

C# Visuale

Creazione dinamica degli oggetti

Definizione e inizializzazione di una variabile di tipo Button:

```
Button btn;
```

```
btn = new Button();
```

La dichiarazione e l'istanza di una variabile è uguale per tutti i tipi di oggetti.

Es: Label lbl; lbl = new Label();

Sintassi generale per tutti gli oggetti

```
TipoOggetto nome_variabile;                    nome_Variabile = new  
TipoOggetto();
```

Alcune proprietà degli oggetti:

```
nome_variabile.Name = "Nome"; //Nome dell'oggetto
```

```
nome_variabile.Location = new Point(CoordinateX, CoordinateY);
```

```
//Posizione nel form dell'oggetto
```

```
nome_variabile.Text = "Testo"; //Testo dell'oggetto
```

```
nome_variabile.Font = new Font(lbl.Font, FontStyle.Bold); //Stile del font  
Grassetto
```

```
nome_variabile.Font = new Font(lbl.Font, FontStyle.Italic); //Stile del font  
Corsivo
```

```
nome_variabile.Font = new Font(lbl.Font, FontStyle.Underline); //Stile del font  
Sottolineato
```

```
nome_variabile.Font = new Font(lbl.Font.FontFamily, 18); //Grandezza del  
font
```

```
nome_variabile.ForeColor = Color.Red; //Colore del testo
```

```
nome_variabile.BackColor = Color.LightBlue; //Colore dello sfondo
```

`nome_variabile.AutoSize = true; //Imposta le dimensioni dell'oggetto automaticamente`

`nome_variabile.Size = new Size(larghezza, altezza); //Imposta la grandezza dell'oggetto`

`btn.FlatStyle = FlatStyle.Flat; //Imposta i bordi del pulsante in bidimensionale`

`btn.FlatAppearance.BorderColor = Color.Red; //Cambio il colore dei bordi del pulsante`

`this.AutoSize = true; //Imposta dimensioni automatiche al form s`

Aggiunta oggetto al form:

`this.Controls.Add(nome_variabile);`

Creazione di un evento click per un pulsante:

`btn.Click += new EventHandler(btn_Click);`

Sintassi generale per tutti gli oggetti

nome_variabile.nome_evento += EventHandler(nome metodo);

Utilizzo del sender nei metodi degli eventi di un pulsante:

`Button btn = (Button)sender;`

Sintassi generale per tutti gli oggetti:

tipoOggetto nome_variabile = (tipoOggetto)sender;

Accedere a un qualsiasi oggetto del form da qualunque parte del programma:

`this.Controls["nome_controllo"].(proprietà che si desidera modificare)`

Rimuovere degli oggetti dal form in base al tipo e al nome dell'oggetto:

```
for (int i = this.Controls.Count-1; i >= 0; i--)
{
    if (this.Controls[i] is (tipo oggetto) && this.Controls[i].Name != "Nome
oggetto")
    {
        this.Controls.RemoveAt(i);
    }
}
```

Foreach:

Sintassi

foreach (**TipoVariabile** **nomeVariabile** in **array/lista da scorrere**)

```
{
    /* Istruzioni varie*/
}
```

foreach (Control c in this.Controls)

```
{
    if (c is Button)
    {
        if (c.Name == "btnGioca")
            c.Enabled = !flag;
        else
            c.Enabled = flag;
    }
}
```

```
}
```

Spiegazione: questo foreach scorre tutti i controlli, gli oggetti, presenti nel form. All'interno verifichiamo se la variabile di tipo controllo c , in questo caso, è di tipo Button e poi svolgiamo delle istruzioni.

```
foreach (Button b in this.Controls.OfType<Button>())
```

```
{
```

```
    if (b.Name == "btnGioca")
```

```
    {
```

```
        b.Enabled = !flag;
```

```
    }
```

```
    else
```

```
    {
```

```
        b.Enabled = flag;
```

```
        b.BackColor = Color.White;
```

```
    }
```

```
}
```

Spiegazione: questo foreach scorre solo i controlli di tipo Button presenti nel form. `this.Controls.OfType<Button>()` questo serve per far sì che durante l'esecuzione vengano presi in considerazione soltanto gli oggetti di tipo Button e che gli altri tipi vengano ignorati dal foreach.