



ListBox e ComboBox



- Forniscono un elenco da cui l'utente può selezionare
- Vari stili, con scelta basata su
 - Quantità di dati da visualizzare
 - Spazio disponibile
 - Possibilità di aggiungere elementi all'elenco da parte dell'utente

Tipi di elenchi



- ListBox (riquadro di riepilogo) 
 - Prefisso lst
 - Riquadro di riepilogo con o senza barre di scorrimento
- ComboBox (casella combinata) 
 - Prefisso cbo
 - Si può consentire all'utente di aggiungere nuovi elementi
 - L'elenco si può “dispiegare” per mostrare gli elementi (casella combinata a discesa)

Le CheckedListBox non vengono trattate

Come appaiono gli elenchi



ListBox →

DropDown ComboBox

Simple ComboBox

DropDown List ComboBox

Scegliere il tipo di elenco



- Molti elementi, poco spazio
 - Dropdown ComboBox (casella combinata a discesa)
 - Dropdown List ComboBox (elenco a discesa)
- Pochi elementi
 - ListBox (riquadro di riepilogo)
- L'utente deve poter aggiungere elementi
 - Simple ComboBox (casella combinata semplice)
 - Dropdown ComboBox (casella combinata a discesa)

Proprietà Name e Text



- **ListBox**
 - Il nome (proprietà Name) viene visualizzato in progettazione
 - La proprietà Text è accessibile solamente in esecuzione e serve ad ottenere o cercare il testo dell'elemento correntemente selezionato.
- **ComboBox**
 - La proprietà Text è visualizzata e accessibile in progettazione ed in esecuzione
 - Se l'utente inserisce un nuovo dato nel ComboBox, VB lo memorizza temporaneamente in Text

Proprietà DropDownStyle



- Usata per indicare il tipo di ComboBox
 - A discesa = DropDown
 - Semplice = Simple
 - Elenco a discesa = DropDownList

Items Collection



- L'elenco di elementi in un ListBox o in un ComboBox è una *raccolta*
- Le raccolte sono oggetti con proprietà e metodi per
 - Aggiungere elementi
 - Eliminare elementi
 - Riferirsi a singoli elementi
 - Contare gli elementi
 - Eliminare tutti gli elementi

Proprietà Index



- Lo zero è il primo indice utilizzato per riferirsi ai singoli elementi della raccolta
- Posizione di un elemento in un elenco
 - 1° elemento Index = 0 (1-1=0)
 - 2° elemento Index = 1 (2-1=1)
 - 3° elemento Index = 2 (3-1=2)
- Si può quindi affermare che . . .
 - n° elemento Index = n-1

Proprietà SelectedIndex



- Usata per identificare l'elemento correntemente selezionato nell'elenco
- Se nessun elemento è selezionato allora SelectedIndex vale -1
- Viene usata per selezionare un elemento nell'elenco

Proprietà Items.Count



- Usato per determinare il numero di elementi nell'elenco

Si ricordi che:

Items.Count è il numero di elementi corrente nell'elenco *MA* la proprietà SelectedIndex e l'indice saranno di 1 inferiori.

Per esempio, se ci sono 20 elementi nell'elenco:

Items.Count=20

MA

Index =19 per riferirsi all'ultimo elemento nell'elenco

Usare la raccolta Items



- Usare l'indice dell'elemento per riferirsi ad uno specifico elemento (esistente) della raccolta
- Ricordarsi che il primo elemento della raccolta ha indice zero

```
lstSchools.Items(5) = "USC"
```

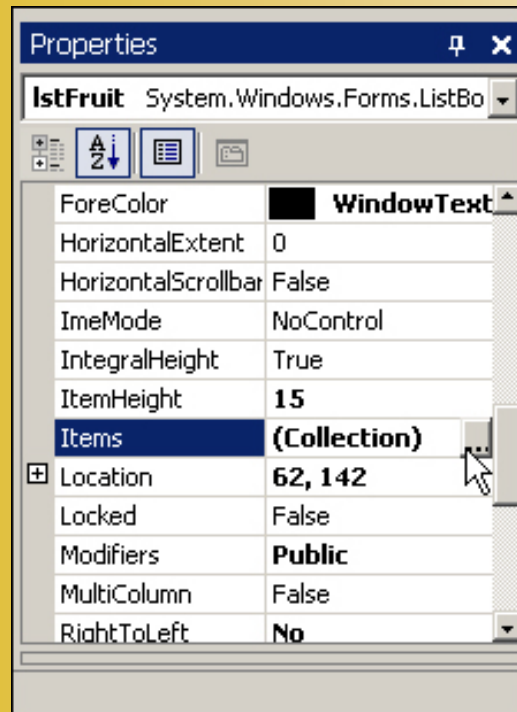
' La linea successiva si riferisce all'elemento selezionato
`strSelectedFlavor = lstFlavor.Items(lstFlavor.SelectedIndex)`

