

Contrôles

Deux façons pour ajouter un contrôle :

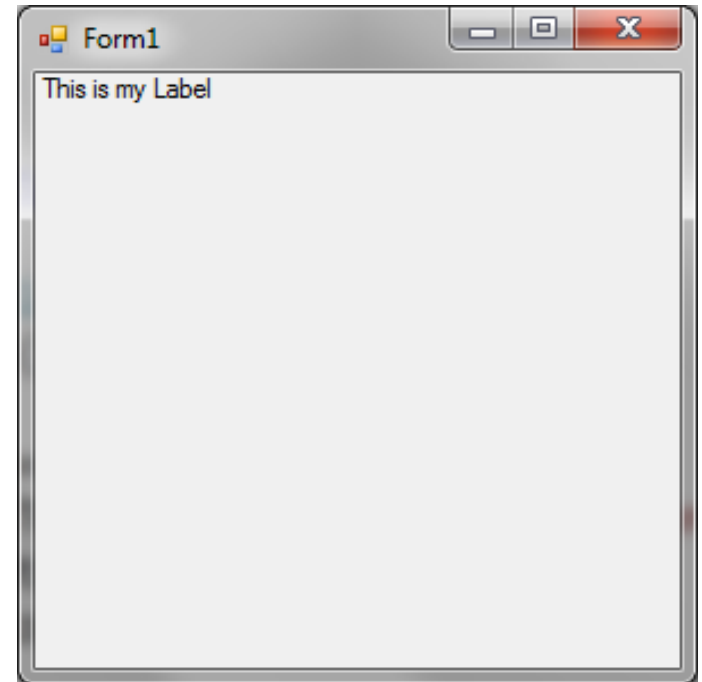
1. Ajouter le contrôle via le code source
 - Créer un objet du type du contrôle souvent privé:
 - Instancier l'objet déclaré:
 - Personnaliser le contrôle en modifiant ses propriétés:
 - Ajouter le contrôle à la collection des contrôles du formulaire:
2. Faire un Drag & Drop du contrôle à partir de la Toolbox (boîte à outils)

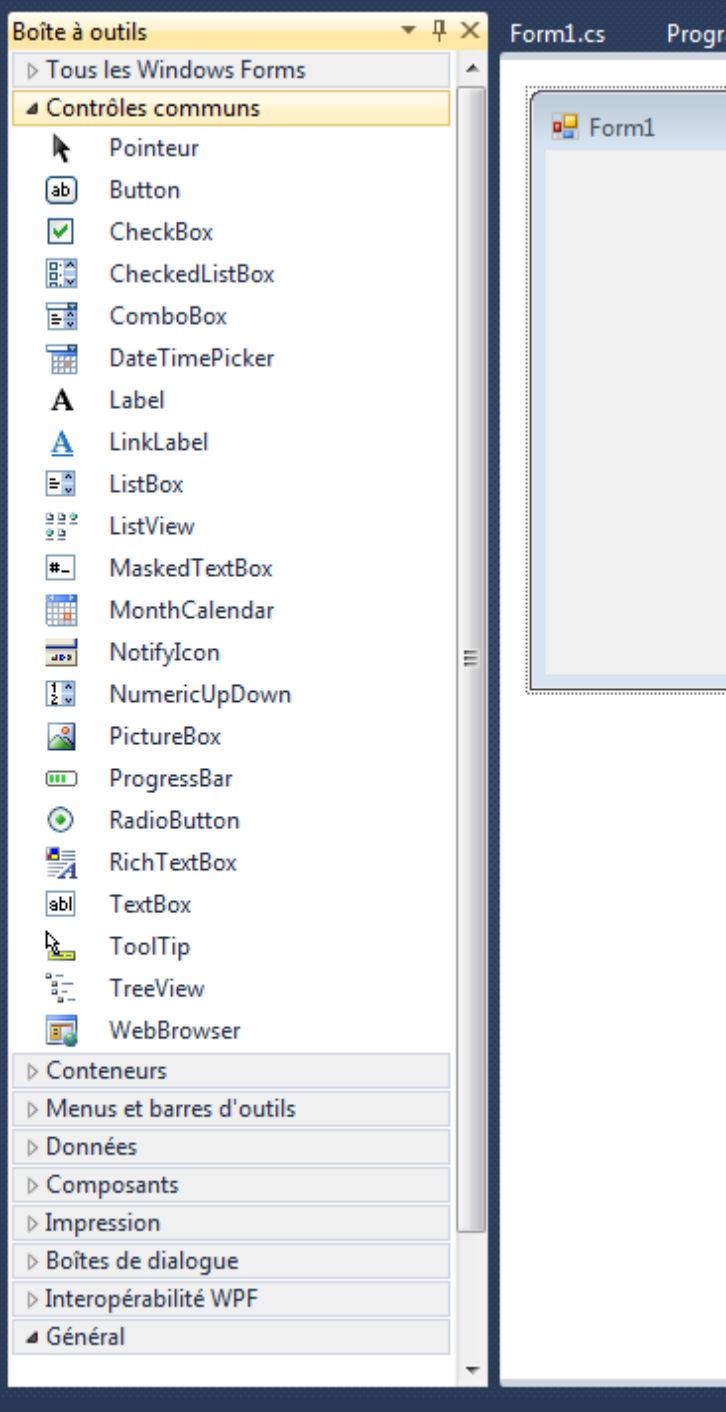
Contrôles via code source

```
public partial class Form1 : Form
{
    private Label myLabel;
    public Form1()
    {
        myLabel = new Label();
        myLabel.Text = "This is my Label";
        this.Controls.Add(myLabel);
        InitializeComponent();
    }
}
```

