilità alla precedente versione 5.

Chi usa PHP e per che cosa

PHP è il linguaggio *server-side* più utilizzato per costruire siti e servizi web (Figura 1.3). Le ragioni di questo successo sono legate alla semplicità della sintassi (che richiama quella ben nota del linguaggio C, da cui derivano anche Java e JavaScript), al fatto di essere stato gratuitamente disponibile dall’alba di Internet e al fatto che sono state sviluppate solide applicazioni per la realizzazione di siti web. Infatti, la maggior parte dei siti web è costruita utilizzando CMS (*Content Management System*) scritti in PHP così come in PHP sono realizzati i popolari Facebook e Wikipedia e la piattaforma di e-learning Moodle.

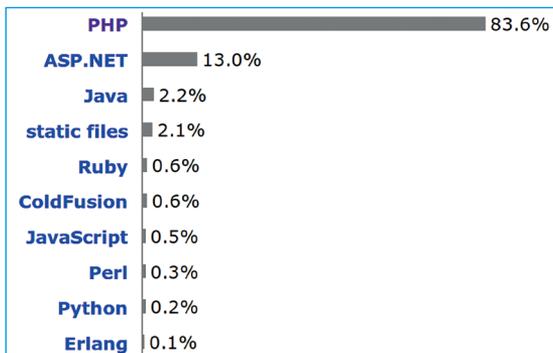


Figura 1.3 Percentuale dei siti web che usano i vari linguaggi di programmazione server-side (Fonte: <https://w3techs.com>, 13 agosto 2018).

L’ecosistema PHP

Intorno al linguaggio di programmazione PHP ruota un vasto ecosistema costituito da:

- un *programma* che interpreta ed esegue codice scritto nel linguaggio PHP;

- una CLI (l'interprete a linea di comando con cui interagirai a fine capitolo);
- un insieme di *librerie*, che raccolgono codice scritto da altri sviluppatori e che puoi utilizzare nei tuoi programmi;
- *framework* applicativi per lo sviluppo rapido di web app di grandi dimensioni;
- piattaforme open source pronte da installare e personalizzare, come i tre CMS più utilizzati, Wordpress, Joomla! e Drupal, e le piattaforme per l'e-commerce Magento e Prestashop (in Figura 1.4 sono riportati i loro loghi);
- una *comunità* di appassionati, professionisti e aziende interessate all'evoluzione del linguaggio (<https://phpcommunity.org>, <http://php.net>, <https://2018.phpday.it>).



Figura 1.4 L'ecosistema di PHP comprende alcune delle principali piattaforme per la creazione di siti web.

L'ambiente di sviluppo XAMPP

XAMPP è un pacchetto software che permette di installare con pochi clic il cosiddetto *full stack*, ovvero tutti gli strumenti necessari (e anche qualcosa in più) per sviluppare in PHP: un *server web*, che fornisce le pagine HTML e i cosiddetti *asset* (immagini, font, fogli di stile, file JavaScript ecc.), il motore (o interprete) PHP e MariaDB, un potente DBMS per la gestione di *basi di dati*. Con il browser vai alla pagina <https://www.apachefriends.org>, cerca l'immagine di Figura 1.5 e fai clic sul pulsante corrispondente al tuo sistema operativo per scaricare l'ultima versione di XAMPP.



Figura 1.5 I pulsanti per il download di XAMPP dal sito <https://www.apachefriends.org>.

Una volta scaricato l'eseguibile avvialo. Vediamo in dettaglio solo i passaggi per Windows, ma per macOS e Linux l'installazione è analoga. Procedi facendo clic sul pulsante *Next*, confermando tutte le impostazioni proposte, fino alla finestra di Figura 1.6 e deseleziona alcuni programmi per alleggerire l'installazione: tieni solo Apache (il web server), MySQL (il DBMS), PHP e phpMyAdmin. Nella schermata successiva lascia come destinazione la directory **C:\xampp** che permette di modificare i file anche senza i privilegi di amministratore.

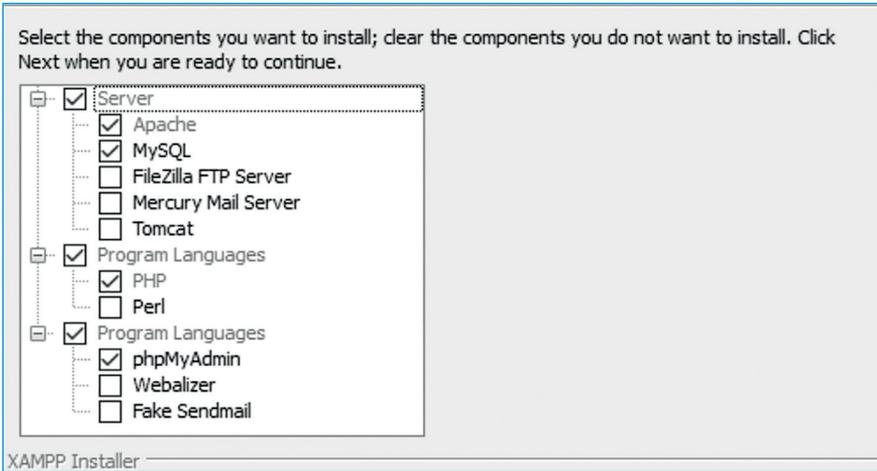


Figura 1.6 Installazione di XAMPP: la scelta dei componenti.

