

Corso di PHP

3 - Le strutture di controllo

M. Malatesta 3-Strutture di controllo-07

1
24/03/2017

Prerequisiti

- Istruzioni semplici
- Controllo del flusso di programma

M. Malatesta 3-Strutture di controllo-07

2
24/03/2017

Introduzione

In molti casi le sole istruzioni viste finora non sono sufficienti a risolvere certi problemi.

È possibile far effettuare alla macchina delle scelte in modo da alterare il flusso di esecuzione sequenziale del programma?

È possibile far eseguire ripetitivamente parti di programma in base ad una condizione?

In questa Unità daremo una risposta a questi quesiti.

M. Malatesta 3-Strutture di controllo-07

3
24/03/2017

Informazioni generali

In questa Unità presentiamo le strutture di controllo di PHP per realizzare selezioni di vario tipo e cicli. In questo modo, possiamo alterare l'esecuzione sequenziale dei programmi e realizzare programmi anche molto complessi.

È sempre indispensabile che la progettazione dei programmi segua la tecnica **top-down**, che prevede la scomposizione di un problema in sottoproblemi, via via più semplici, da risolvere in un secondo momento

M. Malatesta 3-Strutture di controllo-07

4
24/03/2017

Le strutture di controllo

Come abbiamo detto in precedenza, vogliamo studiare gli strumenti che consentono di alterare il flusso sequenziale di esecuzione del programma.

Questi strumenti prendono il nome di **strutture di controllo**.

Le strutture di controllo sono:

- 1) **Sequenza**
- 2) **Selezione**
- 3) **Iterazione**

M. Malatesta 3-Strutture di controllo-07

5
24/03/2017

La sequenza

Tutte le istruzioni presenti nella parte eseguibile del programma si dice che costituiscono una **SEQUENZA**, nel senso che **costituiscono un blocco che può essere considerato come un'unica istruzione**

Quando si vuole considerare un blocco di istruzioni come se fosse una sola istruzione si utilizza la sintassi seguente:

```
{ istruzione;  
  istruzione;  
  .....  
}
```

La **SEQUENZA** è caratterizzata dalla coppia di parentesi graffe aperta "{" e chiusa "}".

ATTIVITA': scrivere una applicazione PHP che calcoli la somma dei valori interi contenuti in due variabili.

M. Malatesta 3-Strutture di controllo-07

6
24/03/2017

La sequenza

```
<?php
<?php
$a = 9;
$b = 6;
if ($a<$b)
{ $primo=$a;
  $secondo=$b;
}
else
{ $primo=$b;
  $secondo=$a;
}
echo "primo=".$primo." secondo=".$secondo;
?>
```

Le istruzioni vengono eseguite una dopo l'altra, come se il gruppo di istruzioni fosse un **unico blocco**.

M. Malatesta 3-Strutture di controllo-07

7
24/03/2017

La selezione

Quando si deve eseguire una scelta sulla base di una *condizione* si usano le strutture di controllo selettive

Le strutture di controllo **selettive** si servono del risultato di un **test** sulla *condizione* per stabilire quale percorso dovrà seguire il programma durante l'esecuzione.

Un **test** è la valutazione del risultato di una **espressione logica** o **espressione booleana** e quindi può dare come valori **vero** (*true*) o **falso** (*false*).

M. Malatesta 3-Strutture di controllo-07

8
24/03/2017

La selezione

Selezione doppia

Quando si vuole che il programma segua un percorso diverso a seconda del valore di una espressione logica (vera o falsa) si utilizza la **selezione doppia** che ha la sintassi seguente:

```
if (espressione-logica)  
    istruzione1;  
else istruzione2;
```

Se *espressione-logica* risulta **VERA** (diversa da 0) viene eseguita *istruzione1*, altrimenti viene eseguita *istruzione2*

ATTIVITA': scrivere una applicazione PHP che calcoli il massimo tra due numeri..

