

Tipi di dato, variabili, istruzioni

Esercizio 1

Implementare un servizio che estrae due numeri random tra 1 e 100, stampa a video i due numeri ed esegue le seguenti operazioni:

- se almeno uno dei due numeri è minore di 50, stampa la somma dei primi 50 numeri, ad eccezione del numero 5 (il numero 5 è scartato nella somma)
- se entrambi i numeri sono maggiori di 50, estrae un altro numero compreso tra 0 e 2. A seconda del numero estratto esegue le seguenti operazioni:
 - 0 → stampa la scritta “zero”
 - 1 → stampa la scritta “uno”
 - 2 → stampa la scritta “due”

Utilizzare la funzione rand() per l'estrazione dei due numeri.

Soluzione

Il programma va implementato per passi. Prima di tutto, va scritta l'intelaiatura dell'intero programma:

Innanzitutto, va aperto il tag di apertura e chiusura del programma php:

```
<?php
?>
```

Poi vanno estratti i due numeri a caso, usando la funzione rand(). La funzione rand riceve in ingresso due numeri, che indicano l'intervallo all'interno del quale effettuare l'estrazione. Quindi ad esempio rand(0,100), estrae un numero tra 0 e 100. Invochiamo la funzione rand due volte, per l'estrazione dei due numeri e assegnamo i risultati delle estrazioni a due variabili:

```
<?php
$num1 = rand(0,100);
$num2 = rand(0,100);
?>
```

Poi, stampiamo a video i due numeri:

```
<?php
$num1 = rand(0,100);
$num2 = rand(0,100);
```

```
echo $num1."<br>";  
echo $num2."<br>";
```

```
?>
```

Abbiamo utilizzato l'operatore . (punto) per concatenare due stringhe e il tag html
 per effettuare l'invio accapo.

A questo punto dobbiamo controllare che almeno uno dei due numeri sia < 50. Per fare questo usiamo l'istruzione IF e l'operatore logico OR (il cui simbolo è ||). L'operatore or ha la seguente sintassi:

```
condizione1 || condizione2
```

L'operatore or verifica che almeno una delle due condizioni sia vera. Se almeno una delle due condizioni è vera, allora ritorna vero come risultato. Nel nostro caso le due condizioni da verificare sono:

```
($num1 < 50) || ($num2 < 50)
```

Possiamo mettere queste due condizioni dentro l'istruzione IF. In questo modo, se almeno una delle due condizioni è vera, allora è eseguito il corpo di istruzioni dentro l'istruzione IF. Usiamo le parentesi graffe, per racchiudere le istruzioni dentro il blocco IF. Se non mettessimo le parentesi graffe, solo la prima istruzione sarebbe eseguita.

Per comodità e per buona prassi di programmazione, definiamo una variabile \$n a cui assegniamo il valore 50. In questo modo, useremo \$n al posto di 50 e se in futuro dovremo cambiare il codice (ad esempio dovremmo controllare i numeri fino a 60), basterà cambiare il valore di \$n.

```
$n = 50;
```

Passiamo quindi all'IF:

```
if( $num1 < $n || $num2 < $n)  
{  
    // istruzioni  
}
```

Una buona prassi di programmazione consiste nell'aggiungere i commenti al codice nelle parti più importanti. Abbiamo usato l'operatore // per scrivere il commento "istruzioni", che però è troppo generico. Scriviamo invece cosa dovrà fare il programma.

```
if( $num1 < $n || $num2 < $n)  
{  
    // calcola la somma dei primi $n numeri, ad eccezione del numero 5  
}
```

Scriveremo le istruzioni dentro la IF dopo. Per ora procediamo con la costruzione dell'intelaiatura del codice.

Se entrambe le condizioni sono false, cioè se entrambi i numeri sono maggiori di \$n, l'operatore or restituisce falso e quindi tutta l'istruzione IF ritorna falso, per cui non è eseguito il blocco di istruzioni dentro l'istruzione IF.

