# PRIMO APPROCCIO A VISUAL STUDIO - EVENTI E CODICE

## *Eventi e Codice*

Sono chiamati **Eventi** tutte le *Azioni che l'Utente può effettuare sui Controlli* durante la *Fase di Esecuzione*.

- Durante la *Fase di Esecuzione* (ossia ***mentre il programma è in funzione***), l'utente può operare e interagire in tanti modi con i Controlli presenti sulla Form. Esempi di tali "azioni", chiamate tecnicamente *Eventi*, sono: "fare **click** su un pulsante", "fare **doppio-click** su una immagine", "**premere un tasto della tastiera** per scrivere in una casella di testo", "**trascinare** un oggetto su un altro", "fare **click-destro** su un oggetto", "**selezionare** una voce da un elenco", ecc.

Con il termine **Codice**, indicheremo una **Sequenza di Istruzioni** (scritte in linguaggio C#) che il computer può eseguire.

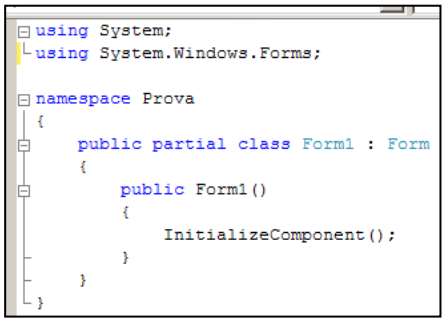
In *Fase di Progettazione*, con C#, è possibile **Associare, ad un Evento, una specifico Blocco di Codice**.

- Per il nostro esempio, lascia sulla Form un unico pulsante e chiamalo *plsSaluto*. In C# puoi associare all'**Evento** "*click sul pulsante plsSaluto*", un **Blocco di Codice** con delle istruzioni che ordinano a C# di visualizzare, in una finestrina, il messaggio "SALUTO TUTTI I MIEI AMICI!". Vediamo come puoi realizzare questo…

In *Fase di Esecuzione*, quando **l'utente provoca un Evento**, allora **C# esegue automaticamente il Blocco di Codice** ad esso associato.

- Se il programma fosse già completo e funzionante, allora noteresti che un "*click sul pulsante plsSaluto*" (evento) provoca l'esecuzione delle istruzioni (codice) e la comparsa della finestrina con il messaggio "SALUTO TUTTI I MIEI AMICI!".

Per digitare il Codice e associarlo agli Eventi si utilizza la **Finestra Codice,** attivabile con il **comando Visualizza / Codice** (o F7). L'ambiente **C# genera Automaticamente una Parte del Codice**, parte indispensabile al funzionamento del programma**.**

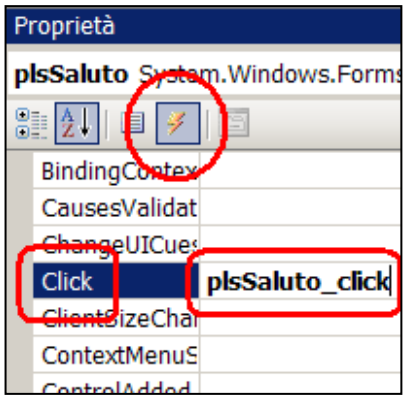


```
using System;
using System.Windows.Forms;

namespace Prova
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
    }
}
```

- Il Codice in C#, come per tutti i Linguaggi di Programmazione, và scritto secondo precise regole (*sintassi*) che approfondiremo in dettaglio più avanti. C# inserisce automaticamente alcune righe di codice: *esse sono necessarie per un corretto funzionamento il programma e non vanno modificate o cancellate*. Approfondiremo questo aspetto in un secondo momento.

Prima di poter scrivere il Codice da associare ad un Evento, è anzitutto necessario **Specificare quale Controllo e quale Evento** si desidera "programmare" e **generare il Blocco di Codice** da associare all'evento.



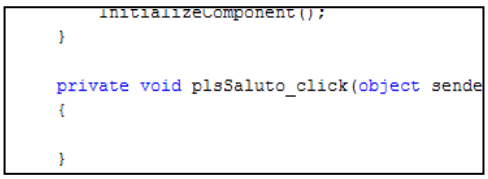

Per far questo, si opera sulla Form: anzitutto si *Seleziona il Controllo* da programmare e, dalla Finestra Proprietà, si accede alla **Lista degli Eventi** di quel controllo, tramite **l'icona Eventi** ⚡.

Nella Lista degli Eventi *si individua l'Evento* desiderato e, nella casella adiacente, si **effettua un doppio-click:** questo provoca la **generazione automatica del Blocco di Codice** da associare all'evento**.**

Al blocco **viene attribuito automaticamente un Nome**, nella forma standardizzata "***nome-controllo_nome-evento***".

- Nel nostro esempio, il *nome-controllo* è "plsSaluto" e il *nome-evento* è "click" per cui il Nome generato automaticamente sarà "**plsSaluto_click**"

Al termine dell'operazione, C# apre automaticamente la *Finestra del Codice* dove possiamo notare come **abbia generato automaticamente il Blocco di Codice**, anche se ancora "vuoto", ossia privo di istruzioni al suo interno.



```
        InitializeComponent();
    }

    private void plsSaluto_click(object sende
    {

    }
```

- Tutte le *Istruzioni* del Blocco di Codice devono essere scritte fra le due **Parentesi Graffe { … }** del blocco stesso.

