

PHP, il primo script

Utilizzando le tecnologie Open Source e la vostra piattaforma Linux, ecco come immergersi nel popolare mondo della programmazione con il linguaggio PHP, tra i più diffusi nel Web



La nascita di PHP si colloca nel lontano 1995 quando il suo creatore Rasmus Lerdorf cominciò a lavorare a un sistema di script chiamato **Personal Home Page** (PHP). L'improvvisa domanda per gli script spinse Rasmus a sviluppare ulteriormente il linguaggio e nel 1997 venne rilasciata la versione 2.0 con molti miglioramenti compiuti da programmatori di tutto il mondo. La versione 2.0 divenne rapidamente popolare e spinse un team di sviluppatori ad affiancare Rasmus nell'ulteriore sviluppo del linguaggio. La versione 3.0 del 1997 vide una riscrittura del motore di parsing, e nel 1998 fu stimato che più di 50.000 utenti sfruttavano PHP nelle loro pagine Web. Tale versione comportò anche la modifica del nome in quello che conoscete oggi: PHP: **Hypertext Preprocessor**. Avanti veloce di un anno e, con una base stimata di un milione di utenti, PHP stava diventando uno dei linguaggi più popolari del mondo. Lo sviluppo continuava a un ritmo incalzante con migliaia di funzioni introdotte. Due degli sviluppatori principali, Zeev Suraski e Andi Gutmans, decisero di ripensare il modo in cui PHP operava e riscrissero ancora una volta il motore di parsing, rilasciandolo nella versione 4.0 come **Zend** scripting engine. Pochi mesi dopo il rilascio della 4.0, Netcraft stimò le installazioni di PHP in 3,6 milioni di domini. La versione 4.0 rappresentò un deciso salto in avanti a livello di programmazione e in ambito aziendale,

ma il linguaggio aveva ancora i suoi svantaggi, principalmente dovuti alla giovane età. La versione 5.0 venne rilasciata al mondo nel 2004, e con questa arrivò una miriade di miglioramenti che portarono il linguaggio a maturità e comportarono un picco di installazioni: si pensa che PHP al giorno d'oggi sia usato in più di 20 milioni di domini, ed è riportato come il modulo Apache più popolare, disponibile in almeno il 54% delle installazioni globali del Web server. La versione 6.0 si avvicina al rilascio pubblico, e migliorerà ulteriormente il supporto, le funzionalità e la maturità del linguaggio. Con siti come Wikipedia, Facebook, Flickr e Digg basati su PHP, non serve chiedersi da dove derivi la sua popolarità; vi mostreremo quindi com'è semplice iniziare con questo linguaggio di scripting dinamico.

“La versione 6 di PHP si avvicina al rilascio e migliorerà la maturità del linguaggio”

Setup e installazione

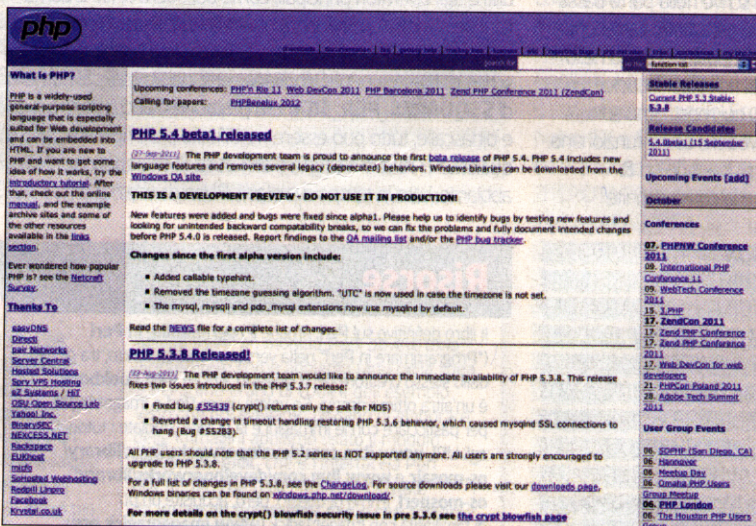
La maggior parte delle ultime distribuzioni Linux contengono già PHP, quindi in questo tutorial

