

Php, un linguaggio di scripting lato server

by ComputerLucaWorld

inizio stesura 24 gennaio 2012

Per iniziare

Php è un linguaggio lato server, il cui interprete che effettivamente legge le singole istruzioni è presente su un server di tipo remoto.

Un server è un software composto da più moduli software in cui una parte si chiama application Server, che è il modulo software che ospita l'interprete di php. L'interprete è un modulo software che accetta in input una sequenza di istruzioni, verifica la sua sintassi e la esegue solo se è corretta, mostrando il risultato dell'elaborazione. Se non è corretta, avvisa dell'errore, invitando il programmatore a correggerlo.

LA NOSTRA PRIMA PAGINA IN PHP

Uno script php deve cominciare sempre con questi due tag

<?php (tag di apertura)

istruzioni php

?> (tag di chiusura)

Vediamo un esempio di una pagina web html contenente codice php.

N.B. Per essere interpretata dal server come php, deve essere salvata con estensione .php

```
<html>
<body>
<?php
echo "Hello World";
?>
</body>
</html>
```

Ogni istruzione php deve terminare con la virgola. In questo esempio usiamo la funzione echo che utilizziamo per "stampare", ovvero scrivere del testo sullo schermo che deve essere messo tra virgolette.

Le variabili in php

Le variabili in qualsiasi linguaggio di programmazione si possono paragonare a scatole che possono contenere stringhe, valori numerici, array (vedremo meglio più avanti questo tipo complesso di variabile).

Tutte le variabili in php devono cominciare con il simbolo del dollaro \$.

Vediamo quindi un esempio pratico per capirci:

```
$casa="luca";
```

Questa semplice istruzione dichiara una variabile di nome "casa" che contiene il testo "luca".

```
$casa=10;
```

Questa semplice istruzione dichiara una variabile di nome "casa" di tipo numerico perchè contiene il valore numerico 10.

Regole da seguire per i nomi delle variabili:

- Deve cominciare con una lettera o con il segno _ (underscore);
- Può contenere solo caratteri alfanumerici
- Non può contenere spazi.

Manipolare le stringhe

Dopo aver definito una variabile possiamo stamparne il suo contenuto sullo schermo. Vediamo come fare:

```
<?php  
$casa="Sono un testo lunghissimo";  
echo $casa;  
?>
```

Queste due piccolissime istruzioni permettono di creare una variabile casa di tipo stringa, che conterrà quindi un testo. Utilizzando la funzione echo verrà stampato sullo schermo il contenuto della variabile.

Immaginiamo ora di avere due variabile contenente in una una prima parte di un nome e in un'altra una seconda parte. Usiamo il simbolo "." (il punto) per concatenare due stringhe. Vediamo in concreto come fare.

```
<?php  
$n1="lu";  
$n2="ca";  
echo $n1.$n2;  
?>
```

Si crea una variabile che contiene il testo "lu" e una che contiene il testo "ca". Si stampa poi il contenuto delle due variabili.

Lunghezza di una stringa

Immaginiamo di voler conoscere la lunghezza di una determinata stringa. Dovremo utilizzare la funzione strlen. Vediamo un esempio pratico:

```
<?php
$casa="luca";
echo strlen($casa);
?>
```

Sullo schermo verrà stampato il numero 4 perchè la parola "luca" presenta 4 caratteri.

La funzione strpos()

Questa funzione si usare per cercare un carattere o un testo all'interno di una stringa. Se viene trovato, scriverà la posizione del primo risultato della ricerca. Se non viene trovato, restituirà il valore FALSE.

If a match is found, this function will return the character position of the first match. If no match is found, it will return FALSE.

Let's see if we can find the string "world" in our string:

Vediamo un esempio pratico

```
<?php
echo strpos("Hello world!","world");
?>
```

Questo codice darà come output 6.

Perchè?

Php comincia a contare da 0 e non da 1.

Gli operatori del linguaggio php

Operatori aritmetici

Operatore	Descrizione	Esempio	Risultato
+	Addizione	x=2 x+2	4
-	Sottrazione	x-3	-1
*	Moltiplicazione	x*3	6
/	Divisione	2/3	0,6666

% Modulo (fornisce il resto di una divisione)

++	Incremento	x=5	x++ x=6
--	Decremento	x=5	x-- x=4

Operatori di assegnazione

Operatore	Esempio	E' lo stesso di
=	x=y	x=y
+=	x+=y	x=x+y
-=	x-=y	x=x-y
=	x=y	x=x*y
/=	x/=y	x=x/y

. = x.=y x=x.y
%= x%=y x=x%y

Operatori di confronto

Operatore	Descrizione	Esempio
==	è uguale a	5==8 returns false
!=	non è uguale a	5!=8 returns true
<>	non è uguale a	5<>8 returns true
>	è maggiore di	5>8 returns false
<	è minore di	5<8 returns true
>=	Maggiore o uguale to	5>=8 returns false
<=	minore o uguale a	5<=8 returns true

Operatori logici

Operatore	Descrizione	Esempio
&&	and	x=6 y=3 (x < 10 && y > 1) returns true
	or	x=6 y=3 (x==5 y==5) returns false
!	not	x=6 y=3 !(x==y) returns true

Facciamo ragionare il computer

Strutture di controllo

Quando scriviamo del codice, vogliamo a volte fare in modo che il computer scriva una determinata parola diversa a seconda della situazione che si verifica.

Dobbiamo usare in questo caso delle strutture di controllo

La struttura di controllo if

Si usa la seguente sintassi

```
if (condizione da verificare) {  
azioni da eseguire se la condizione è vera;  
}
```

La struttura if... else

Si usa per eseguire un certo codice se la condizione è vera e un altro codice se la condizione è falsa.

```
if (condition){  
codice da essere eseguito se la condizione da verificare risulta vera;  
else{
```

```
codice da essere eseguito se la condizione da verificare risulta falsa;  
}
```

```
</body>  
</html>
```

La struttura di controllo switch

Use the switch statement to select one of many blocks of code to be executed.

Si usa per decidere quale codice eseguire al variare del valore di una certa espressione.

Sintassi

```
switch (n)  
{  
case label1:  
    code to be executed if n=label1;  
    break;  
case label2:  
    code to be executed if n=label2;  
    break;  
default:  
    code to be executed if n is different from both label1 and label2;  
}
```

This is how it works: First we have a single expression n (most often a variable), that is evaluated once. The value of the expression is then compared with the values for each case in the structure. If there is a match, the block of code associated with that case is executed. Use break to prevent the code from running into the next case automatically. The default statement is used if no match is found.

Example

```
<html>  
<body>  
  
<?php  
switch ($x)  
{  
case 1:  
    echo "Number 1";  
    break;  
case 2:  
    echo "Number 2";  
    break;
```

```
case 3:  
  echo "Number 3";  
  break;  
default:  
  echo "No number between 1 and 3";  
}  
?>  
  
</body>  
</html>
```