

# Laboratorio Progettazione Web

## **Il linguaggio PHP – Le Istruzioni**

Andrea Marchetti– IIT-CNR

[andrea.marchetti@iit.cnr.it](mailto:andrea.marchetti@iit.cnr.it)

AA 2014/2015

# Ambiente di test

- Cercare PHP online per avere un interprete PHP

# 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 $x=$x+1;`
  - `++$x; // equivalente a $x=$x+1;`
  - `$x--; // equivalente a $x=$x-1;`
  - `--$x; // equivalente a $x=$x-1;`

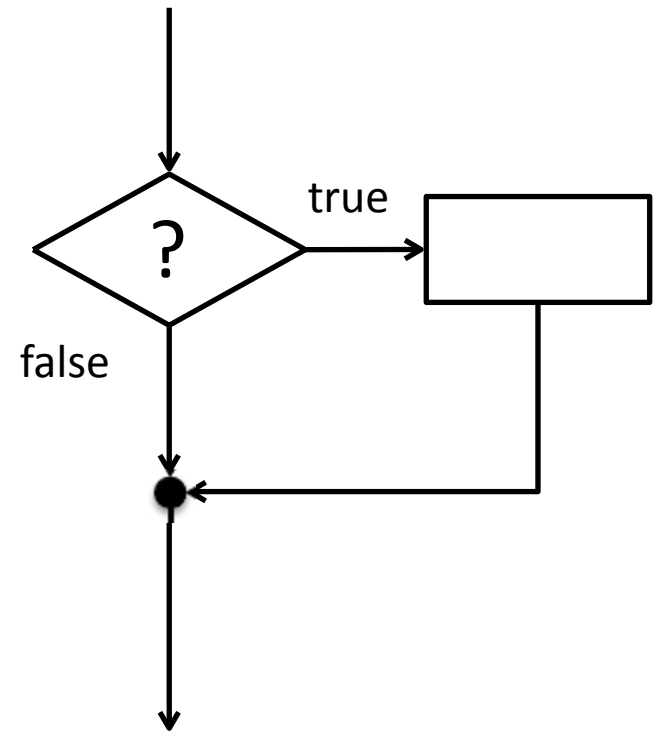
# Controllo del Flusso

- Istruzioni condizionali
  - if
  - if else
  - if elseif
  - ?:
  - switch
- Cicli/Iterazioni
  - Definite: for
  - Indefinite: while, do while
- Interruzione di flusso
  - Break, Continue, Exit

# istruzione if

```
if (condizione) {  
    istruzioni da eseguire se la  
    condizione è vera  
}
```

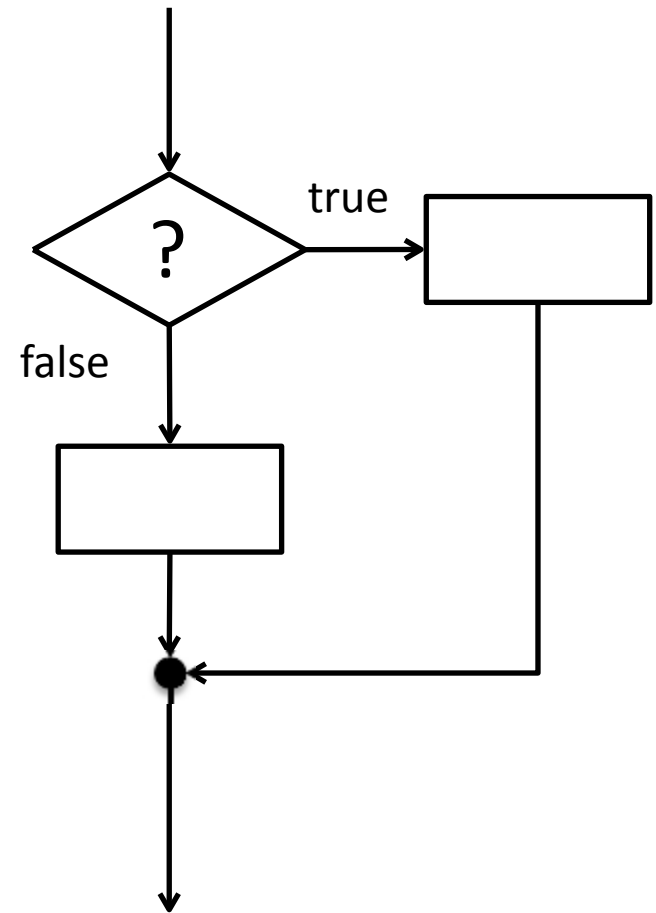
```
/* controllo che il denominatore  
   sia != 0; */  
if ($a != 0) $c = 1/$a;
```