si assumerà che sia già installato e correttamente configurato nel vostro server Web di fiducia. Anche se potete lanciare PHP da riga di comando, userete qui un browser (e quindi un Web server) per gli esempi. Potete caricare i file PHP in un server Web su Internet (se ne avete uno), oppure potete sfruttare un'installazione base di Apache sulla vostra macchina locale, senza dover aggiornare i file via FTP a ogni modifica. Se necessitate di istruzioni

per l'installazione o la configurazione, fate riferimento al manuale ufficiale di PHP, alla pagina <http://php.net/manual/en/install.unix.php>; in alternativa, esistono miriadi di guide per l'installazione per qualsiasi distribuzione Linux: una rapida ricerca con Google vi darà tutte le risposte che cercate.

Dall'inizio

E ora la parte divertente: lavorare e scrivere il vostro primo script PHP. Storicamente scrivereste un banale "Hello world", ma potrebbe risultare un po' noioso, quindi



The screenshot shows the official PHP website. The main content area features several news items and announcements. Key highlights include:

- PHP 5.4 beta1 released:** A notice about the first beta release of PHP 5.4, mentioning new language features and deprecated behaviors.
- PHP 5.3.8 Released:** An announcement of the immediate availability of PHP 5.3.8, highlighting a security fix for a buffer overflow in the openssl extension.
- Upcoming conferences:** A list of events including the PHPWorld 2011, Zend PHP Conference 2011, and the PHPNW Conference 2011.
- Stable Releases:** A section detailing the current stable version (5.3.8) and its release date.
- Community Links:** A sidebar with links to various PHP-related resources, including the PHP manual, community forums, and local user groups.

Il sito Internet di PHP potrebbe non essere il più accattivante del mondo, ma ci tornerete sempre più spesso viste le tante informazioni che ospita

passerete subito alla generazione di testo dinamico con la funzione **date()** di PHP. Prima di addentrarvi nella foresta del linguaggio, dovrete capire come l'interprete legge il vostro codice PHP e genera l'output relativo. Uno dei vantaggi di PHP sta nel poter inserire il codice direttamente nelle pagine HTML statiche, e in questo modo l'intera pagina viene passata all'interprete. È estremamente importante sapere che tutti i file PHP devono avere l'estensione **.php**: se scrivete codice PHP in file HTML o HTML non verrà interpretato, e visualizzerete semplicemente il codice stesso. Perché il codice PHP possa essere estratto dal resto della pagina, dev'essere racchiuso entro delimitatori. PHP eseguirà il codice trovato al loro interno e tutto il resto verrà ignorato. I delimitatori standard (e più usati) sono **<?php** per indicare l'inizio del codice e **?>** per la fine. Esistono altre opzioni per i delimitatori come i famosi 'short tag', ma alcuni di questi possono avere ripercussioni su XML e XHTML, quindi il consiglio è di usare sempre i tag standard. Tenendo a mente tutto questo, aprite il vostro editor di testo preferito e scrivete

```
<?php echo 'Benvenuti nel magico mondo di PHP'; ?>
```

Salvate il file come **welcome.php** nella root del vostro server Web; una volta salvato, aprite il browser e puntate a tale file con un indirizzo del tipo **http://localhost/welcome.php**. Dovreste vedere semplicemente la scritta *Benvenuti nel magico mondo di PHP* nel browser. Se invece vedete il codice, significa che non avete impostato il server Web per interpretare correttamente i file PHP: tornate sui vostri passi e, aiutandovi anche con eventuali guide, assicuratevi di aver configurato tutto correttamente. Se vedere il testo senza codice PHP, siete pronti per il passo successivo.

