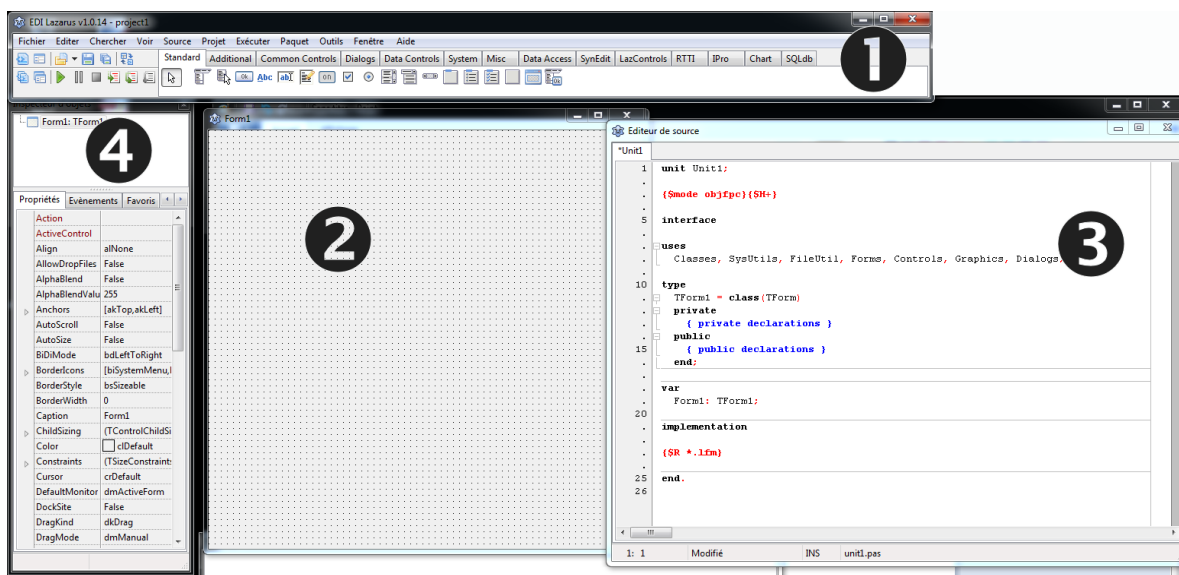


## Présentation de l'environnement Lazarus :

Pour lancer l'environnement **Lazarus**, lance sur le disque Commun\\Scribe, le fichier **T:\PC\Lazarus\Lazarus.exe**  
 Tu obtiendras cet environnement composé de quatre parties, en déplaçant et en redimensionnant les fenêtres, donne-lui cet aspect :



- ❶ **La barre de menus et d'outils** : sert à positionner des éléments sur une fiche, à sauvegarder les fichiers et à générer l'application : on dit « Compiler »
- ❷ **La fiche** : Représente visuellement ce que sera l'application
- ❸ **L'éditeur de source** : permet d'écrire le programme, le code de l'application
- ❹ **L'inspecteur d'objets** : Permet de définir les propriétés d'un objet : une fiche, un bouton, une image sont des objets

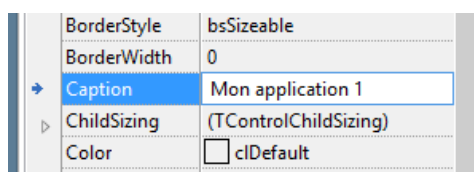
## Avant de commencer, donner un nom à son application, à sa fiche et les sauvegarder :

▪ **Clique sur la fiche (❷)**, elle s'appelle pour l'instant « Form1 », nous allons la renommer « MainF » en utilisant l'inspecteur d'objet (❹)

Cherche La propriété « Name » dans l'inspecteur d'objet (❹), elle porte la valeur «Form1 ». Remplace « Form1 » par « MainF », ce sera le nom de la fiche de ton application, valide par la touche Entrée du clavier.

A cette fiche MainF, correspondra un fichier de code « **MainU.pas** ».