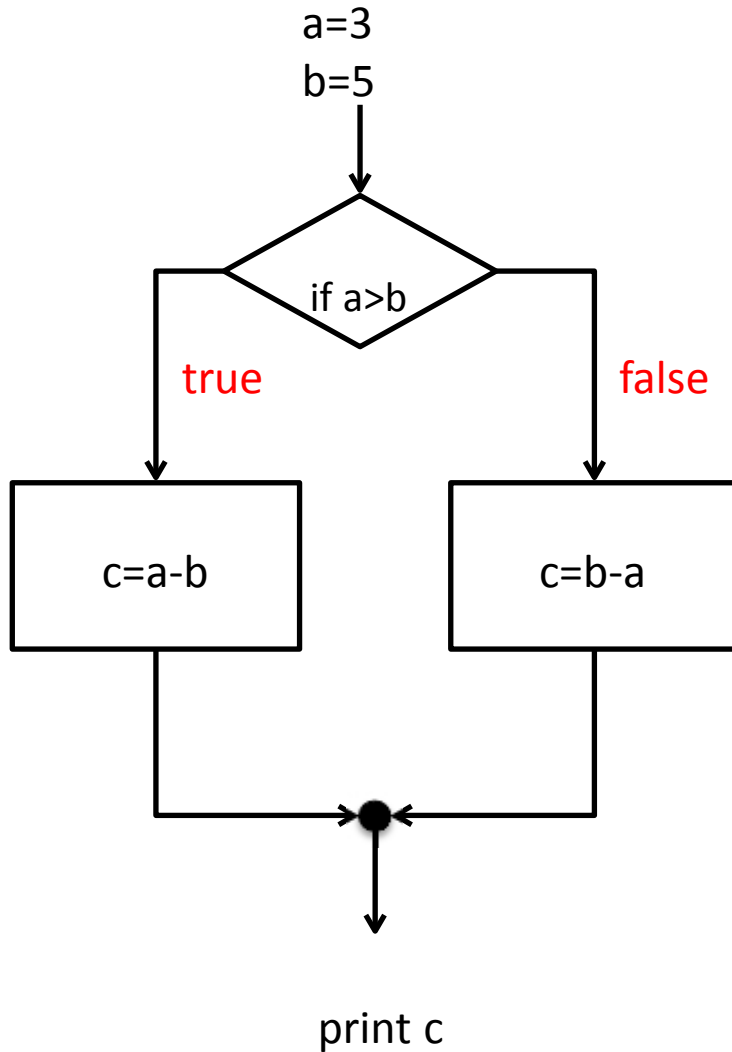
# istruzione if else

```
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;

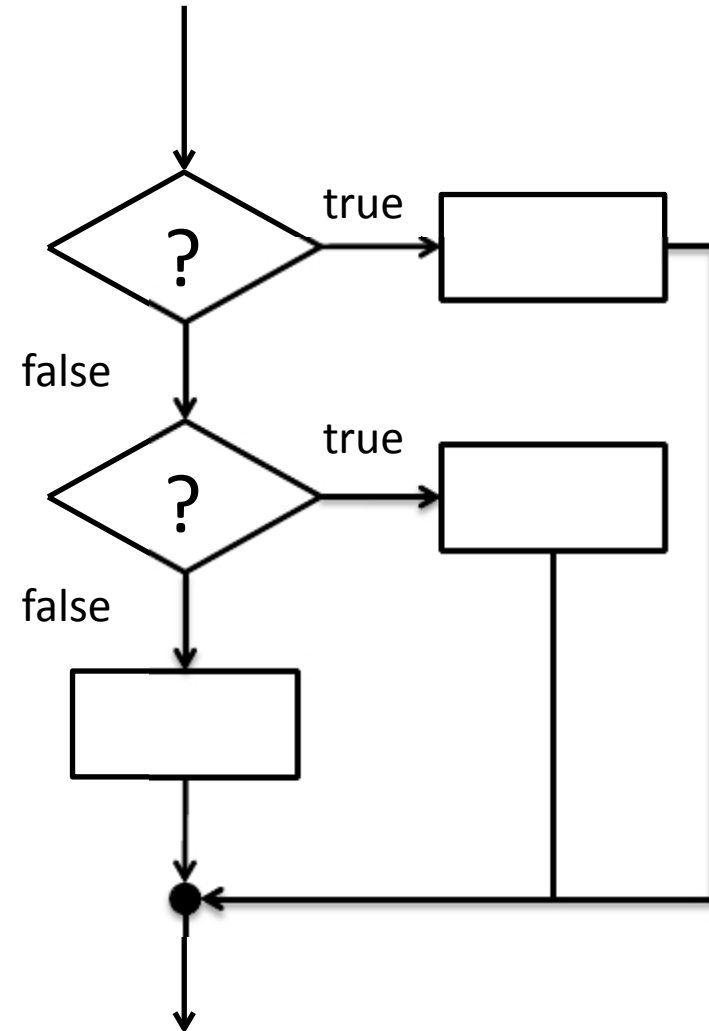
```
<?php  
$a=3;  
$b=5;
```