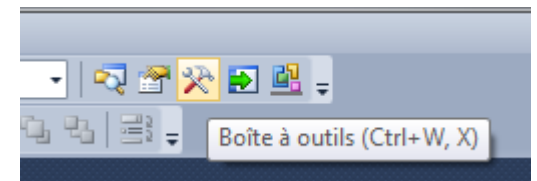
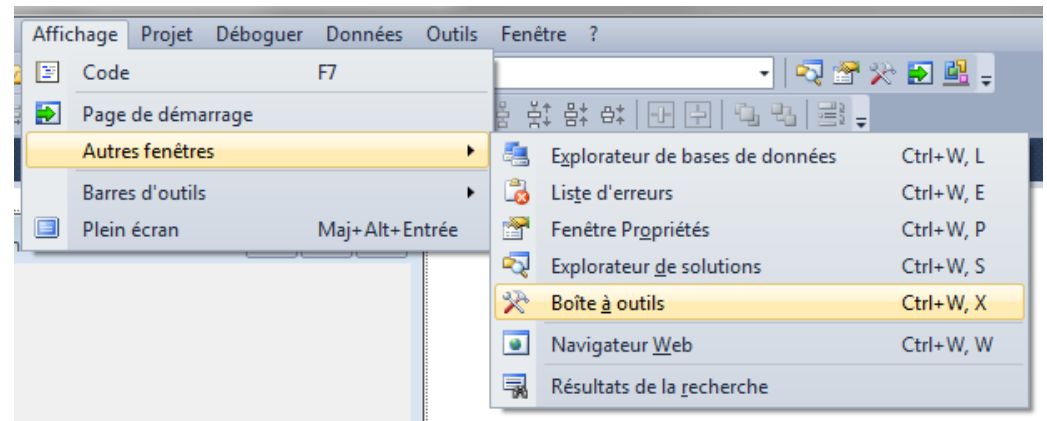
Ou

```
private System.Windows.Forms.Label myLabel;
private void InitializeComponent()
{
    myLabel = new System.Windows.Forms.Label();
    myLabel.Text = "This is my Label";
    this.Controls.Add(myLabel);
}
```



