Laboratorio Progettazione Web
Il linguaggio PHP – Lezione 6

Andrea Marchetti – IIT-CNR
andrea.marchetti@iit.cnr.itA
2012/2013

Assegnamento

- L'assegnamento ad una variabile è il classico variabile = valore;
  - $a = 3 * 2;
- Ci sono altre forme di assegnamento
  - $x +=10; // equivalente a $x = $x + 10;
  - $x -=10; // equivalente a $x = $x - 10;
  - $x++; // equivalente a $x=Sx+1;
  - ++$x; // equivalente a $x=Sx+1;
  - $x--; // equivalente a $x=$x-1;
  - --$x; // equivalente a $x=$x-1;

Data e ora

- In molti programmi dinamici occorre fornire la data e l'ora correnti.
- Sono disponibili varie funzioni PHP per reperire dal server queste informazioni.
  - getdate() che restituisce un array contente data e ora corrente
  - date("formato") che restituisce la data nel formato definito.

```
$dataoggi=date("d/m/Y");
echo $dataoggi; // Visualizzerà la data odierna
```

Istruzione date()

date("*formato*") dove *formato* può contenere

Anno	Mese	Giorno	Ora	Minuti	Secondi
Y=anno su 4 cifre	**n**=mese numerico	**d**=giorno del mese su due cifre	**H**=ora su due cifre	**i**=minuti	**s**=secondi
y=anno su 2 cifre	**m**=mese numerico su due cifre	**j**=giorno del mese	**G**=ora		
	F=mese testuale	**w**=giorno della settimana			
	M=mese testuale su tre lettere	**l**=giorno della settimana testuale			
		D=giorno della settimana su tre lettere			

ESEMPIO: date(“d/m/Y");

Comandi su linee multiple

- A volte specie per produrre codice HTML si richiede di usare molti comandi echo
- PHP offre due alternative
- Uso degli apici

```
echo “Questo è il titolo

Questo è il sottotitolo
Qui inizia il paragrafo”;

$p = “Una lunga
stringa su più righe”;
```

Comandi su linee multiple

- Uso dell'operatore <<< detto heredoc

```
echo <<< THE_END
Questo è il titolo

Questo è il sottotitolo
Qui inizia il paragrafo
THE_END;

$p = <<< XYZ
Una lunga
stringa su più righe
XYZ;
```

- Utile se voglio mantenere la formattazione

Controllo del flusso

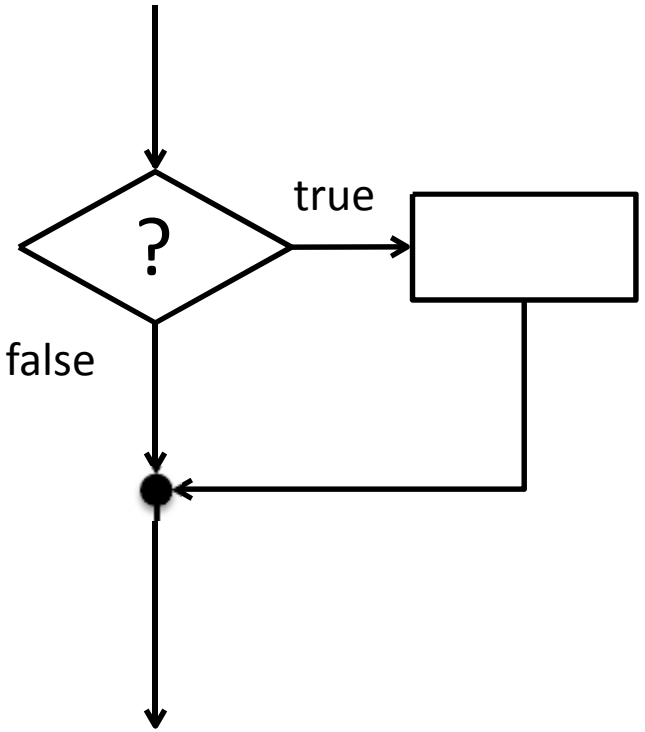
- Istruzioni condizionali
  - if
  - if else
  - if elseif
  - ?:
  - switch
- Cicli/Iterazionni
  - definite
    - for
  - indefinite
    - while
    - do while

