

# Thème numéro 14

## Points essentiels traités dans le thème :

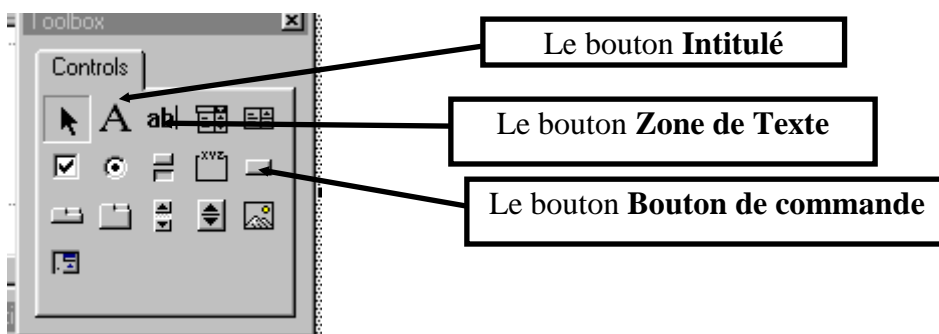
1. Création d'une boîte de dialogue
2. Ecriture de procédures de gestion d'évènements

## 14.1. Création d'une boîte de dialogue

- Lancer Excel sur un classeur vide
- Dans ce classeur, créez une boîte de dialogue, comme suit :
  - Lancez l'Editeur Visual Basic avec **Developpeur**→**Code**→**Visual Basic** (**Developer**→**Code**→**Visual Basic**)
  - Activez la sous-rubrique **Insertion**→**Boîte de dialogue** (**Insert**→**UserForm**) dans le menu contextuel du projet (obtenu par un clic droit sur l'icône du projet **Project-VBAProject**)

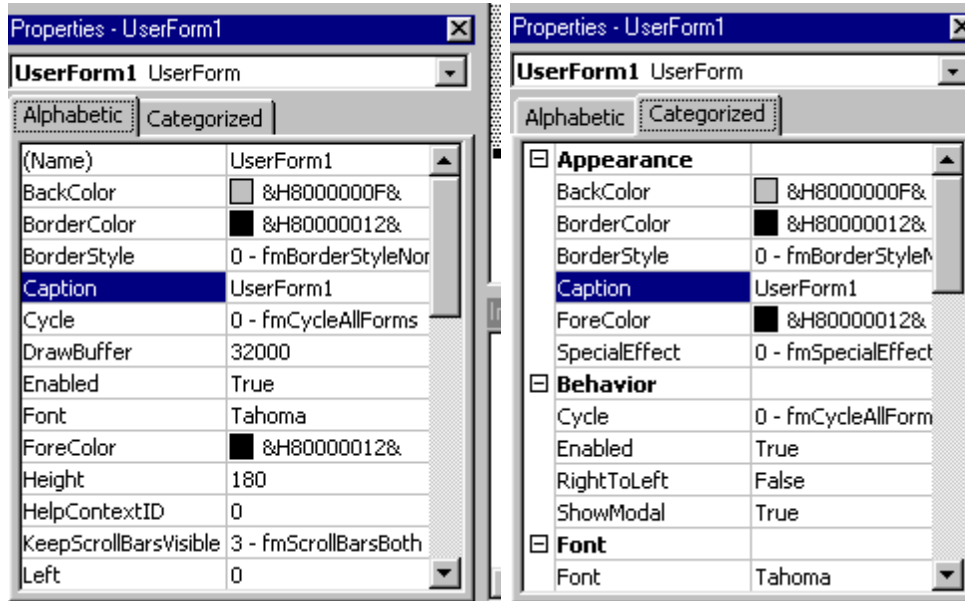
*Il apparaît une boîte de dialogue nommée **UserForm1**, couverte d'un pointillé en quadrillage. Cette boîte de dialogue est vide. Un clic en n'importe quel point de cette boîte (ou l'ouverture du menu **Affichage** → **Boîte à outils**) fait apparaître la **boîte à outils** dont les éléments qui nous intéresseront sont détaillés ci-après :*

