

Laboratorio Progettazione Web

Il linguaggio PHP – Le Istruzioni

Andrea Marchetti– IIT-CNR

andrea.marchetti@iit.cnr.it

AA 2015/2016

Ambiente di test

- PHP online per avere un interprete PHP
 - [Php Online](#)
 - [PHP Tester](#)
 - [PHP Fiddle](#)
 - [Write PHP online \(simple\)](#)

Assegnamento

nome variabile = espressione;

```
$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;
```

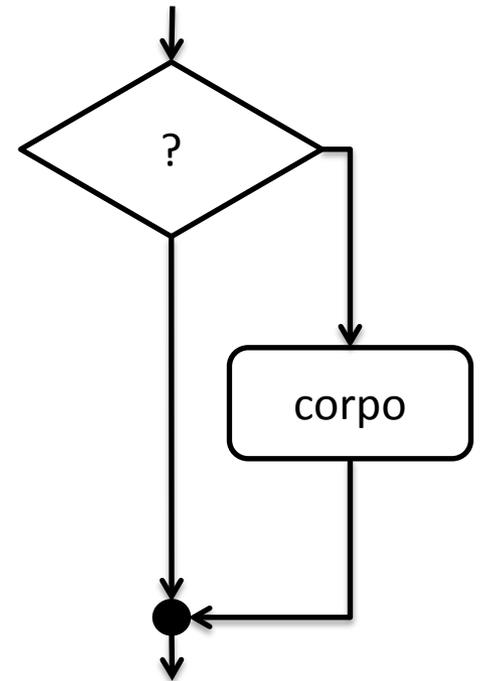
```
--$x; // equivalente a $x=$x-1;
```

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

IF

```
if (condizione) { // exp booleana  
    sequenza di istruzioni  
    da eseguire se la condizione  
    è vera  
}
```

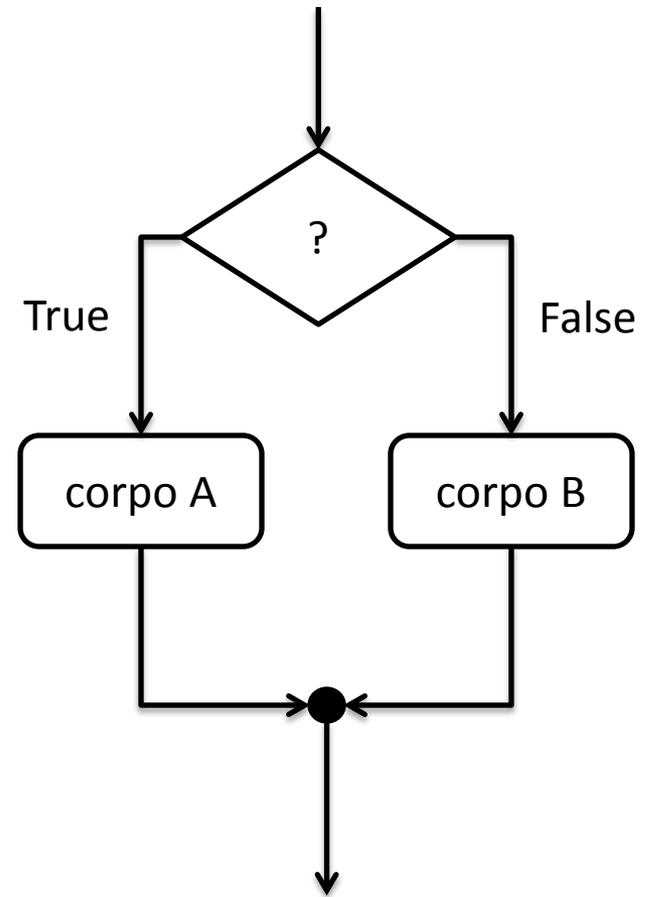


IF

```
$febbraio = 28;  
$anno     = 2016;  
  
if ($anno%4==0) {  
    $febbraio = 29;  
}  
  
print ($febbraio);
```

