TUTO « EXCELLONS » : Manipuler les formes (Shapes) par VBA

L'un des aspects des plus remarquables d'EXCEL réside dans sa capacité à dessiner **une multitude de formes graphiques** plus ou moins complexes, de **pouvoir gérer leurs tailles, leurs couleurs, leurs positionnements, leurs comportements…**

Ce tutoriel présente de façon concise ce qu'il est nécessaire de connaître **pour manipuler les formes par programmation VBA**.

(Les copies-écrans du présent document sont issues d'EXCEL version 365)

L'icône signale une remarque réclamant une **attention particulière**.

Table des matières

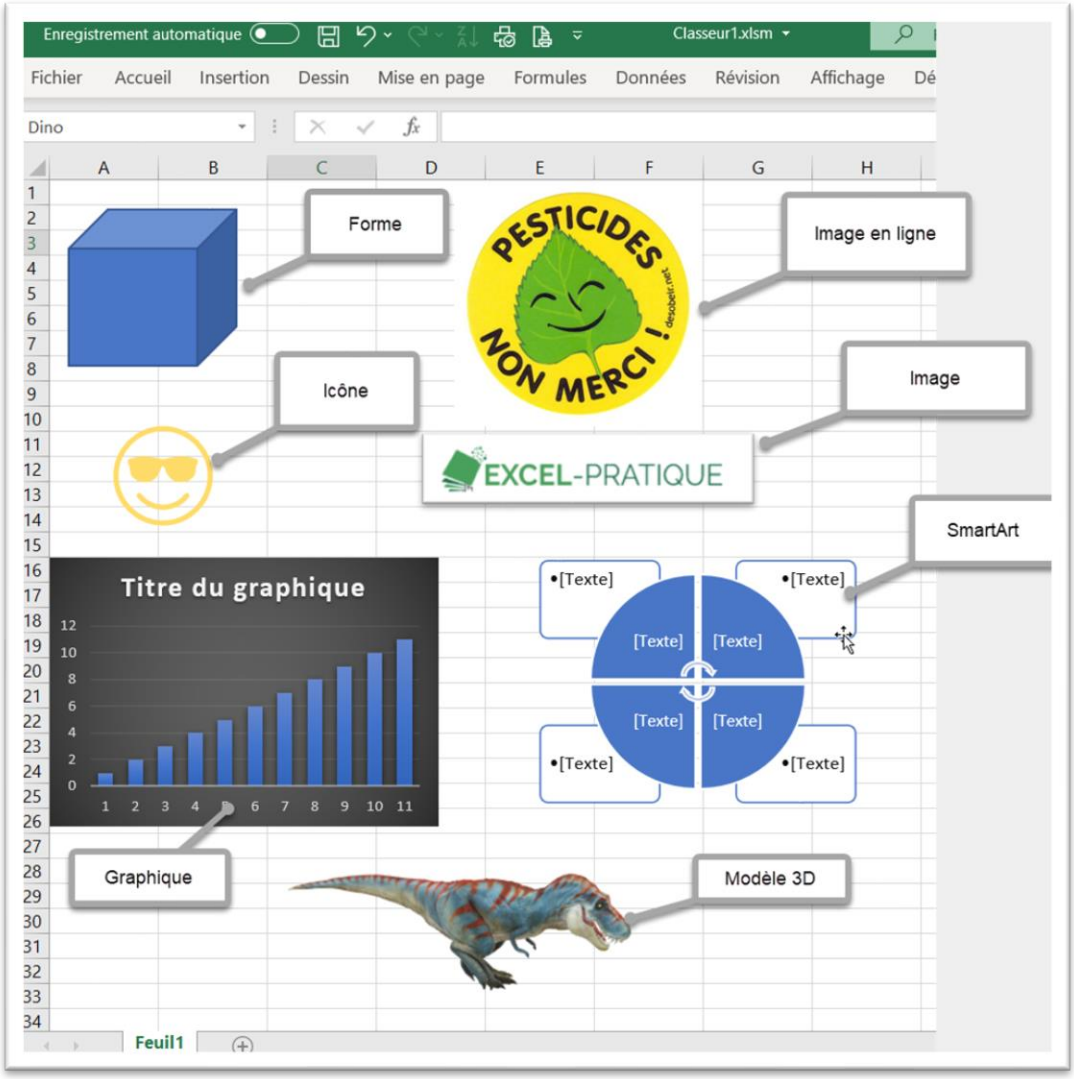
QU'EST-CE QU'UNE FORME (SHAPE) POUR EXCEL 3
UTILISONS L'ENREGISTREUR DE MACROS 4
LISTONS TOUTES LES FORMES CONTENUES DANS UNE FEUILLE 4
REFERENCONS LES FORMES 5
Référençons par index 5
Référençons par nom 5
Affectons un nom explicite 5
POSITIONNONS LES FORMES 6
Positionnons la forme lors de sa création 6
Repositionnons une forme 6
Repositionnons une forme par rapport à une cellule 6
Gérons les propriétés « Placement » 6
DIMENSIONNONS LES FORMES 7
Dimensionnons la forme lors de sa création 7
Redimensionnons une forme 7
Redimensionnons une forme par rapport à la taille d'une cellule 7
Redimensionnons un ensemble de formes avec les méthodes 'ScaleWidth-ScaleHeight' 8
INSERONS UNE NOUVELLE FORME (SHAPE) 9

AddChart2 (AddChart) – pour ajouter un graphe 9
AddCallout – pour ajouter une légende 10
AddConnector – pour ajouter un trait ou un connecteur 11
AddCurve – pour ajouter une courbe de Bézier 12
AddFormControl – pour ajouter un contrôle de formulaire 13
AddLabel/AddTextbox – pour ajouter une zone de texte 14
AddOLEObject – pour ajouter un lien vers un document externe 15
AddPicture2/AddPicture – pour ajouter une image 16
AddPolyline – pour ajouter un polygone ou une ligne brisée 17
AddShape – pour ajouter une forme automatique 19
AddSmartArt – pour ajouter une forme SmartArt 20
AddTextEffect – pour ajouter une forme WordArt 22
Add3DModel – pour ajouter un modèle 3D 23
AGISSONS SUR LES COULEURS ET LE REMPLISSAGE 24
AJOUTONS ET FORMATONS DU TEXTE 26
ASSIGNONS UNE MACRO 28
RELIONS DES FORMES AVEC UN CONNECTEUR 29

QU'EST-CE QU'UNE FORME (SHAPE) POUR EXCEL

Pour EXCEL, la **classe « Shape »** (traduite ici par **« Forme »**) comprend une **multitude d'objets graphiques**.

