



A

ISTRUZIONI
DI BASE

Nella prima parte del libro acquisirete le nozioni base sulla scrittura di codice PHP.

Programmare significa creare una lista di istruzioni che un computer può eseguire per svolgere un compito specifico. Potete paragonare queste istruzioni a una ricetta che contiene i passaggi da seguire per creare un piatto. Ogni singolo passo di codice PHP è un'**istruzione**.

Poiché PHP è stato progettato per creare siti web in grado di creare dinamicamente pagine HTML, ciò che apprenderete in questa prima parte del libro si concentrerà su come viene utilizzato PHP per creare pagine HTML.

Un sito web è spesso composto da migliaia di righe di codice, quindi è importante organizzarlo con cura. Questa parte introduce due concetti che raggruppano determinati insiemi di istruzioni correlate.

- Le **funzioni** raggruppano le istruzioni necessarie per eseguire una singola attività.
- Gli **oggetti** raggruppano istruzioni che rappresentano concetti: per esempio articoli visualizzati su un sito, prodotti venduti da un sito o utenti iscritti a un sito.

Gli argomenti di questa parte formano le basi per tutti gli altri contenuti del libro.

Prima di affrontare il primo capitolo, ci sono alcune nozioni fondamentali da acquisire.

COME INSTALLARE IL SOFTWARE E GLI ESEMPI DI CODICE

Per creare siti utilizzando PHP e un database MySQL su un computer desktop o laptop, sarà necessario installare del software. Una volta installato, potrete scaricare il codice di esempio del libro dal nostro sito web all'indirizzo:
<http://phpandmysql.com/code/>

COME I FILE PHP CONTENGONO UN MIX DI CODICE HTML E PHP

Poiché il linguaggio PHP viene utilizzato per creare pagine HTML, in genere un file PHP contiene una combinazione di codice HTML e PHP, quindi è necessario comprendere come l'interprete PHP rileva la differenza tra questi due tipi di codice.

COME PHP È USATO PER CREARE CODICE HTML

Una delle istruzioni più comuni che darete all'interprete PHP è di aggiungere contenuti alla pagina HTML da inviare al visitatore. Questa istruzione viene utilizzata in ogni esempio di questa parte.

COME AGGIUNGERE COMMENTI AL CODICE PHP

I commenti non vengono eseguiti dall'interprete PHP, ma aiuteranno, voi e altri, a capire che cosa dovrebbe fare il codice, e quindi è importante imparare ad aggiungerli. In tutto il libro troverete all'interno del codice dei commenti, che spiegano che cosa fanno determinate istruzioni di codice.

INSTALLAZIONE DEL SOFTWARE E DEI FILE

Gli strumenti che seguono installano, su un computer desktop o laptop, tutto il software necessario per creare siti web basati su database.

Per questo libro, occorre installare:

- un **server web** che esegua un interprete PHP; questo libro usa **Apache** (il server più utilizzato);
- **MySQL** o **MariaDB**, un software per database;
- **phpMyAdmin** per gestire il database.

Invece di scaricare e installare singolarmente ognuno di questi programmi, gli strumenti descritti di seguito li scaricano e li installano al vostro posto.

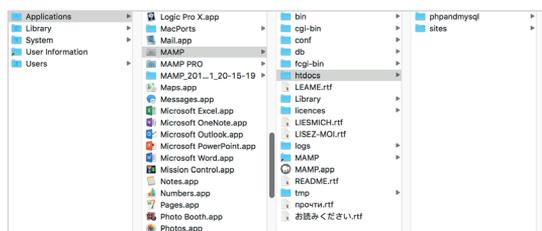
Si consiglia inoltre di utilizzare un editor di codice come descritto in: <http://notes.re/php/editors>

INSTALLAZIONE SU MAC

Si consiglia gli utenti Mac di installare il software richiesto utilizzando uno strumento chiamato MAMP. Il link per scaricarlo e le istruzioni per usarlo sono disponibili all'indirizzo: <http://notes.re/php/mamp>.

Installando MAMP su Mac (usando le impostazioni predefinite), viene creata la cartella: `/Applications/MAMP`

All'interno di questa cartella ne è presente un'altra di nome `htdocs`. Tutte le pagine web che si scrivono con PHP devono essere collocate all'interno di questa cartella, che è la **radice del documento**.