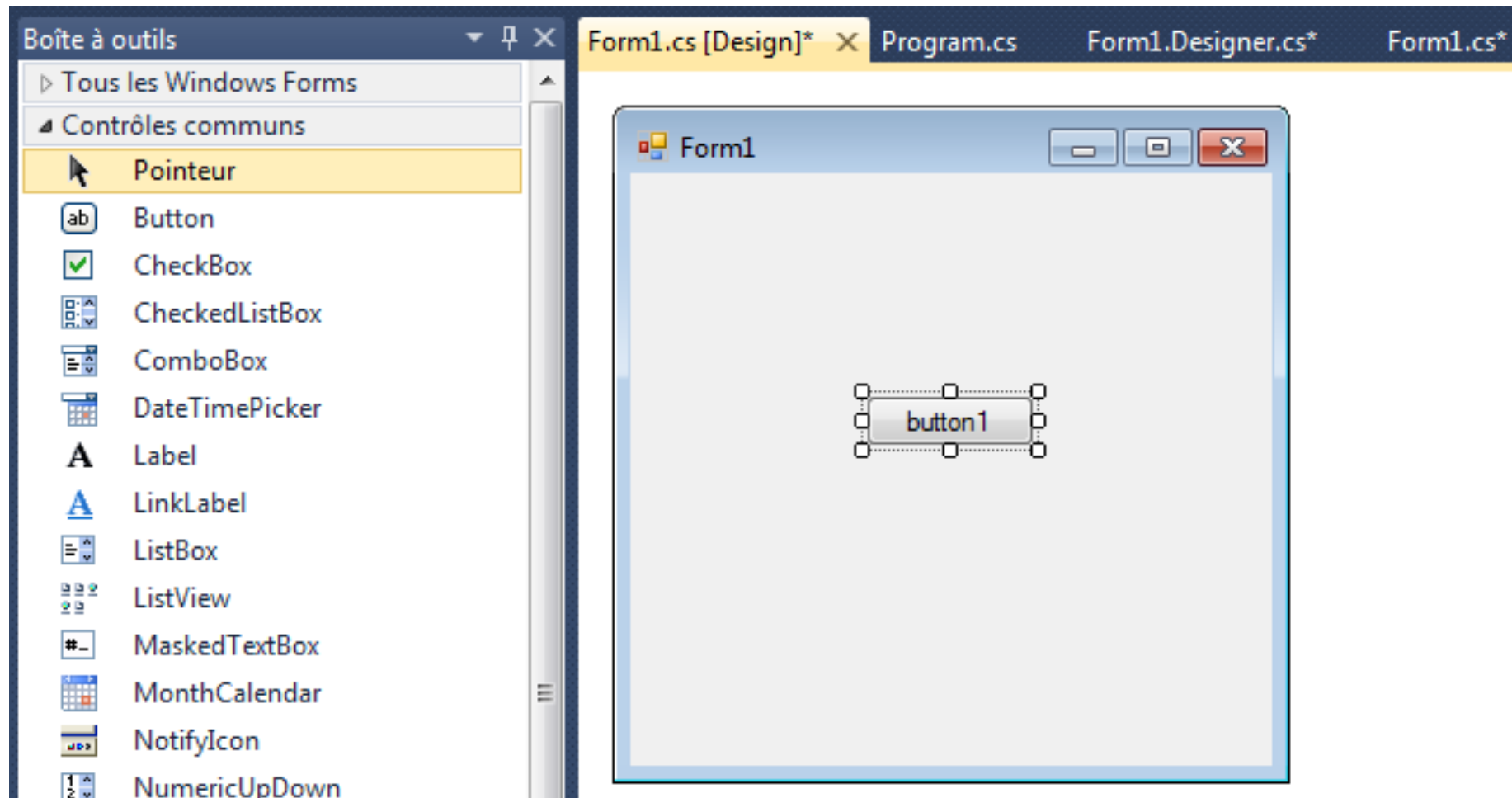


Contrôles : ToolBox