Dans la figure ci-dessous, quelques objets graphiques membres de la classe « Forme » :



Une forme (shape) peut être un **objet « simple »** comme une forme « automatique », une icône, une image ou bien un **objet « complexe »** comme un graphique, un objet **SmartArt**, un **modèle 3D**…

Une partie des **méthodes et propriétés de la classe « Shape »** décrites dans ce document sont utilisables pour **toutes les formes**.

(Cependant, ayons en tête que certaines propriétés et méthodes de la classe « Shape » ne concernent que **certaines d'entre elles**, par exemple, voir plus loin la méthode **ScaleWidth**).

UTILISONS L'ENREGISTREUR DE MACROS

Du fait de la **richesse** et de la **diversité** des méthodes et propriétés exposées par les formes (shapes), nous **gagnons beaucoup de temps** en utilisant, dans un premier temps, **l'enregistreur de macros**.

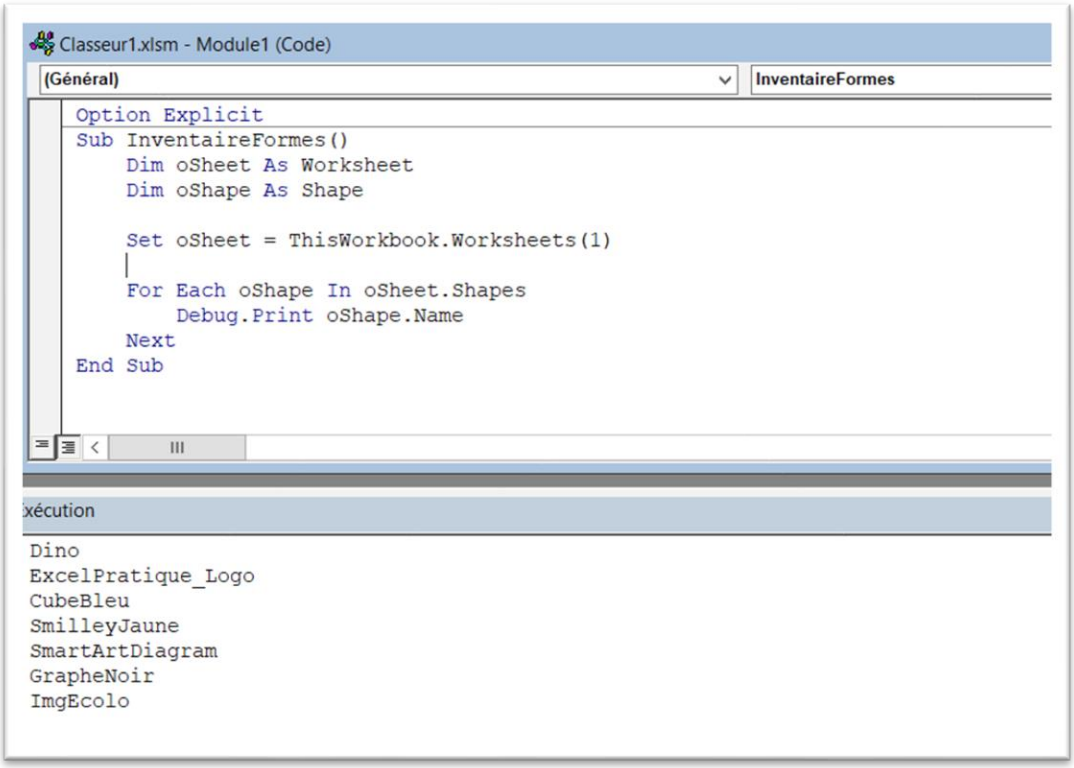
Dans un second temps, nous corrigerons le code généré pour l'**adapter** et/ou le **simplifier**.

Pour une bonne utilisation de l'enregistreur de macros, reportons-nous à la **fiche Excellons « Bonnes Pratiques VBA – Enregistreur de Macros ».**

LISTONS TOUTES LES FORMES CONTENUES DANS UNE FEUILLE

L'objet **'Shapes'** d'une **feuille de calcul** regroupe la collection de **toutes les formes** (shapes) présentes dans une feuille.

Le fragment de code ci-dessous **énumère** dans **la fenêtre d'exécution** les **noms de toutes les formes** présentes dans la feuille « Feuil1 » de l'exemple :



Pour afficher la fenêtre d'exécution : **Ctrl+G**

REFERENCONS LES FORMES

Référençons par index

Nous pouvons **référencer une forme par son index** dans la collection des « Shapes ». Par exemple, pour nous référer à la forme d'index 3 :

```
Set oShape = oSheet.Shapes(3)
```

Cette méthode de référencement n'est ni très pratique ni très explicite, aussi nous privilégierons le référencement pas nom.

Référençons par nom

Lorsque nous créons une forme dans une feuille (manuellement ou par code), EXCEL lui affecte **un nom par défaut**. Ce nom reprend le **type de forme** et y ajoute une **numérotation chronologique**. Par exemple :

```
« Image 7 »
```

Nous **référençons cette forme par nom** de la façon suivante :

```
Set oShape = oSheet.Shapes("Image 7")
```

Aux noms affectés par défaut par EXCEL, nous pouvons préférer que nos formes portent des noms explicites.

Affectons un nom explicite

Nous **affectons préférentiellement un nom explicite** à nos formes en modifiant la propriété **« Name »** de la forme.

Par exemple :

```
Set oShape = oSheet.Shapes("Image 7")
oShape.Name = "Excellons_Logo"
```

POSITIONNONS LES FORMES

Positionnons la forme lors de sa création

Lorsque nous créons une forme dans une feuille par code en utilisant l'une des méthodes disponibles (Addxxx, voir plus loin), nous devons indiquer **sa position exprimée en points** pour son coin supérieur gauche :

Addxxx (..., ***gauche***, ***haut,*** ...)

```
Set oShape = oSheet.AddShape(msoShapeBevel, 5, 5, ...)
```

Ainsi, pour ajouter un rectangle biseauté positionné à 5 points à gauche et en haut de la feuille

Pour convertir les centimètres en points, nous disposons de la fonction '**Application.CentimetersToPoints'** . Par exemple, pour positionner la forme à 1 centimètre à gauche et 1 centimètre en haut de la feuille nous écrirons :

```
Set oShape = oSheet.AddShape(msoShapeBevel, Application.CentimetersToPoints(1), Application.CentimetersToPoints(1), ...)
```

Repositionnons une forme

Pour repositionner une forme, nous modifions ses propriétés **« Left »** et **« Top »** en indiquant leurs valeurs **en points** (ou en centimètres en utilisant la fonction '**Application.CentimetersToPoints'**). Par exemple, pour repositionner la forme à 50 points à gauche et en haut de la feuille :

```
oShape.Left = 50
oShape.Top = 50
```

Repositionnons une forme par rapport à une cellule

Nous pouvons positionner une forme par rapport à une cellule en se référençant aux propriétés **« Left »** et **« Top »** de la cellule. Par exemple, pour repositionner la forme sur le coin haut-Gauche de la cellule « D4 » :

```
Set oSheet = ThisWorkbook.Worksheets(1)
Set oShape = oSheet.Shapes("Excellons_Logo")
Set oRange = oSheet.Range("D4")

oShape.Left = oRange.Left
oShape.Top = oRange.Top
```

Gérons les propriétés « Placement »

La propriété **« Placement »** de l'objet **« Shape »** nous permet de gérer le positionnement des formes.

