

INTRODUZIONE

Questo libro vi aiuterà a creare siti web utilizzando il linguaggio di programmazione PHP e ad archiviare in un database come MySQL i dati che il sito web utilizza.

PHP è un linguaggio di programmazione progettato per essere eseguito su un server web in modo tale che, quando viene richiesta una pagina web, il server possa generare una pagina HTML da inviare a quell'utente specifico. Ciò significa che le pagine possono essere adattate alle singole persone. Questo è un requisito per qualsiasi sito che consente agli utenti di eseguire le seguenti attività.

- **Registrarsi o accedere**, perché il nome, l'email e la password di ogni utente sono differenti.
- **Fare un acquisto**, perché i dettagli di ordine, pagamento e consegna di ogni cliente sono unici.
- **Eseguire ricerche nel sito web**, perché i risultati della ricerca sono personalizzati per ogni utente.

Il linguaggio PHP è stato progettato per funzionare con database come MySQL che possono memorizzare dati, come il contenuto delle pagine visualizzate dal sito, i prodotti venduti dal sito o le informazioni sugli iscritti al sito. Utilizzando PHP, imparerete a creare pagine web che consentano ai componenti di aggiornare i dati archiviati nel database. Alcuni esempi.

- **I sistemi di gestione dei contenuti** consentono ai proprietari del sito di aggiornare i contenuti del sito utilizzando un modulo e tali aggiornamenti vengono mostrati ai visitatori senza dover scrivere nuovo codice.
- **Gli shop online** consentono ai proprietari di elencare i prodotti in vendita e ai clienti di fare acquisti.
- **I social network** consentono ai visitatori di registrarsi e accedere, di creare profili utente, di caricare i propri contenuti e di visualizzare pagine su misura per i loro interessi.

Poiché le informazioni visualizzate su questi siti sono archiviate in un database, sono noti come **siti web basati su database**.

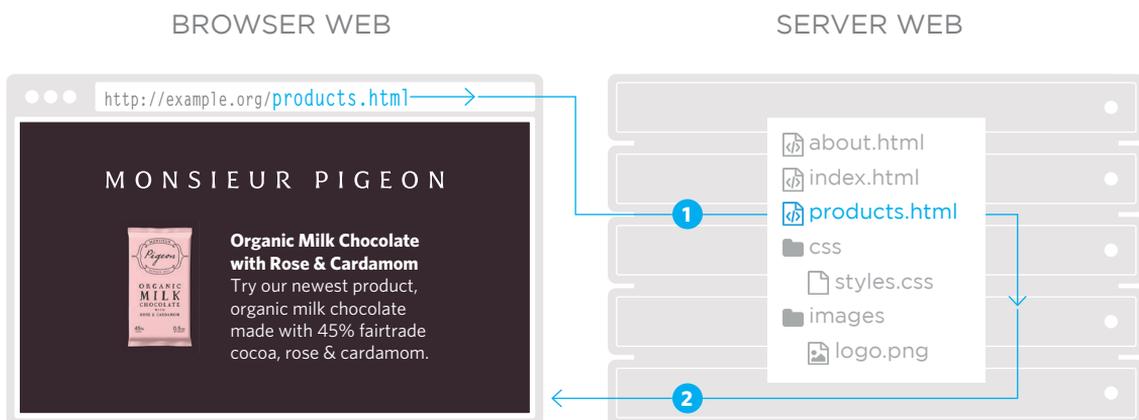
SITI WEB STATICI O DINAMICI

Quando un sito viene creato utilizzando solo codice HTML e CSS, ogni visitatore vede lo stesso contenuto, perché a tutti vengono inviati gli stessi file HTML e CSS.

1. Quando il browser richiede una pagina creata utilizzando file HTML e CSS, la richiesta viene inviata al **server web** che ospita il sito.
2. Il server web trova il file HTML richiesto dal browser e glielo invia. Può anche inviare del codice CSS per definire lo stile della pagina, dei media (come le immagini), del codice JavaScript e altri file utilizzati dalla pagina.

Poiché a tutti i visitatori vengono inviati gli stessi file HTML, vedranno tutti lo stesso contenuto. Pertanto, questo tipo di sito è noto come **sito web statico**.