☞ A questo punto non resta che scrivere le Istruzioni del Codice fra le due *Parentesi Graffe*. Nel nostro esempio usiamo l'Istruzione **MessageBox.Show** che, quando eseguita, provoca la **visualizzazione di un messaggio in una finestrina**:

```
private void plsSaluto_click ( .... )
{
        MessageBox.Show ( "SALUTO TUTTI I MIEI AMICI" );
}
```

Il Codice inserito fra le Parentesi Graffe **verrà eseguito** da C# solo quando, in Fase di Esecuzione, **l'utente provocherà l'Evento indicato, sul Controllo indicato**.


☞ Se Esegui il programma (con F5), apparirà la Form con il pulsante *plsSaluto*: se fai *click* sul pulsante, allora *provochi l'evento click* sul *controllo plsSaluto* e C# esegue il Codice (ossia l'istruzione MessageBox.Show) visualizzando "SALUTO TUTTI I MIEI AMICI!"


Nel linguaggio C#, **Tutte le Istruzioni devono essere Terminate da un Punto e Virgola** ( **";"** ). Inoltre, C# è un linguaggio **case-sensitive** (sensibile alle maiuscole) ossia *non è la stessa cosa scrivere in Maiuscolo o in Minuscolo*.

☞ Ad esempio, se scrivessimo "messageBOX.show" anziché "MessageBox.Show", l'ambiente C# lo segnalerebbe come errore e il programma non potrebbe essere eseguito.

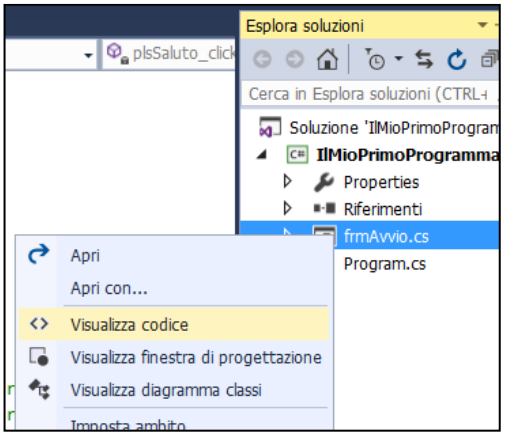
Il *Blocco di Codice* associato ad un Evento prende il nome di **Sottoprogramma-Evento.**

### *Salvataggio e Apertura del Progetto*

Per **Salvare un Progetto C#** si utilizza il pulsante  o il **menù File / comando Salva Tutto** (*CTRL-MAIUS-S*).

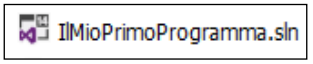
Il Progetto C# è composto da *più Files* e *più Cartelle interne*, tutti memorizzati nella Cartella in cui il progetto è salvato.

☞ Se vuoi spostare o copiare il progetto, devi agire sempre *sull'intera cartella del Progetto* e non sui singoli files. Le **cartelle BIN e OBJ** vengono comunque *rigenerate* alla riapertura del progetto. Per **inviare un esercizio** in C# ti conviene, quindi, rimuovere le cartelle BIN e OBJ e comprimere l'intera cartella del progetto in unico file zippato.




Per **Aprire un Progetto C#** si utilizza il il pulsante  o il **menù File / comando Apri Progetto/Soluzione** (*CTRL-MAIUS-O*).

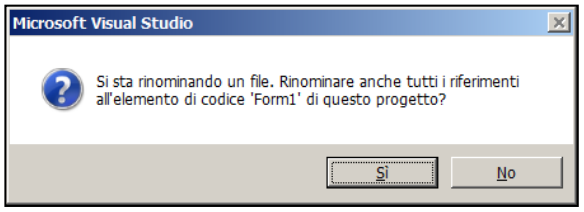
In alternativa, è possibile lanciare il **File di Soluzione (".sln")** presente nella cartella del progetto stesso.




Una volta aperto il progetto, è utile visualizzare la **Finestra Esplora Soluzioni**, che *elenca tutti i "componenti" del progetto*, inclusa la Form.

Un click-destro sulla Form, visualizza un menù pop-up dal quale è possibile accedere rapidamente alla **Finestra di Progettazione della Form** o alla **Finestra del Codice della Form**.

Dallo stesso menù, è possibile **Rinominare la Form** (ossia *Modificare il Nome della Form*). Nell'attribuire il nuovo nome è necessario **lasciare invariata l'estensione** (ossia la parte "**.cs**" del nome stesso) e **rispondere SI alla successiva finestra** che chiede se modificare anche *tutti i riferimenti alla form stessa*.




Come per gli altri controlli, anche per la Form è utile che il nome sia *preceduto da un prefisso*, il prefisso **frm**.

☞ Negli esempi e gli esercizi più semplici, si utilizza solo una form. Più avanti, però, realizzerai *progetti con più form*: abituati fin da ora a denominare correttamente la form iniziale, chiamandola, per esempio, **frmAvvio**.