Elle peut prendre les valeurs :

- **xlFreeFloating** = la forme conserve son positionnement lorsque les cellules sous-jacentes sont redimensionnées.
- **xlMove** = la forme se déplace avec les cellules sous-jacentes (sans être redimensionnée).
- x**lMoveAndSize** = la forme se déplace et se redimensionne avec les cellules sous-jacentes.

DIMENSIONNONS LES FORMES

Dimensionnons la forme lors de sa création

Lorsque nous créons une forme dans une feuille par code en utilisant l'une des méthodes disponibles (Addxxx, voir plus loin), nous devons indiquer **ses dimensions exprimées en points** :

Addxxx (…, ***Largeur***, ***Hauteur,*** …)

Ainsi, pour ajouter un rectangle biseauté de 100 points sur 100 :

```
Set oShape = oSheet.AddShape(msoShapeBevel, …, …, 100,100)
```

Redimensionnons une forme

Pour redimensionner une forme, nous modifions ses propriétés **« Width »** et **« Height »** en indiquant leurs valeurs **en points** (ou en centimètres en utilisation la fonction '**Application.CentimetersToPoints'**).

Par exemple, pour redimensionner la forme à une largeur et une hauteur de 100 points :

```
oShape.Width = 100
oShape.Height = 100
```

Redimensionnons une forme par rapport à la taille d'une cellule

Nous pouvons redimensionner une forme par rapport à la taille d'une cellule (ou d'une plage de cellules) en se référençant aux propriétés **« Width »** et **« Height »** de la cellule (ou de la plage).

Par exemple, pour redimensionner la forme à la taille de la plage « D4:F6 » :

```
Set oSheet = ThisWorkbook.Worksheets(1)
Set oShape = oSheet.Shapes("Excellons_Logo")
Set oRange = oSheet.Range("D4:F6")

oShape.Width = oRange.Width
oShape.Height = oRange.Height
```

Gérons les propriétés « LockAspectRatio »

La propriété **« LockAspectRatio »** de l'objet **« Shape »** nous permet d'indiquer comment la forme doit se comporter lors d'un redimensionnement.

Elle peut prendre les valeurs suivantes :

- **msoTrue** = la forme se redimensionne en conservant ses proportions. Si, par exemple, nous redimensionnons la largeur de la forme, sa hauteur se redimensionne proportionnellement.
- **msoFalse** = la forme ne conserve pas ses proportions lorsque nous la redimensionnons. Si, par exemple, nous redimensionnons la largeur de la forme, sa hauteur reste inchangée.

Redimensionnons un ensemble de formes avec les méthodes 'ScaleWidth-ScaleHeight'

Avec les méthodes **« ScaleWidth »** et **« ScaleHeight »** nous **diminuons** ou **augmentons** proportionnellement la taille des formes en largeur (ScaleWidth) ou en hauteur (ScaleHeight).

Ce fragment de code, **diminue la taille des formes** d'une même feuille de la moitié de leurs tailles actuelles :

```
Sub ShrinkAllShapes()
   Dim oSheet As Worksheet
   Dim oShape As Shape

   Set oSheet = ThisWorkbook.Worksheets(1)
   'On parcourt toutes les formes présentes dans la feuille
   For Each oShape In oSheet.Shapes
      'On conserve les proportions de la forme
      oShape.LockAspectRatio = msoTrue
      'Dans le cas où la forme ne supporte pas la méthode ScaleWidth, on ignore l'erreur
      On Error Resume Next
      'On diminue la taille de la forme de moitié = 50%
      oShape.ScaleWidth 0.5, msoTrue
      'On rétablit la gestion des erreurs
      On Error GoTo 0
   Next
End Sub
```

Certaines formes ne supportent pas la méthode **'ScaleWidth'** utilisée, c'est la raison pour laquelle l'instruction '**On Error Resume Next'** est utilisée dans ce fragment de code.

Pour la gestion des erreurs d'exécution, reportons-nous au **document Excellons « Bonnes Pratiques VBA – Gestion des erreurs »**

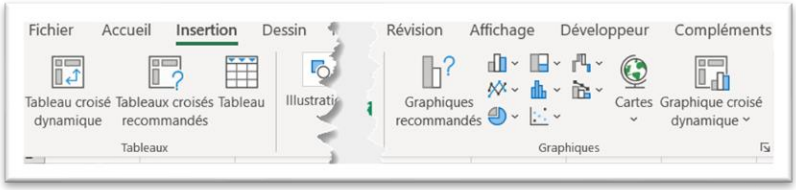
INSERONS UNE NOUVELLE FORME (SHAPE)

L'objet **'Shapes'** expose un ensemble de **méthodes** permettant d'insérer par code **une nouvelle forme** dans une feuille.

Chacune de ces méthodes insère un **type de forme (shape) particulier**.

AddChart2 (AddChart) – pour ajouter un graphe

(Raccourci Alt+S+….)



La méthode **AddChart2** a été introduite avec la **version 2013 d'EXCEL**. La méthode **AddChart** a été, quant à elle, introduite dans la **version 2007 d'EXCEL**.

La différence entre ces deux méthodes tient au fait qu'Addchart2 dispose de 2 arguments supplémentaires :

AddChart (XlChartType, gauche, haut, largeur, hauteur)

AddChart2 (***Style***, XlChartType, gauche, haut, largeur, hauteur, ***NewLayout***)

Pour créer complètement un graphique par code VBA, nous devrons valoriser des propriétés n'appartenant pas à **l'objet 'Shape'** mais à **l'objet 'Chart'** (par exemple, les données devant composer le graphe).