Coloro che gestiscono siti web statici devono conoscere i linguaggi HTML e CSS. Se i proprietari vogliono aggiornare il testo di una pagina, il codice HTML deve essere aggiornato manualmente e ricaricato sul server web.



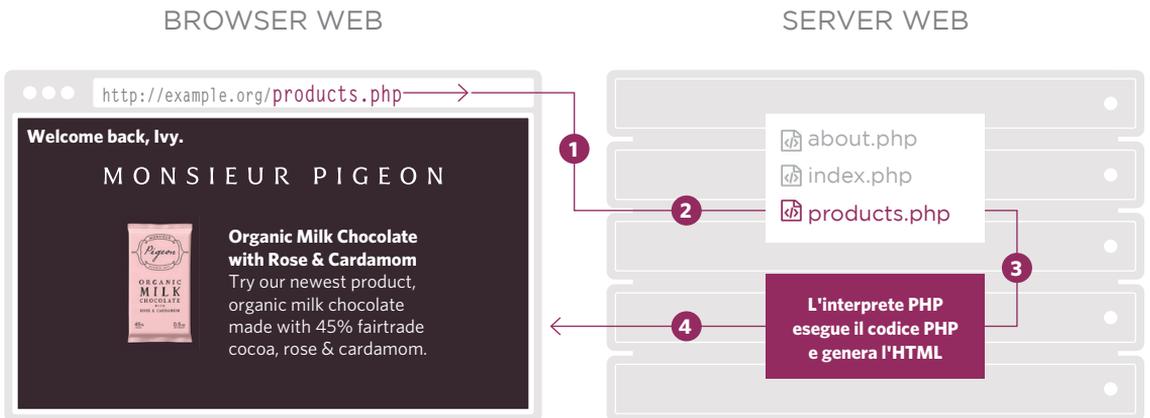
Questo libro presuppone la conoscenza dei linguaggi HTML e CSS. Abbiamo dedicato un libro a tale argomento: <http://htmlandcssbook.com>.

Quando i siti web vengono creati utilizzando PHP, ogni visitatore vede contenuti diversi, perché la pagina PHP crea il codice HTML che viene inviato.

Siti web come eBay, Facebook e i siti di news spesso visualizzano nuove informazioni a ogni accesso. Se visualizzate in un browser il sorgente delle pagine, vedrete il codice HTML, ma il programmatore non avrà aggiornato a mano il codice a ogni vostro accesso.

Questo tipo di sito è noto come **sito web dinamico** perché la pagina HTML che viene restituita all'utente viene creata utilizzando istruzioni scritte in un linguaggio come PHP.

1. Un browser richiede una pagina di un sito creato con PHP; la richiesta viene inviata al server web.
2. Il server web legge il file PHP.
3. Il codice PHP presente nel file viene eseguito tramite un software chiamato **interprete PHP** e viene creata una pagina HTML per quell'utente.
4. Il server web invia al suo browser la pagina HTML che ha creato per quell'utente. Non conserva una copia del file; alla richiesta successiva, creerà una nuova pagina HTML per quell'utente.



Il codice PHP non viene inviato al browser web, ma viene utilizzato per creare una pagina HTML la quale viene poi mandata al browser. Poiché il codice PHP viene eseguito sul server web, si parla di **programmazione lato server**.

PHP può essere utilizzato per creare pagine HTML su misura o **personalizzate** per ogni singolo visitatore. Ciò può includere la visualizzazione del nome del visitatore, degli argomenti cui è interessato o dei post dei suoi amici.

PHP: IL LINGUAGGIO E L'INTERPRETE

L'interprete PHP è un software che viene eseguito sul server web. Possiamo dirgli cosa fare usando codice scritto in linguaggio PHP.

Il software è ciò che ci permette di utilizzare un computer per fargli eseguire determinate attività senza sapere esattamente come le eseguirà. Alcuni esempi.

- I programmi di posta elettronica consentono di inviare e ricevere mail senza sapere come i computer le accettano e trasmettono.
- Photoshop consente di modificare le immagini senza conoscere come il computer le manipola.