Se entrambe le condizioni sono false, il nostro esercizio richiede che occorre verificare che entrambi i numeri siano maggiori di \$n. In questo caso, possiamo usare l'operatore logico and (il cui simbolo è &&), che ritorna vero se entrambe le condizioni sono vere:

(condizione1) && (condizione2)

Se almeno una delle due condizioni è falsa, cioè se almeno uno dei due numeri è minore di \$n, l'operatore ritorna falso. Nel nostro caso, appendiamo all'IF del controllo precedente un ELSE (cioè altrimenti) e un ulteriore controllo IF:

```
if( $num1 < $n || $num2 < $n)
{
    // calcola la somma dei primi $n numeri, ad eccezione del numero 5
}
else if($num1 > $n && $num2 > $n)
{
    // istruzioni
}
```

Potremmo anche non mettere l'ELSE, ma ciò rallenterebbe il programma. Infatti, l'idea di IF...ELSE è che se è verificata una condizione, allora eseguo un'istruzione, altrimenti eseguo un'altra istruzione. Se non mettiamo l'ELSE, e quindi usiamo IF...IF, il programma verifica entrambe le condizioni.

Sostituiamo ora il generico commento "istruzioni" con quanto il programma deve fare:

```
else if($num1 > $n && $num2 > $n)
{
    // estrai un numero tra 0 e 2
    // controlla il numero e stampa a video la stringa corrispondente
}
```

A questo punto possiamo scrivere il corpo di ciascun blocco IF. Partiamo dal primo IF. Occorre stampare la somma dei primi \$n numeri, ad eccezione del numero 5. Per ora non consideriamo l'eccezione e provvediamo a scrivere la somma dei primi \$n numeri. Usiamo l'istruzione FOR. La sintassi di questa istruzione è la seguente:

```
for(istruzione_iniziale; condizione; post_istruzione)
    // istruzione
```

Il FOR come prima cosa esegue l'istruzione iniziale, che è eseguita una volta sola, poi verifica che condizione sia verificata, se sì, esegue l'istruzione. Dopo esegue la post_istruzione. Di nuovo verifica la condizione, se è verificata, esegue l'istruzione ed esegue la post_istruzione. E così via finché la condizione è verificata. Quando la condizione non è più verificata, il ciclo FOR termina. Di solito il ciclo FOR si usa nel seguente modo:

```
for($i = 0; $i < $n; $i++)
```

```
{
  //istruzione
}
```

Il ciclo crea una variabile \$i, inizializzata a 0, poi verifica che \$i sia minore di un certo numero \$n precedentemente definito, esegue l'istruzione e poi incrementa \$i di 1 (\$i++).

Nel nostro caso, occorre fare la somma dei primi \$n numeri. Quindi, inizializziamo una variabile \$somma, che inizialmente vale 0 e che poi via via conterrà la somma dei vari numeri:

```
$somma = 0;
```

Usiamo il ciclo FOR per calcolare la somma. Dobbiamo fare un ciclo che va da 1 a 50 e ad ogni passo, aggiunge a \$somma, il numero corrente:

```
for($i = 1; $i < $n; $i++)
{
    $somma = $somma + $i;
}
```

La variabile che mantiene il numero corrente è \$i, quindi ad ogni ciclo, dobbiamo sommare alla variabile \$somma il valore di \$i.

Stampiamo infine il valore di \$somma, fuori dal ciclo FOR:

```
echo $somma."<br>";
```

Consideriamo ora il fatto che il numero 5 non va sommato alla variabile \$somma. In questo caso bisogna fare un controllo: se \$i è uguale a 5 occorre continuare il ciclo, senza effettuare la somma. Possiamo utilizzare l'istruzione continue, che si può usare solo dentro i cicli e permette di saltare tutte le istruzioni che seguono e ritornare al ciclo:

```
for(istruzione_iniziale; condizione; post_istruzione)
{
    if(condizione)
        continue;
    //istruzione
}
```

La continue funziona nel seguente modo: se la condizione è verificata, è eseguita la continue, che salta direttamente all'esecuzione della post_istruzione, senza eseguire l'istruzione dentro il FOR.