```
$c = ($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 <code>\$a=4;\$b=2;</code>
<code>==</code>	uguale	<code>\$a==\$b;//False</code>
<code>===</code>	identico(uguale anche il tipo)	<code>\$a=== \$b;//False</code>
<code>!=</code>	differente	<code>\$a!= \$b;//True</code>
<code>!==</code>	non identico	<code>\$a!== \$b;//True</code>
<code>&gt;</code>	maggiore	<code>\$a&gt;\$b;//True</code>
<code>&lt;</code>	minore	<code>\$a/\$b;//False</code>
<code>&gt;=</code>	maggiore uguale	<code>\$a&gt;=\$b;//True</code>
<code>&lt;=</code>	minore uguale	<code>\$a&lt;=\$b;//False</code>

# Operatori logici

Utili per combinare espressioni booleane in espressioni booleane 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 ottimizzazioni di valutazione del primo argomento
	come or con ottimizzazioni di valutazione del primo argomento

# Istruzione switch

- Questa istruzione è utile quando una variabile può assumere un numero multiplo 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/iterazione

- L'iterazione è un comando che consente di ripetere più volte l'esecuzione di un blocco di istruzioni
- Abbiamo due tipi di iterazioni
  - **Definite (For)**: quando all'inizio dell'iterazione si conosce il numero di cicli
  - **Indefinite (While)**: quando il numero delle iterazioni dipende da un evento non noto a priori

# Ciclo definito For

Si usa quando il numero di volte in cui dovrà essere eseguito il ciclo è noto all'inizio dell'iterazione

```
$x = rand(1,10);  
for ($i=1;$i<=$x;$i++) {  
    print("Ciao\n");  
    ...  
}
```