Alla fine dell'installazione appare una finestra con un avviso di sicurezza di Windows. Fai clic per consentire l'accesso al sistema da parte dell'applicazione Apache. Nel pannello di controllo di XAMPP (Figura 1.7) puoi controllare avvio, arresto e configurazione dei moduli mediante i pulsanti *Actions*. Avvia Apache (①), che si occupa di eseguire PHP quando occorre (puoi avviare anche MariaDB che ci servirà negli ultimi capitoli del libro).



Figura 1.7 Pannello di controllo di XAMPP con Apache e MariaDB (MySQL) avviati.

Fai clic sul pulsante *Admin* relativo ad Apache (②). Se tutto funziona si apre una finestra nel browser predefinito con la pagina di benvenuto di XAMPP (Figura 1.8).

ra 1.8). Qui fai clic su *PHPIInfo* e accedi alla pagina informativa di PHP, dove puoi trovare dettagli interessanti su PHP.

Nel pannello di controllo di XAMPP fai clic sul pulsante *Explorer* (③) e apri la cartella **htdocs** (in Windows l'indirizzo completo di questa directory è **C:\xampp\htdocs**) dove metterai i programmi sviluppati nel libro. Ti consigliamo di scrivere i programmi tu stesso. In alternativa, puoi scaricarli dall'indirizzo <http://bit.ly/apo-bm-php> e scompattarli nella cartella **htdocs** in modo da avere una cartella per ogni capitolo del libro: **cap02**, **cap03**, e così via.



Figura 1.8 La pagina di benvenuto di XAMPP con evidenziato il link alla pagina di informazioni di PHP.

Torneremo al browser più avanti, per ora accedi alla console PHP. Nella finestra di controllo di XAMPP, fai clic sul pulsante *Shell* (④). L'interprete dei comandi del tuo sistema operativo si apre in una finestra nera con un cursore lampeggiante. Digita all'interno il seguente comando (tranne **#**) e premi Invio per eseguirlo:

```
# php -a
Interactive shell
php >
```

Nel descrivere i comandi da digitare riportiamo anche il *prompt* (il carattere **#** o la scritta **php >**), che è il segnale di attesa o "pronto" del sistema e che non va digitato.

NOTA

Il prompt **php >** ti informa che la shell interattiva di PHP ha preso il controllo al posto della console del sistema operativo (**#**). Digita il comando PHP che segue e premi Invio:

```
php > echo "hello, world";
hello, world
```

Complimenti, hai scritto ed eseguito il tuo primo codice PHP!

NOTA

Nel 1974 Brian Kernighan scrive in C il programma **"hello, world"** che diventerà famoso con la pubblicazione del libro scritto con Dennis Ritchie, *Il linguaggio C*, una pietra miliare della programmazione. Da allora, quando si impara un nuovo linguaggio, la prima cosa che si fa è scrivere ed eseguire un programma che ha come output questa frase di saluto.

Per chiudere la finestra digita due volte **exit** + Invio: la prima per uscire dall'interprete PHP e la seconda dalla shell del sistema operativo.

```
php > exit  
# exit
```

Per chiudere tutto, nella finestra di XAMPP, fai clic sul pulsante *Stop* in corrispondenza di Apache e poi chiudi la finestra.

Editor di testo

Per scrivere le istruzioni di un programma vero e proprio hai bisogno di un programma di scrittura di testo che permetta di salvare il *codice sorgente* PHP su file per poterlo modificare ed eseguire più volte. Esistono molti *editor* open source e gratuiti, ricchi di *plugin* (o estensioni) che facilitano la scrittura di sorgenti nei diversi linguaggi. Noi facciamo riferimento ad Atom, che puoi installare scaricandolo da <https://atom.io>. Altri editor validi sono Notepad++ (<https://notepad-plus-plus.org>) e Visual Studio Code (<https://code.visualstudio.com>).

Che cosa hai imparato

- Che cosa vuol dire programmare un computer, che cosa sono un algoritmo e un programma.
- Che cos'è PHP, com'è nato e quali sono le sue caratteristiche principali.
- Gli elementi che costituiscono l'ecosistema PHP.
- Ambiti di utilizzo del linguaggio e riferimenti alla comunità PHP.
- Come installare l'ambiente di sviluppo XAMPP e a cosa servono gli editor di testo come Atom.
- Come avviare la console a linea di comando e l'interprete PHP.
- Scrivere ed eseguire nella console la tua prima istruzione PHP.