

# Gestion d'événements

- Si on double click sur un contrôle, une liaison est créée entre le contrôle et son événement par défaut:
  - `CheckedChanged` pour une `CheckBox`
  - `TextChanged` pour un `TextBox`
  - `Click` pour un bouton
  - ...

# Gestion d'événements

- dans FormX.designer.cs, l'événement est lié à une méthode (handler) qui le traitera :


```
private void InitializeComponent()  
{  
    //...  
    this.button1.Click += new System.EventHandler(this.button1_Click);  
    //..  
}
```

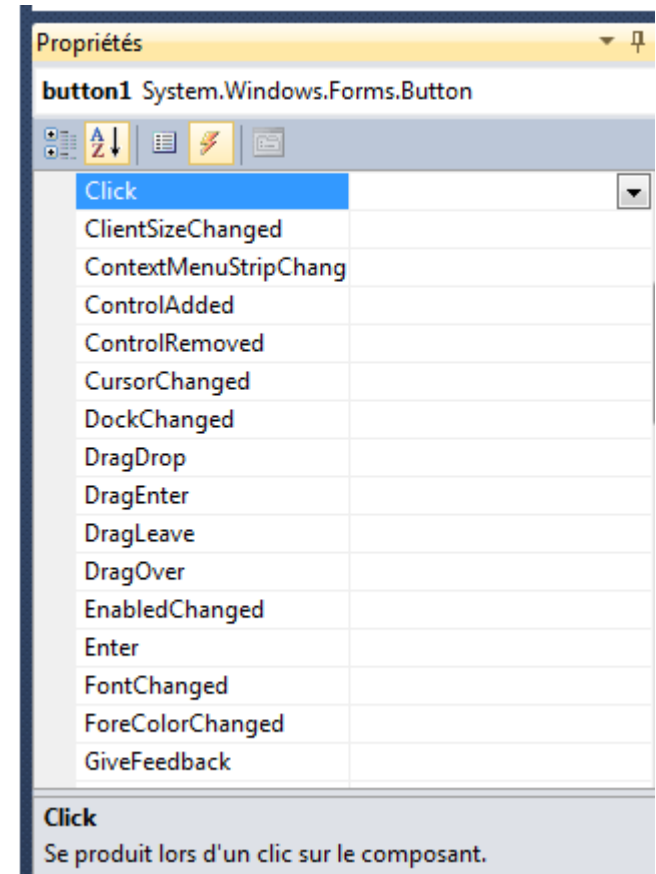
- dans FormX.cs, on implémentera cette méthode :

```
public partial class Form1 : Form  
{  
    public Form1()  
    {  
        InitializeComponent();  
    }  
    private void button1_Click(object sender, EventArgs e)  
    {  
    }  
}
```

# Gestion d'événements

En général, pour gérer un événement d'un contrôle, il faut:

- Sélectionner le contrôle dans le Designer
- Dans l'onglet **propriétés** cliquer sur l'icone: 
- Double-cliquer sur l'événement souhaité
  - Le handler est alors créé et lié
  - On est basculé où il faut implémenter le traitement



# Evénements communs

Evénement	Description
Click	Se produit suite à un clic sur le contrôle.
DoubleClick	Se produit à la suite d'un double-clic sur le contrôle.
Enter/Leave	Se produit lorsque l'utilisateur entre/quitte le contrôle
Move	Se produit lorsque le contrôle est déplacé (la propriété location est modifié)
MouseDown / MouseUp	déclenché quand le pointeur est sur le contrôle et que le bouton de la souris est enfoncé / relâché
MouseClicked / MouseDoubleClick	Se produit après un clic / double-clic de souris sur le contrôle.
...	

Rq : En utilisant le paramètre **MouseEventArgs** on peut déterminer quel bouton a été enfoncé (clic gauche ou clic droit).

# Boutons

- Classe de base : ButtonBase
- Trois dérivées :
  - Button
  - CheckBox
  - RadioButton
- Si des RadioButton sont regroupés dans un même GroupBox (container) on peut sélectionner un seul à la fois.

# Champs de texte

- Classe de base : TextBoxBase
- Trois dérivées :
  - MaskedTextBox : Utilise un masque (motif) pour distinguer entre une entrée d'utilisateur correcte et incorrecte.
  - TextBox: pour du texte non formaté
  - RichTextBox: pour du texte formaté

# Champs de texte

## TextBox: Propriétés

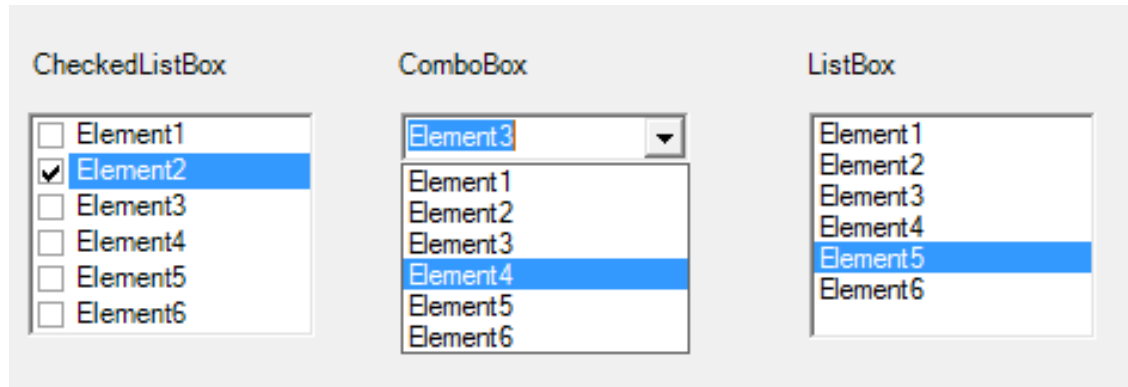
- `CharacterCasing`: Permet de forcer la casse.
- `MaxLength`: Limiter le nombre de caractères maximum.
- `Multiline`: Permet de saisir du texte multi-ligne
- `PasswordChar`: Pour des raisons de sécurité masquer les caractères saisis par le caractère spécifié.
- `ReadOnly`: Déterminer si le texte est en lecture seule.
- `WordWrap`: Pour déterminer si on passe à la ligne suivante automatiquement (pour le multi-ligne uniquement)