### La boîte à outils

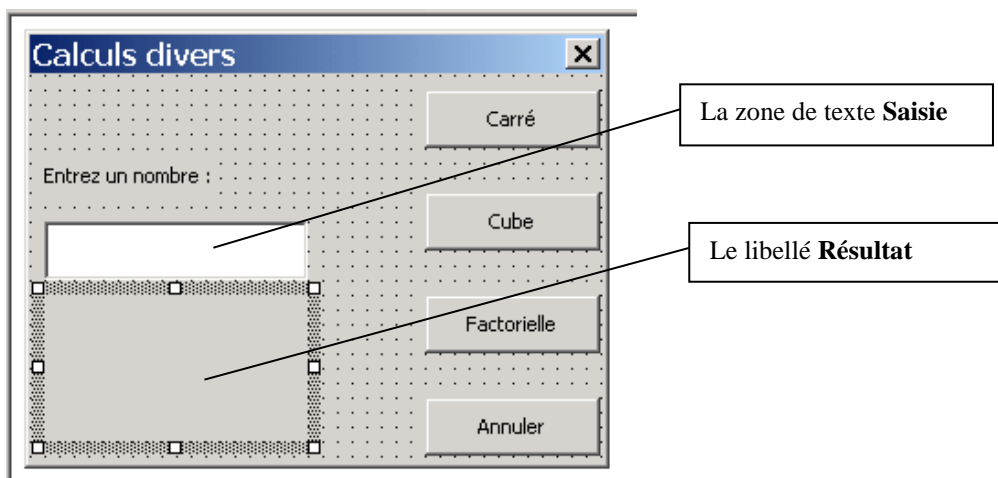


*Attention : le symbole placé à la suite du **Bouton de commande** est un **Bouton à bascule**. Ce n'est pas celui qui nous intéresse.*

Dans le volet inférieur gauche de l'Editeur apparaissent les propriétés de la boîte de dialogue. Choisissez, en cliquant dessus, l'onglet **Par catégories (Categorized)**



Voici, avant de poursuivre, l'aspect de la boîte de dialogue que nous voulons obtenir :



- Cliquez sur le mot **Titre (Caption)** pour le mettre en vidéo inverse, puis cliquez sur la même ligne, à la fin du titre actuel qui est UserForm1, et remplacez ce dernier par **Calculs divers**. Au fur et à mesure de votre frappe, vous voyez le titre se modifier dans la barre de titre de la boîte de dialogue.

- Créez un bouton, que vous placerez près du coin inférieur droit de la boîte de dialogue. Pour ce faire :
  - Première méthode :  
Faire un **glisser-déposer (drag and drop)** de l'icône du bouton de commande de la **boîte à outils**, en cliquant sur cette icône, puis en amenant le pointeur de souris à l'endroit voulu sans relâcher le bouton, enfin en relâchant le bouton de la souris lorsque vous serez satisfait de l'emplacement. Cette technique présente l'avantage de créer un bouton de taille standard.
  - Seconde méthode :
    - Cliquez sur l'icône représentant le bouton dans la **boîte à outils**. Cette icône reste enfoncée, montrant ainsi que l'outil est sélectionné
    - enfoncer le bouton gauche de la souris sur l'angle supérieur gauche du futur tracé du bouton, puis déplacer la souris en maintenant le bouton enfoncé. Un rectangle se dessine. Relâcher le bouton de la souris lorsque vous serez satisfait de la taille du bouton.

Les propriétés du bouton apparaissent immédiatement dans le volet **Propriétés (Properties)**. Renommez-le **Annuler**, en modifiant sa propriété **Nom (Name)**, et faites-lui afficher le texte **Annuler**, en modifiant sa propriété **Titre (Caption)**.

*La propriété **Name** fixe le nom sous lequel l'objet est connu de VBA. La propriété **Caption** fixe le texte affiché par l'objet.*

- Dans la catégorie **Comportement (Behavior)**, mettez à **Vrai (True)** la propriété **Annuler (Cancel)**. Ceci a pour effet de rendre la frappe de la touche **[Echap]** équivalente à un clic sur ce bouton.
- Cliquez sur le bouton **Zone de texte** de la boîte à outils. Ce bouton reste enfoncé (sa couleur est plus claire, et ses bords changent d'aspect).
- Créez une zone de texte, vers le milieu de la partie gauche de la boîte de dialogue. Donnez-lui le nom **Saisie** (propriété **Nom**).