Ogni volta che utilizzate un software, questo è in grado di eseguire le stesse attività, ma può eseguirle con dati diversi.

- Un programma di posta elettronica può essere utilizzato per creare, inviare, ricevere e archiviare messaggi, ma il contenuto e il destinatario di ogni messaggio possono essere diversi.
- Photoshop può applicare un filtro, ridimensionare o ritagliare un'immagine, e questo con qualsiasi immagine.

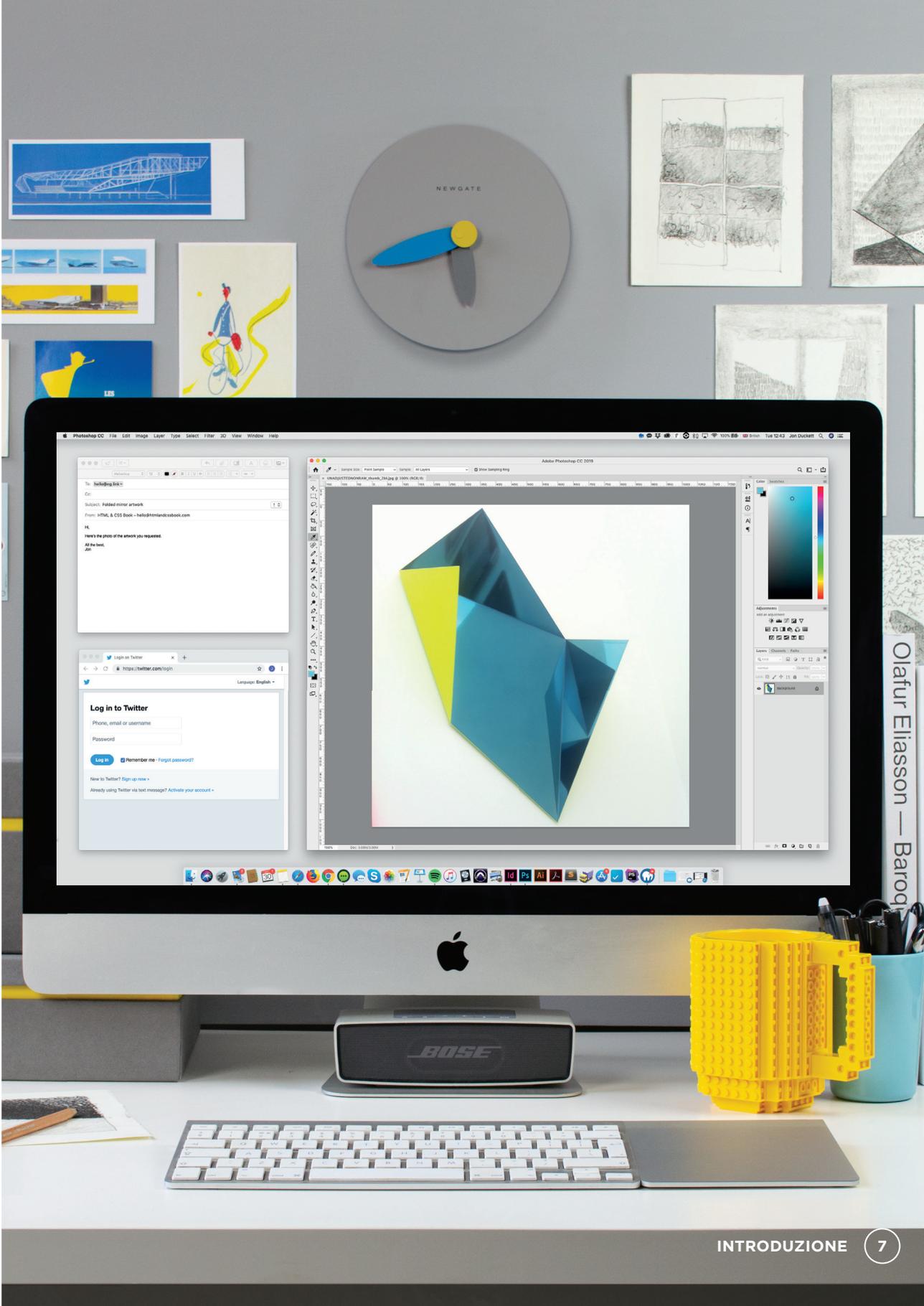
Entrambi questi software, per svolgere queste attività, hanno un'interfaccia grafica con cui l'utente può interagire.