M. Malatesta 3-Strutture di controllo-07

9
24/03/2017

La selezione

Selezione doppia

```
<?php  
$a = 9;  
$b = 4;  
if ($a>$b)  
{  
    echo $a, " <b>è maggiore di </b>", $b;  
}  
else  
{  
    echo $a, " <b>è minore o uguale di</b> ", $b;  
}  
?>
```

L'espressione logica risulta **VERA** se $a > b$

M. Malatesta 3-Strutture di controllo-07

10
24/03/2017

La selezione

Selezione doppia

Una particolarità della selezione doppia è quella poter di utilizzare la parola chiave **elseif**.

```
if (espressione-logica1)  
    istruzione1;  
elseif (espressione-logica2)  
    istruzione2;  
else istruzione3;
```

Se *espressione-logica1* è falsa, con **elseif** si valuta *espressione-logica2*. Dopo *istruzione2*, segue eventualmente **else**, che fa eseguire *istruzione3* nel caso in cui anche *espressione-logica2* sia falsa.

M. Malatesta 3-Strutture di controllo-07

11
24/03/2017

La selezione

Operatore ternario

Un caso particolare di selezione doppia si può rappresentare con l'**operatore ternario** "?:".

L'operatore ternario ha la sintassi:

ident = (*espress-logica*) ? *espressione1* : *espressione2*;

Se *espress-logica* è **VERA**, ad *ident* assegna *espressione1*, se risulta **FALSA** gli assegna *espressione2*.

Costituisce un modo **abbreviato** per effettuare una selezione doppia.

M. Malatesta 3-Strutture di controllo-07

12
24/03/2017

La selezione

Operatore ternario

ATTIVITA': scrivere tramite l'operatore ternario le seguenti s.d.c. selettive:

- 1) **if** (\$primo > \$secondo) \$a=\$primo-\$secondo; **else** \$a=\$secondo-\$primo;
(\$primo>\$secondo) ? \$a=\$primo-\$secondo : \$a=\$secondo-\$primo;
- 2) **if** (\$x>0) \$modulo_x=\$x; **else** \$modulo_x=-\$x;
\$modulo_x=(\$x>0)? \$x : (-\$x);
- 3) **if** (\$y==\$z) \$x=\$u*\$x+\$a; **else** \$x=\$u*\$x+\$b;
\$x=(\$u*\$x+(\$y==\$z)? \$a:\$b)

M. Malatesta 3-Strutture di controllo-07

13
24/03/2017

La selezione

Selezione semplice

In alcuni casi, la struttura selettiva si può utilizzare nella forma seguente (**selezione semplice**):

if (*espressione-logica*)
istruzione;

Se *espressione-logica* risulta
VERA viene eseguita
istruzione

ATTIVITA': scrivere una applicazione PHP che calcoli il minimo fra tre numeri.

M. Malatesta 3-Strutture di controllo-07

14
24/03/2017

La selezione

Selezione semplice

```
<?php
$a = 6;
$b = 9;
$c = 4;
if ($a < $b)
{ $min = $a; }
else
{ $min = $b; }
if ($c < $min)
{ $min = $c; }
echo "min($a, $b, $c)=", $min;
```

Selezione doppia

Selezione
semplice

M. Malatesta 3-Strutture di controllo-07

15
24/03/2017

Espressioni logiche

Osservazioni

- il valore numerico 0,
- una stringa che contiene '0'
- una stringa vuota
- un valore NULL, cioè una variabile:
 - che non è stata definita;
 - che è stata eliminata con **unset()**;
 - a cui è stato assegnato il valore NULL esplicitamente

L'interprete PHP considera
false tutte le situazioni
indicate a fianco

Pertanto qualsiasi numero, intero o decimale purchè diverso da 0,
qualsiasi stringa non vuota, se usati come espressione condizionale
saranno considerati veri.