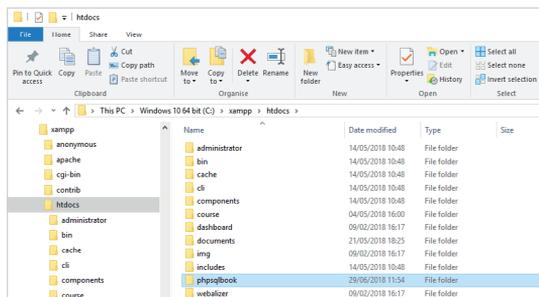


INSTALLAZIONE SU PC / LINUX

Si consiglia gli utenti PC e Linux di installare il software richiesto utilizzando uno strumento chiamato XAMPP. Il link per scaricarlo e le istruzioni per usarlo sono disponibili all'indirizzo: <http://notes.re/php/xampp>.

Installando XAMPP su PC o Linux (con le impostazioni predefinite), viene creata la cartella: `c:\xampp\`

All'interno di questa cartella ne è presente un'altra di nome `htdocs`. Tutte le pagine web che si scrivono con PHP devono essere collocate all'interno di questa cartella, che è la **radice del documento**.



COME LE PAGINE PHP COMBINANO INSIEME IL CODICE HTML E PHP

Molte pagine PHP combinano codice HTML e PHP. Il codice PHP viene racchiuso tra tag PHP. I tag PHP di apertura e chiusura, oltre a tutto il codice PHP che contengono, formano **blocchi PHP**.

TAG DI APERTURA

<?php

Il tag di apertura indica che l'interprete PHP deve iniziare a elaborare il codice prima di inviare il contenuto al browser.

TAG DI CHIUSURA

?>

Il tag di chiusura indica che l'interprete PHP può interrompere l'elaborazione del codice finché non incontra un altro tag di apertura <?php.

PHP è un tipo di **linguaggio di scripting**. I linguaggi di scripting sono progettati per essere eseguiti in uno specifico ambiente; PHP è stato creato per funzionare con un interprete PHP su un server web.

Una singola pagina PHP è spesso chiamata **script**.

Dovreste trattare tutto il codice PHP come se facesse distinzione tra maiuscole e minuscole.

Sebbene alcune parti del linguaggio non facciano distinzione tra maiuscole e minuscole, distinguendole otterrete meno errori.

Una pagina PHP è un file di testo (proprio come un file HTML). L'estensione del file è `.php`, e chiede al server web di inviare il file all'interprete PHP, in modo che possa seguire le istruzioni PHP in esso contenute.

La seguente pagina PHP contiene due elementi.

- **Il codice PHP tra i tag PHP (in viola).**

L'interprete PHP elabora tutto il codice scritto all'interno dei tag PHP.

- **Il codice HTML all'esterno dei tag PHP (in bianco).**

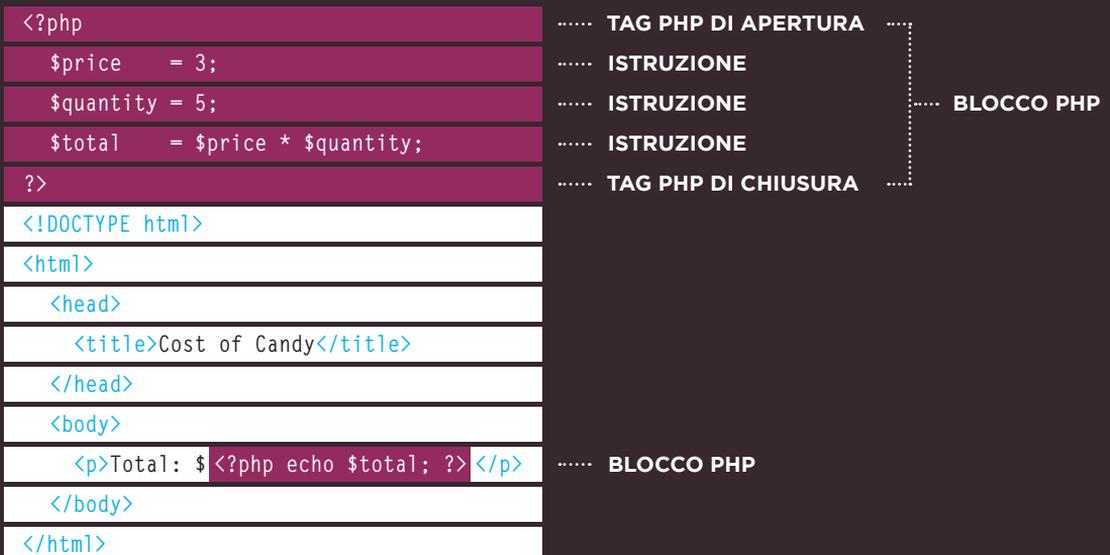
Viene aggiunto automaticamente al file HTML che viene inviato al browser (perché l'interprete PHP non ha bisogno di toccarlo in alcun modo).