Anche l'interprete PHP è un software, ed è parte di un server web. Ma, invece di usare un'interfaccia utente grafica, si può dire all'interprete PHP cosa deve svolgere usando del codice scritto nel linguaggio di programmazione PHP.

Quando si crea una pagina web usando il linguaggio PHP, questo eseguirà sempre le stesse attività utilizzando, ogni volta che viene richiesta la pagina, dati diversi. Per esempio, un sito scritto utilizzando PHP potrebbe avere:

- una pagina di accesso, che ogni utente utilizza per accedere al sito, anche se l'indirizzo e-mail e la password utilizzati per accedere sono diversi;
- una pagina dell'account, che ogni utente può utilizzare per visualizzare i dettagli del proprio account; anche se centinaia di persone utilizzano contemporaneamente la pagina, vedranno solo le proprie informazioni.

Ciò è possibile perché le istruzioni necessarie per eseguire queste attività sono sempre le stesse per ogni utente, ma i dati che ognuno fornisce o visualizza possono essere diversi.



Olafur Eliasson — Baroque

ESECUZIONE DI UN COMPITO CON DATI DIVERSI

I linguaggi di programmazione consentono di creare regole che indicano a un computer come svolgere un'attività. I dati utilizzati dal programma possono essere diversi ogni volta che l'attività viene eseguita.

Quando si usa un linguaggio di programmazione, bisogna dare al computer istruzioni precise, dicendogli esattamente che cosa deve fare. Queste istruzioni sono molto diverse da quelle che si potrebbero dare a una persona affinché esegua un compito.

Immaginate di voler acquistare cinque barrette di cioccolato e di doverne calcolare il prezzo totale. Per calcolarlo, moltiplicate il prezzo di una singola barretta per il numero di barrette che desiderate acquistare. Potreste esprimere questa regola nel modo seguente:

$$\text{totale} = \text{prezzo} \times \text{quantità}$$

Ecco alcuni esempi di calcolo del costo totale delle caramelle.

- Se una barretta di cioccolato costa \$1 e ne acquistate 5, il totale è \$5.
- Se il prezzo di una singola barretta fosse \$1,50, la regola è la stessa, ma il totale sarebbe \$7,50.
- Volendo acquistare 10 barrette a \$2 ciascuna, la regola è la stessa, ma il totale sarebbe \$20.

I valori utilizzati al posto delle parole *totale*, *prezzo* e *quantità* possono cambiare, ma la regola utilizzata per calcolare il costo totale delle caramelle resta la stessa.

Quando si usa codice PHP per creare una pagina web, occorre prima comprendere:

- il compito che volete ottenere;
- i dati, che possono cambiare ogni volta che viene eseguita un'attività.

Quindi potete fornire all'interprete PHP istruzioni dettagliate su come eseguire l'attività e utilizzate i nomi per rappresentare i valori che possono cambiare. Se avete detto all'interprete PHP che:

$$\begin{aligned}\text{prezzo} &= 3 \\ \text{quantità} &= 5\end{aligned}$$

potrete usare la regola:

$$\text{totale} = \text{prezzo} \times \text{quantità}$$

La parola *totale* rappresenterebbe un valore di 15. La volta successiva che eseguirete la pagina, potrete dare a prezzo o quantità valori diversi e potrete calcolare il nuovo totale usando la stessa regola.

I programmatori chiamano le parole che rappresentano valori col nome di **variabili**, perché il valore che rappresentano può variare a ogni esecuzione del programma.

totale = prezzo x quantità



$$\$9 = 3 \times 3$$

CHE COS'È UNA PAGINA PHP?

Spesso una pagina PHP contiene una combinazione di codice HTML e PHP. Può essere utilizzata per inviare una pagina HTML al browser.