Riempire un elenco

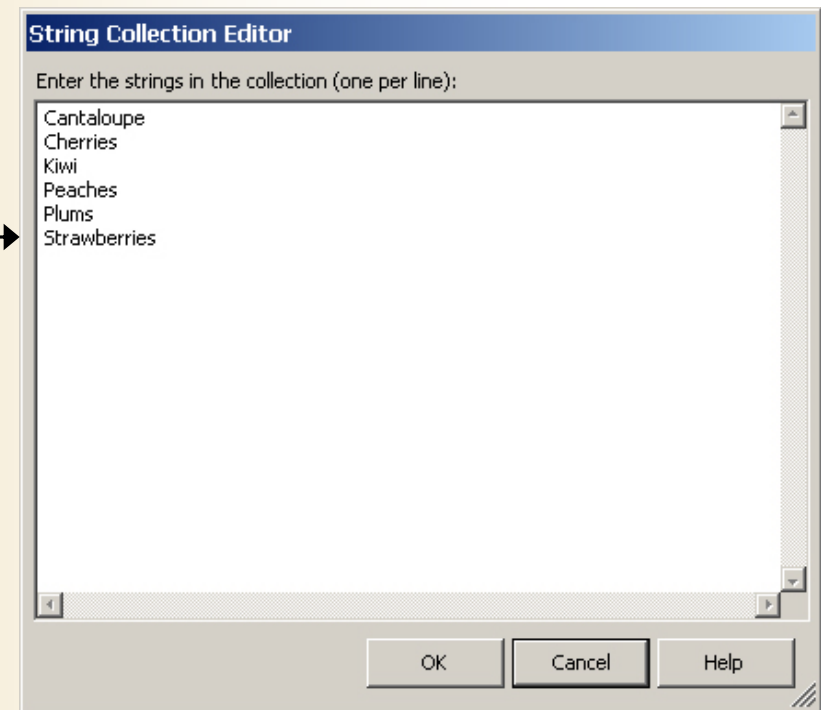


- Nella finestra delle proprietà in progettazione
 - Proprietà Items
 - Fare click sui puntini per aprire l'editor della raccolta di stringhe
 - Terminare la digitazione di ogni singolo elemento con l'INVIO
- In esecuzione
 - `Items.Add(elemento)`
 - `Items.Insert(posizione, elemento)`

Riempire un elenco in progettazione



Click sul
pulsante
con i
puntini per
aprire





Metodo Items.Add

- Usato per aggiungere nuovi elementi in coda all'elenco in esecuzione
- Forma generale `Object.Items.Add(ItemValue)`
- Esempi

```
lstNames.Items.Add("Brandon")  
lstDeptNums.Items.Add(100)  
cboMajors.Items.Add(txtMajors.Text)
```

'Per aggiungere quanto scritto nel combo
`cboGrade.Items.Add(cboGrade.Text)`

Metodo Items.Insert



- Usato per aggiungere nuovi elementi in una posizione specifica (indice) (i seguenti sono spostati verso il basso)
- Forma generale

```
Object.Items.Insert(IndexPosition, ItemValue)
```

- Esempi

```
lstDeptNums.Items.Insert(1, 100)  
cboGrade.Items.insert(0, "Freshman")  
cboMajors.Items.Insert(11, txtMajors.Text)
```

Metodo Items.RemoveAt



- Usato per rimuovere specifici elementi dall'elenco indicando la loro posizione
- Forma generale

```
Object.Items.RemoveAt(IndexPosition)
```

- Esempi

```
lstDeptNums.Items.RemoveAt(1)
```

' Per rimuovere l'elemento selezionato

```
cboGrade.Items.RemoveAt(cboGrade.SelectedIndex)
```




Metodo Items.Remove

- Usato per rimuovere il primo elemento della lista con il contenuto specificato
- Forma generale

```
Object.Items.Remove(TextString)
```

- Esempi

```
lstDeptNums.Items.Remove("USC")
```

' Per rimuovere l'elemento selezionato

```
cboGrade.Items.Remove(cboGrade.Text)
```

Metodo Clear



- Svuota l'intero elenco del riquadro di riepilogo o della casella combinata, rimuovendo tutti gli elementi

- Forma generale `Object.Items.Clear()`

- Esempi
`lstDeptNums.Items.Clear()`
`cboGrade.Items.Clear()`

Eventi dei ListBox e ComboBox



- Evento TextChanged
 - Ha luogo quando l'utente scrive nel combo (ad ogni pressione di un tasto)
- Evento SelectedIndexChanged
 - Ha luogo quando cambia l'elemento selezionato
- Evento Enter (ricevimento del focus)
- Evento Leave (perdita del focus)