Ogni singolo comando all'interno dei tag PHP è chiamato **istruzione**. La maggior parte delle istruzioni inizia su una nuova riga e termina con un punto e virgola. Dopo un'istruzione potete omettere il punto e virgola solo:

- nell'ultima riga di un blocco PHP;
- se il blocco PHP contiene solo un'istruzione.

Includere un punto e virgola alla fine di ogni istruzione aiuta comunque a evitare errori.

Questa pagina calcola il costo di 5 sacchetti di caramelle a \$3 ciascuno e memorizza il prezzo nella variabile `$total`. Quindi scrive il valore della pagina HTML. Ritroverete il codice PHP nel Capitolo 1.



PHP INVIA AL BROWSER IL TESTO E IL CODICE HTML

Il comando echo chiede all'interprete PHP di aggiungere testo o codice alla pagina HTML che sta creando per il browser.

Qualsiasi testo e/o codice HTML racchiuso tra virgolette dopo un comando echo viene inviato al browser, in modo che possa essere visualizzato nella pagina. Dopo il comando echo si possono utilizzare virgolette singole o doppie, ma le virgolette di apertura e chiusura devono **corrispondere**.

La prima virgoletta indica all'interprete PHP dove inizia il testo che deve aggiungere alla pagina; il secondo indica dove esso finisce. Il testo è noto come **stringa letterale**. Il punto e virgola alla fine della riga indica all'interprete PHP la fine dell'istruzione.

```
echo '<b>Hello!</b>';
```

IN VIA AL BROWSER TESTO E CODICE DA VISUALIZZARE

Per visualizzare una virgoletta *nel* testo che state inviando al browser, fatele precedere una barra rovesciata. In tal modo chiedete all'interprete PHP di non trattare come codice la virgoletta che segue. Questo si chiama **escape** delle virgolette.

Di seguito, il comando echo usa le virgolette per scrivere un link HTML. L'URL nell'attributo href deve essere tra virgolette, quindi quelle virgolette devono essere sottoposte a escape. Il codice seguente scrive il link HTML `PHP`.

```
echo "<a href=\"http://notes.re/php\">PHP</a>";
```

VIRGOLETTE DI APERTURA DEL COMANDO ECHO LE VIRGOLETTE CON ESCAPE OSPITANO L'ATTRIBUTO VIRGOLETTE DI CHIUSURA DEL COMANDO ECHO

Potete anche visualizzare le virgolette inserendo il testo e il codice HTML tra virgolette singole.

Tutto funziona perché l'interprete PHP cerca l'apice corrispondente per trovare la fine del testo.

```
echo '<a href="http://notes.re/php">PHP</a>';
```

VIRGOLETTE DI APERTURA DEL COMANDO ECHO ATTRIBUTO HTML TRA DOPPIE VIRGOLETTE VIRGOLETTE DI CHIUSURA DEL COMANDO ECHO

SCRIVERE IL CONTENUTO NELLA PAGINA

PHP

section_a/intro/echo.php

```
<!DOCTYPE html>
<html>
  <head>
    <title>echo Command</title>
    <link rel="stylesheet" href="css/styles.css">
  </head>
  <body>
    <h1>The Candy Store</h1>
    <h2><?php echo 'Ivy\s'; ?> page</h2>
    <?php echo '<p class="offer">Offer: 20% off</p>' ?>
  </body>
</html>
```

①

②

OUTPUT



Nel box di codice a sinistra:

- il percorso in alto a destra corrisponde al file nel codice disponibile al download;
- i numeri corrispondono ai passi descritti di seguito.

1. Il comando echo usa le virgolette singole per scrivere il nome dell'utente seguito da 's'. Il carattere barra rovesciata viene utilizzato per l'escape del ' tra il nome e la lettera s.

2. Il comando echo aggiunge un paragrafo alla pagina. L'elemento <p> ha l'attributo class.

Poiché il testo e il codice scritti sulla pagina sono inseriti tra virgolette singole, l'attributo HTML può utilizzare quelle doppie.