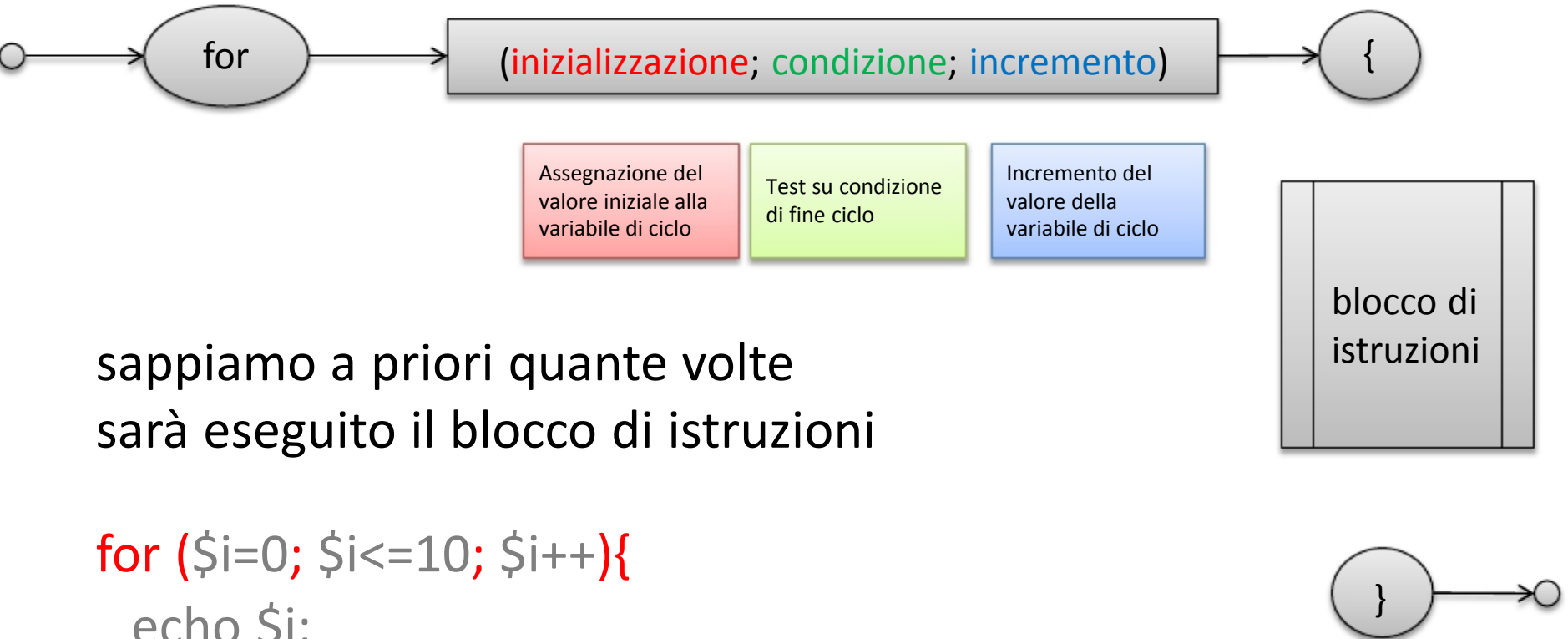


# Ciclo definito For

Cosa succede se nel corpo del for vado a modificare la variabile di iterazione?

```
for ($i=1;$i<=10;$i++) {  
    print("Ciao\n");  
    $i=1;  
}
```

# Ciclo definito for



sappiamo a priori quante volte sarà eseguito il blocco di istruzioni

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

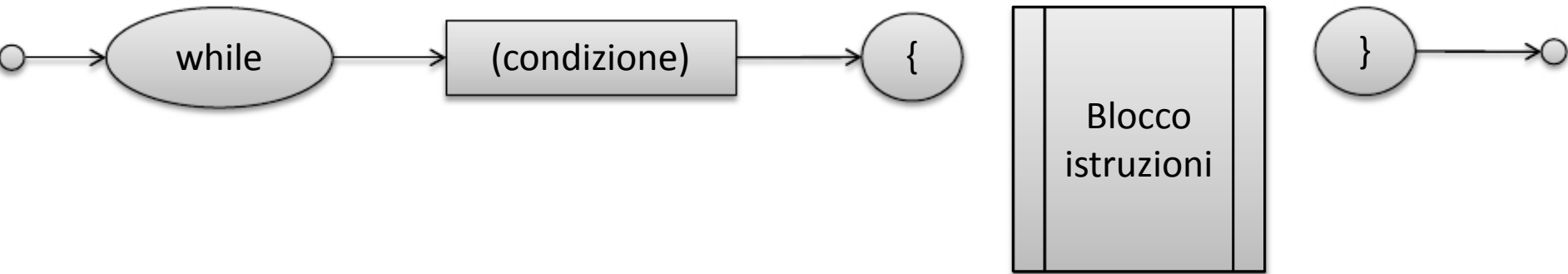
# Ciclo indefinito While

Si usa quando non conosciamo a priori il numero di volte in cui dovrà essere eseguito il ciclo

```
$x = rand(1,10);  
while ($x!=3) {  
    print($x. "-");  
    $x=rand(1,10);  
}
```