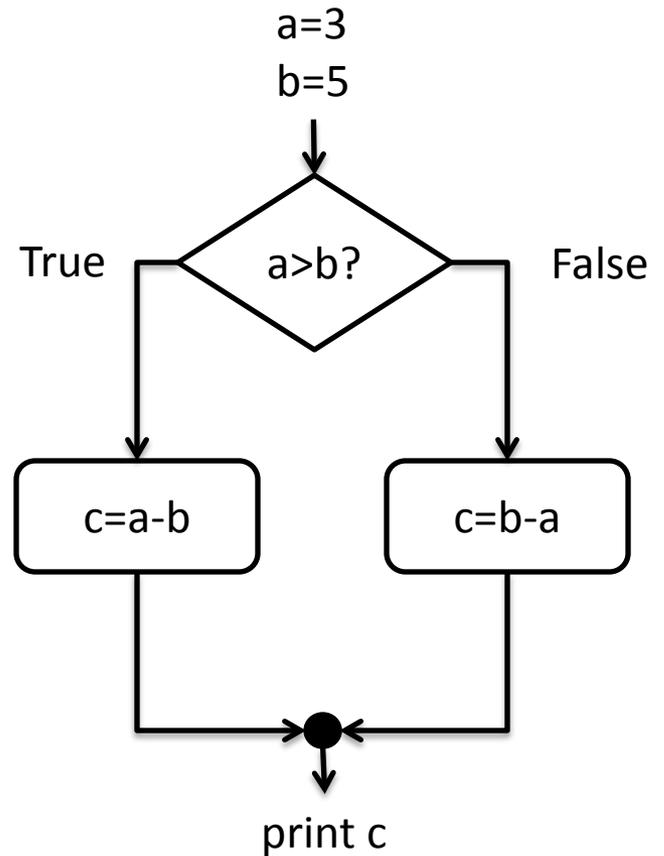
IF ... ELSE

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



IF ... ELSE

```
$a=3;  
$b=5;  
if($a>$b){  
    $c=$a-$b;  
}  
else{  
    $c=$b-$a;  
}  
echo "La differenza è $c";
```



Istruzioni condizionali

- Se il corpo IF e/o ELSE contiene una sola istruzione le parentesi possono essere omesse

```
$a=3;  
$b=5;  
  
if($a>$b)    $c=$a-$b;  
else        $c=$b-$a;  
  
echo "La differenza è $c";
```

Operatore ? :