M. Malatesta 3-Strutture di controllo-07

16
24/03/2017

Espressioni logiche

Osservazioni

```
if ($a = 4)
    echo "$a è uguale a 4"; // stampa questa frase
...
if ($a = 0)
    echo "$a è uguale a 4"; // NON stampa questa frase
```

Analogamente ad altri linguaggi (C, C++, Java) vale questa regola: le istruzioni di assegnazione hanno un valore, in particolare hanno il valore che viene assegnato alla variabile

M. Malatesta 3-Strutture di controllo-07

17
24/03/2017

La selezione

Selezione multipla

Quando la scelta deve essere fatta tra molti valori si usa quest'altra struttura di controllo:

```
switch (espressione-intera)
{
    case valore-1:
        istruzione-1; break;
    case valore-2:
        istruzione-2; break;
    .....
    default: istruzione;
}
```

Questa struttura di controllo prende il nome di **SELEZIONE MULTIPLA**

La selezione multipla serve a selezionare un dato valore tra diversi possibili. Se ***espressione_intera*** vale *valore-1* viene eseguita *istruzione1*, se vale *valore-2* viene eseguita *istruzione2* e così via. Se nessun valore corrisponde al valore di *espressione-intera*, la parola **default** consente di trattare le situazioni di mancata corrispondenza.

ATTIVITA': scrivere una applicazione PHP che selezioni tramite una **switch()** un valore a scelta fra 1,2 o 3 immessi da tastiera e lo stampi.

M. Malatesta 3-Strutture di controllo-07

18
24/03/2017

La selezione

Selezione multipla

```
<?php
$a = 2;
switch ($a)
{ case 1: echo "Hai immesso il valore 1";
  break;
  case 2: echo "Hai immesso il valore 2";
  break;
  case 3: echo "Hai immesso il valore 3";
  break;
  default: echo "Il valore deve essere fra 1 e 3!";
}
?>
```

Quando viene trovata una corrispondenza di valori, viene eseguita l'istruzione

break fa saltare il controllo alla fine dell'istruzione **switch**.

Viene eseguita nel caso di mancata corrispondenza

M. Malatesta 3-Strutture di controllo-07

19
24/03/2017

L'iterazione

Quando si deve eseguire ripetutamente un blocco di istruzioni si usano le strutture di controllo iterative.

Le strutture di controllo **iterative** servono ad eseguire i **cicli**.

Un ciclo è la ripetizione di un dato blocco di istruzioni. Nelle strutture di controllo iterative è spesso necessario utilizzare una variabile detta **contatore** per tenere traccia del numero di volte che il ciclo viene effettuato.

M. Malatesta 3-Strutture di controllo-07

20
24/03/2017

L'iterazione

Iterazione predefinita

Un primo tipo di costrutto iterativo è il seguente:

```
for (contatore=inizio; contatore<=fine; contatore++)  
    istruzione;
```

Questa struttura di controllo esegue ripetutamente *istruzione* per ciascuno dei valori che *contatore* assume, partendo da *inizio* fino a *fine*. L'istruzione *contatore++* serve ad incrementare il valore di *contatore*.

Questa struttura prende il nome di **ITERAZIONE PREDEFINITA** e si usa quando il programmatore conosce a priori il numero di volte che il ciclo deve essere eseguito

L'**ITERAZIONE PREDEFINITA** esegue il ciclo fintantochè la condizione al centro risulta **VERA**. Quando diviene **FALSA**, ossia quando *contatore*>*fine* il ciclo termina

Condizione di ripetizione

M. Malatesta 3-Strutture di controllo-07

21
24/03/2017

L'iterazione

Iterazione predefinita

Un altro tipo di iterazione predefinita è

```
for (contatore=inizio; contatore>=fine; contatore- -)  
    istruzione;
```

Questa è analoga alla precedente, ma si usa quando il valore *inizio* è maggiore del valore *fine*. Per indicare che il contatore questa volta deve indietreggiare, si usa l'istruzione *contatore - -*.

L'**ITERAZIONE PREDEFINITA** esegue il ciclo fintantochè la condizione al centro risulta **VERA**. Quando diviene **FALSA**, ossia quando *contatore*<*fine*, il ciclo termina

Condizione di ripetizione

ATTIVITA': scrivere una applicazione PHP che calcoli la somma di N numeri interi letti da input.