- Créez un intitulé, au-dessus de votre zone de texte, et faites-lui afficher le texte **Entrez un nombre**, en modifiant sa propriété **Texte** ( **Caption** )
- Créez un intitulé en-dessous de la zone de texte. Vous lui donnerez le nom **Resultat** et lui ferez afficher une chaîne vide, en modifiant ses propriétés **nom** (**name**) et **texte** (**caption**).

*Attention : si vous orthographiez correctement **Résultat**, vous devrez conserver la même orthographe dans le code VBA.*

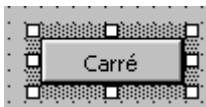
- Créez enfin trois boutons, l'un au-dessus de l'autre, au-dessus du bouton **Annuler**. Vous leur donnerez pour noms et leur ferez afficher respectivement **Carré**, **Cube** et **Factorielle**.

*Attention : Le problème de l'orthographe du nom n'a pas d'importance pour un nom de bouton. En effet, VBA utilisera le nom déclaré dans la propriété Name pour créer le nom de la procédure de gestion d'événement associée au bouton.*

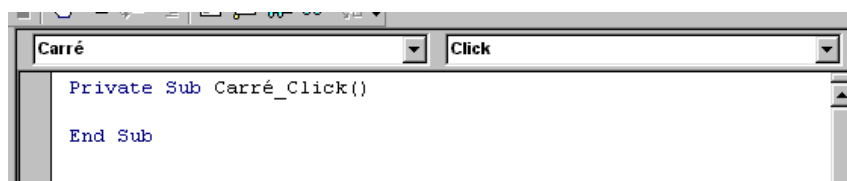
## 14.2. Création de procédures de gestion d'évènements

Nous allons désormais affecter aux trois boutons **Factorielle**, **Carré** et **Cube** des procédures qui réaliseront les calculs demandés.

- Sélectionnez le bouton **Carré**, par un seul clic. Il prend l'apparence suivante :



Un clic droit sur le bouton ouvre alors un menu contextuel. Activez la rubrique **Code** ( **View Code** ). Vous voyez apparaître à la place de la boîte de dialogue une fenêtre de saisie de code Visual Basic dans laquelle la procédure qui sera appelée par un clic sur le bouton commence à prendre forme :



En haut de cette fenêtre figurent le nom de l'objet ( **Carré** ) et le nom de l'événement associé à la procédure ( **Click** ).

Cette procédure est **d'ores et déjà affectée au bouton**

- Introduisez entre la ligne **Sub Carré\_Click()** et la ligne **End Sub** les lignes suivantes (attention, ne tentez pas d'introduire le losange, utilisé ici pour mettre en relief les lignes que vous devez introduire) :

◆ `Dim p As Integer`

◆ `Dim r As Long`

*Ces deux instructions déclarent une variables entière **p** et une variable de type entier long **r**.*

◆ `If Not IsNumeric(Saisie.Value) Then`

◆ `Resultat.Caption = "Ceci n'est pas un nombre"`

*Si la zone de modification **Saisie** contient autre chose que des chiffres et/ou un signe -, le programme modifie la propriété **caption** du libellé **Resultat**, ce qui a pour effet de faire afficher le texte **Ceci n'est pas un nombre** dans la zone réservée au libellé.*

◆ `Else`

*Ici commence le traitement d'un champ numérique*

◆ `On Error GoTo tropgrand`

*Une erreur peut désormais provenir de l'introduction d'une valeur numérique trop grande pour être transformée en un **Integer**. Dans ce cas, VBA activera le traitement d'erreur qui se trouve derrière l'étiquette **tropgrand**.*

◆ `P = CInt(Saisie.Value)`

*On range dans **p** le résultat de la transformation en **Integer** de la chaîne de caractères saisie dans la zone de texte **Saisie**.*

◆ `On Error Goto 0`

*Cette instruction ramène le système de gestion des erreurs à son état initial. En effet VBA considère 0 comme un numéro de ligne. Du fait que le numéro de ligne 0 est interdit, VBA reconnaît qu'il s'agit du cas particulier de réinitialisation du système de gestion des erreurs.*