Sotto, a sinistra, potete vedere una pagina PHP che contiene un mix di codice HTML e PHP:

- il codice HTML è in blu;
- il codice PHP è in viola.

Quando l'interprete PHP apre il file, questo:

- copia tutto il codice HTML direttamente in un file HTML temporaneo che crea per quell'utente;
- poi segue qualsiasi istruzione scritta in codice PHP (che spesso genera contenuti per la pagina HTML).

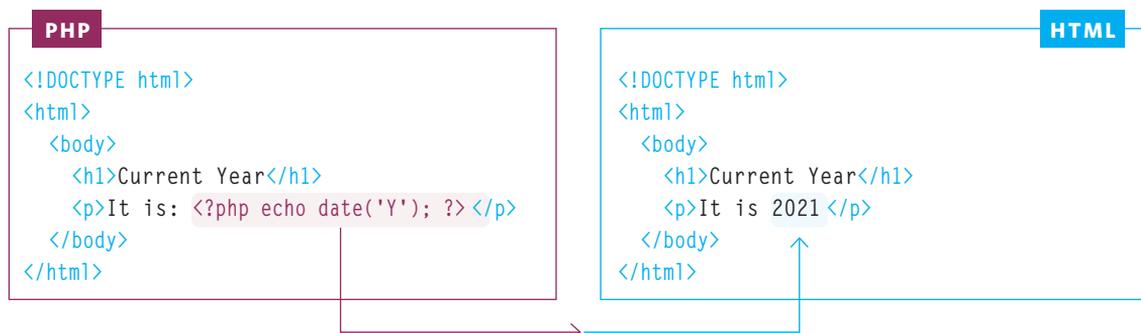
Il codice PHP che segue determina l'anno corrente e lo scrive all'interno dei tag di apertura `<p>` e chiusura `</p>`.

Il codice PHP può eseguire attività semplici, come operazioni aritmetiche o ottenere la data corrente, o più complesse, come utilizzare le informazioni inviate tramite un modulo HTML per aggiornare i dati archiviati in un database.

Quando l'interprete PHP ha terminato l'elaborazione del file PHP, invia al browser la pagina HTML temporanea creata per quell'utente, quindi elimina la pagina HTML temporanea.

Sotto, potete vedere la pagina HTML che verrebbe inviata al browser una volta che l'interprete PHP avrà eseguito il codice PHP.

L'interprete PHP ha determinato l'anno in corso e lo ha visualizzato nella pagina HTML che ha creato.



L'interprete PHP determina l'anno corrente e lo scrive all'interno dei tag di paragrafo.

Di solito ogni pagina esegue la stessa attività a ogni richiesta, ma può funzionare con informazioni diverse per ogni utente del sito.

Un sito web PHP è costituito da un insieme di pagine PHP e ogni pagina svolge un'attività specifica. Per esempio, un sito che consente l'accesso agli iscritti potrebbe avere:

- una pagina di accesso – per consentire agli iscritti l'accesso al sito;
- una pagina del profilo – per consentire agli iscritti di visualizzare il proprio profilo.

Ogni volta che viene richiesta una di queste pagine, deve essere in grado di elaborare dati specifici del componente corrente. Pertanto, la pagina deve:

- contenere istruzioni su come completare l'attività che la pagina dovrebbe svolgere;
- assegnare un nome a ogni dato che potrebbe cambiare ogni volta che viene richiesta la pagina.

In PHP, le variabili usano un nome per rappresentare un valore che può cambiare ogni volta che viene richiesta la pagina.

Il codice PHP può dire all'interprete PHP:

- quale nome di variabile usare per un dato che può cambiare *ogni* volta che viene richiesta la pagina;
- quale valore usare a *questa* richiesta di pagina.

Una volta che la pagina HTML è stata inviata all'utente, l'interprete PHP dimentica tutti i valori memorizzati nelle variabili, quindi può eseguire la stessa operazione (ma con valori diversi) per la persona successiva che richiede la pagina.