- (condizione) ? espressione1 : espressione 2

```
$a=3;
```

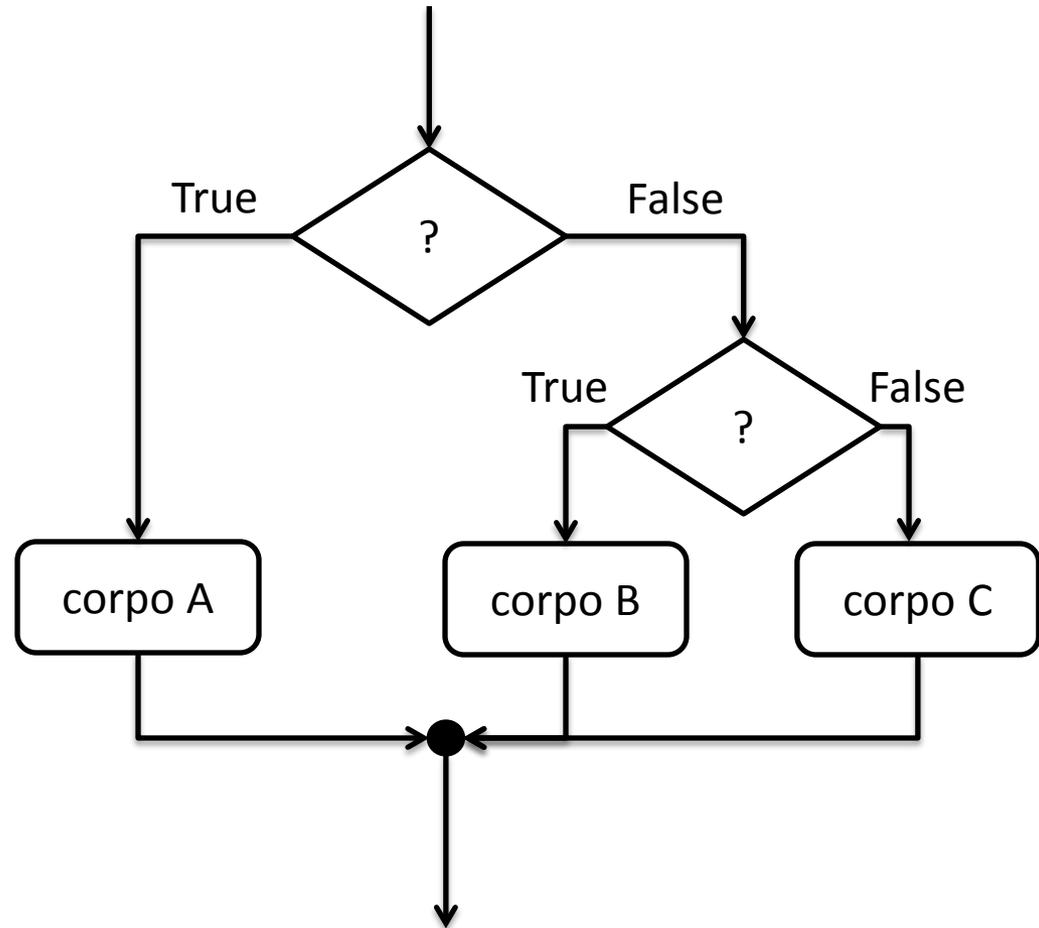
```
$b=5;
```

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

```
echo "La differenza è $c";
```

IF ... ELSEIF

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



Operatori di confronto

- Utili per creare **espressioni booleane** ovvero condizioni

```
if ($a%2 == 0) echo "$a è pari";
```

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

Operatori logici

- Utili per combinare **espressioni booleane**

```
if ($a%2==0 and $a>0) echo "$a è 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. Se il primo è falso non si valuta il secondo
	come or con ottimizzazione di valutazione del primo argomento. Se il primo è vero non si valuta il secondo

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

- Questa istruzione è utile quando una variabile può assumere più di due valori

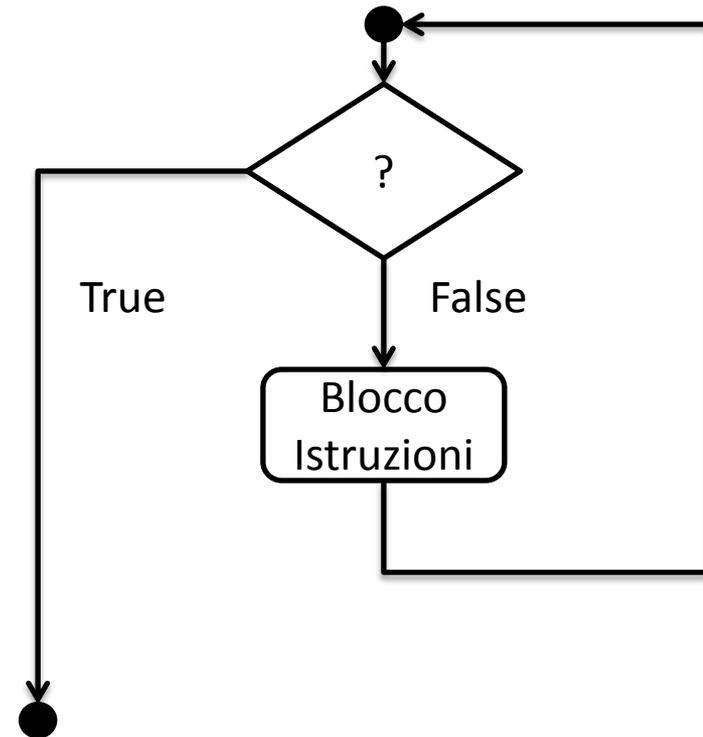
```
switch(espressione) {  
    case valore1 : istruzioni; break;  
    case valore2 : istruzioni; break;  
  
    default: istruzioni;  
}
```

SWITCH

```
$giornoSettimana = rand(1,7);  
switch($giornoSettimana ) {  
    case 1 : $giorno = "Lunedì";      break;  
    case 2 : $giorno = "Martedì";     break;  
    case 3 : $giorno = "Mercoledì";   break;  
    case 4 : $giorno = "Giovedì";     break;  
    case 5 : $giorno = "Venerdì";     break;  
    case 6 : $giorno = "Sabato";      break;  
    case 7 : $giorno = "Domenica";    break;  
    default: print("errore");  
}  
print("Il $giornoSettimana è $giorno");
```

Istruzioni di ciclo/iterazione

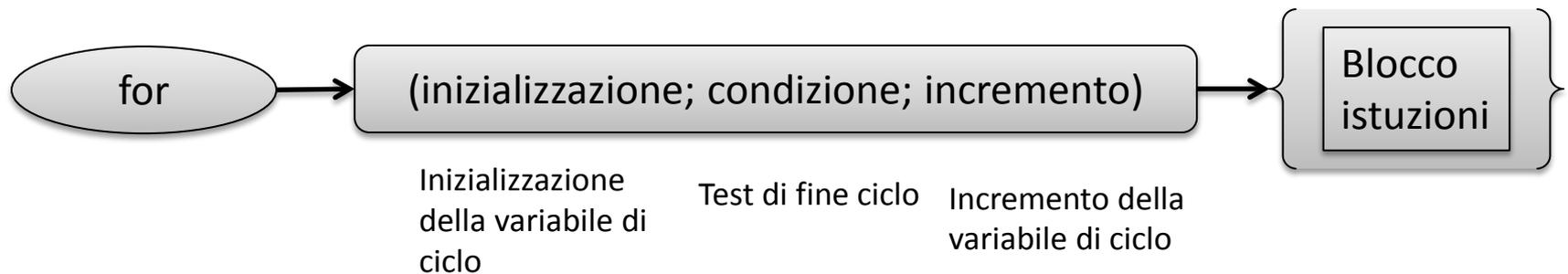
- L'iterazione è un comando che consente di ripetere più volte l'esecuzione di un **blocco di istruzioni**
- Attenzione a non creare dei cicli infiniti



Istruzioni di ciclo/iterazione

- 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



- Il blocco di istruzioni viene ripetuto per un numero di volte noto
- Per evitare cicli infiniti assicurarsi che nel blocco istruzioni non ci sia un'istruzione che modifichi la variabile di ciclo

Ciclo definito For

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