istruzione if

if (*condizione*) {

istruzioni da eseguire se la condizione è vera

}

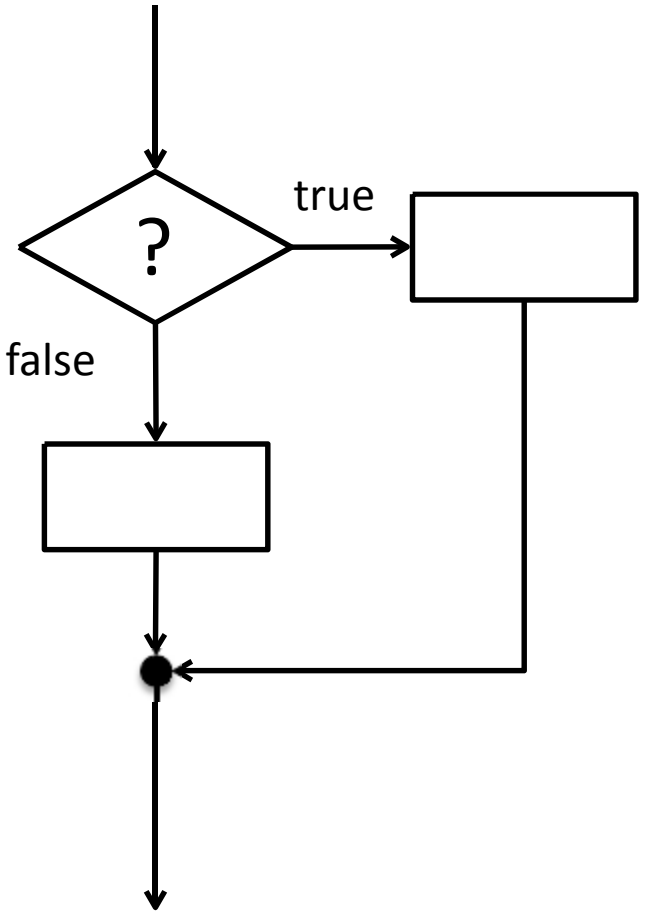



```
/* controllo che il denominatore
   sia != 0;  */
if ($a != 0)  $c = 1/$a;
```

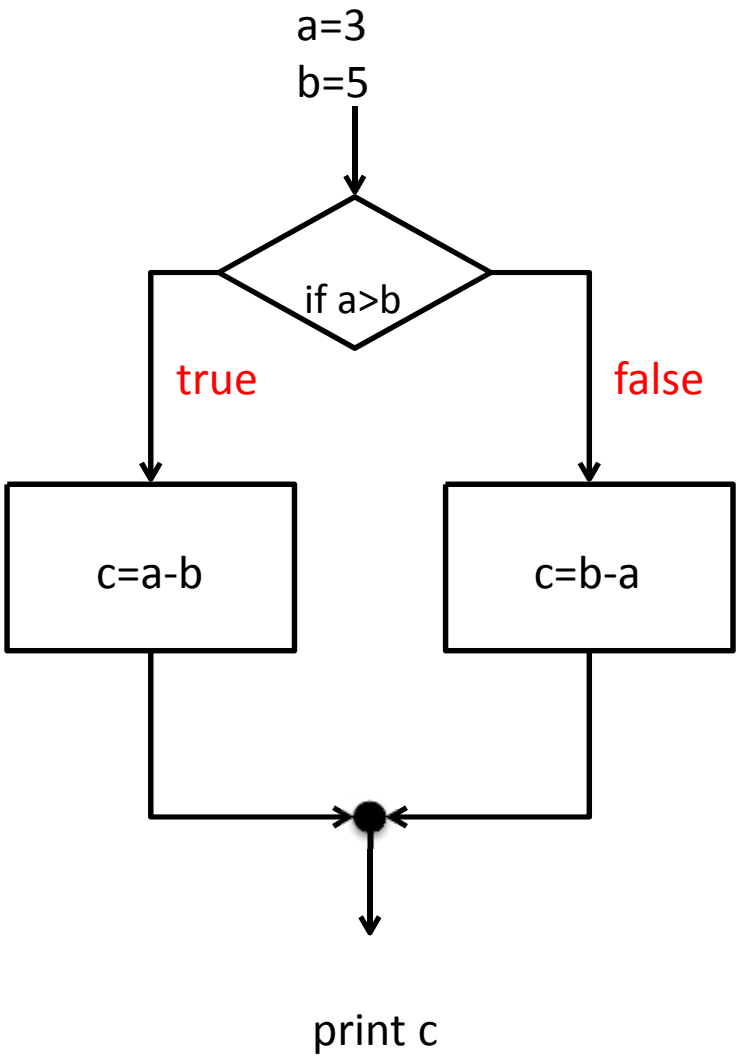
istruzione if else

if (*condizione*) {
    istruzioni da eseguire se la condizione è vera
}
else {
    istruzioni da eseguire se la condizione è falsa
}