Per archiviare i dati più a lungo, basta memorizzarli in un database, per esempio MySQL, come vedremo nella prossima pagina.

```
PHP
<!DOCTYPE html>
<html>
  <body>
    <h1>My Profile</h1>
    <p>Name: <?php echo $username; ?></p>
  </body>
</html>
```

```
HTML
<!DOCTYPE html>
<html>
  <body>
    <h1>My Profile</h1>
    <p>Name: Ivy Stone</p>
  </body>
</html>
```

L'interprete PHP ottiene il valore memorizzato nella variabile \$username e lo scrive all'interno dei tag di paragrafo.

CHE COS'È MYSQL?

MySQL è un tipo di database. I database archiviano i dati in modo strutturato, in modo da potervi accedere facilmente e aggiornare le informazioni in essi contenute.

I fogli di lavoro come Excel memorizzano le informazioni in una griglia di colonne e righe. I programmi possono quindi utilizzare tali dati per eseguire calcoli o manipolarli utilizzando formule.

MySQL è un software che memorizza le informazioni in modo simile; in **tabelle** composte da colonne e righe. È possibile utilizzare PHP per accedere e aggiornare le informazioni memorizzate nel database.

Un database può contenere più tabelle. Ogni tabella di solito contiene un tipo di dati che il sito deve memorizzare. Di seguito sono riportati due esempi di tabelle di database che contengono:

- gli iscritti (o utenti) di un sito;
- gli articoli visualizzati dal sito.

All'interno di ogni tabella, i **nomi delle colonne** descrivono il tipo di informazioni che ciascuna colonna della tabella contiene:

- la tabella `member` memorizza il nome, il cognome, l'indirizzo e-mail, la password, la data di iscrizione e l'immagine del profilo di ciascun iscritto;
- La tabella `article` memorizza il titolo, il riepilogo, il contenuto, la data di creazione di ciascun articolo e alcuni valori discussi nella pagina a destra.

Ogni **riga** contiene i dati descrittivi di un oggetto rappresentato nella tabella:

- tabella `member` – ogni riga rappresenta un iscritto;
- tabella `article` – ogni riga contiene un articolo.

NOME TABELLA	NOME COLONNA				COLONNA			
member	<code>id</code>	<code>forename</code>	<code>surname</code>	<code>email</code>	<code>password</code>	<code>joined</code>		<code>picture</code>
	1	Ivy	Stone	ivy@eg.link	\$2y\$10\$MAdTTCA0Mi0w	2021-01-01	20:28:47	ivy.jpg
	2	Luke	Wood	luke@eg.link	\$2y\$10\$NN5HEAD3atar	2021-01-02	09:17:21	NULL
	3	Emiko	Ito	emi@eg.link	\$2y\$10\$/RpRmiUMStji	2021-01-02	10:42:36	emi.jpg

article	<code>id</code>	<code>title</code>	<code>summary</code>	<code>content</code>	<code>created</code>	<code>category_id</code>	<code>member_id</code>	<code>image_id</code>	<code>published</code>
	1	Systemic	Brochure	<p>This	2021-01-01	1	2	1	1
	2	Polite	Poster	<p>These	2021-01-02	1	1	2	1
	3	Swimming	Architect	<p>This	2021-01-02	4	1	3	1

RIGA

Usando PHP, potete svolgere le seguenti operazioni.

- **Trarre dati dal database e visualizzare** queste informazioni in una pagina web.
- **Aggiungere nuove righe di dati.** Per creare un nuovo articolo, aggiungete una riga alla tabella `article` e fornite i dati che devono essere archiviati in ciascuna colonna.
- **Cancelare righe di dati.** Per eliminare un articolo, rimuovete l'intera riga che rappresenta l'articolo.
- **Modificare i dati di una riga.** Per aggiornare l'indirizzo email di un iscritto, dovete trovare la riga della tabella `member` che lo rappresenta, quindi aggiornare il valore nella colonna `email`.

Notate come entrambe le tabelle inizino con una colonna `id`. Ogni riga della tabella ha, in questa colonna, un valore univoco (che inizia da 1 e per ogni riga viene incrementato di 1). I valori nella colonna `id` consentono di indicare al database con quale riga di dati si desidera lavorare. Per esempio, potreste volere l'iscritto il cui `id` è 2 o l'articolo in cui `id` è 1.