Sintassi, tipi dati e funzioni

Noterete che il codice è terminato da un punto e virgola prima del delimitatore. PHP usa il punto e virgola per indicare la fine di una riga di codice o di un'istruzione: senza quello, PHP non saprebbe dove terminare

la valutazione del codice e lo script non funzionerebbe. PHP è molto permissivo quando si tratta di formattazione, potete aggiungere tutti gli spazi e le righe che volete (a meno che siano contenute in stringhe), permettendovi di essere piuttosto liberi nel formattare il codice come preferite. PHP supporta diversi tipi di dato, offrendovi un'enorme flessibilità nello scrivere i programmi. Citando Wikipedia, "In informatica un tipo di dato (o semplicemente tipo) è un nome che indica l'insieme di valori che una variabile

Perché scegliere PHP

PHP è da molto tempo una delle scelte preferite dagli sviluppatori Web. Non solo perché ha una base utenti impressionante, e quindi un ottimo supporto per lo sviluppo e il debug e una quantità enorme di codice disponibile liberamente, ma anche perché gli host che lo supportano sono decisamente diffusi ed economici. La relativa semplicità del linguaggio è stata uno dei motivi per la sua diffusione. PHP può essere molto "di bocca buona" quando programmate. Ad esempio,

non dovete dichiarare le variabili (o il loro tipo) prima di usarle o istanziarle, ed esistono diversi metodi che vi permettono di saltare i tradizionali modi di programmare. Il metodo più facile per capire il perché della popolarità di PHP è cominciare a usarlo. Se avete già lavorato con un altro linguaggio in precedenza, vedrete come sia lineare capire PHP. Se invece è il vostro primo linguaggio, resterete sorpresi dalla velocità con la quale otterrete i primi risultati!

o il risultato di un'espressione possono assumere e le operazioni che su tali valori si possono effettuare". PHP supporta tipi di dato numerici, stringhe e tipi complessi come Array e Oggetti. Nelle distribuzioni standard di PHP esistono più di 1.000 funzioni, che spaziano da cose semplici come le funzioni su data e ora a concetti più avanzati quali LDAP e funzionalità per MySQL. Per tutto quello che manca (o che volete migliorare) potete semplicemente scrivere la vostra funzione. In questo numero vi concentrerete tuttavia sulle basi e sul mantenere le cose semplici.

Script flessibili

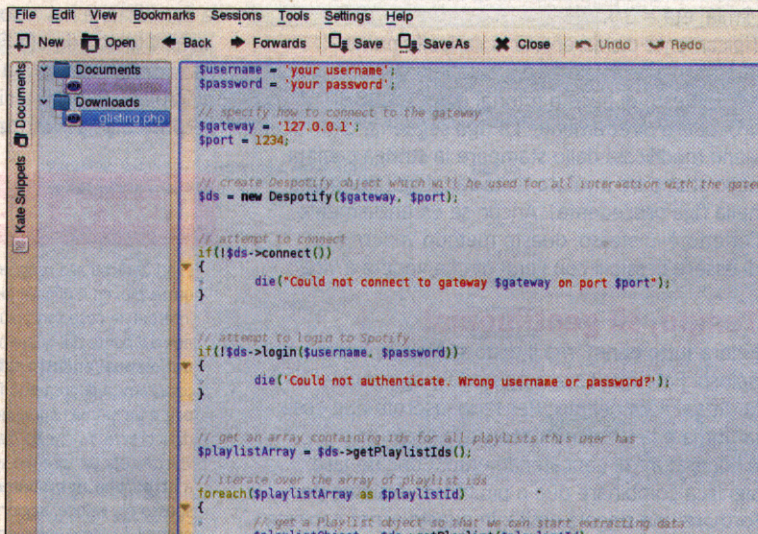
Nel primo esempio avete semplicemente indicato a PHP di visualizzare una stringa specifica attraverso la funzione **echo**. Il valore della stringa può provenire da diversi posti: un database, il valore di uscita di una funzione, un file sul server o addirittura l'interazione dell'utente con il sito. Nel cablare il valore nel codice siete più o meno legati a quello che la stringa può rappresentare: procederete dunque a assegnare la stringa a una variabile, quindi aprite nuovamente l'editor e salvate il codice seguente in un file chiamato **welcome2.php**.

```
<?php
$testo = 'Benvenuti nel magico mondo di PHP';
```

