

A photograph of a silver laptop on a desk. The laptop is open, and its screen is dark. The keyboard is visible. The background is dark and out of focus. The text "Php per neofiti" is overlaid on the top half of the image.

Php per neofiti

Php per neofiti

Antonio Mercurio

Php le basi: per principianti

Facile da capire

antonio mercurio

Questo libro è in vendita presso <http://leanpub.com/phplebasiperprincipianti>

Questa versione è stata pubblicata il 2016-08-09



This is a [Leanpub](#) book. Leanpub empowers authors and publishers with the Lean Publishing process. [Lean Publishing](#) is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

© 2016 antonio mercurio

Tweet This Book!

Please help antonio mercurio by spreading the word about this book on [Twitter](#)!

The suggested tweet for this book is:

[Ho appena acquistato Php le basi: per principianti](#)

The suggested hashtag for this book is [#php_lebasi](#).

Find out what other people are saying about the book by clicking on this link to search for this hashtag on Twitter:

https://twitter.com/search?q=#php_lebasi

Dedico questo libro a tutti coloro che ogni giorno in silenzio si isolano per lasciare spazio all'unico suono delle dita che scorrono veloci sulla tastiera per offrirci software sempre migliore ed utile.

Indice

Part1	i
Introduzione	ii
Cosa serve per questo libro	iii
Condivi il libro	iv
Supporto	v
Come inserire codice php	1
Il comando echo	3
Le variabili, stringhe e numeri	4
Come effettuare comparazioni	5
Gli operatori logici	6

Part1

Introduzione

Grazie carissimo lettore per aver acquistato questo libro. Se non hai mai programmato in php per te questo è il luogo giusto per imparare le basi. Il linguaggio sarà semplice e con degli esempi per capire il codice. Io mi chiamo Antonio Mercurio e sono un appassionato di tecnologie opensource ed in particolare dei linguaggi orientati allo sviluppo web. Spero che da questo libro imparerai a conoscere questo linguaggio e a muovere i primi passi. Php è un linguaggio server-side cioè consente di dialogare con un server remoto sul quale abbiamo un nostro database contenente i nostri dati come i dati anagrafici degli utenti, indirizzi e qualsiasi cosa debba essere conservata ed utilizzata quando serve. Per poter aggiornare i dati di un database usiamo il codice php in congiunzione con il linguaggio sql, ch  il linguaggio che contiene comandi che inseriti in query(interrogazioni) consentono di aggiornare, inserire, cancellare dati dal database. Uno dei linguaggi che ci consente di interagire con sql eseguendo delle query   php. Php   anche un linguaggio pseudo-oggetti cio  consente di programmare secondo la tipologia definita procedurale, ossia eseguendo le istruzioni una dopo l'altra oppure utilizzare il paradigma della programmazione moderna detta ad oggetti. In questo libro vedremo solo il metodo procedurale.

Cosa serve per questo libro

Per questo libro hai bisogno solo di comprendere le basi html perchè durante le nostre spiegazioni faremo uso di questa sintassi ed è meglio saperne qualcosa. Inoltre potrai seguire le mie spiegazioni su qualunque sistema operativo. Avrai bisogno di un editor che può essere qualunque editor opensource. Ti consiglio komodo edit, atom, che sono multiplatforma e notepad+++ per Windows. Io personalmente uso PHP storm che è un ide e cioè un sistema di sviluppo integrato che offre molte funzionalità avanzate ma il software è a pagamento. Inoltre dovreste installare in locale un server funzionante con supporto php, mysql e apache. Vi consiglio su mac il pacchetto MAMP, mentre per Windows e Linux va bene Xampp. Il primo lo trovate sul sito seguente:

MAMP¹, mentre il secondo qui: [xampp](http://apachefriends.org)².

In questo libro non spiego l'uso di un editor anche perchè siete liberi di scegliere quale desiderate e neanche di Xampp e MAMP per i quali ci sono le pagine di supporto sui siti ufficiali molto semplici.

¹<http://www.mamp.info>

²<http://apachefriends.org>

Condivi il libro

Se il libro ti è piaciuto condividilo sui social ed inoltre per twitter puoi menzionarmi usando @tonyubuntu.

Supporto

Se hai problemi di qualunque genere o vuoi fornirmi suggerimenti scrivimi su tonymercurio75@gmail.com.

Come inserire codice php

Aprirete il vostro editor e create una nuova cartella e chiamatela per esempio test e al suo interno create un file con estensione php e denominatelo test.html e il vostro editor aggiunge in automatico l'estensione php e quindi alla fine avrete un file di nome test.html.php. Ora al suo interno inserite i tags html, cioè dovrete avere una struttura della pagina come la seguente.

```
1  <!DOCTYPE html>
2  <html lang="it">
3  <head>
4      <title>La mia pagina di prova</title>
5      <meta charset="utf-8">
6
7  </head>
8
9  <body>
10
11
12  <?php
13
14
15
16
17
18
19
20
21  ?>
22
23  </body>
24
25  </html>
```

Questi sono i normali tags html che servono per racchiudere codice puro html che viene scritto tra i due tags body.