Avant		Après	
KeyPreview	False	KeyPreview	False
Left	1532	Left	1532
Menu		Menu	
➔ Name	Form1	➔ Name	MainF
ParentBiDiMode	True	ParentBiDiMode	True
ParentFont	False	ParentFont	False
PixelsPerInch	96	PixelsPerInch	96



De la même manière, donne l'inspecteur d'objet (❹), la valeur « Mon Application1 » à la propriété « Caption » de la fiche. Valide par entrée.

Crée un dossier **Lazarus** dans ton dossier **Perso** sur le réseau, dans ce dossier **Lazarus** crée un dossier **MonAppli1**. Tous les éléments de l'application seront sauves dans ce dossier **Perso\Lazarus\MonAppli1**

▪ Utilise la commande **Fichier | Enregistrer** sous de **Lazarus**, enregistre le projet (l'application) sous le nom **MonAppli1.lpi** dans le dossier **Perso > Lazarus > MonAppli1**

▪ Enregistre ensuite l'unité **unit1.pas** sous le nom **MainU.pas** dans le dossier **Perso > Lazarus > MonAppli1** ce fichier **MainU.pas** contiendra le code de l'application **MonAppli1.exe**

### Première compilation (génération de l'application) :

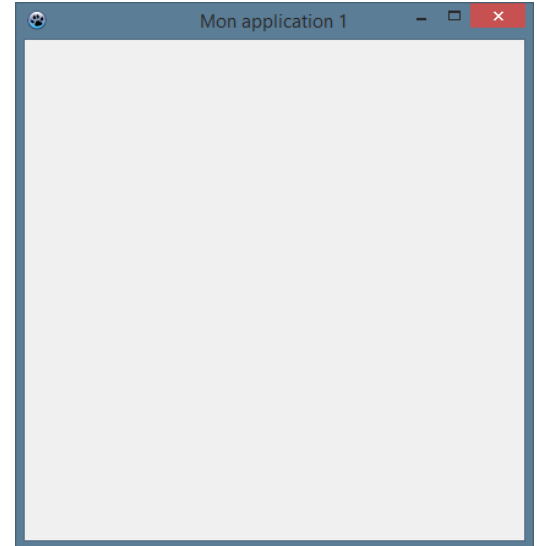
Lazarus est maintenant prêt à compiler ton application ( qui est vide, ce n'est qu'une fiche ou fenêtre sans autre fonction).



Pour compiler, utilise la commande « Triangle vert » de la barre d'outils ou la commande de menu **Exécuter | Compiler**

La compilation dure environ 20 secondes, **Lazarus** va créer une application nommée **MonAppli1.exe** puisque c'est le nom que tu lui as donné dans le dossier **Perso > Lazarus > MonAppli1**

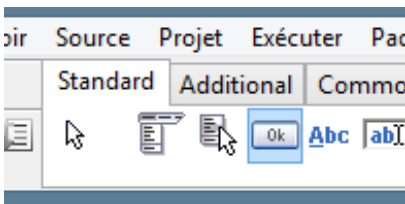
**Lazarus** lance ensuite automatiquement l'application **MonAppli1.exe** qui est représentée ci-contre à droite. C'est une simple fiche, vide, que tu peux déplacer, ouvrir, fermer, redimensionner... Il reste à lui donner des fonctions.



Tu devras systématiquement fermer cette fiche « **Mon application 1** » pour reprendre le contrôle de **Lazarus**. **Une fois qu'elle a été fermée**, tu peux également relancer l'application en utilisant directement le fichier de l'application : **Perso > Lazarus > MonAppli1 > MonAppli1.exe**.