◆ `r = CLng(p) * p`

*Le premier **p** est transformé explicitement en **Long** par **CLng**, ce qui permet à l'ensemble du calcul de s'effectuer dans le type **Long**, sans quoi il s'effectuerait dans le type **Integer** et il serait impossible de calculer le carré d'un nombre supérieur à **181**. En effet le carré d'un nombre >181 serait supérieur à **32767** qui est le plus grand **Integer** possible.*

◆ `Resultat.Caption = "Le carré de " + Str(p) + " est " + Str(r)`

*Cette instruction affiche dans le libellé **Resultat** le résultat de la transformation en chaîne de caractères des variables **p** et **r**, encadrés par des textes explicatifs*

◆ `End If`

*Cette instruction termine l'instruction **If Not IsNumeric...** commencée huit lignes plus haut.*

◆ `Exit Sub`

*Cette instruction marque la fin de la procédure, et a pour but d'éviter d'exécuter à tort la séquence de traitement d'erreur qui suit.*

◆ `tropgrand:`

*Ceci est l'étiquette à laquelle se branchera le système de gestion des erreurs si le nombre introduit dans la zone **Saisie** ne peut être convenablement transformé en un **Integer** par la fonction **Cint**.*

◆ `Resultat.Caption = "Ce nombre est trop grand"`

*Cette ligne constitue la séquence de traitement d'erreurs introduite par l'étiquette **tropgrand**. Puisque la fin de la procédure suit immédiatement, une instruction **Resume** n'est pas nécessaire.*

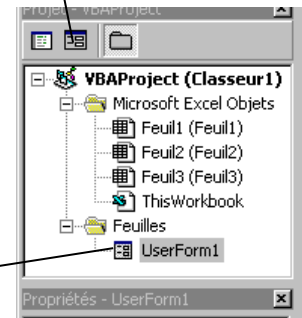
◆ `End Sub`

*Cette instruction, qui déclare la fin de la procédure, n'a pas à être introduite, l'éditeur VBA l'avait déjà introduite automatiquement.*

Le bouton **voir l'objet**

- Cliquer sur le bouton **Voir l'objet (View Object)** dans la fenêtre de l'explorateur de projet. Ceci fait réapparaître le formulaire, qui est toujours l'objet couramment sélectionné dans cet explorateur de projet.

L'objet **UserForm1** est toujours sélectionné



- Sélectionnez à présent le bouton **Cube**, puis faites sur ce bouton un clic droit et activez la rubrique **Code (View Code)** du menu contextuel qui apparaît. Vous voyez comme précédemment apparaître une fenêtre de saisie de code, qui contient l'ossature de la procédure **Cube\_Click** :

```
◆ Private Sub Cube_Click()
◆
◆ End Sub
```

Introduisez entre ces deux instructions les instructions qui figurent ci-dessous. La plupart pourront résulter d'un copier-coller à partir de la procédure **Carré\_Click**. Attention : ici l'élévation au cube d'un **Integer** pourra donner un résultat trop grand pour un **Long**, aussi il conviendra de protéger également contre les erreurs le calcul du cube. L'instruction **On Error Goto 0** devra donc être repoussée d'une ligne vers le bas. N'oubliez pas également de modifier la ligne qui calcule **r** et celle qui affiche le résultat (*le cube de...* et non plus *le carré de...*)

```
◆ Dim p as Integer
◆ Dim r As Long
◆ If Not IsNumeric(Saisie.Value) Then
◆   Resultat.Caption = "Ceci n'est pas un nombre"
◆ Else
◆   On Error Goto tropgrand
◆   p = Cint(Saisie.Value)
◆   r = CLng(p) * p * p
◆   On Error Goto 0
◆   Resultat.Caption = "le cube de " + Str(p) + " est " + Str(r)
◆ End If
◆ Exit Sub
◆
◆ tropgrand:
◆   Resultat.Caption = "Ce nombre est trop grand"
```