La condizione dipende da un operazione interna al ciclo

# Ciclo indefinito while



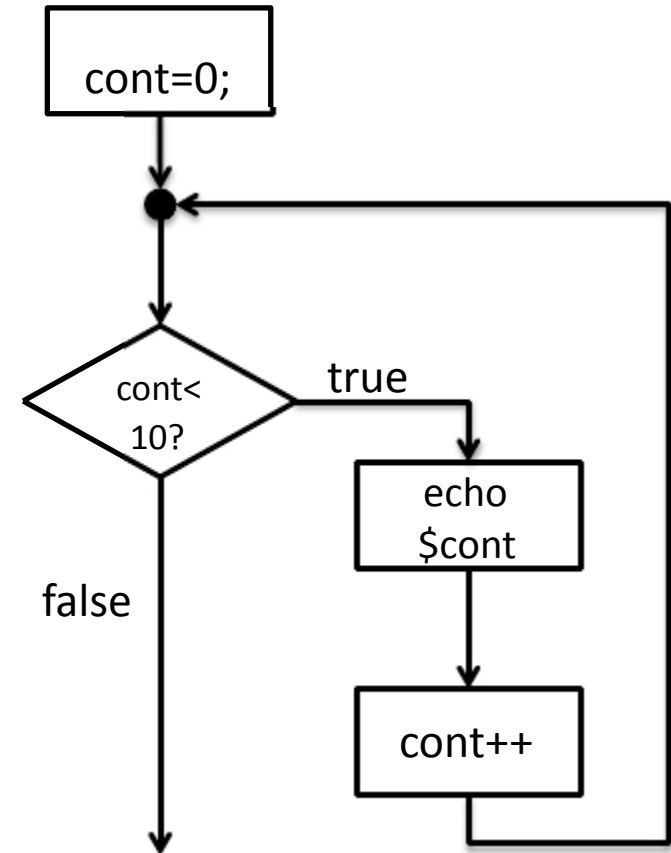
- Il blocco di istruzioni viene ripetuto fintanto che la condizione è TRUE
- Per evitare cicli infiniti assicurarsi che nel blocco istruzioni ci sia un'operazione che farà scattare la condizione a FALSE
- Ricordarsi di inizializzare prima del while la variabile che determina la condizione

# Ciclo indefinito while

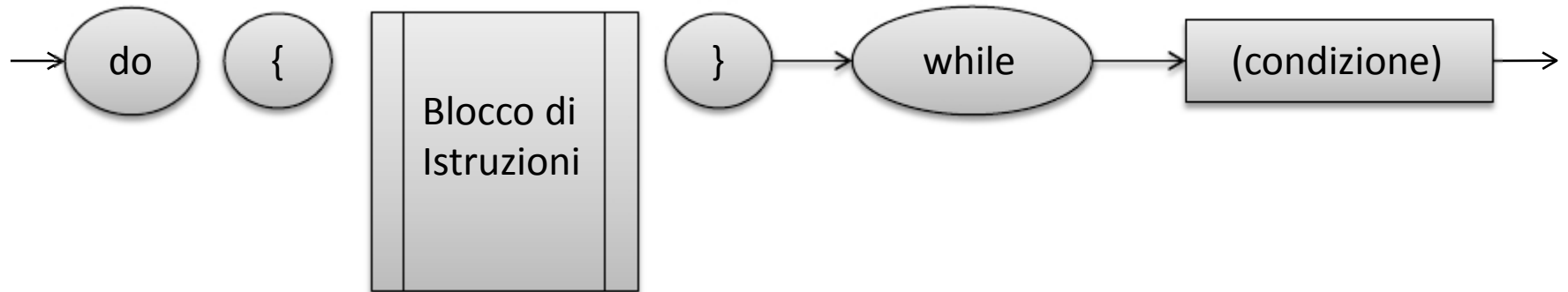
```
<?php
$cont=0;           // inizializzazione contatore
                   // all'esterno del ciclo

while ($cont<10) { // test del contatore
    echo $cont;
    $cont++;       // aggiornamento contatore
                   // all'interno del ciclo
}

?>
```