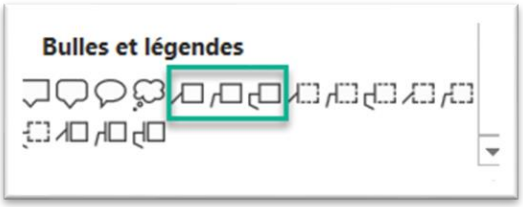
Ce fragment de code illustre l'**articulation** entre l'objet '**Shape'** et l'objet '**Chart'** :

```
Sub AddGraph()
  Dim oSheet As Worksheet
  Dim oShape As Shape
  Dim oChart As Chart
  Set oSheet = ThisWorkbook.Worksheets(2)
  'On insère le graphe dans la feuille avec la méthode AddChart2
  Set oShape = oSheet.Shapes.AddChart2(333, xl3DPie, , , , True)
  With oShape
    'On référence l'objet Chart
    Set oChart = oShape.Chart
    With oChart
      'On affecte la source de données du graphe
      .SetSourceData oSheet.Range("P3:P12")
      'On donne un titre au graphe
      .ChartTitle.Text = "MON GRAPHE"
    End With
  End With
End Sub
```

(Ce document est consacré à **l'objet formes 'Shapes'**, il ne **traite pas en détail de l'objet 'Chart'**)

AddCallout – pour ajouter une légende

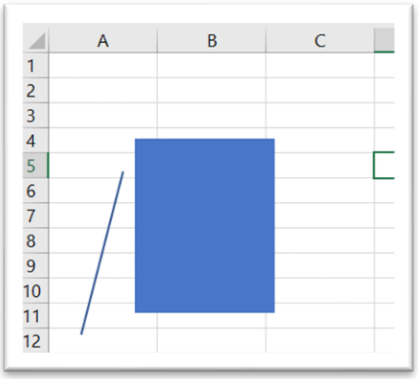
(Raccourci Alt+S+ZI+ER…) :



Ce fragment de code insère une légende avec lien à droite :

```
Sub AddLegend()
   Dim oSheet As Worksheet
   Dim oShape As Shape

   Set oSheet = ThisWorkbook.Worksheets(1)
   Set oShape = oSheet.Shapes.AddCallout(msoCalloutOne, 50, 50, 80, 100)
End Sub
```

Résultat obtenu :



Dans l'énumération des types exposés par la méthode '**AddCallout'** la valeur '**msoCallMixed'** n'est pas fonctionnelle.

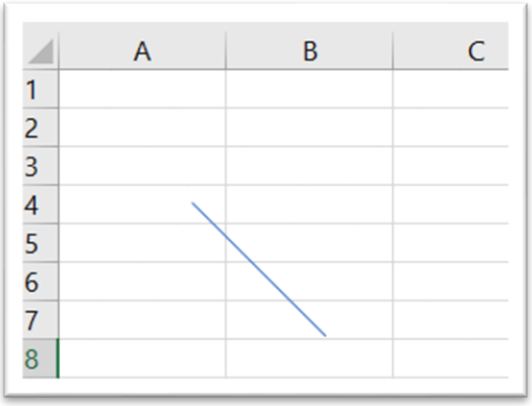
AddConnector – pour ajouter un trait ou un connecteur

(Raccourci Alt+S+ZI+ER….)

Ce fragment de code insère un trait :

```
Sub AddTrait()
   Dim oSheet As Worksheet
   Dim oShape As Shape

   Set oSheet = ThisWorkbook.Worksheets(1)
   Set oShape = oSheet.Shapes.AddConnector(msoConnectorStraight, 50, 50, 100, 100)
End Sub
```

Résultat obtenu :



La méthode '**AddConnector**' permet également de **relier deux formes** entre-elles (voir plus loin dans ce document « **RELIONS DES FORMES AVEC UN CONNECTEUR**).

AddCurve – pour ajouter une courbe de Bézier

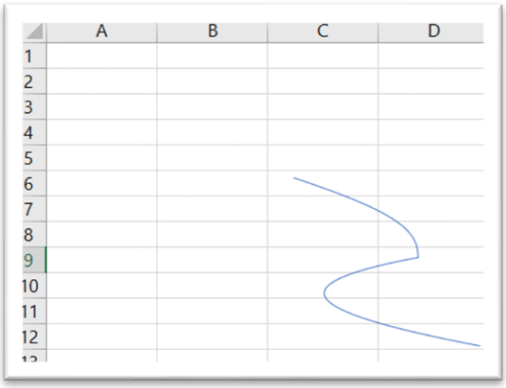
(Raccourci Alt+S+ZI+ER….)

Les points de la courbe doivent être fournis à la méthode '**AddCurve'** dans un **tableau de valeurs**.

Par exemple, ce fragment de code insère une courbe en S avec des coordonnées de début (210,130) et de fin (315,225) :

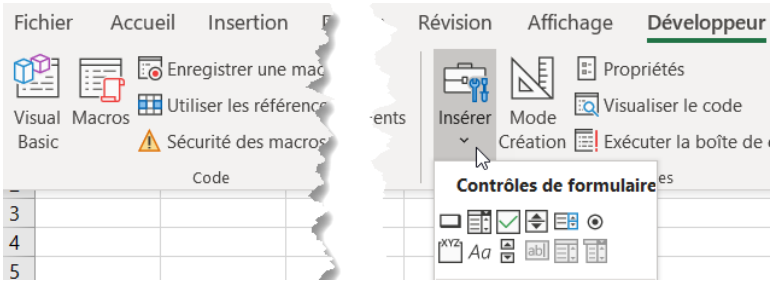
```
Sub AddCourbeBezier()
   Dim oSheet As Worksheet
   Dim oShape As Shape
   Dim aPoints(1 To 7, 1 To 2) As Single
   aPoints(1, 1) = 210
   aPoints(1, 2) = 130
   aPoints(2, 1) = 270
   aPoints(2, 2) = 150
   aPoints(3, 1) = 280
   aPoints(3, 2) = 160
   aPoints(4, 1) = 280
   aPoints(4, 2) = 175
   aPoints(5, 1) = 200
   aPoints(5, 2) = 190
   aPoints(6, 1) = 210
   aPoints(6, 2) = 205
   aPoints(7, 1) = 315
   aPoints(7, 2) = 225

   Set oSheet = ThisWorkbook.Worksheets(1)
   Set oShape = oSheet.Shapes.AddCurve(aPoints())
End Sub
```

Résultat obtenu :



AddFormControl – pour ajouter un contrôle de formulaire

(Raccourci Alt+V+B….)

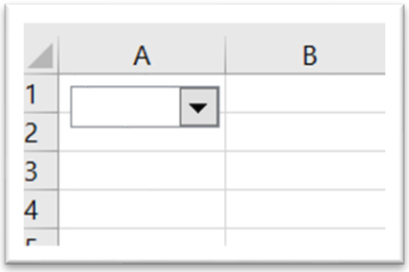


Avec la méthode '**AddFormControl',** nous indiquons le **type de contrôle** à insérer par la précisant par l'une des valeurs exposées par l'énumération **XlFormControl**.