# Listes

- Classe de base : `ListControl`
- Deux dérivées :
  - `ListBox` : permet d'afficher une liste d'éléments dans la quelle l'utilisateur peut sélectionner en cliquant.
    - Dérivée : **`CheckedListBox`** : Affiche un objet `ListBox` dans lequel une case à cocher est affichée à gauche de chaque élément.
  - `ComboBox` : affiche un champ d'édition de texte associé à une `ListBox`, permettant à l'utilisateur de sélectionner des éléments dans la liste ou d'entrer du texte.

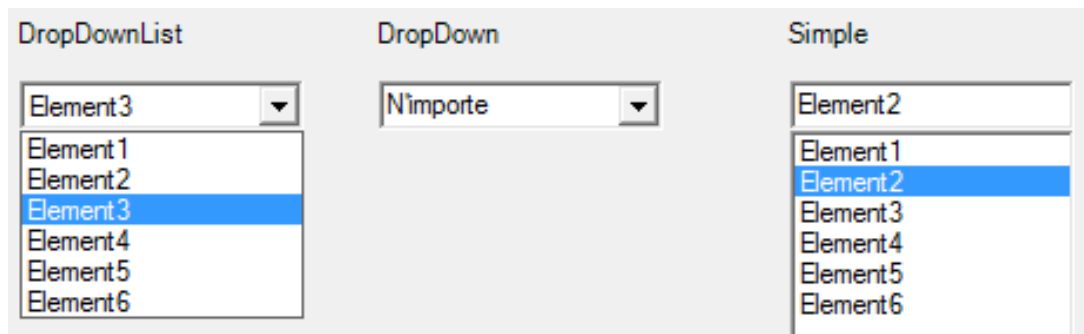


# Listes



## ComboBox: propriétés

- Sorted: les éléments sont triés automatiquement
- MaxDropDownItems: Nombre d'éléments à afficher
- DropDownStyle: Simple, DropDown, ou DropDownList



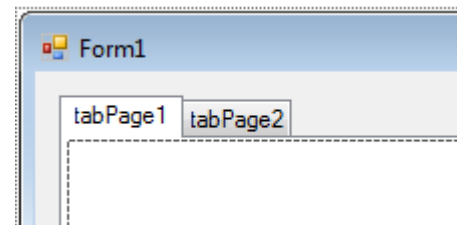
# Listes

## ListBox

- MultiColumn: Pouvoir afficher des données multi-colonnes dans la liste.
- SelectionMode: Nombre max d'éléments à sélectionner en même temps.
- SelectedIndex: Index de l'élément sélectionné, 1 par défaut.
- Items: Liste de tous les éléments de la liste (liste à manipuler pour ajouter/modifier/supprimer un élément de la liste via code source).

# TabControl

- Outil d'organisation (conteneurs) pour séparer fonctionnellement les parties d'une application
- Il se compose d'une collection de TabPage
- Pour ajouter des TabPages et pouvoir personnaliser leurs apparences et leurs comportements on utilise la propriété TabPages
- Pour indiquer qu'on veut faire un highlight sur le titre du TabPage situé sous la souris il faut mettre la propriété HotTrack à true



# Boîtes de dialogue

Boite de dialogue	utilité
MessageBox	Affiche un simple message à l'utilisateur et attend une réponse
ColorDialog	Permet de sélectionner une couleur prédéfinie ou créer une couleur personnalisée. La propriété Color retourne la couleur sélectionné par l'utilisateur
FontDialog	Permet de sélectionner un style, une taille, ...
OpenFileDialog / FolderBrowserDialog	Naviguer dans le système de fichier pour sélectionner un fichier / répertoire
PageSetupDialog PrintDialog PrintPreviewDialog	Choisir une imprimante, personnaliser l'impression, voir un aperçu puis imprimer un document.
SaveFileDialog	Naviguer dans le système de fichier pour choisir un chemin de sauvegarde

# Menus

- Pour créer un menu principal on ajoute une instance de **MenuStrip** au formulaire
- On dispose alors d'un assistant pour créer rapidement un menu arborescent
  - Un menu se compose d'une arborescence d'objets **ToolStripMenuItem**
- Les événements sur les éléments du menu sont gérés d'une manière similaire à ceux d'un contrôle classique

# Menus

- Propriétés fréquemment utilisé pour personnaliser un élément du menu

Propriété	Description
Checked / CheckOnClick	Spécifier respectivement si on veut afficher une case à cocher à coté de l'élément du menu et changer son état automatiquement après chaque clique cet élément
ToolTipText	Pour afficher une description de l'élément du menu si on passe la souris dessus
ShortcutKeys / ShowShortcutKeys	Spécifier le raccourci clavier pour l'élément du menu / indiquer ou pas ce raccourci à l'utilisateur
Text	Texte affiché pour représenter l'élément du menu

# Menus Contextuels

- Définition: Menu affiché quand l'utilisateur fait un clique droit sur un contrôle
- Etapes pour créer un menu contextuel:
  - Créer une instance de type **ContextMenuStrip**
  - Créer l'arborescence souhaité (objets de type **ToolStripMenuItem**)
  - Associer le menu créé à la propriété **ContextMenuStrip** du contrôle désiré