M. Malatesta 3-Strutture di controllo-07

22
24/03/2017

L'iterazione

Iterazione predefinita

```
<?php
$n=10;
$s=0;
for ($i=0; $i < $n; $i++)
{
    $s+=$i;
}
echo "La somma vale: ", $s;
?>
```

s è assunto come **accumulatore** con valore iniziale uguale a 0.

i è assunto come **contatore** con valore iniziale uguale a 0. La condizione di uscita dal ciclo è $i == n$

M. Malatesta 3-Strutture di controllo-07

23
24/03/2017

L'iterazione

Iterazione precondizionata

Quando **NON** si conosce a priori il numero di volte che un ciclo deve essere ripetuto, si usa la seguente:

```
while (espress-logica)
    istruzione;
```

Questa struttura di controllo esegue il ciclo fintantoche *espressione-logica* risulta **VERA**. Esce quando diventa **FALSA**.

Ovviamente, se *espressione-logica* è falsa già all'inizio, la while **NON SARA' ESEGUITA NEMMENO UNA VOLTA**

Questa struttura prende il nome di **ITERAZIONE PRECONDIZIONATA** poiché la condizione da verificare si trova prima del blocco di istruzioni da ripetere

ATTIVITA': scrivere una applicazione PHP che dato un intero *N* da input, ne calcoli il divisore più grande.

M. Malatesta 3-Strutture di controllo-07

24
24/03/2017

L'iterazione

Iterazione precondizionata

```
<?php
$n=18;
$i = $n-1;
while ($i > 1)
{ if ($n%$i==0)          /* il primo divisore è sicuramente il massimo */
    break;              /* quindi interrompe l'esecuzione del ciclo */
    $i--;
}
$div=($n>1)? $i:1;        /* se N vale 1 il massimo divisore è 1 */
echo "Il divisore massimo di ", $n, " e' ", $div;

?>
```

Operatore ternario

M. Malatesta 3-Strutture di controllo-07

25
24/03/2017

L'iterazione

Iterazione postcondizionata

In altri casi, e sempre quando NON si conosce a priori il numero di volte che un ciclo deve essere ripetuto, si usa la seguente:

```
do
{ istruzione;
} while (espress-logica);
```

Questa struttura di controllo esegue il ciclo fino a quando *espressione-logica* risulta **VERA**. Esce quando diventa **FALSA**.

Ovviamente, anche se *espressione logica* è falsa già all'inizio, la **do-while** **VERRA' ESEGUITA ALMENO UNA VOLTA**.

Questa struttura prende il nome di **ITERAZIONE POSTCONDIZIONATA** poiché la condizione da verificare si trova dopo il blocco di istruzioni da ripetere

ATTIVITA': scrivere una applicazione PHP che estragga 1000 numeri casuali e stampi il massimo tra i valori usciti.

M. Malatesta 3-Strutture di controllo-07

26
24/03/2017

L'iterazione

Iterazione postcondizionata

```
<?php
$max=0;
$i=0;
do
{ $n = rand();
  if ($n>$max)
    $max=$n;
  $i++;
}
while ($i<1000);
echo "Il massimo e' ", $max;
?>
```

Estrae un valore casuale

M. Malatesta 3-Strutture di controllo-07

27
24/03/2017

L'iterazione

Ciclo foreach

Esiste una variante dell'iterazione predefinita che serve a trattare strutture dati particolari (array e collezioni) in cui ogni elemento è caratterizzato dalla coppia (**nome, valore**).

La sintassi è la seguente

```
foreach ($arr as $chiave => $valore)
{
    istruzione;
}
```

Per ogni elemento della collezione *\$arr*, il ciclo estrae *\$chiave* e *\$valore*.

Vedremo più avanti in dettaglio l'utilità di questa struttura di controllo.

M. Malatesta 3-Strutture di controllo-07

28
24/03/2017

Argomenti

- Le strutture di controllo
- La selezione
 - Selezione doppia
 - Operatore ternario
 - Selezione semplice
 - Selezione multipla
- La sequenza
- L'iterazione
 - Iterazione predefinita
 - Iterazione preconditionata
 - Iterazione postcondizionata
 - Ciclo **foreach**

M. Malatesta 3-Strutture di controllo-07

29
24/03/2017

Altre fonti di informazione

- P.Camagni, R.Nicolassy – PHP ed. HOEPLI Informatica

M. Malatesta 3-Strutture di controllo-07

30
24/03/2017