Ce fichier peut se déplacer d'un ordinateur à un autre, par le réseau, par une clef USB, par un CD-Rom, on l'appelle l'exécutable de l'application

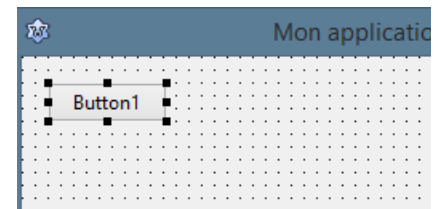
### Ajouter un bouton sur ta fiche :



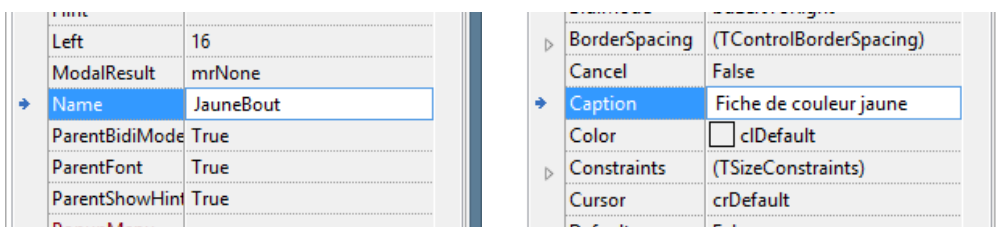
Un peu de couleur ! Nous allons ajouter un bouton dont la fonction sera de faire passer la couleur de la fiche de l'application de la couleur jaune lorsque l'on cliquera dessus.

Sélectionne d'abord l'objet « **Bouton** » dans la barre d'outils de **Lazarus**, voir ci-contre à gauche.

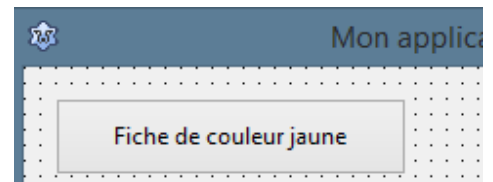
Clique ensuite sur la fiche de ton application pour y déposer un bouton (il s'appelle par défaut « Bouton1 »), déplace le bouton dans le coin supérieur gauche de la fiche, voir ci-contre à droite.



Nous allons affecter à la propriété « **Name** » de ce bouton la valeur « **JauneBout** » et à sa propriété « **Caption** » (**le texte qu'il affiche**), la propriété « Fiche de couleur jaune ». Voir ci-dessous.



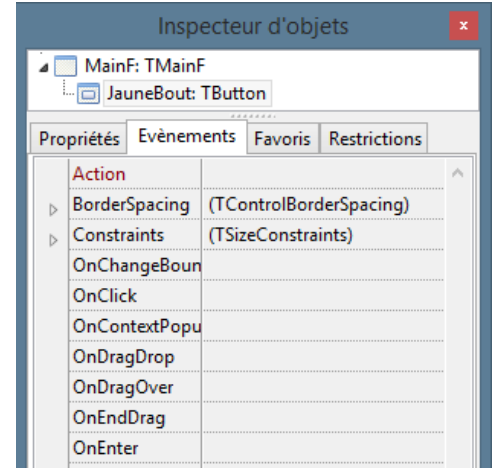
Le bouton est trop petit pour afficher tout le texte, redimensionne-le en utilisant les poignées (les carrés noirs) jusqu'à ce que « Fiche de couleur jaune » soit visible comme sur l'image ci-contre à droite.



- Compile l'application, le bouton est fonctionnel, tu peux cliquer dessus, il change de couleur mais n'aboutit à rien : il faut affecter à ce bouton un événement qui va changer la couleur de la fiche en jaune.

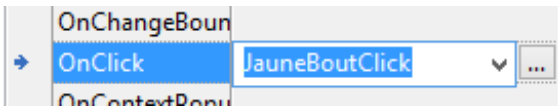
### Affecter un événement au bouton :

Reviens à **Lazarus**, sur la fiche de ton application, sélectionne le bouton « **JauneBout** » en le cliquant, ses propriétés sont alors affichées dans l'inspecteur d'objet (④).



Clique sur l'onglet « **Evènements** » pour voir tous les événements qui peuvent être associés au bouton « **JauneBout** ».

L'événement qui nous intéresse ici est l'événement « **OnClick** » qui se produit quand on clique sur le bouton « **JauneBout** ».