- Recommencer les opérations qui précèdent pour introduire le code lié au bouton Factorielle. Ce code, très semblable à celui établi lors du travail sur **le thème 11**, est donné ci-dessous :

```

◆ Dim p as Integer
◆ Dim r As Long
◆ If Not IsNumeric(Saisie.Value) Then
◆   Resultat.Caption = "Ceci n'est pas un nombre"
◆ Else
◆   On Error Goto tropgrand
◆   p = Cint(Saisie.Value)
◆   If p < 0 Then
◆     Resultat.Caption = "Pas de factorielle"
◆   Else
◆     r = 1
◆     While p > 1
◆       r = r * p
◆       p = p - 1
◆     Wend
◆     Resultat.Caption = "la factorielle de " + Saisie.Value + _
◆       " est " + Str ( r )
◆   End If
◆ End If
◆ Exit Sub
◆
◆ tropgrand:
◆   Resultat.Caption = "Ce nombre est trop grand"

```

Comme précédemment, ces instructions devront être introduites entre les instructions :

```

◆ Private Sub Factorielle_Click
◆ End Sub

```

déjà mises en place par l'éditeur VBA

- Il n'y a pas d'instruction **On Error Goto 0** car l'emplacement de cette instruction devrait coïncider avec la fin de la procédure, marquée par **Exit Sub**. Or la sortie d'une procédure réinitialise le système de gestion des erreurs de VBA. L'instruction **On Error Goto 0** serait donc inutile.



- Il faut enfin introduire le code lié au bouton **Annuler**. Ce code est extrêmement simple :

```
◆ Private Sub Annuler_Click  
◆     UserForm1.Hide  
◆ End Sub
```

- La méthode **Hide** de l'objet **UserForm1** fait disparaître cet objet de l'écran (mais il ne le détruit pas réellement, il sera toujours possible d'en relancer l'exécution).

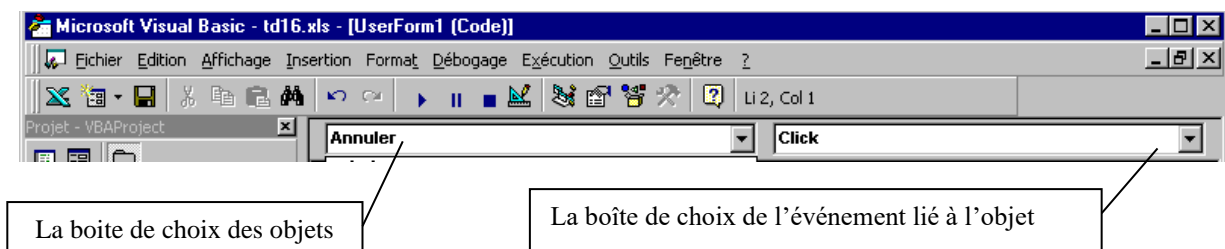
Si vous n'avez pas commis d'erreur, la boîte de dialogue que vous venez de créer est prête à réaliser les calculs demandés.

- **Sauvegardez votre classeur**

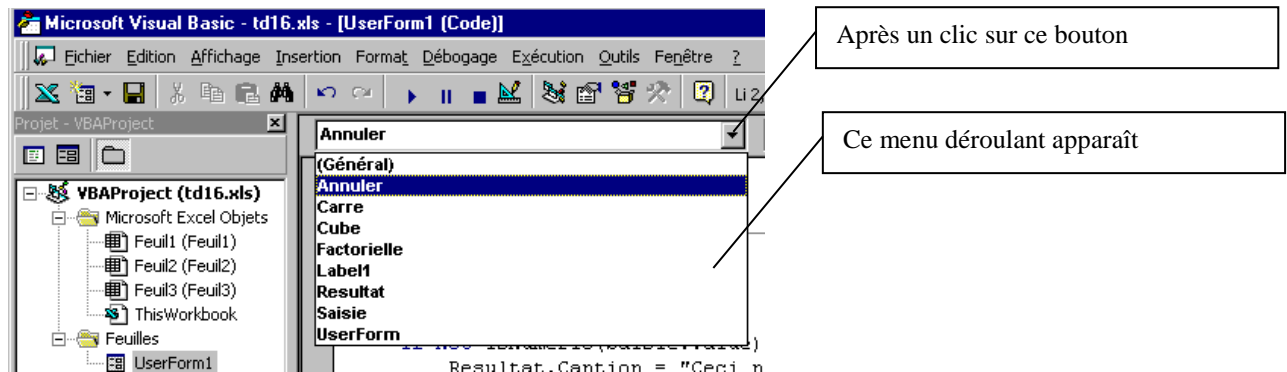
## 14.3. Mise au point de votre travail

Comme les diverses procédures que vous avez créées sont mises en oeuvre grâce à des clics sur des boutons, il serait malcommode de les démarrer directement en pas-à-pas. C'est pourquoi vous allez introduire des *points d'arrêt* dans ces procédures.