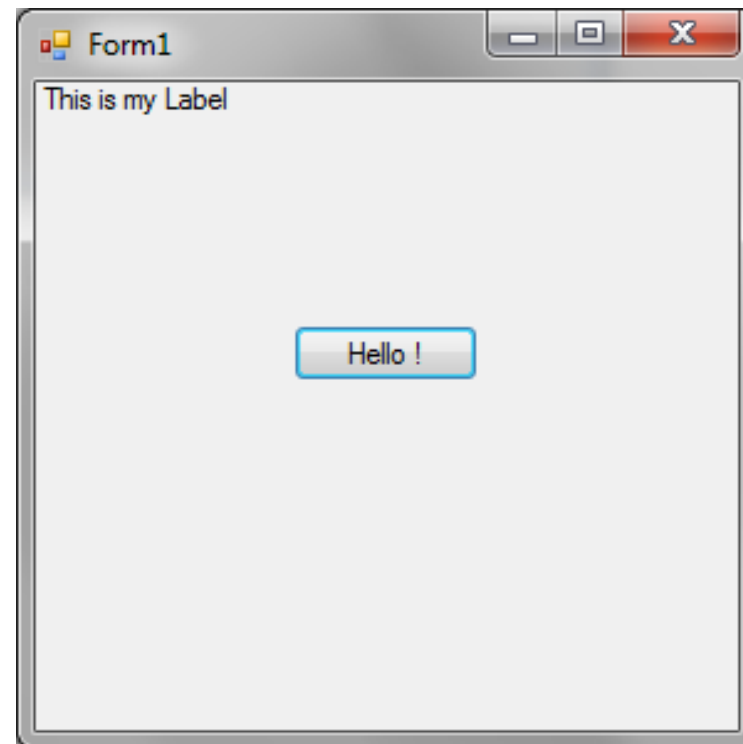
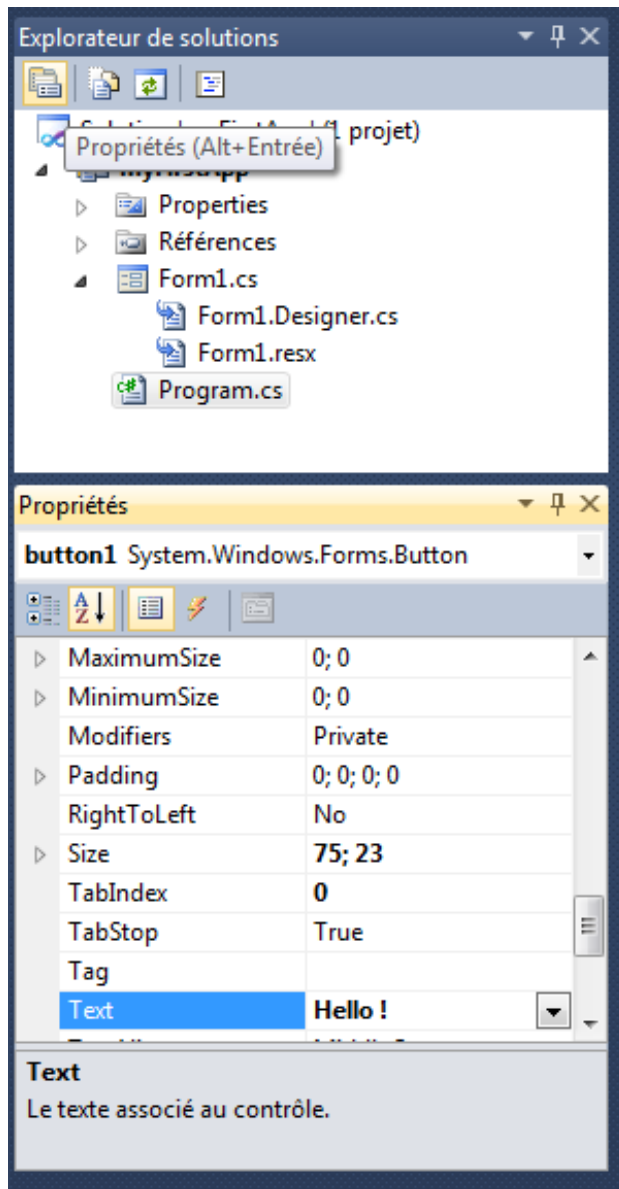