Double-clique sur le rectangle vide à droite de « **OnClick** », Lazarus crée un nouvel événement qu'il appelle « **JauneBoutClick** », il reste à

associer du code à cet événement, les choses sérieuses commencent !

### Associer du code à un événement dans l'éditeur de source :

Quand tu as double-cliqué sur **OnClick**, Lazarus a généré automatiquement du code dans l'éditeur de source (⑤).

Voici ce code :

```
.
.
.
.
34 |
35 | end;
```

Il reste à indiquer entre les instructions **begin** et **end**, que la couleur de la fiche doit devenir jaune.

La fiche s'appelle « **MainF** » sa propriété qui commande sa couleur est « **Color** », la couleur jaune est désignée par « **clYellow** ».

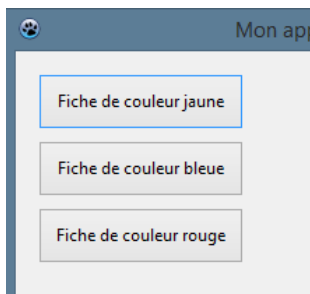
Le code qui correspond à affecter à la propriété **Color** de **MainF**, la valeur **clYellow** est

**MainF.Color:=clYellow ;** ( saisis cette ligne de code dans l'éditeur de source (⑥) entre **begin** et **end**)

La ligne de code se termine par un point-virgule, comme ceci :

```
.
.
.
.
34 | MainF.Color:=clYellow ;
35 | end;
```

- Compile l'application et vérifie le fonctionnement du bouton qui doit changer la couleur de la fiche en jaune.



### Créer deux autres nouveaux boutons :

Crée deux autres boutons, similaires au premier et qui fonctionnent de la même manière dont les caractéristiques sont les suivantes :

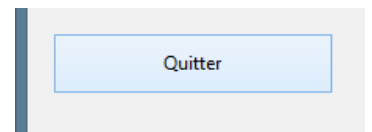
- Bouton « **BleuBout** » qui fait passer la couleur de la fiche à bleue quand on le clique ( la couleur bleue est désignée par `clBlue` ). Il affiche le texte « Fiche de couleur bleue ».
- Bouton « **RougeBout** » qui fait passer la couleur de la fiche à rouge quand on le clique ( la couleur rouge est désignée par `clRed` ). Il affiche le texte « Fiche de couleur rouge ».

- Compile l'application et vérifie le fonctionnement des trois boutons

### Ajouter un bouton « Quitter » :

La commande pour fermer l'application est **Close**; ( Close signifiant fermer ).

Crée un bouton appelé « **CloseBout** » qui va fermer **MonAppli1.exe** et qui affichera le texte « **Quitter** ».



- Compile l'application et vérifie le fonctionnement du bouton « **Quitter** »

### Ajouter des commentaires au code :

Quand on écrit du code, il est nécessaire de le commenter pour se souvenir ensuite précisément de la tâche que l'on affectée à ce code.

Pour commenter du code, il suffit de placer deux slashes // après le code et de saisir le texte de ton choix.

Exemple ci-contre à droite.

```

35 |
    | procedure TMainF.JauneBoutClick(Sender: TObject);
    | begin
38 |     MainF.Color:=clYellow; // La fiche devient jaune
    | end;
40 |

```

- Commente les événements « **OnClick** » des trois boutons : « **JauneBout** », « **RougeBout** » et « **BleuBout** ».

### Changer les dimensions de la fenêtre :

La propriété qui commande la largeur de la fiche en pixels (1 pixel=1 point sur l'écran) est « **Width** », « largeur en anglais »

La propriété qui commande la hauteur de la fiche en pixels est « **Height** », « hauteur en anglais »

La commande suivante réglera donc la largeur de la fiche à 530 pixels :

```

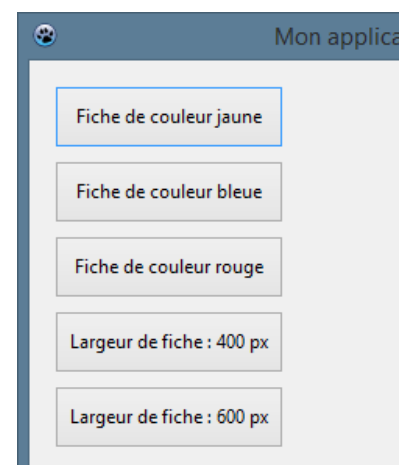
    |
    | MainF.Width:=530;
108 |