Ora il codice php si inserisce all'interno dell'area body della pagina html tra due tags php che sono i seguenti <?php e ?>.

All'interno di questi due tags il codice php verrà interpretato dall'interprete php. Il vostro programma di navigazione interpreta prima il codice html e poi quello di qualsiasi altro linguaggio. Fra i due tags php inserite il seguente codice

```
echo "ciao mondo";
```

Al di fuori dei tags php e quindi tra <body> e </body> scrivete **Ciao mondo** e salvate ed aprite la pagina nel vostro programma di navigazione. Dovreste vedere due frasi Ciao mondo. Per il momento non spieghiamo il comando echo che abbiamo inserito ma dovete solo sapere che il codice php si deve inserire fra i due tags php di cui sopra. Può essere inserito anche in un file con solo codice php. Create un nuovo file nel vostro editor con estensione php e lo chiamate solo test ed inserite il seguente codice e quindi avrete una pagina simile

```
1  <?php
2
3  $n=15;
4
5  $n1=15.2;
6
7  $n2='12';
8
9  echo gettype($n). "<br />";
10
11 echo gettype($n1). "<br />";
12
13 echo gettype($n2);
```

In pratica si tratta di un file php che contiene solo codice php. I>Quando create un file con estensione php e che contiene solo codice php potete tralasciare il tag ?> di chiusura che non è obbligatorio. Non lo abbiamo detto ma in locale le pagine php devono essere inserite nella cartella del server che se usate xampp si chiama htdocs ed anche se usate mamp. Come possiamo richiamare questo nostro file test.php?. Siccome questo libro presuppone che siate in locale sul vostro pc dovete digitare il seguente indirizzo sul vostro browser localhost/cartellacorso/test.php ed avrete la scritta **ciao mondo**. Su mac con l'uso di Mamp dovete invece digitare di norma anche il nome della porta su cui si avvia il server e cioè localhost:8888/cartellacorso/test.php. In pratica abbiamo richiamato sempre la pagina appena creata usando solo codice php racchiuso fra i tags <?php e ?>. In sostanza abbiamo creato una pagina php pura senza codice html. Il risultato è lo stesso ma spesso ci troveremo a creare pagine pure php e a richiamarle.

Ovviamente cartellacorso va sostituita con il nome della cartella del progetto che abbiamo creato in partenza.

Il comando echo



In php ogni riga di codice termina con un ;

Immaginiamo di voler stampare un messaggio a video, oppure il risultato di un'espressione. Per farlo usiamo una funzione interna di php definita

echo. Ecco un esempio che potete inserire in una pagina di prova che chiamate echo.php.

```
echo 'ciao mondo';
```

Abbiamo inserito un messaggio che poi è una stringa alfabetica all'interno di singoli apici. L'interprete php leggerà il contenuto all'interno e

tramite la funzione echo lo stamperà. Possiamo inserire un testo anche all'interno di doppi apici e il risultato non cambia.

```
'echo "ciao mondo";
```

In pratica abbiamo usato una funzione interna di php echo che stampa a video una frase. Possiamo provare a far stampare il risultato di un calcolo.

```
echo 3*2;
```

Vi darà risultato 6.

Ricordate sempre il punto e virgola alla fine di ogni riga di codice. Ovvio che possiamo anche stampare una espressione matematica avendo in stampa il risultato. Per esempio la seguente

```
1 <?php
2
3 echo (5*4)+2;
```

se provate a lanciare la pagina dovrebbe fornirvi in stampa il risultato delle espressioni come se fosse una calcolatrice.

Le variabili, stringhe e numeri

Le variabili sono contenitori logici che racchiudono dei tipi di dati. I tipi di dati possono essere stringhe, numeri, booleani, ed altri. Immaginiamo di voler racchiudere all'interno di un contenitore un nome di persona. I nomi sono una sequenza di caratteri alfabetici che possiamo salvare all'interno appunto di una variabile. Per dichiarare una variabile in php si usa il simbolo del dollaro seguito dal nome che vogliamo attribuire al nostro contenitore. Supponiamo di voler racchiudere il mio nome all'interno della variabile chiamata nome allora in php scriviamo `$nome="tony"`; Ricordate che ogni riga di codice php termina sempre con il punto e virgola. Questa che abbiamo appena creato è una variabile di tipo stringa anche se solo alfabetica mentre le stringhe possono contenere anche numeri e simboli. Per esempio racchiudiamo nella stringa di nome stringa una frase in questo modo

```
$stringa="ciao mondo da @tonyubuntu";
```