Ce fragment de code insère un contrôle de liste déroulante :

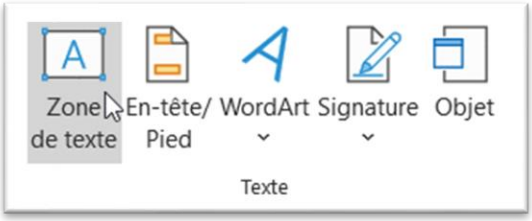
```
Sub AddCombobox()
   Dim oSheet As Worksheet
   Dim oShape As Shape

   Set oSheet = ThisWorkbook.Worksheets(1)
   Set oShape = oSheet.Shapes.AddFormControl(xlDropDown, 5, 5, 55, 15)
End Sub
```

Résultat obtenu :



AddLabel/AddTextbox – pour ajouter une zone de texte

(Raccourci Alt+S+8….)

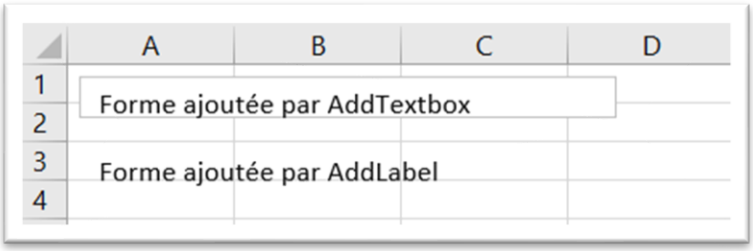


Les méthodes '**AddTextBox'** et **'AddLabel',** permettent d'insérer une zone de texte ou de libellé.

Ce fragment de code insère une zone de texte et une zone de libellé :

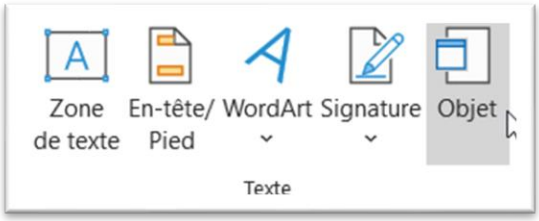
```
Sub AddTextBox_Label()
   Dim oSheet As Worksheet
   Dim oShape As Shape

   Set oSheet = ThisWorkbook.Worksheets(1)
   Set oShape = oSheet.Shapes.AddTextBox(msoTextOrientationHorizontal, 5, 5, 200, 15)
   oShape.TextEffect.Text = "Forme ajoutée par AddTextbox"
   Set oShape = oSheet.Shapes.AddLabel(msoTextOrientationHorizontal, 5, 30, 200, 15)
   oShape.TextEffect.Text = "Forme ajoutée par AddLabel"
End Sub
```

Résultat obtenu :



AddOLEObject – pour ajouter un lien vers un document externe

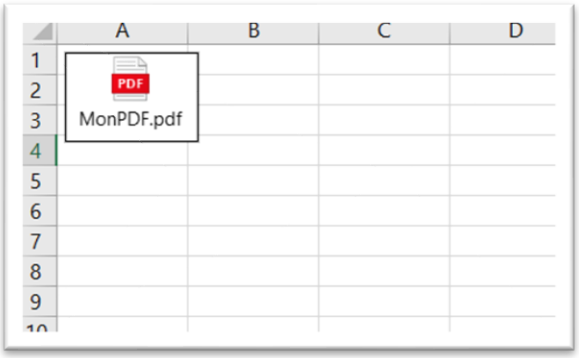
- Ajoute un lien à un document externe (Word, PDF, Excel, Texte…) Raccourci Alt+S+ZT+W) :



Ce fragment de code insère un lien sous forme d'une icône vers un document PDF intégré au classeur :

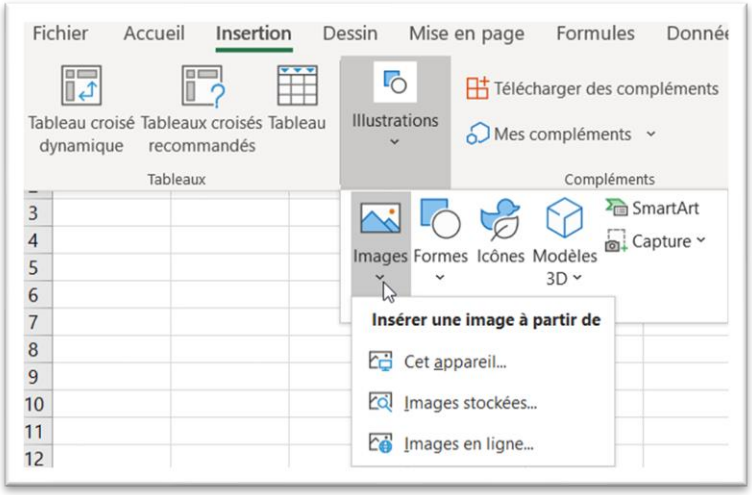
```
Sub AddPDF()
   Dim oSheet As Worksheet
   Dim oShape As Shape
   Dim oRange As Range

   Set oSheet = ThisWorkbook.Worksheets(1)
   Set oShape = oSheet.Shapes.AddOLEObject(, ThisWorkbook.Path & "\MonPDF.pdf", , , , , , 5, 5, 200, 200)

End Sub
```

Résultat obtenu :



AddPicture2/AddPicture – pour ajouter une image

(Raccourci Alt+S+ZI+I1…)



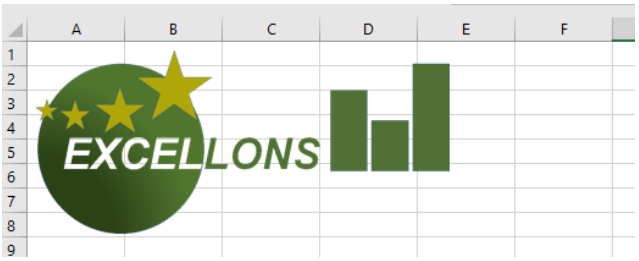
La seule différence entre les 2 méthodes : la méthode **AddPicture2** expose le paramètre supplémentaire « **compress** » (indiquant si l'image doit être compressée ou non).

Pour convertir les centimètres en points, nous disposons de la fonction '**Application.CentimetersToPoints**' .