Tip



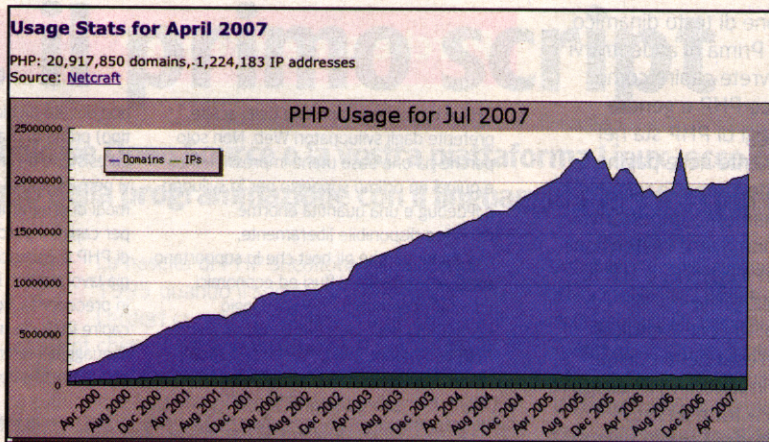
Quando possibile, usate l'indentazione: vi aiuterà a capire e leggere meglio il codice sulla pagina. Alcuni editor lo fanno automaticamente, ma molti sviluppatori usano il tab o due-quattro spazi.



Dopo aver scritto sempre con il blocco note, tutti prima o poi si rendono conto di quant'è bella la vita a colori. Gli editor avanzati sono il paradiso per gli occhi dei programmatori, davvero!



Potete chiaramente vedere la rapida crescita d'uso di PHP e che il numero di installazioni è ancora in aumento



Tip



Utilizzate un editor di testi che abbia l'evidenziazione della sintassi per PHP: vi aiuterà a identificare velocemente il vostro codice e le funzioni al suo interno. Esistono parecchi programmi disponibili per lo sviluppo, quindi date un'occhiata in giro e scegliete quello che vi si confà maggiormente!

```
echo $testo;
?>
```

Lanciando questo script non dovreste vedere alcuna differenza dal primo, ma guardando il codice troverete la prima riga che inizia con **\$testo**: questa viene chiamata variabile. Le variabili in PHP possono contenere un preciso pezzo di dati in un dato momento. Tale dato può cambiare e può essere di qualsiasi tipo in qualsiasi punto. Le variabili iniziano sempre con

\$ seguito dal nome della variabile. Il nome può iniziare solamente con una lettera o un underscore, ma il resto può contenere anche

numeri. Una nota importante: le variabili sono case sensitive, quindi **\$Testo** e **\$testo** sono due variabili diverse. Nello script sopra, dichiarate e assegnate il valore contemporaneamente. Alcuni linguaggi non permettono di farlo, ma PHP è molto flessibile nella programmazione. L'assegnamento di valore è semplicemente il processo del copiare un valore nella variabile assegnata, come

```
$testo = 'Benvenuti nel magico mondo di PHP';
$mia_eta = 30
```

Tipicamente dichiarereste la variabile prima di assegnarle il valore, ma data la natura e il contesto di questo tutorial è accettabile procedere direttamente. La riga seguente dello script viene modificata dallo stampare la stringa cablata nello stampare il valore della variabile (assegnata nella riga precedente). Anche se essenzialmente l'effetto è lo stesso, questo metodo vi permette di essere flessibili con l'output che vedete a video.

Tempo, sii gentiluomo!