```
// Stampa tabelline
$n = 2;

for ($i=1;$i<=10;$i++){
    $prodotto=$i*$n;
    print( "$i*$n=$prodotto</br>" );
}
```

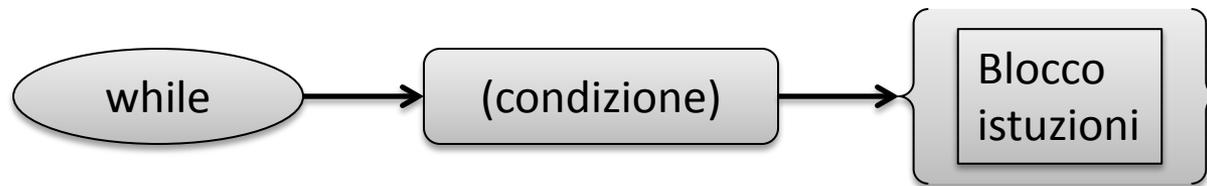
Ciclo definito For

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

```
// Stampa tabelline
$n = 2;

for ($i=1;$i<=10;$i++) {
    $prodotto=$i*$n;
    print ( "$i*$n=$prodotto</br>" );
    $i=1;
}
```

Ciclo indefinito WHILE



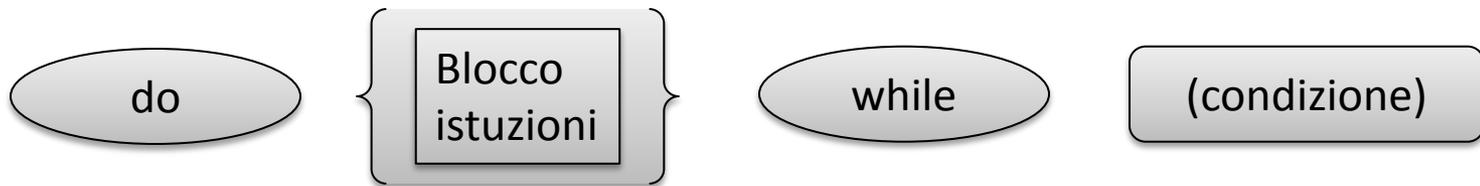
- Il blocco di istruzioni viene ripetuto fintanto che la condizione è TRUE
- Per evitare cicli infiniti assicurarsi che nel blocco istruzioni ci sia un'operazione che farà scattare la condizione a FALSE
- Ricordarsi di inizializzare prima del while la variabile che determina la condizione

Ciclo indefinito While

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

```
// Conta quanti numeri pari escono
$x = 0; // inizializzazione variabile ciclo
$cont = 0;
while ($x%2==0) {
    print("$x<br>");
    $cont++;
    $x=rand(1,100); // influenza il ciclo
}
print("Sono usciti $cont numeri pari");
```

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

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
- *break* consente di uscire dal ciclo ignorando la condizione e riprendere l'esecuzione dall'istruzione successiva al ciclo
- *exit* consente di terminare il programma

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>	nuovalinea
<code>\t</code>	carattereditabulazione
<code>\"</code>	doppioapice
<code>\'</code>	apicesingolo
<code>\\</code>	backslash
<code>\xXX</code>	codicesadecimaleda00aFFdiuncarattere(adesempio\xA9 perilcarattere©)

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");
```