Ce fragment de code insère le logo « Excellons » compressé en position 5 points du haut et de la gauche de la feuille, et ayant une longueur de 9 cm et une largeur de 4 cm :

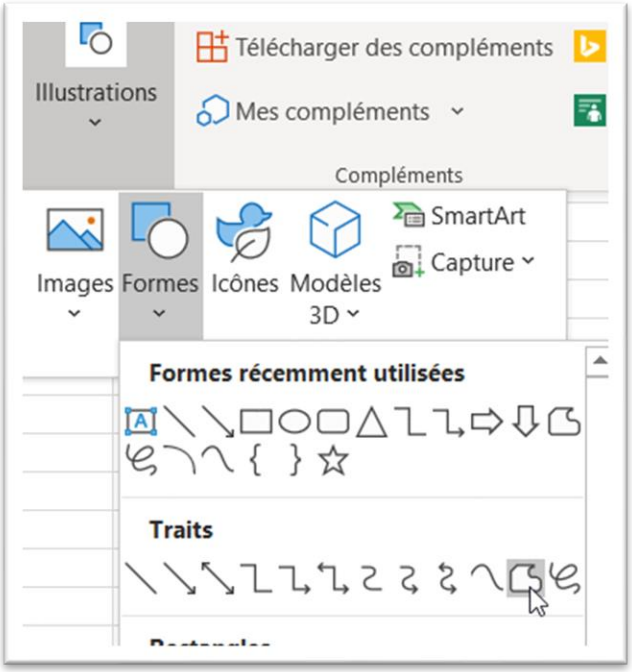
```
Sub AddImage()
   Dim oSheet As Worksheet
   Dim oShape As Shape
   Dim oRange As Range

   Set oSheet = ThisWorkbook.Worksheets(1)
   Set oShape = oSheet.Shapes.AddPicture2(ThisWorkbook.Path & "\Logo_Excelons.png", msoFalse, msoTrue, 5, 5 _
                , Application.CentimetersToPoints(9), Application.CentimetersToPoints(4), True)
End Sub
```

Résultat obtenu :



AddPolyline – pour ajouter un polygone ou une ligne brisée

(Raccourci Alt+S+ZI+ER…)



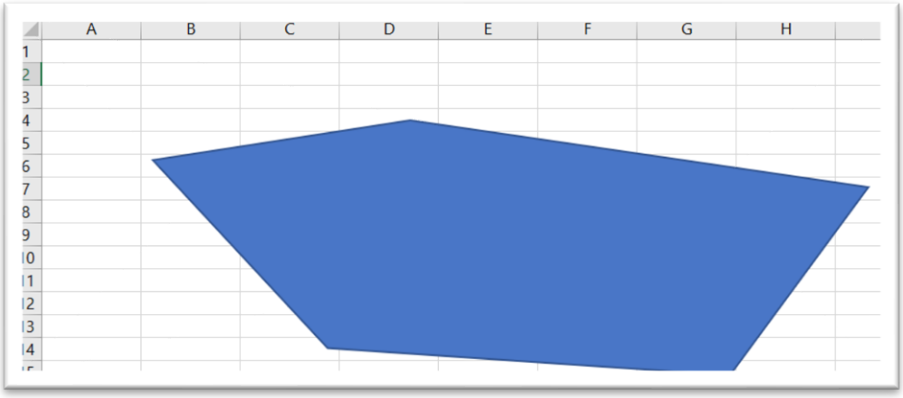
Les points de la figure voulu sont à fournie à la méthode '**AddPolyLine'** dans un **tableau de valeurs**.

Pour constituer une figure fermée, le **dernier point** fourni doit posséder **les mêmes coordonnées** que le **premier point** fourni.

Ce fragment de code trace **un pentagone** (figure de 5 côtés).

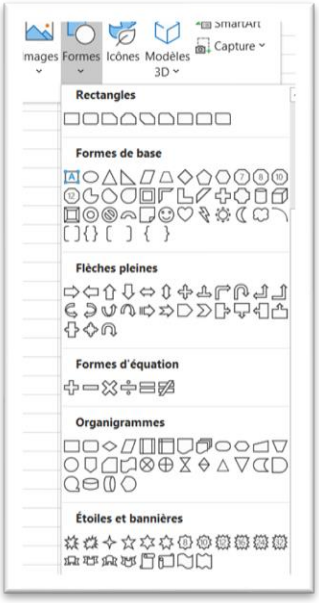
```
Sub TracePentagone()
   Dim oSheet As Worksheet
   Dim oShape As Shape
   Dim oRange As Range
   Dim aPoints(1 To 6, 1 To 2) As Single

   'On affecte les coordonnées des sommets du pentagone
   aPoints(1, 1) = 520
   aPoints(1, 2) = 93
   aPoints(2, 1) = 433
   aPoints(2, 2) = 211
   aPoints(3, 1) = 180
   aPoints(3, 2) = 194
   aPoints(4, 1) = 70
   aPoints(4, 2) = 76
   aPoints(5, 1) = 232
   aPoints(5, 2) = 51
   aPoints(6, 1) = 520 'le dernier sommet possède des coordonnées identiques au premier sommet
   aPoints(6, 2) = 93  '-> pour fermer le polygone

   Set oSheet = ThisWorkbook.Worksheets(2)
   'On insère le pentagone dans la feuille
   Set oShape = oSheet.Shapes.AddPolyline(aPoints)
End Sub
```

Résultat obtenu :



AddShape – pour ajouter une forme automatique

(Raccourci Alt+S+ZI+ER…)

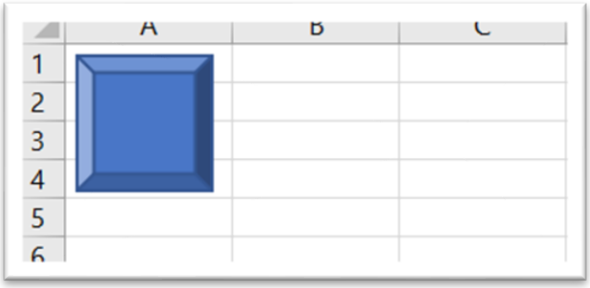


Le type de la forme désirée est indiqué par l'argument **« Type »** de la méthode **AddShape** (voir l'énumération MsoAutoShapeType comprenant plus de 180 formes différentes).

Ce fragment de code insère une forme de type **« Bouton en relief »** :

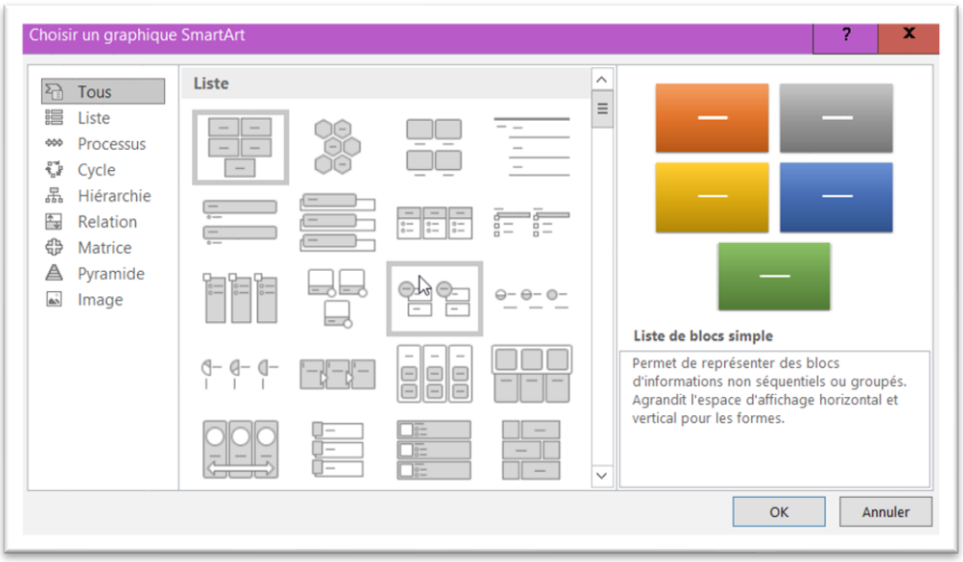
```
Sub AddFormeBasique()
   Dim oSheet As Worksheet
   Dim oShape As Shape
   Dim oRange As Range

   Set oSheet = ThisWorkbook.Worksheets(3)
   Set oShape = oSheet.Shapes.AddShape(msoShapeBevel, 5, 5, 50, 50)
End Sub
```