Finora tutto bene... ma il testo statico è piuttosto noioso. Fate qualcosa a riguardo e aggiungete la data e l'ora nel mix. Per farlo, userete due cose: la prima è la funzione **date()** e la seconda è l'operatore di concatenamento. Concatenare significa combinare due o più "cose" assieme per formare una singola entità; in questo script capirete come concatenare il messaggio di benvenuto con la data. Per farlo, aprite il solito editor e salvate il codice nel file **welcome-date.php**:

```
<?php
$testo = 'Benvenuti
nel magico mondo
di PHP. Oggi è il ';
echo $testo . date('j F') .
' e sono le ' . date('H:i');
?>
```

Una volta lanciato, dovrete vedere la riga di testo con la data corrente e l'orario integrato. È importante notare che la data e l'ora arrivano dal server e **non** dal vostro browser, come invece farebbe un programma JavaScript

(o altro linguaggio lato client). La prima riga del codice è praticamente identica allo script precedente, modificata solamente per riflettere la natura dinamica dell'output. La seconda riga è dove ha luogo tutta la magia (conosciuta anche come concatenazione). Essenzialmente, e parlando in parole povere, la seconda riga dice "stampa il testo seguito da un comando data, quindi altro testo e chiudi con un altro comando data". Potreste chiedervi com'è stata

specificata la data da stampare: viene fatto tutto con il parametro che passate alla funzione **date()**. Tale funzione accetta parametri in input per

rappresentare esattamente la stringa voluta in output: attualmente accetta più di 35 parametri diversi, ognuno rappresentante un "pezzo" diverso di data e/o ora. Nell'esempio si è spezzata la data e l'ora in due chiamate a funzione, ma potrebbe essere perfettamente accettabile fonderle in una sola. I più lungimiranti tra di voi avranno notato che tra le due chiamate a date c'è del testo ancora una volta cablato. Potete assegnarlo a una variabile per aumentare la flessibilità, e nel caso la riga sarà simile a:

```
echo $testo . date('j F') . $altro_testo . date('H:i');
```

Per una lista completa dei parametri per date, controllate la pagina di manuale ufficiale:

<http://it2.php.net/manual/en/function.date.php>

Risorse essenziali

I libri dedicati alla programmazione in PHP non si contano ormai più, ed è difficile dire quale sia il migliore. Sul fronte Internet, le cose sono più definite: sicuramente <http://php.net> è il punto dal quale partire in assoluto, e [www.php.net/manual/en/intro-whatcando.php](http://php.net/manual/en/intro-whatcando.php) vi darà una panoramica di cosa può fare PHP per voi. <http://phpsec.org> è un'ottima risorsa per tutte le problematiche di sicurezza. Anche se decidete di acquistare un libro, vi troverete sempre più spesso a far riferimento a <http://php.net/manual> per la documentazione online, spesso più veloce, aggiornata e completa di commenti ed esempi. L'ultimo link è egualmente importante se il vostro obiettivo è installare PHP in un server pubblico, perché vi darà importanti dritte sul come metterlo in sicurezza.

PHP 6: cosa significa?

Con la futura versione PHP 6 ci si aspetta grandi cose. Uno dei miglioramenti è un supporto maggiore per le stringhe Unicode, ampliando il set di caratteri disponibili per l'internazionalizzazione. Per gli sviluppatori avanzati, verrà introdotto il supporto ai Namespace; con l'avanzata delle funzionalità

Web 2.0, la versione 6 supporterà nativamente il protocollo SOAP e le caratteristiche per XML verranno rinforzate ed estese. Un pugno di caratteristiche verranno eliminate dal core, inclusi `magic_quotes()`, `register_globals()`, `register_long_arrays()` e `safe_mode()`. La ragione principale è legata alla sicurezza: alcune funzioni

lasciavano aperti potenziali buchi di sicurezza mentre altre portavano a pratiche di programmazione poco efficaci. Potete già scaricare la versione di sviluppo di PHP 6 per provarla, ma al momento della stesura non c'è una data ufficiale per il rilascio, e anche una volta rilasciato, ci si aspetta una discreta attesa prima che sia disponibile al pubblico.