La **condizione** è una **espressione booleana**



Esempio if else




```
<?php
$a=3;
$b=5;

if($a>$b)
{
    $c=$a-$b;
}
else
{
    $c=$b-$a;
}

echo "La differenza è $c";
?>
```

istruzione if else

Se abbiamo una sola istruzione le parentesi {} possono essere omesse

if (*condizione*) istruzione a;
else istruzione b;

```
<?php
$a=3;
$b=5;

if($a>$b)  $c=$a-$b;
else       $c=$b-$a;

echo "La differenza è $c";
?>
```

Operatore ? :

Operatore ? :

(condizione) ? espressione1 : espressione2;

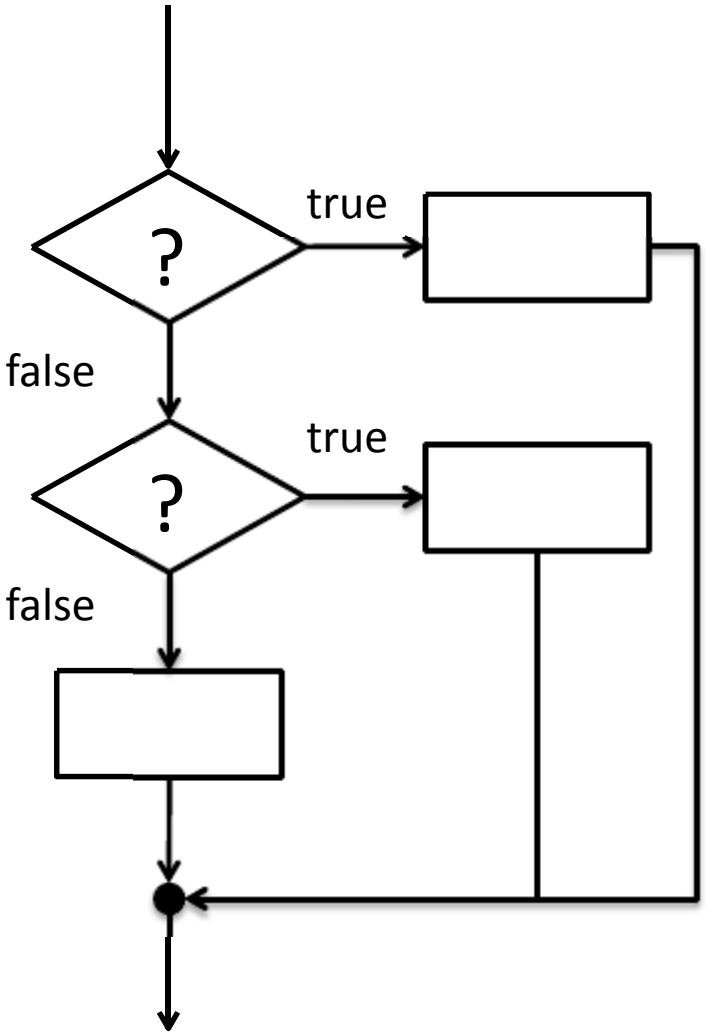
```
<?php
$a=3;
$b=5;

$c = ($a>$b) ? $a-$b : $b-$a;

echo "La differenza è $c";
?>
```

istruzione elseif

```
if (condizione1) {
    istruzioni da eseguire se la
    condizione1 è vera
}
elseif (condizione2) {
    istruzioni da eseguire se la
    condizione1 è falsa e la
    condizione2 è vera
}
else
{
    istruzioni da eseguire se
    entrambi le condizioni sono
    false
}
```



Operatori di confronto

Utili per creare espressioni booleane ovvero condizioni
if ($a%2 == 0) echo “La variabile A contiene un numero pari”;

