

Videocorso PHP




Introduzione al linguaggio
e richiami di programmazione

Alessandro Flora

La sintassi di PHP

Come creiamo uno script PHP

- Apriamo in VS Code la cartella del server e premiamo il tasto nuovo file 
- Inseriamo il nome dello script, per esempio **index.php** (NB: se non chiamiamo lo script index dovremo recarci all'indirizzo localhost/nomeScript.php)
- Scriviamo il codice all'interno dei delimitatori **<?php?>** senza preoccuparci di definire la funzione main (o una relativa classe per chi è abituato a Java)



index.php

La sintassi di PHP

Come codifichiamo lo script PHP

- All'interno del codice delimitiamo tutto il codice PHP dai delimitatori `<?php?>`
- Tutto ciò al di fuori da essi sarà interpretato come se fosse un file HTML

🐼 index.php

```
1 <?php
```

```
2
```

```
3     // qui inseriamo il codice
```

```
4
```

```
5 ?>
```

```
6
```

```
7 <!-- qui inseriamo l'HTML (lezioni successive) -->
```

I commenti analogamente a C e Java si indicano con:

- `//` (riga singola oppure uno per riga)
- `/* */` (più righe)

Ci servirà per far interagire PHP e HTML

La sintassi di PHP

Come dichiariamo le variabili

- Il loro nome inizia obbligatoriamente con il simbolo \$
- Il loro nome è case sensitive e inizia obbligatoriamente con una lettera
- Non sono tipizzate e dunque le inizializziamo assegnando un valore

🐘 index.php

```
1 <?php
```

```
2
```

```
3     $var = 5;
```

```
4
```

```
5 ?>
```

→ Come in C e in Java le istruzioni terminano con il punto e virgola

La sintassi di PHP

I tipi dei dati contenuti dalle variabili


- In quanto linguaggio tipizzato le variabili non hanno un tipo, i dati naturalmente sì
- Possiamo riassegnare le variabili cambiandone il dato contenuto (e dunque il tipo)

index.php

```
1 <?php
2
3 $var = 5; // numero intero -> int
4
5 $var = 2.3; // numero reale -> float
6
7 $var = 'Hello, World!'; // stringa -> str
8
9 $var = true; // booleano -> bool
10 $var = false;
11
12 $var = []; // vettore -> array
13 $var = array();
14
15 $var = new Object(); // oggetto -> object
16
17 ?>
```

 In PHP gli apici e le virgolette sono equivalenti

 Le notazioni sono equivalenti, argomento della prossima lezione

 Non nel programma obbligatorio, OOP trattata nella lezione A1

La sintassi di PHP

Gli operatori matematici

- Servono per fare interagire tra di loro delle variabili nelle operazioni

index.php

```
1  <?php
2
3  $a = 1;
4  $b = 2;
5
6  // OPERATORI MATEMATICI
7
8  $a + $b; // somma
9  $a - $b; // differenza
10 $a * $b; // prodotto
11 $a / $b; // quoziente
12 $a % $b; // resto della divisione intera
13 $a ** $b; // elevamento a potenza di $a alla $b
14
15 ?>
```



È utile per verificare la divisibilità (per esempio se un numero è pari questo avrà resto zero rispetto alla divisione intera per 2)



Può essere usato anche per estrarre radici ricordando che la radice n-esima equivale alla potenza $1/n$

La sintassi di PHP

Gli operatori relativi alle stringhe e agli incrementi

- Servono per fare interagire tra di loro delle variabili nelle operazioni

🐘 index.php

```
1  <?php
2
3      $a = 1; $b = 2;
4      $c = 'c'; $d = 'd';
5
6      $c.$d; // concatenazione -> otteniamo 'cd'
7
8      $a++; $a--; // incremento e decremento unitari
9      $b *= 2; // il valore di $b viene raddoppiato
10     $b /= 2; // il valore di $b viene dimezzato
11     $a += 4; // il valore di $a viene incrementato di 4
12  ?>
```

La sintassi di PHP


Gli operatori di confronto

- Servono per ottenere una risposta pari a **true** o **false** rispetto a una condizione

index.php

```
1  <?php
2
3      $a = 1; $b = 2;
4
5      $a == $b; // uguaglianza
6      $a != $b; // disuguaglianza
7
8      $a <= $b; // minore o uguale
9      $a > $b; // maggiore
10
11     $res = $a == $b; // $res sarà false (in questo caso)
12
13     !$res; // otteniamo l'opposto del valore di $res (in questo caso true)
14
15  ?>
```

\$res potrebbe essere utile per salvare il valore dell'uguaglianza a un punto del codice e riusarlo in seguito



La sintassi di PHP

Le espressioni booleane: operatori logici

index.php

```
1  <?php
2
3      $a = 1; $b = 2;
4
5      // OR logico -> o una condizione o l'altra
6      $a > $b || $b > $a;
7      $a > $b or $b > $a;
8
9      // AND logico -> entrambe le condizioni
10     $a == $b && $a > 0;
11     $a == $b and $a > 0;
12
13     // XOR logico -> o la prima condizione o la seconda ma non entrambe
14     $a > $b xor $a > 0;
15
16  ?>
```

or ,	true	false
true	true	true
false	true	false
and , &&	true	false
true	true	false
false	false	false
xor	true	false
true	false	true
false	true	false

La sintassi di PHP

Come interagiamo e vediamo l'output

- Il comando echo ci consente di far interfacciare PHP e HTML; serve per stampare come HTML qualunque cosa gli sia posta successivamente (variabili, valori, ...)

index.php

```
1  <?php
2
3      echo 'Hello, World!';
4      echo $variabile;
5      echo $variabile.' altra stringa';
6      echo "stringa contenente una $variabile";
7
8  ?>
```

La sintassi di PHP


Il costrutto if - else if - else

- Serve per eseguire o meno un'istruzione o un blocco di codice dato un booleano

index.php

```
1  <?php
2
3  $a = 2; $b = 3;
4
5  if($a > $b) {
6      echo "a è maggiore di b";
7  } else if($a < $b) {
8      echo "b è maggiore di a";
9  } else
10     echo "a e b sono uguali";
11
12  ?>
```

Il risultato è un valore booleano, possiamo volendo anche usare una variabile booleana come condizione della selezione/ciclo



La sintassi di PHP

Il costrutto while

- Serve per eseguire un numero non precisato di volte un blocco di istruzioni

index.php

```
1  <?php
2
3  $a = 2; $b = 3;
4  $diff = 0;
5
6  while($a < $b) {
7      $diff++;
8      $a++;
9  }
10
11  ?>
```

È fondamentale ricordarsi di modificare la condizione di prosecuzione del ciclo onde evitare che questo prosegua all'infinito

La sintassi di PHP

Il costrutto do - while

- Analogo al while, controlla però dopo l'esecuzione delle istruzioni

index.php

```
1  <?php
2
3  $a = 2; $b = 3;
4  $eseguito = false;
5
6  do {
7      $eseguito = true;
8  } while($a > $b);
9
10 ?>
```



Alla fine dell'esecuzione dello script **\$eseguito** avrà valore **true** perché prima di controllare la condizione espressa nel while è già stato eseguito una volta il blocco di codice

La sintassi di PHP

Il costrutto for

- Analogo al while, accorpa la condizione iniziale, di prosecuzione e di passo

index.php

```
1  <?php
2
3  $somma = 0;
4
5  for($i=0;$i<10;$i++) {
6      $somma += $i;
7  }
8
9  ?>
```

Allo fine dell'esecuzione dello script **\$somma** avrà
valore **45**