Mettere insieme tutto quanto

Prima abbiamo scritto che uno dei punti di forza per la diffusione di PHP è la capacità di fondere pezzi di codice in HTML statico con facilità. Questo diventa evidente quando volete creare sezioni 'dinamiche' all'interno di una pagina altrimenti statica, ad esempio con lo script di cui sopra. Potete tranquillamente inserire il codice nel template esistente (se ne avete uno). Ecco come potrebbe essere:

```
<html>
...
<body>
<div id="welcome">
    <?php
    $testo = 'Benvenuti
    nel magico mondo
    di PHP. Oggi è il ' .
    date('j F') .
    ' e sono le ' . date('H:i');
    ?>
</div>
...
</body>
</html>
```

Questo è un rudimentale pezzo di codice HTML e potete chiaramente vedere dov'è stato inserito il codice PHP da stampare come testo dinamico nella pagina. Il blocco PHP può stare ovunque nella pagina e può essere ripetuto varie volte: non importerà averne 5, 10 o più blocchi di codice all'interno dell'HTML.

Una cosa meravigliosa è che internamente PHP "comunicerà" tra i vari blocchi di codice nella pagina; perciò se impostate il valore di una variabile nel primo blocco di codice in cima alla pagina, questa sarà disponibile fino all'ultimo blocco in fondo alla pagina. Con questo potete fare meraviglie quando state modificando il contenuto in base ai valori di una variabile impostata in altri punti della pagina. Un esempio molto usato è un sistema di Login/Logout nel quale a un utente viene presentato il form di accesso in base al valore del suo 'stato' attuale.

A voi la palla!

Con il vostro primo script avete ora un'esperienza diretta di come sia facile usare PHP in un sito Web. Da qui, perché non proseguire l'esempio della data, per esempio modificando i parametri in input con qualcosa di diverso, o addirittura provando a integrare il codice in un sito esistente? Date un'occhiata alla documentazione ufficiale (che trovate all'indirizzo Web www.php.net/manual/en/) e scoprirete quali altre funzioni PHP vi offre: resterete sorpresi da quanto potete ottenere dalla sola installazione standard.

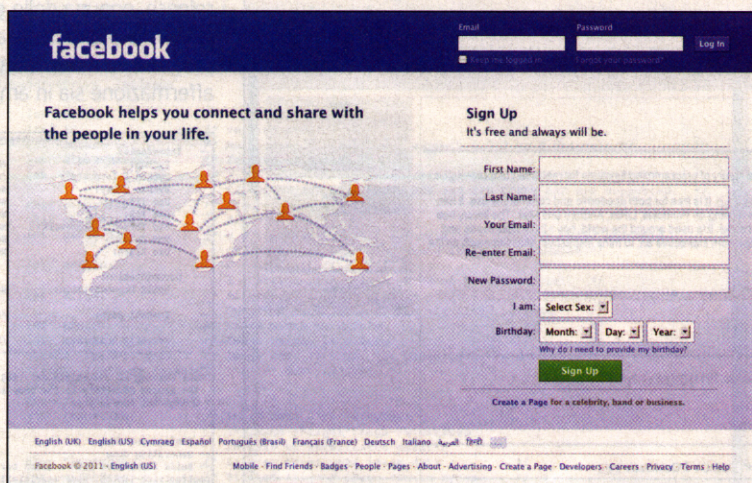
Vale la pena registrare l'URL come segnalibro, perché mano a mano che proseguirete

"Resterete sorpresi di quanto è possibile ottenere dall'installazione standard"

nell'uso di PHP, lo userete sempre più come utile fonte di riferimento. E ovviamente continuate a sperimentare, solo così si può imparare. **LXP**

Più avanti

Tra due numeri vi addenterete nel mondo degli Array e delle funzioni. Imparerete come costruire uno script che modifichi l'output dinamicamente in base al giorno della settimana e consoliderete gli skill appresi in questo numero.



Facebook ama il linguaggio PHP a tal punto da essere scritto in una versione ottimizzata per la piattaforma, chiamata HipHop