Vous pouvez remarquer que les quatre procédures que vous avez créées se trouvent, en fait, l'une à la suite de l'autre, dans un unique fragment de code, qui n'est pas considéré par VBA comme un module, mais plutôt comme le code lié à la boîte de dialogue **UserForm1**. Vous pouvez naviguer entre les procédures liées à chacun des objets présents dans la boîte de dialogue grâce à une boîte de choix déroulante située au-dessus de la zone d'affichage du code :



Si vous cliquez sur le bouton orné d'une flèche dirigée vers le bas à droite de cette boîte, vous voyez apparaître la liste des différents objets présents dans votre boîte de dialogue :



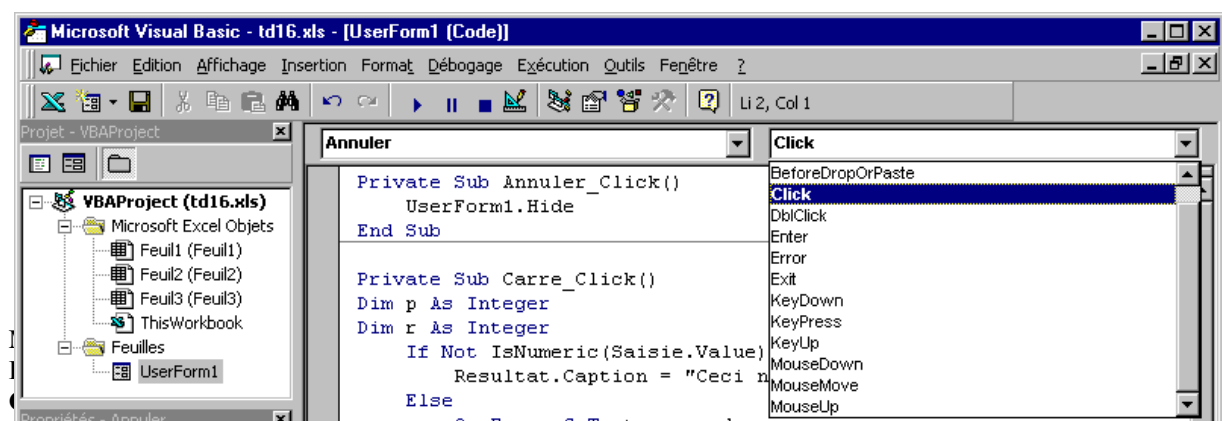
Seuls les boutons **Annuler**, **Carre**, **Cube** et **Factorielle** ont des codes qui leur sont liés. Ces codes sont tous liés à l'événement **Click** dont le nom apparaît dans la **boîte de choix d'événement** située à droite de la **boîte de choix d'objet**.

**Attention** : le choix dans la **boîte de choix d'objet** d'un objet auquel n'est affecté aucun code entraînerait la création d'une amorce de procédure d'événement liée à cet objet :

- **Click** pour un objet **Label** ou l'objet **UserForm** lui-même
- **Change** pour un objet **TextBox** tel que notre objet **Saisie**.


Ces amorces de procédures sont vides et leur présence ne gêne aucunement le fonctionnement de la boîte de dialogue. Toutefois elles encombreraient inutilement la zone d'affichage du code. Si, à la suite d'une fausse manoeuvre, vous en créez une, vous pouvez purement et simplement la supprimer du code.

Choisissez l'objet **Annuler**. Si vous développez la boîte de choix d'événement grâce à un clic sur le bouton qui se trouve à sa droite, vous voyez apparaître la liste des événements liés à l'objet, auxquels il serait possible de lier des procédures :



C'est l'événement **Click** qui est sélectionné par défaut. Vous pouvez d'ailleurs voir le texte de la procédure **Annuler\_Click** s'afficher en haut de la zone de texte. Refermez à présent le menu déroulant, en cliquant à nouveau sur le bouton qui l'a ouvert, puis :

- placez le curseur d'insertion dans la ligne contenant l'instruction **UserForm1.Hide**, et activez la rubrique de menu **Debug**→**Basculer le point d'arrêt**, qui se nomme en Anglais **Debug**→**Toggle Breakpoint**.