Come vedete abbiamo appena creato una stringa di nome stringa che contiene una frase con caratteri e simboli. Immaginiamo invece di voler salvare in memoria una variabile di tipo numerico ecco come potremmo definirla

```
$numero=17;
```

Come vedete la variabile numerica si chiama numero ed il suo contenuto è 17 e quindi un numero intero. In effetti i numeri possono essere anche a virgola mobile detti double e si dichiarano così

```
$numero=17.5;
```

Per separare i decimali si usa il punto. Abbiamo visto quindi numeri, stringhe ma esistono anche i booleani cioè valori che possono assumere lo stato di vero o falso cioè true oppure false. Potremmo usare queste variabili quando dobbiamo verificare tramite confronto che una espressione sia vera o falsa quindi true oppure false. Possiamo dichiarare i booleani così `$true=true`;

`$false=false`; La prima espressione è vera mentre la seconda falsa. In pratica per creare una variabile usiamo il simbolo del dollaro seguito dal nome della variabile che deve iniziare sempre con una lettera minuscola o maiuscola che sia o con un simbolo `_` ma mai con un numero. Poi segue il simbolo di uguale che si definisce assegnazione, perchè serve ad assegnare un valore alla variabile. Se non assegnate un valore ad una variabile si dice non inizializzata e se cercherete di stamparla vi darà messaggio di errore.

Come effettuare comparazioni

In php possiamo usare degli operatori di comparazione per effettuare operazioni di confronto. Per esempio dichiariamo due variabili numeriche e vogliamo confrontare se sono uguali, oppure se una è maggiore dell'altra. Dichiariamo due variabili numeriche come segue

```
1 $n1=12;  
2 $n2=15;
```

poi lanciamo questo comando per vedere se n1 è maggiore di n2.

```
echo ($n1>$n2);
```

Vedrete una pagina vuota. In pratica essendo falso che n1 sia maggiore di n2 non vi stamperà nulla. Mentre se scriviamo così

```
echo ($n1<$n2);
```

vi darà come risultato 1 che vuol dire vero. In pratica abbiamo usato gli operatori di comparazione minore (<) e maggiore (>). Se volessimo sapere invece se n1 sia uguale ad n2 dobbiamo scrivere così

```
echo ($n1==$n2);
```

Quindi abbiamo usato il simbolo di uguaglianza. Per sapere se n1 sia diverso da n2 dobbiamo usare l'operatore !=. In pratica il punto esclamativo indica negazione e quindi sta a significare che non è uguale, come se fosse not = che vuol dire non uguale. Quindi scriviamo

```
echo $n1!=$n2;
```

dovrebbe fornirvi come risultato 1. Infatti è vero che n1 sia diverso da n2. Esiste anche un altro operatore di paragone ed è === che vuol dire equivalente. Cioè possiamo stabilire se per esempio n1 sia uguale come valore ed identico come tipo di variabile. Nel nostro esempio di prima vediamo di modificare le due variabili numeriche così

```
1 $n1=12;  
2 $n2=12;  
3  
4 echo $n1=== $n2;
```

dovrebbe fornirvi vero cioè 1. Questo perchè n1 è un intero e vale 12 come anche n2.

In pratica equivalenza vuol dire essere identici come tipo di variabile e identici come valore. Ovviamente se volessimo capire se n1 sia non identico allora usiamo l'operatore !== che vuol dire non identico.

Gli operatori logici

Gli operatori logici sono gli stessi che conosciamo dalla matematica e sono AND e OR. L'operatore and vuol dire e mentre OR oppure. Scriviamo il seguente codice

```
1 $n1=10;  
2 $n2=11;  
3  
4 echo $n1<20 and $n2>10;  
5 echo $n1<20 or $n2>1;
```

Se caricate la pagina vi darà in entrambi i casi risultato 1 in quanto entrambe le espressioni sono vere. Analizziamo il codice. La prima riga e la seconda definiscono due nuove variabili di valore 10 e 11 e nella terza riga stiamo chiedendo all'interprete php di dirci se n1 è minore di 20 e allo stesso tempo n2 sia maggiore di 1. In effetti devono verificarsi entrambe le condizioni per essere vero, e lo sono perchè n1 vale 10 ed è minore di 20 e allo stesso tempo n2 vale 11 ed è maggiore di 1. Mentre la seconda espressione chiede all'interprete php se n1 sia minore di 20 oppure n2 sia maggiore di uno. Quindi basta che si verifichi o l'una o l'altra. Inoltre sappiate che l'operatore and si può scrivere anche && e l'operatore or così ||.