# Ciclo indefinito do while



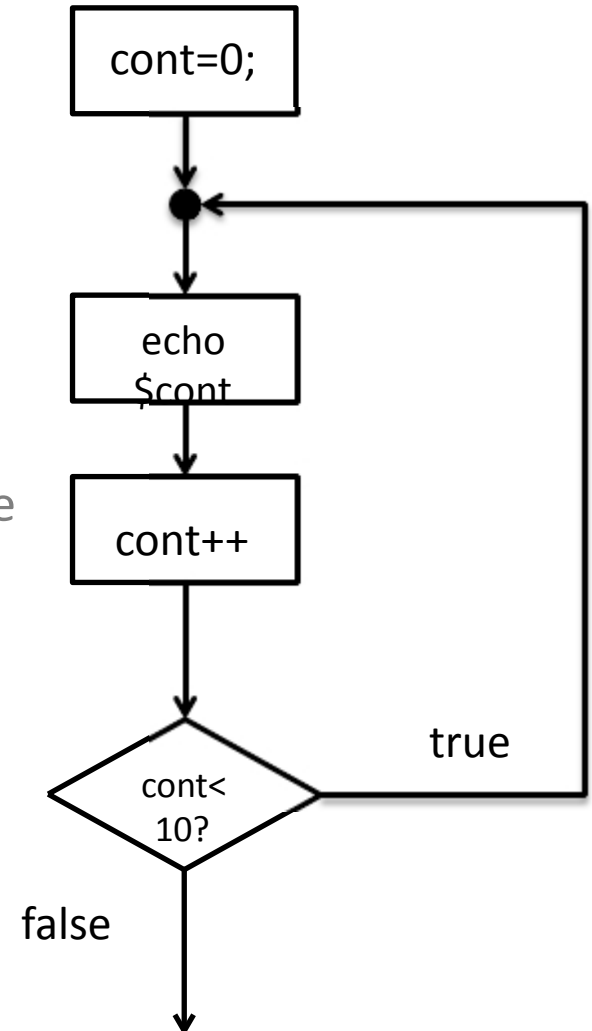
Rispetto all'istruzione while la condizione di uscita viene testata in fondo e non all'inizio, quindi almeno una volta il blocco di istruzioni viene eseguito

# Ciclo indefinito while

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

do {
    echo $cont;
    $cont++;       // aggiornamento contatore
                   // all'interno del ciclo
} while ($cont<10) // test del contatore

?>
```



# Continue, break, exit

- *continue* è impiegato all'interno delle strutture di iterazione per saltare il resto del ciclo corrente e riprendere l'esecuzione dalla verifica della condizione di controllo e quindi dall'inizio dell'iterazione successiva



# 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>

# Caratteri di escaping nelle stringhe

Carattere	Significato
<code>\n</code>	nuova linea
<code>\t</code>	carattere di tabulazione
<code>\"</code>	doppio apice
<code>\'</code>	apice singolo
<code>\\</code>	backslash
<code>\xXX</code>	codice esadecimale da 00aF di un carattere (ad esempio <code>\xA9</code> per il carattere ©)

# Esercizio

- Scrivere un programma che estratto un numero da uno 1 a 10 casualmente generi un giudizio secondo il seguente schema
  - Voto minore di 5 => giudizio insufficiente
  - Voto uguale a 6 => giudizio sufficiente
  - Voto tra 7 e 8 => giudizio buono
  - Voto superiore ad 8 => voto ottimo

# Esercizio

- Scrivere un programma che stampa tutti i prefissi di una stringa preassegnata
- Esempio se considero la stringa «cane» tutti i prefissi sono: c,ca,can,cane
- Fare uso delle funzioni
  - `strlen($stringa)` calcola la lunghezza di una stringa
  - `substr($stringa,0,2)` estrae una sottostringa da una data
- Cercare la definizione delle due funzioni online

# Soluzione

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

Esercizio: data una stringa riscriverla al contrario

# Esercizio

- Estrarre numeri casuali tra -100 e 100 fintanto che ne ottengo esattamente 10 positivi pari
- Stampare i numeri dispari estratti e il numero totale di estrazioni

# Soluzione

```
/* Impostazione iniziale */
$numEstrazioni    = 0;
$numPositiviPari  = 0;

while ($numPositiviPari < 10){ // condizione di fine it.
    $x = rand(-100,100);
    $numEstrazioni++;
    if($x%2==0 and $x>0) $numPositiviPari++;
    elseif (!$x%2==0) print("Numero dispari $x\n");
}
print("Sono state effettuate $numEstrazioni estrazioni");
```