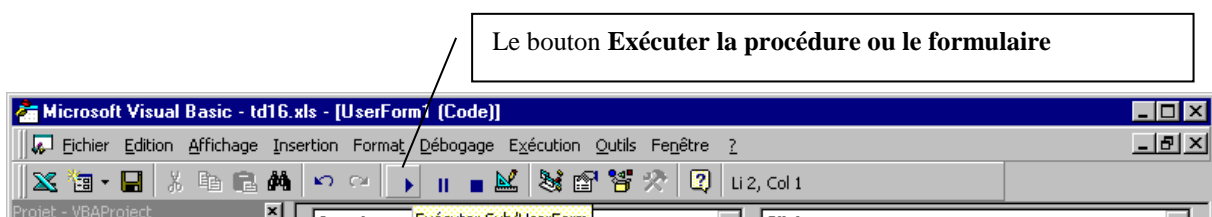
(vous pouvez également cliquer sur le bouton représentant une main levée dans la barre d'outils de mise au point, ou frapper la touche [F9]), ou enfin cliquer dans la marge  du texte du module

L'instruction apparaît à présent en blanc sur fond brun, avec un gros point brun dans la marge. C'est ainsi qu'on reconnaît d'un coup d'oeil les instructions sur lesquelles est installé un **point d'arrêt**.

```
Private Sub Annuler_Click()
    UserForm1.Hide
End Sub
```

Réalisez ensuite la même opération pour les instructions **If Not IsNumeric...** qui se trouvent dans les procédures associées à l'événement **Click** des trois autres boutons **Carre**, **Cube** et **Factorielle**.

Activez à présent la rubrique de menu **Exécution**→**Exécuter le formulaire** ( **Run**→**Run Userform** ), ou appuyez sur la touche F5, ou encore cliquez dans la barre d'outils sur le bouton orné d'une flèche bleue orientée vers la droite



Vous voyez apparaître la version définitive de votre boîte de dialogue. Vous pouvez désormais garnir la zone de texte, et cliquer sur l'un des boutons. Après un clic sur l'un des boutons **Annuler**, **Carré**, **Cube** ou **Factorielle**, vous verrez immédiatement apparaître le code de gestion de l'événement **OnClick** correspondant, et le mode **pas à pas** sera activé. Continuez comme d'habitude, avec [F8] pour avancer pas à pas dans la procédure.

Lorsque toutes les procédures fonctionneront correctement, vous supprimerez les points d'arrêt, soit individuellement par les mêmes manœuvres que celles qui vous ont permis de les créer, soit d'une seule fois en activant la rubrique de menu

**Débogage** → **Effacer tous les points d'arrêt**  
**(Debug** → **Clear All Breakpoints)**

Cette dernière rubrique n'est accessible que si on a fermé la boîte de dialogue, soit en cliquant sur son bouton **Annuler**, soit en cliquant sur la croix de Saint-André à droite de sa barre de titre.

## 14.4. Création d'un bouton pour afficher le formulaire

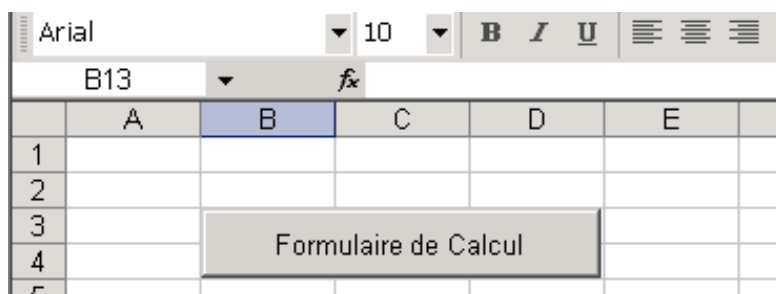
Retourner sur une feuille EXCEL, en cliquant sur le bouton correspondant de la barre de tâches

Activer la rubrique **Développeur** → **Contrôles** → **Insérer**

Créez à partir de cette **boîte à outils** un **Bouton de commande** sur la feuille EXCEL (le **drag and drop** ne fonctionne pas ici, vous devrez dimensionner le bouton à la souris). Dans la boîte de dialogue qui apparaît pour « affecter une macro » à votre bouton, cliquer sur « annuler » pour fermer cette fenêtre.

