



Corso base - Lezione 8

http://www.vbsimple.net/base/bas_08.htm

- [Uno sguardo sommario alla logica di Boole.](#)
- [Introduzione al ciclo di controllo IF...THEN...ELSE...END IF.](#)
- [Oggetto Checkbox.](#)

La programmazione in ogni linguaggio si basa sull'effettuazione di certi test e sulla determinazione di certi valori. Per esempio quando dobbiamo eseguire una semplice operazione aritmetica innanzitutto è necessario valutare il segno posto tra i due numeri. Faremo un controllo **SE** è un segno + **ALLORA** aggiungeremo il secondo numero al primo, se non si tratta di un segno di addizione controlleremo **SE** è un - indicante la sottrazione e via dicendo. Questi nostri test sono stati studiati dal matematico inglese [George Boole](#) che ha sviluppato una logica detta [logica booleana](#) secondo la quale ogni nostro interrogativo può essere risolto con una risposta SI o NO. Non esistono terze soluzioni.

La determinazione di queste certezze permette al programmatore di scrivere codici che, analizzando certe situazioni, rispondono in una maniera condizionata dai test all'interno del programma.

Il corrispondente in programmazione delle domande che hanno per risposta un SI o un NO booleano è un ciclo di test. Il più semplice ciclo di test è il ciclo IF, traduzione inglese del nostro SE. A questo segue sempre una valutazione di una condizione ovvero un test, una risposta ben determinata ed un'azione correlata alla risposta, comandata dal THEN, voce inglese dell'avverbio ALLORA.

La sintassi più semplice di un ciclo If è la seguente:

If [condizione](#) **Then** *istruzione*
scritto tutto su una riga.

SE, per esempio, IL SEGNO È +, **ALLORA** fai la somma dei valori.
In Visual Basic si scriverebbe **If** *Segno* = "+" **Then** *Somma* = *Valore1* + *Valore2*.

Se dovessimo impartire al programma un'istruzione di 2 o più righe dovremmo usare un'altra sintassi, leggermente diversa:

```
If condizione Then  
    istruzione1  
    istruzione2  
    ....  
    istruzioneN  
End If
```

La regola fondamentale è che se si utilizza questa seconda versione, quindi con più azioni,


tutte le azioni devono stare su una riga diversa da quella in cui si valuta la condizione. Inoltre per ogni IF aperto ci deve essere un END IF che ne delimita la chiusura.



Esiste anche una terza versione di questo ciclo:


```
If condizione Then  
    istruzioni...  
Else  
    istruzioni...  
End If
```

La parola ELSE indica l'avverbio **ALTRIMENTI** e delimita una seconda serie di istruzioni. Se il risultato del test è un SI verrà eseguito il primo corpo di istruzioni. Se invece il risultato è NO verrà eseguito soltanto il gruppo seguente al termine ELSE.

Il ciclo di controllo IF...THEN...ELSE....END IF sarà approfondito nel corso Intermedio.

Nel capitolo precedente abbiamo visto come inserire un pulsante  per effettuare delle determinate operazioni legate al momento del click su di esso.

Questa volta introdurremo un nuovo oggetto: la casella di controllo, detta [CheckBox](#)  la cui funzione principale è comandare processi per i quali esistono due sole soluzioni. Faremo allora un controllo se la casella ha un segno di spunta; nel caso ci sia il segno di spunta, la proprietà **Value** conterrà 1 corrispondente alla [costante](#)  vbChecked. Nel caso contrario la proprietà Value avrà il valore 0, costante vbUnchecked.

Passiamo subito alla pratica inserendo nel nostro form del Progetto1 una CheckBox  e denominiamola AttivaCheck. Lo stato di questa casella di controllo determinerà il funzionamento del pulsante che permette il conteggio dei click. Nel momento in cui verrà tolto il segno di spunta dalla casella il pulsante verrà disabilitato. Quando verrà rimesso il segno sulla casella di controllo il pulsante ritornerà disponibile.

Per questo motivo imposteremo il valore iniziale della casella di controllo a 1 - vbChecked. Impostiamo allora a tale valore la proprietà **Value** della casella. Apriamo la finestra del codice e aggiungiamo queste istruzioni:

```
1. Private Sub AttivaCheck_Click()  
2.     If AttivaCheck.Value = vbChecked Then  
3.         PulsanteConta.Enabled = True  
4.     Else  
5.         PulsanteConta.Enabled = False  
6.     End If  
7. End Sub
```

La routine appena creata è quella che risponde all'evento Click del controllo AttivaCheck. All'interno d'essa abbiamo un controllo della proprietà *Value* di *AttivaCheck*. Se tale valore è vbChecked allora verrà impostata la proprietà *Enabled* del pulsante *PulsanteConta* a True (Vero), in modo da rendere funzionante il pulsante. Se la proprietà Value **non** è vbChecked verrà bloccato il pulsante impostando la proprietà *Enabled* di *PulsanteConta* a False (Falso).

Eseguiamo il programma ed osserviamone il funzionamento.

Nel momento in cui toglieremo il segno di spunta dalla casella di controllo il pulsante diventerà grigio e non sarà possibile premerlo.



Per rendere nuovamente il pulsante funzionante dobbiamo cliccare nuovamente sulla casella di controllo. Solo quando il pulsante sarà attivato potrà rispondere ai nostri click.

L'oggetto CheckBox non introduce novità né nel funzionamento, né nelle proprietà e tantomeno negli eventi da esso generati.

Viene utilizzato in quei casi nei quali si vogliono dare due scelte soltanto, che verranno regolate dallo stato della sua proprietà Value.

[Fibia FBI](#)

11 Novembre 2000



[Torna alla settima lezione](#)

[Vai alla nona lezione](#)

