

ISTRUZIONI DI CONTROLLO: IF-ELSE

Un programma viene eseguito rispettando l'ordine in cui le istruzioni vengono scritte dal programmatore, partendo dalla prima istruzione e fino a quando non si raggiunge l'ultima istruzione. Alcune volte però, in base al verificarsi o meno di una certa condizione, l'istruzione da eseguire potrebbe variare. Per esempio posso calcolare il risultato dell'operazione X/Y solo se $Y > 0$. In particolare in questi casi si vuole deviare il flusso del programma in base al risultato di una condizione (nell'esempio precedente la condizione è $Y > 0$).

Per fare questo, basta porre delle condizioni che, se verificate, eseguono un pezzo di codice o altrimenti ne eseguono un altro; queste istruzioni particolari che permettono di incanalare il flusso si chiamano *strutture di controllo condizionale* o, più genericamente, *istruzioni condizionali* o *controlli* e si scrivono come **if-else** dove l'if viene affiancato da una condizione descritta tra parentesi tonde. Se la condizione è vera allora vengono eseguite tutte le istruzioni che susseguono l'if fino all'ultima istruzione prima dell'else (se questo è stato inserito). A differenza dell'if, il costruito **else** non prende nessuna condizione esplicita in quanto viene eseguito solo se la condizione all'interno dell'if è falso. Infatti, significa *altrimenti* e non può esistere se prima non c'è un **if**. Se all'interno dell'if e dell'else si vogliono inserire più istruzioni allora queste vanno chiuse nelle parentesi graffe.

Più nel dettaglio, la struttura di un programma con le istruzioni condizionali ha la forma

```
[... altre istruzioni ...]  
if (condizione)  
    istruzione  
[... altre istruzioni ...]
```

In questo caso se la condizione risulta vera, fa eseguire l'istruzione immediatamente successiva, altrimenti (se la condizione è falsa) si salta l'istruzione (od il blocco di istruzioni) facenti parti dell'if e si procede nell'esecuzione delle *altre istruzioni* successive, che possono essere la prosecuzione del programma o un semplice **else**, ad indicare la possibile alternativa all'if:

```
if (espressione)  
    istruzione1  
else  
    istruzione2
```

ESEMPIO 1: la divisione tra due numeri si scrive:

```
if ($y > 0){  
    $ris = $x / $y;  
    echo "$ris";  
}  
else  
    echo "$ris";
```

Questi costrutti consentono dunque di gestire il flusso del programma decidendo di eseguire una parte di codice oppure no (in questo caso nel programma abbiamo inserito solo un **if**), di fare eseguire una parte di codice in alternativa ad un'altra (in questo caso nel programma abbiamo inserito **if - else**) o di fare una scelta esclusiva tra più parti di codice (in questo caso nel programma abbiamo inserito **if - else if - else**).

```
if (espressione)  
    istruzione1  
else if (espressione)  
    istruzione2  
else  
    istruzione3
```

ESEMPIO 2: Programma che in base al voto preso all'esame stampa se è superato, superato con riserva oppure non superato.

```
if ($risEsame >=18)  
    echo "Complimenti hai superato l'esame";  
else if ($risEsame >=18)  
    echo "Devi sostenere l'orale per questo esame";  
else  
    echo "Non hai superato l'esame";
```

Generalmente dopo un **if** o un **else** viene eseguita **solo** la prima istruzione più vicina, regola che può creare dei problemi quando ci sono due o più istruzioni, ma alla quale si può porre rimedio ponendo tutte le istruzioni tra due parentesi graffe (vedi ESEMPIO 1).

IF ANNIDATI. Bisogna porre particolare attenzione all'uso dei blocchi if annidati, come per esempio:

```
if (risultato_esame < 18)
    if (risultato_esame < 15)
        echo "Non hai superato l'esame";
else echo ("Sei stato ammesso all'orale");
```

In questo caso potrebbe essere difficile capire quale sia il significato dell'ultimo else. Se non capiamo esattamente il codice, conviene ricordarsi la regola secondo la quale *"l'else si riferisce sempre all'if più vicino, a meno che non sia diversamente specificato da eventuali parentesi graffe"*, ma è comunque buona norma usare sempre le parentesi graffe anche quando si tratta di eseguire una sola istruzione, magari indentando (mettendo opportune spaziature) correttamente il codice. L'esempio precedente risulta sicuramente più chiaro se strutturato in questo modo:

```
if (risultato_esame < 18)
{
    if (risultato_esame < 15)
    {
        printf ("Non hai superato l'esame");
    } else {
        printf ("Sei stato ammesso all'orale");
    }
}
```