- Glisser le composant désiré
- personnaliser

Contrôles : ToolBox



Contrôles : ToolBox



Contrôles : ToolBox

- Code généré :

```
private void InitializeComponent()
{
    this.button1 = new System.Windows.Forms.Button();
    this.SuspendLayout();
    //
    // button1
    //
    this.button1.Location = new System.Drawing.Point(104, 98);
    this.button1.Name = "button1";
    this.button1.Size = new System.Drawing.Size(75, 23);
    this.button1.TabIndex = 0;
    this.button1.Text = "Hello !";
    this.button1.UseVisualStyleBackColor = true;
    //
    // Form1
    //
    this.AutoScaleMode = new System.Drawing.SizeF(6F, 13F);
    this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
    this.ClientSize = new System.Drawing.Size(284, 262);
    this.Controls.Add(this.button1);
    this.Name = "Form1";
    this.Text = "Form1";
    this.ResumeLayout(false);
}

private System.Windows.Forms.Button button1;
```

Types de contrôles

- **Command** : Button, LinkLabel, NotifyIcon, Toolbar
- **Menu** : MainMenu, ContextMenu
- **Selection** :CheckedListBox, ComboBox, DomainUpDown, ListBox, ListView, NumericUpDown, TreeView
- **Text** : TextBox, RichTextBox, Label, LinkLabel, StatusBar
- **Graphics** : PictureBox, ImageList
- **Value** : CheckBox, CheckedListBox, RadioButton, Trackbar
- **Date** : DateTimePicker, MonthCalendar
- **Common Dialog** : ColorDialog, FontDialog, OpenFileDialog, PrintDialog, PrintPreviewDialog, SaveFileDialog
- **Grouping** : Panel, GroupBox, TabControl

Contrôles: Propriétés

Propriété	Description
Anchor	détermine comment un contrôle est redimensionné avec son parent
Dock	Obtient ou définit les bordures de contrôle qui sont ancrées à son contrôle parent
BackColor	Couleur de fond
Background Image	Image de fond
Enabled	Activé ou grisé (true ou false)
Font	Propriétés du texte
TabIndex	Ordre de tabulation du contrôle dans son conteneur
Text	Texte du contrôle
Visible	Visibilité du contrôle (true ou false)
Size	La hauteur et la largeur du contrôle.

MessageBox

- La classe **MessageBox** sert à afficher dans une fenêtre pop-up un message donnant des informations ou des instructions
 - peut contenir du texte, des boutons et des symboles.
- C'est une classe statique
- On appelle la méthode **Show**.
 - Elle est surchargée et possède 21 implémentations.
 - Plus on lui passe d'arguments, plus la MessageBox sera complète.

`MessageBox.Show(|`

▲ 7 sur 21 ▼ System.Windows.Forms.DialogResult MessageBox.Show(**string text**, string caption, MessageBoxButtons buttons, MessageBoxIcon icon)

Affiche un message avec le texte, la légende, les boutons et l'icône spécifiés.

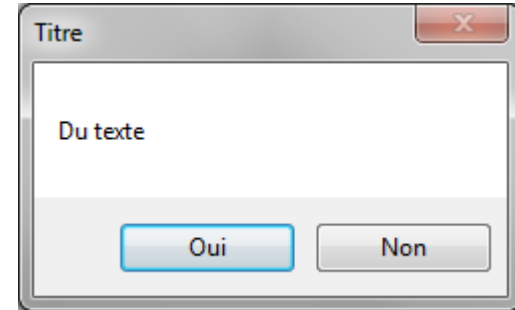
text: Texte à afficher dans le message.

MessageBox

- **Show** retourne un DialogResult qui peut valoir (suivant quel bouton est cliqué dans le pop-up) :
 - **None** : cela signifie que l'exécution de la boîte de dialogue se poursuit.
 - **OK**
 - **Cancel** (annuler)
 - **Abort** (abandonner)
 - **Retry** (réessayer)
 - **Ignore** (ignorer)
 - **Yes** (oui)
 - **No** (non)
- DialogResult est une énumération, donc pour accéder à ses valeurs, on fait par exemple DialogResult.OK.

MessageBox

```
public Form1()
{
    InitializeComponent();
    MessageBox.Show("Du texte", "Titre", MessageBoxButtons.YesNo);
}
```



```
public Form1()
{
    InitializeComponent();
    MessageBox.Show("Du texte", "Titre",
        MessageBoxButtons.RetryCancel, MessageBoxIcon.Error);
}
```