Sebbene, dopo il comando echo, sia possibile utilizzare sia le virgolette singole sia quelle doppie è meglio sceglierne una e basta. Questo libro utilizza principalmente le virgolette singole per contenere attributi HTML, come mostrato qui.

NOTA Se dopo un comando echo usate le virgolette, l'interprete PHP verifica se il testo contiene nomi di variabili (che vedremo alle pp.32-36). In tal caso, scriverà il valore contenuto nella variabile. Questo non avviene con le virgolette singole (p. 52).

PROVATE Nel Passo 1, sostituite il nome Ivy con il vostro nome e salvate il file. Quando aggiornerete la pagina, la frase di benvenuto utilizzerà il vostro nome.

COMMENTI

È buona norma aggiungere commenti che descrivono il codice PHP per ricordarvi che cosa fa il codice quando tornerete a lavorarvi dopo qualche tempo e per aiutare gli altri a comprenderne il funzionamento.

I **commenti a riga singola** iniziano con:

- due barre //
- il simbolo di cancelletto #.

Questi caratteri chiedono all'interprete PHP di ignorare anche il codice PHP di quella riga, fino a un tag di chiusura ?>.

```
echo "Welcome"; // Visualizza il saluto  
echo "Welcome"; # Visualizza il saluto
```

COMMENTO A RIGA SINGOLA

I **commenti su più righe** consentono di aggiungere descrizioni e/o note più dettagliate al codice.

Una barra e un asterisco /* indicano all'interprete PHP di ignorare tutto il testo fino al successivo asterisco seguito dalla barra */.

```
echo "Welcome";  
/*  
Dopo il messaggio di benvenuto:  
- Aggiungi la foto profilo accanto al nome  
- Crea un collegamento alla pagina del profilo  
*/
```

COMMENTO
SU PIÙ RIGHE

AGGIUNTA DI COMMENTI AL CODICE

PHP

section_a/intro/comments.php

```
1 <?php
  /*
  Questa pagina mostra il nome dell'iscritto
  e dettagli su un'offerta corrente
  */
  ?>
  <!DOCTYPE html>
  <html>
    <head>
      <title>Adding Comments to Your Code</title>
      <link rel="stylesheet" href="css/styles.css">
    </head>
    <body>
      <h1>The Candy Store</h1>
      2 <h2><?php echo 'Welcome Ivy'; // Mostra il nome ?></h2>
      <?php echo '<p class="offer">Offer: 20% off</p>' ?>
    </body>
  </html>
```

OUTPUT



Questo esempio è molto simile al precedente, ma aggiunge commenti al codice.

1. La pagina inizia con un commento su più righe che descrive il funzionamento del codice.

2. Dopo il messaggio di benvenuto, un commento a riga singola indica che cosa viene visualizzato.

I commenti non vengono aggiunti al codice HTML che viene inviato al browser; sono visibili solo nel codice PHP.

PROVATE Nel Passo 1, aggiungete un'altra riga di testo nel commento.

PROVATE Nel Passo 2, sostituite i caratteri della doppia barra con un # (simbolo di cancelletto).

NOTA In questo libro sono presenti molti commenti per aiutare a descrivere che cosa fanno le singole righe di codice degli esempi. I programmatori esperti raramente usano commenti riga per riga come facciamo in questo libro.

IN QUESTA PARTE

ISTRUZIONI DI BASE

1

VARIABILI, ESPRESSIONI E OPERATORI

Ogni volta che una pagina PHP esegue l'attività per cui è stata progettata, può usare valori diversi, quindi è importante rappresentare quei dati nel codice utilizzando **variabili**. Vedremo come utilizzare le **espressioni** e gli **operatori** per lavorare con questi valori.

2

STRUTTURE DI CONTROLLO

Una pagina PHP non eseguirà sempre nello stesso ordine le stesse righe di codice. Le **strutture di controllo** consentono di scrivere regole che l'interprete PHP utilizza per determinare quale riga di codice deve eseguire successivamente.

3

FUNZIONI

Le singole istruzioni necessarie per svolgere un'attività possono essere raggruppate con una **funzione**. Questo non solo aiuta a organizzare il codice, ma evita anche di ripetere le stesse istruzioni se la pagina deve svolgere un'attività più volte.

4

OGGETTI E CLASSI

Il codice viene usato per rappresentare concetti come gli iscritti al sito, i prodotti venduti e gli articoli visualizzati. I programmatori usano **oggetti** e **classi** per raggruppare il codice che rappresenta ciascuno di questi diversi concetti.