Operatore	Descrizione	Esempio$a=4;$b=2;
==	uguale	$a==$b;//False
===	Identico (uguale anche il tipo)	$a===$b;//False
!=	differente	$a!=$b;//True
!==	non identico	$a!==$b;//True
>	maggiore	$a>$b;//True
<	minore	$a/$b;//False
>=	maggiore uguale	$a>=$b;//True
<=	minore uguale	$a<=$b;//False

Operatori logici

Utili per combinare espressioni booleane in espressioni boolenane complesse

if ($a%2 == 0 and $a >=0) { echo “La variabile A contiene un numero pari positivo”; }

Operatore	Descrizione
and	vero se e solo se entrambi gli argomenti sono veri
or	vero se almeno uno è vero
!	vero se l’argomento è falso
xor	vero se solo uno dei due è vero
&&	come and ma con ottimizzazione di valutazione del primo argomento
||	come or con ottimizzazione di valutazione del primo argomento

Istruzione switch

- Questa istruzione è utile quando una variabile può assumere un certo numero di valori noti a priori
  - ad esempio una variabile che prende i giorni della settimana

```
switch (espressione){
    case valore1 : istruzione; break;
    case valore2 : istruzione; break;
    ....
    default: istruzione;
}
```

Switch - esempio

```
<?php
$voto = rand(1,10); // genero un voto da 1 a 10

switch ($voto){
    case 10: echo "Ottimo";       break;
    case   9: echo "Distinto";     break;
    case   8: echo "Buono";        break;
    case   7: echo "Discreto";     break;
    case   6: echo "Sufficiente"; break;
    default: echo "Insufficiente";
}
?>
```

Istruzioni di ciclo o iterazioni

- L'iterazione è una struttura che consente di ripetere più volte l'esecuzione di un blocco di istruzioni
- Abbiamo due tipi di iterazione
  - iterazioni definite: sappiamo a priori il numero di cicli
  - iterazioni indefinite: il numero di iterazioni dipende da un evento non noto a priori

Ciclo definito for

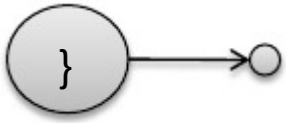
Assegnazione del valore iniziale alla variabile di ciclo

Test su condizione di fine ciclo

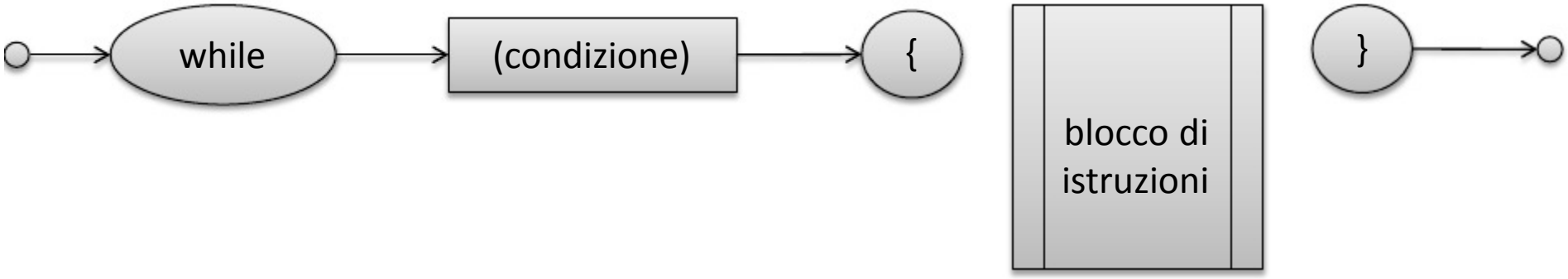
Incremento del valore della variabile di ciclo

sappiamo a priori quante volte
sarà eseguito il blocco di istruzioni

```
for ($i=0; $i<=10; $i++){
   echo $i;
}
```