Demandez l'affichage du code associé en cliquant sur **Développeur** → **Contrôles** → **Visualiser le code**, en anglais  
**(Developer** → **Controls** → **View Code)**

Faites afficher à ce bouton un texte raisonnable, par exemple « Formulaire de Calcul », en modifiant sa propriété **Caption**.



Introduisez comme code, entre les deux instructions

```
Sub Bouton1_Clic()
```

```
End Sub
```

Madame Véronique HEIWY

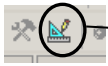
L'usage de cette brochure est réservé exclusivement aux travaux dirigés de l'Université PARIS 1, UFR Gestion-Sorbonne, L2

créées par Visual Basic, l'unique instruction

◆ `UserForm1.Show`

Retournez à la feuille EXCEL, en cliquant sur son icône dans la barre de tâches.

Quittez le mode conception en cliquant une fois sur le bouton orné d'une équerre. Fermez la barre d'outils **Boîte à outils (Toolbox)**.



Le bouton du mode *conception*.

Un clic sur votre bouton provoquera désormais l'exécution de cette instruction `UserForm1.Show`, et fera apparaître votre boîte de dialogue, que vous pourrez faire disparaître grâce à son bouton **Annuler** lorsque vous aurez fini de jouer avec les calculs dont elle est capable.

*Vous pouvez constater que les boutons de commande fournis par la boîte à outils et par la barre d'outils **Formulaires** se comportent différemment :*

- *La création du bouton de la barre d'outils **Formulaires** demande l'affectation à ce bouton d'une macro préexistante*
- *La création du bouton de la **Boîte à outils** ne demande rien de tel, par contre l'affectation d'un code au bouton se fait après coup.*

Vous pouvez maintenant sauver votre classeur.

## Annexe : texte des procédures de gestion d'événements du formulaire UserForm1

```
Private Sub Annuler_Click()  
    UserForm1.Hide  
End Sub  
  
Private Sub Carré_Click()  
    Dim p As Integer  
    Dim r As Long  
    If Not IsNumeric(Saisie.Value) Then  
        Résultat.Caption = "ceci n'est pas un nombre"  
    Else  
        On Error GoTo tropgrand  
        p = CInt(Saisie.Value)  
        On Error GoTo 0  
        r = CLng(p) * p  
        Résultat.Caption = "Le carré de " + Str(p) + " est " + Str(r)  
    End If  
    Exit Sub  
tropgrand:  
    Résultat.Caption = "ce nombre est trop grand"  
End Sub  
  
Private Sub Cube_Click()  
    Dim p As Integer  
    Dim r As Long  
    If Not IsNumeric(Saisie.Value) Then  
        Résultat.Caption = "ceci n'est pas un nombre"  
    Else  
        On Error GoTo tropgrand  
        p = CInt(Saisie.Value)  
        r = CLng(p) * p * p  
        On Error GoTo 0  
        Résultat.Caption = "Le cube de " + Str(p) + " est " + Str(r)  
    End If  
    Exit Sub  
tropgrand:  
    Résultat.Caption = "ce nombre est trop grand"  
End Sub
```

## Suite du texte des procédures de gestion d'événements du formulaire UserForm1

```
Private Sub Factorielle_Click()  
    Dim p As Integer  
    Dim r As Long  
    If Not IsNumeric(Saisie.Value) Then  
        Résultat.Caption = "ceci n'est pas un nombre"  
    Else  
        On Error GoTo tropgrand  
        p = CInt(Saisie.Value)  
        If p < 0 Then  
            Résultat.Caption = "pas de factorielle"  
        Else  
            r = 1  
            While p > 1  
                r = r * p  
                p = p - 1  
            Wend  
            Résultat.Caption = "la factorielle de " + Saisie.Value + _  
                " est " + Str(r)  
        End If  
    End If  
    Exit Sub  
tropgrand:  
    Résultat.Caption = "ce nombre est trop grand"  
End Sub
```