MySQL è un **database relazionale** perché conserva le relazioni tra i tipi di dati archiviati nelle diverse tabelle. Nelle tabelle sottostanti, per esempio, gli articoli sono stati inseriti da diversi utenti iscritti al sito. Nella tabella `article`, il valore nella colonna `member_id` dice quale utente ha inserito l'articolo perché contiene un numero che corrisponde a uno dei valori nella colonna `id` della tabella `member`.

L'articolo con `id` 1 è stato inserito dall'iscritto il cui `id` è 2 (Luke Wood). Gli articoli 2 e 3 sono stati inseriti dall'utente con `id` 1 (Ivy Stone).

Queste relazioni:

- strutturano i dati, assicurando che ogni tabella contenga solo un particolare tipo di dati (iscritto o articolo);
- evitano al database di memorizzare gli stessi dati in più tabelle (con un risparmio di spazio nel database);
- semplifica l'aggiornamento dei dati. Se un iscritto cambia nome, deve essere aggiornato solo nella tabella `member` (non in ogni articolo che ha inserito).

member						
id	forename	surname	email	password	joined	picture
1	Ivy	Stone	ivy@eg.link	\$2y\$10\$MAdTTCA0Mi0w	2021-01-01 20:28:47	ivy.jpg
2	Luke	Wood	luke@eg.link	\$2y\$10\$NN5HEAD3atar	2021-01-02 09:17:21	NULL
3	Emiko	Ito	emi@eg.link	\$2y\$10\$/RpRmiUMStji	2021-01-02 10:42:36	emi.jpg

article								
id	title	summary	content	created	category_id	member_id	image_id	published
1	Systemic	Brochure	<p>This	2021-01-01	1	2	1	1
2	Polite	Poster	<p>These	2021-01-02	1	1	2	1
3	Swimming	Architect	<p>This	2021-01-02	4	1	3	1

STORIA DI PHP

Come per la maggior parte dei software, esistono molte versioni di PHP e MySQL. Le versioni più recenti offrono più funzionalità e sono più veloci rispetto alle versioni precedenti.

PHP è stato creato da Rasmus Lerdorf nel 1994, che ha poi rilasciato il suo codice al pubblico nel 1995, incoraggiando gli utenti a migliorarlo. A quel tempo, la sigla stava per Personal Home Page; ora sta per PHP: Hypertext Processor.

PHP è ora utilizzato sull'80% dei siti web che utilizzano un linguaggio di programmazione per il server.

Siti come Facebook, Etsy, Flickr e Wikipedia sono stati tutti sviluppati per la prima volta in PHP (sebbene attualmente alcuni utilizzino anche altre tecnologie).

I software open source più noti, come WordPress (che alimenta oltre il 35% dei siti web del mondo), Drupal, Joomla e Magento, sono tutti scritti in PHP. Imparare PHP vi aiuterà a lavorare con essi.

A ogni nuova versione di PHP, vengono aggiunte nuove funzionalità. Questo libro tratta le funzionalità fino a PHP 8 incluso, rilasciato a novembre 2020.

.....	1995
..... PHP 1	1996
.....	1997
..... PHP 2	1998
..... PHP 3	1999
.....	2000
..... PHP 4	2001
.....	2002
.....	2003
.....	2004
..... PHP 5	2005
.....	2006
..... PHP 5.1	2007
..... PHP 5.2	2008
.....	2009
..... PHP 5.3	2010
.....	2011
..... PHP 5.4	2012
.....	2013
..... PHP 5.5	2014
..... PHP 5.6	2015
.....	2016
..... PHP 7	2017
..... PHP 7.1	2018
..... PHP 7.2	2019
..... PHP 7.3	2020
..... PHP 7.4	2021
..... PHP 8	

STORIA DI MYSQL

1995	MYSQL 1
1996	
1997	MYSQL 3.2
1998	
1999	PHPMYADMIN
2000	
2001	
2002	
2003	MYSQL 4
2004	
2005	
2006	MYSQL 5
2007	
2008	SUN ACQUISTA MYSQL
2009	MYSQL 5.1
2010	MARIADB ORACLE ACQUISTA SUN
2011	MYSQL 5.5
2012	
2013	MYSQL 5.6
2014	
2015	
2016	MYSQL 5.7
2017	
2018	MYSQL 8
2019	
2020	
2021	

La prima release di MySQL risale al 1995. Le lettere SQL stanno per Structured Query Language, il linguaggio utilizzato per manipolare le informazioni dei database relazionali.