Ciclo indefinito while



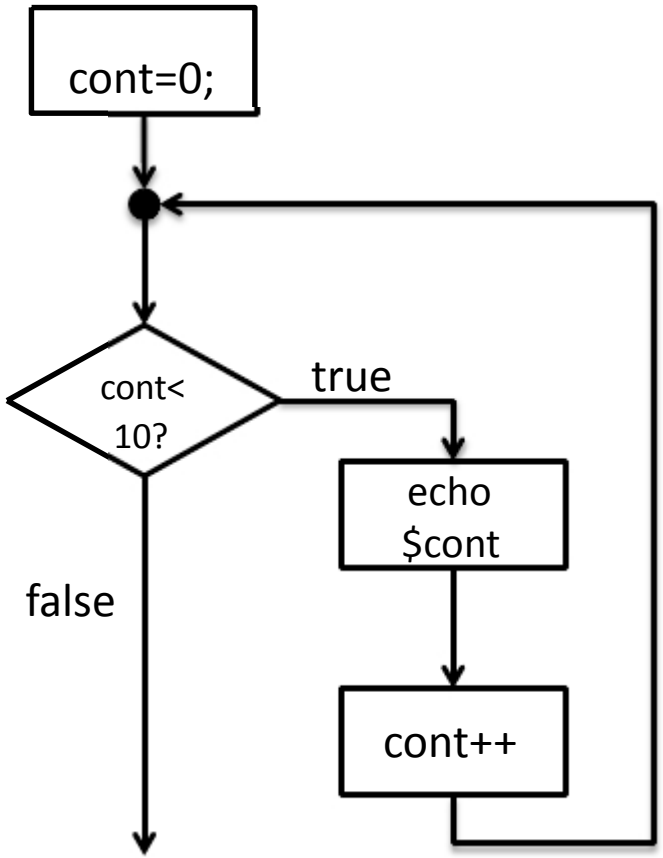
- Il blocco di istruzioni viene ripetuto fino a quando la condizione viene valutata a TRUE
- Per evitare cicli infiniti assicurarsi che nel blocco di istruzioni ce ne sia una che permetta di far scattare la condizione a FALSE
- Ricordarsi inoltre di inizializzare prima del while la variabile che determina la condizione

Ciclo indefinito while

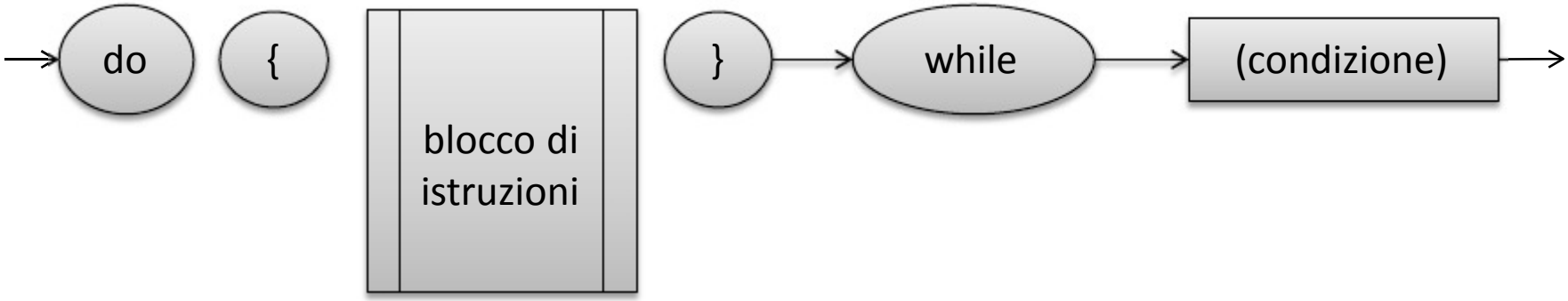
```
<?php
$cont=0;                    // inizializzazione contatore
                               all'esterno del ciclo

while ($cont<10) {    // test del contatore
    echo $cont;

$cont++;              // aggiornamento contatore
                         all'interno del ciclo
}
?>
```



Ciclo indefinito do while



Il blocco di istruzioni viene ripetuto fino a quando l'espressione viene valutata a TRUE; rispetto all'istruzione while la condizione di uscita viene testata in fondo, quindi la sequenza di istruzioni viene eseguita almeno una volta