Résultat obtenu :



AddSmartArt – pour ajouter une forme SmartArt

(Raccourci Alt+S+ZI+RM…)

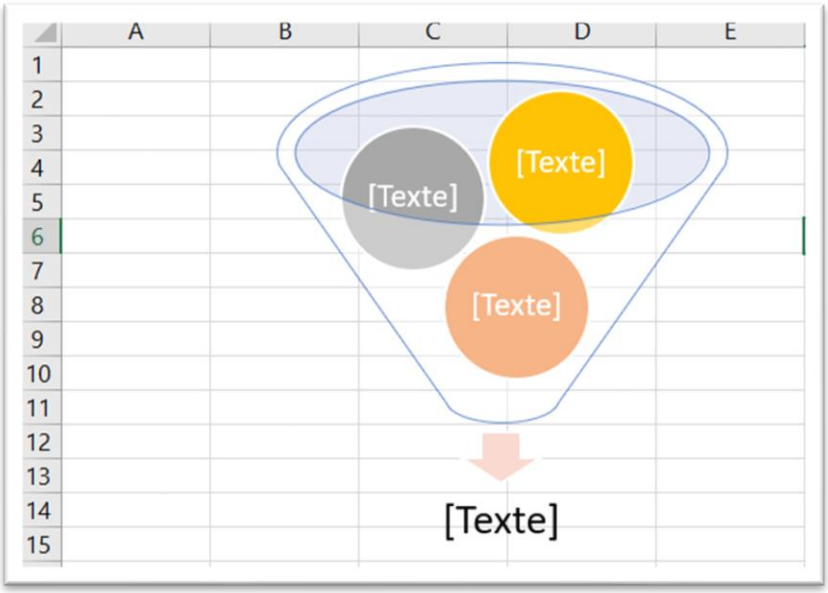


Pour obtenir la forme voulue, nous spécifions l'objet « **SmartArtLayout** » voulu comme argument de la méthode **« AddSmartArt ».**

Le fragment de code suivant, insère la **forme SmartArt « Entonnoir »** :

```
Sub AddCone()
   Dim oSheet As Worksheet
   Dim oShape As Shape
   Dim oSaLayout As SmartArtLayout

   Set oSheet = ThisWorkbook.Worksheets(1)
   'On pointe sur l'illustration 'Entonnoir'
   Set oSaLayout = Application.SmartArtLayouts(68)
   'On créé une nouvelle forme contenant l'illustration voulue
   Set oShape = oSheet.Shapes.AddSmartArt(oSaLayout, 5, 5)
   'On affecte une couleur à la forme
   oShape.SmartArt.Color = Excel.Application.SmartArtColors(4)
End Sub
```

Résultat obtenu :

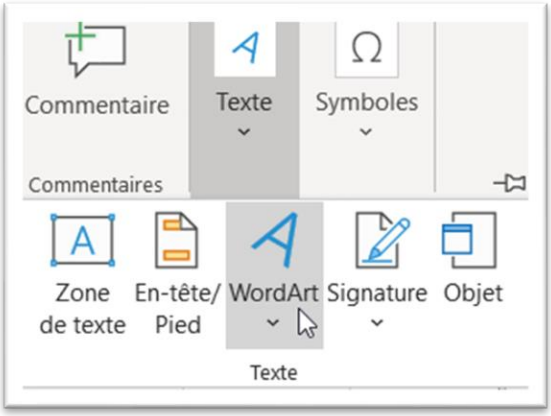


Ce document **traite uniquement de l'objet « Shape »**.

Pour **une référence complète aux formes SmartArt**, reportons-nous au **document Excellons « Astuces – Référence SMARTART »**

AddTextEffect – pour ajouter une forme WordArt

(Raccourci Alt+S+ZT+1)



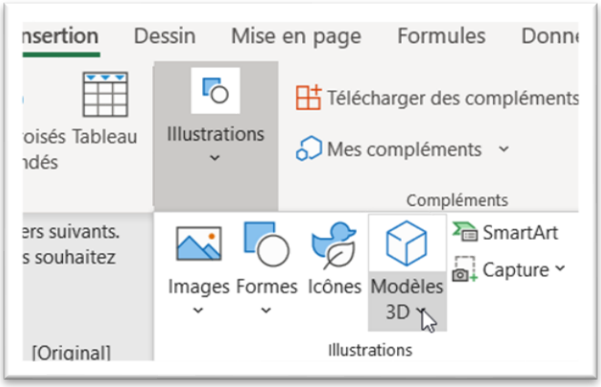
Le fragment de code suivant, insère une **forme WordArt « EXCELLONS.org »** avec effet 7 :

```
Sub AddWordArt()
   Dim oSheet As Worksheet
   Dim oShape As Shape

   Set oSheet = ThisWorkbook.Worksheets(1)
   Set oShape = oSheet.Shapes.AddTextEffect(msoTextEffect7, "EXCELLONS.org", "Arial Black", 36 _
                  , msoFalse, msoFalse, 5, 5)
End Sub
```

<u>Résultat obtenu :</u>

Add3DModel – pour ajouter un modèle 3D

(Raccourci Alt+S+ZI+S3)



Le fragment de code suivant, insère une **forme 3D animée « Volcan »** :

```
Sub Add3D_Volcan()
   Dim oSheet As Worksheet
   Dim oShape As Shape

   Set oSheet = ThisWorkbook.Worksheets(1)
   'On insère la forme 3D
   Set oShape = oSheet.Shapes.Add3DModel( _

"https://cdn.hubblecontent.osi.office.net/models/publish/524bad6218044d5c8e13f4fa93200128/glb/divergent_boundary.glb" _
      , msoFalse, msoTrue, 5, 5, 350, 125)
End Sub
```

<u>Résultat obtenu :</u>

AGISSONS SUR LES COULEURS ET LE REMPLISSAGE

Les formes (shapes) exposent un ensemble de propriétés nous permettant **d'agir sur leurs couleurs** et **remplissage**.