MySQL è stato sviluppato dalla società svedese MySQL AB, che ha reso disponibile MySQL gratuitamente. Michael Widenius, uno dei creatori di MySQL, ha scelto questo nome in onore di sua figlia, My.

MySQL AB è stata poi venduta a Sun Microsystems nel gennaio 2008 e successivamente Oracle ha acquisito Sun nel 2010.

Quando gli sviluppatori di MySQL hanno appreso che Oracle stava per acquisire Sun (e quindi MySQL), temevano che potesse smettere di essere gratuito, quindi hanno creato una versione open source del database, MariaDB (dal nome della figlia minore dello stesso fondatore, Maria).

MySQL e MariaDB sono impiegati da siti come Facebook, YouTube, Twitter, Netflix, Spotify e Wordpress.

phpMyAdmin è uno strumento di amministrazione dei database MySQL e MariaDB. È stato rilasciato nel 1998 come strumento gratuito per facilitare la gestione dei database MySQL, ma funziona anche con MariaDB.

Il codice in questo libro è utilizzabile con MySQL versione 5.5 o MariaDB 5.5 e successive e con phpMyAdmin.

L'ultima versione di MySQL (al momento della pubblicazione di questo libro) è la 8. MySQL 6 non è mai stato rilasciato e la versione 7 non viene trattata perché è stata progettata per essere eseguita su cluster di server, non su personal computer.

ARGOMENTI TRATTATI NEL LIBRO

Questo libro è diviso in quattro parti.
Ecco una panoramica degli argomenti di ogni parte.

A: PROGRAMMAZIONE DI BASE

La prima parte illustra come utilizzare il codice PHP per scrivere istruzioni eseguibili dall'interprete PHP:

- istruzioni di programmazione di base;
- esecuzione di codice diverso in situazioni diverse (per esempio, eseguendo del codice se un utente ha eseguito l'accesso e altro codice in caso contrario);
- come le funzioni possono raggruppare tutto il codice necessario per eseguire una singola attività;
- come le classi e gli oggetti aiutano a organizzare il codice utilizzato per rappresentare le cose del mondo che ci circonda.

B: PAGINE WEB DINAMICHE

La seconda parte del libro introduce una serie di strumenti di PHP che consentono di creare pagine web dinamiche:

- raccogliere i dati inviati da un browser web;
- verificare che gli utenti abbiano fornito i dati necessari a una pagina e nel formato corretto;
- lavorare con tutti i dati inviati;
- elaborare i file che gli utenti hanno caricato;
- rappresentare date e ore in PHP;
- memorizzare temporaneamente i dati in cookie e sessioni;
- risolvere i problemi nel vostro codice.

C: SITI WEB BASATI SU DATABASE

La terza parte illustra come trarre dati da un database e mostrarli nelle pagine web, oltre ad aggiornare i dati archiviati nel database:

- come i database archiviano i dati;
- come impiegare il linguaggio SQL per recuperare o aggiornare i dati contenuti in un database;
- come visualizzare i dati del database in una pagina PHP;
- come utilizzare moduli HTML per consentire ai visitatori di aggiornare i dati memorizzati nel database.

D: ESTENSIONE DELL'APPLICAZIONE

La quarta parte mostra le tecniche pratiche per la creazione di siti web e applicazioni web in PHP. L'applicazione di esempio è un semplice sistema di gestione dei contenuti con funzionalità social:

- migliorare la struttura del codice;
- incorporare codice di altri programmatori;
- inviare e-mail con PHP;
- consentire agli utenti di registrarsi e accedere al sito;
- creare pagine su misura per i singoli iscritti;
- usare URL ottimizzati per i motori di ricerca;
- aggiungere funzionalità social (Mi piace e commenti).