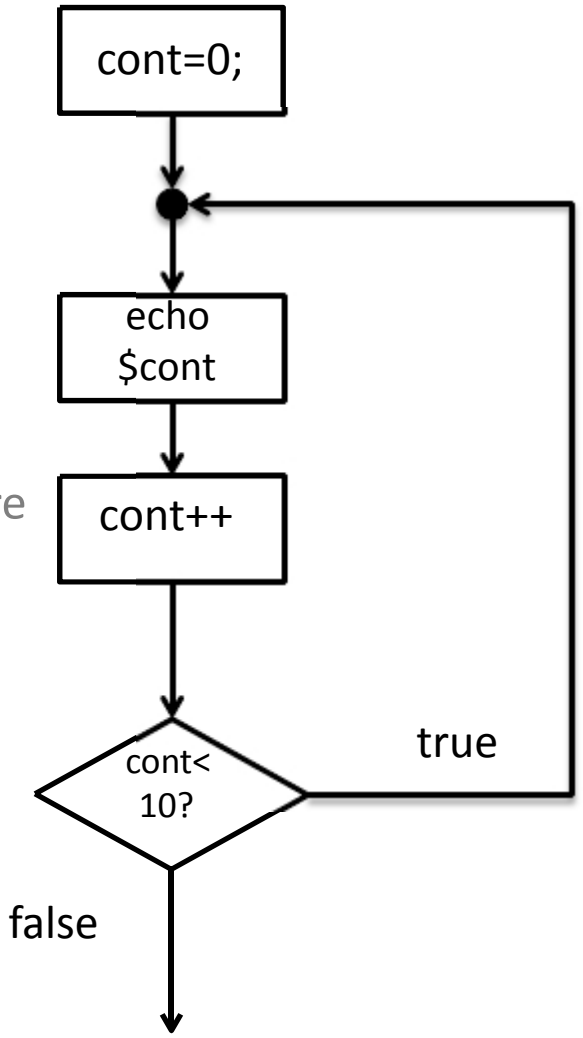
Ciclo indefinito while

```
<?php
$cont=0;                  // inizializzazione del contatore
                             all'esterno del ciclo

do {
    echo $cont;
    $cont++;              // aggiornamento del contatore
                             all'interno del ciclo

} while ($cont<10)        // test del contatore

?>
```



Manuale online

- Manuale online ufficiale:
  - http://www.php.net/manual/en/
- Manuale del linguaggio:
  - http://www.php.net/manual/en/langref.php
- **Descrizione delle singole funzioni**:
  - http://www.php.net/manual/en/funcref.php
  - Funzioni per gestire stringhe
    - http://www.php.net/manual/en/book.strings.php

Caratteri di escaping nelle stringhe

Carattere	Significato
\n	nuova linea
\t	carattere di tabulazione
\"	doppio apice
\'	apice singolo
\\	back slash
\xXX	codice esadecimale da 00 a FF di un carattere (ad esempio \xA9 per il carattere ©)

Esercizio su for

- Scrivere tutti i prefissi di una stringa
- Esempio considero la stringa “cane”, tutti i prefissi sono:
  - c
  - ca
  - can
  - cane
- Fare uso delle funzioni
  - strlen() che calcola la lunghezza di una stringa
  - substr() che estrae una sottostringa
- Cercare la definizione online
  - navigare sul manuale
  - cercare su google: substr php

Soluzione for

```
$stringa="andrea marchetti";

for($lun=1; $lun <= strlen($stringa); $lun++){
   echo substr($stringa,0,$lun)."\n";
}
```

Esercizio su while

- Estrarre numeri casuali tra -1000 e 1000 fintanto che se ne ottengano esattamente 10 pari positivi.

  Stampare a video i numeri dispari e il numero totale di estrazioni

Soluzione while

```
$nEstrazioni = 0;
$nSuccessi=0;  // inizializzo la variabile di iterazione

while ($nSuccessi<10){ // condizione di fine iterazione
   $estrazione = rand(-1000,1000); // estrazione
   $nEstrazioni++;  // aggiorno il numero di estrazioni
/* controllo il risultato dell'estrazione e eventualmente
   incremento la variabile di iterazione */
   if ($estrazione%2 == 0 && $estrazione >=0) $nSuccessi++;
   else echo "Numero negativo o dispari $estrazione\n";
} // chiusura il ciclo

echo "\n\nIl numero di estrazioni(cicli) per ottenere 10 pari
  positivi =$nEstrazioni";
```

Esercizio su if-elseif

- Scrivere un programma in grado di eseguire la conversione in un giudizio di un voto numerico tra 0 e 10, generato casualmente, secondo il seguente schema:
  - voto minore di 5 = giudizio insufficiente
  - voto maggiore di 5 e minore o uguale a 6.5 = giudizio sufficiente
  - voto maggiore di 6.5 e minore o uguale a 7.5 = giudizio buono
  - voto maggiore di 7.5 = giudizio ottimo