```

### Ajout de deux boutons pour modifier la largeur de la fenêtre :

Crée un bouton appelé « **W400Bout** » qui va affecter la valeur 400 pixels à la largeur de la fenêtre. Il affichera le texte « Largeur de fiche : 400 px »

Crée un bouton appelé « **W600Bout** » qui va affecter la valeur 600 pixels à la largeur de la fenêtre. Il affichera le texte « Largeur de fiche : 600 px »



## Agrandir ou réduire la fenêtre à volonté :

Crée un bouton appelé « **WMoinsBout** » qui va diminuer la largeur de la fenêtre de 10 pixels à chaque clic. Il affichera le texte « Largeur Moins »

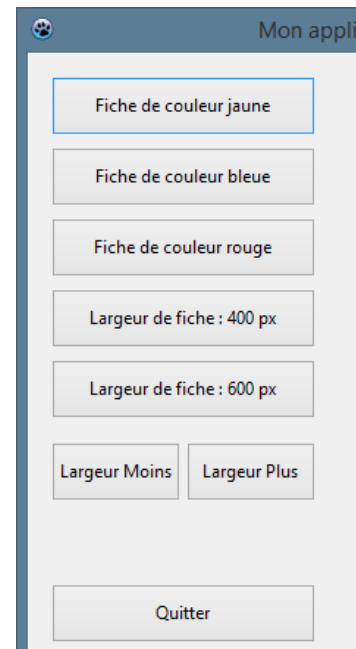
Crée un bouton appelé « **WPlusBout** » qui va augmenter la largeur de la fenêtre de 10 pixels à chaque clic. Il affichera le texte « Largeur Plus »

- Compile l'application et vérifie le fonctionnement des deux boutons

## Commente le code :

Tu as écrits plusieurs lignes de code, il est temps de le segmenter avec des commentaires. Utilise ce modèle, le copier/coller est le meilleur ami du développeur :

```
.
.
. ( TMainF )
.
. // *****
45 //
. // Cette partie du code gère la couleur de la fiche
. //
. // *****
.
50 procedure TMainF.JauneBoutClick(Sender: TObject);
. begin
.   MainF.Color:=clYellow; // La fiche devient jaune
. end;
. et
.
65 // *****
68 // Cette partie du code gère la largeur de la fiche
. //
70 // *****
.
. procedure TMainF.W400BoutClick(Sender: TObject);
. begin
```



## Impose des conditions au code :

Pour la première fois, nous allons imposer une condition à l'exécution du code.

Une condition prend la forme suivante :

**if** {condition} **then** {si la condition est vraie alors exécute le code}.

Exemples :

1/ la fiche ne deviendra rouge que si elle était bleue :

```
.
. 94 |
. 95 | if MainF.Color=clBlue then MainF.Color:=clRed;
.
.
```

2/ La largeur de la fiche est multipliée par deux si sa valeur est plus grande que 500 ( « div 2 » signifie divisé par deux) :

```
.
. if MainF.Width>500 then MainF.Width:=MainF.Width div 2;
.
.
```

En utilisant **if** ..... **then** ..... , modifie le code de l'événement **OnClick** de « **WMoinsBout** » de façon à ce que la largeur de la fenêtre ne soit jamais inférieure à 200 pixels. Ci-dessous, le code original sans **if** ..... **then** .....

```
.
. procedure TMainF.WMoinsBoutClick(Sender: TObject);
. begin
84 |   MainF.Width:=MainF.Width-10; // Diminue la largeur de 10 pixels
85 | end;
```

De même, modifie le code de l'événement **OnClick** de « **WPlusBout** » de façon à ce que la largeur de la fenêtre ne soit jamais supérieure à 1000 pixels.