Nel nostro caso, dobbiamo verificare se \$i sia uguale a 5. Usiamo l'operatore ==, che effettua il confronto:

```
$somma = 0;
for($i = 1; $i < $n; $i++)
{
    if($i == 5)
        continue;
    $somma = $somma + $i;
}
```

```
}  
echo $somma."<br>";
```

A questo punto abbiamo completato la prima parte dell'esercizio. Riportiamo sotto il codice completo, relativo alla prima parte:

```
<?php  
$num1 = rand(0,100);  
$num2 = rand(0,100);  
  
echo $num1."<br>";  
echo $num2."<br>";  
  
$n = 50;  
if( $num1 < $n || $num2 < $n)  
{  
    $somma = 0;  
    for($i = 1; $i < $n; $i++)  
    {  
        if($i == 5)  
            continue;  
        $somma = $somma + $i;  
    }  
    echo $somma."<br>";  
}  
else if($num1 > $n && $num2 > $n)  
{  
    // estrai un numero tra 0 e 2  
    // controlla il numero e stampa a video la stringa corrispondente  
}  
?>
```

Passiamo ora alla seconda parte dell'esercizio. Occorre estrarre un altro numero compreso tra 0 e 2. Usiamo di nuovo al funzione rand:

```
$num3 = rand(0,2);
```

A seconda del numero estratto, deve essere effettuata un'operazione diversa. Visto che i possibili numeri estratti sono tre, possiamo utilizzare l'istruzione SWITCH. Avremmo potuto usare anche l'IF, ma il codice si sarebbe appesantito troppo. L'istruzione SWITCH funziona in questo modo:

```
switch(variabile)  
{  
    case valore1:  
        // istruzioni  
        break;  
    case valore2:
```

```

        istruzioni
        break;
case valoreN:
        istruzioni
        break;
default:
        istruzioni
}

```

L'istruzione esamina il valore della variabile e a seconda del valore esegue il codice in corrispondenza del valore stesso. Se la variabile assume un valore che non è nell'elenco scritto, è eseguita l'operazione di default. L'istruzione break tra un valore e l'altro è obbligatoria.

Nel nostro caso:

```

$num3 = rand(0,2);
switch($num3)
{
    case 0:
        echo "zero<br>";
        break;
    case 1:
        echo "uno<br>";
        break;
    case 2:
        echo "due<br>";
        break;
    default:
        "numero non definito";
}

```

A seconda del numero estratto stampiamo la rappresentazione del numero in caratteri. Il programma completo è il seguente:

```

<?php
$num1 = rand(0,100);
$num2 = rand(0,100);

echo $num1."<br>";
echo $num2."<br>";

$n = 50;
if( $num1 < $n || $num2 < $n)
{
    for($i = 1; $i < $n; $i++)
    {
        if($i == 5)
            continue;
        $somma = $somma + $i;
    }
}

```

```

    }
}
else if($num1 > $n && $num2 > $n)
{
    // estrai un numero tra 0 e 2
    // controlla il numero e stampa a video la stringa corrispondente
    $num3 = rand(0,2);
    switch($num3)
    {
        case 0:
            echo "zero<br>";
            break;
        case 1:
            echo "uno<br>";
            break;
        case 2:
            echo "due<br>";
            break;
        default:
            "numero non definito";
    }
}
?>

```

Esercizio 2

Implementare un servizio che estrae un numero random tra 1 e 10. Se il numero è minore di 5, il programma stampa il doppio del numero, se è maggiore di 5, stampa la metà. Se il numero è uguale a cinque, stampa la somma dei primi 5 numeri. Utilizzare la funzione rand() per generare il numero random.