Reprenons le pentagone créé précédemment.

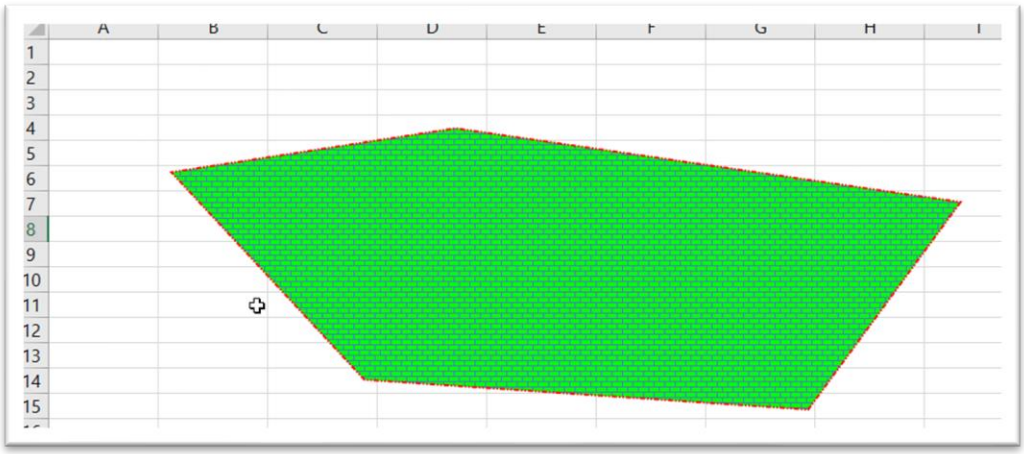
Le fragment de code suivant, **colorise les bordures** de ce pentagone par des **tirets rouges :**

```
Sub ColoreBorduresPentagone()
   Dim oSheet As Worksheet
   Dim oShape As Shape

   Set oSheet = ThisWorkbook.Worksheets(1)
   'On insère la forme 3D
   Set oShape = oSheet.Shapes("Pentagone")

   With oShape
      'On colorise les bordures du pentagone en tirets rouges
      With .Line
         .DashStyle = msoLineDashDotDot
         .ForeColor.RGB = RGB(255, 0, 0)
      End With
   End With
End Sub
```

Le fragment de code suivant, **remplit** le pentagone de **briques vertes :**

```
Sub RemplissagePentagone()
   Dim oSheet As Worksheet
   Dim oShape As Shape

   Set oSheet = ThisWorkbook.Worksheets(1)
   'On insère la forme 3D
   Set oShape = oSheet.Shapes("Pentagone")

   With oShape
      'On remplit le pentagone de briques vertes
      With .Fill
         .Patterned msoPatternHorizontalBrick
         .BackColor.RGB = RGB(0, 255, 0)
      End With
   End With
End Sub
```

Résultat obtenu :



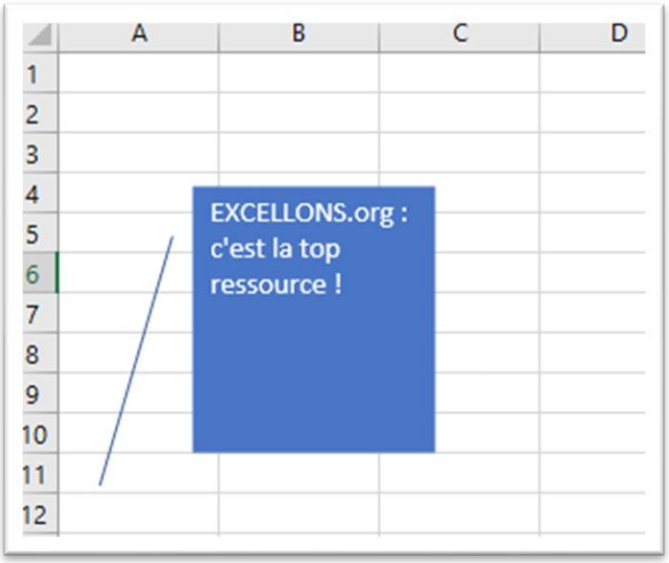
AJOUTONS ET FORMATONS DU TEXTE

Toutes les formes (shapes) exposent une propriété **'TextFrame'** nous permettant d'ajouter du texte et de le mettre en forme.

Reprenons la forme **« Légende »** créée précédemment et **ajoutons du texte** avec le fragment de code :

```
Sub AddTextToLegend()
   Dim oSheet As Worksheet
   Dim oShape As Shape

   Set oSheet = ThisWorkbook.Worksheets(1)
   Set oShape = oSheet.Shapes("Legende")
   oShape.TextFrame.Characters.Text = "EXCELLONS.org : c'est la top ressource !"
End Sub
```

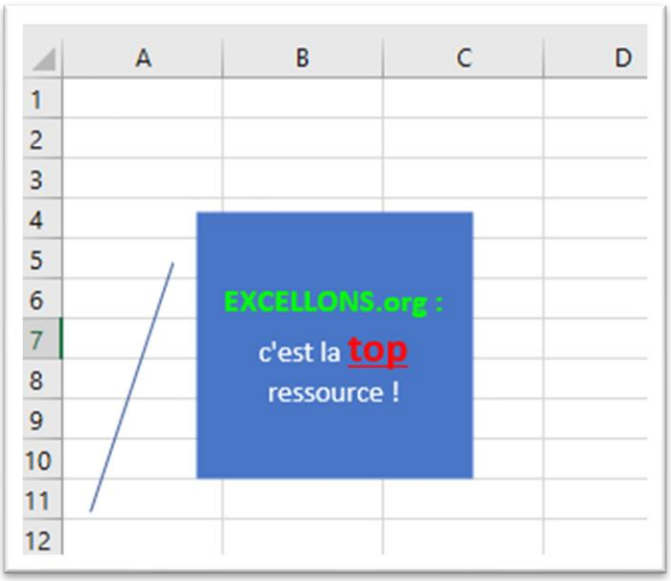
Résultat obtenu :



Formatons le texte inséré précédemment avec le fragment de code :

```
Sub FormatTextToLegend()
   Dim oSheet As Worksheet
   Dim oShape As Shape

   Set oSheet = ThisWorkbook.Worksheets(1)
   Set oShape = oSheet.Shapes("Legende")

   'Formatons les 16 premiers caractères du texte : corps 12, en gras et en vert
   With oShape.TextFrame.Characters(1, 16).Font
      .Color = RGB(0, 255, 0)
      .Size = 12
      .Bold = True
   End With
   'Formatons les 3 caractères à la partir de la position 27 du texte: corps 16, en gras, en rouge et
soulignés
   With oShape.TextFrame.Characters(27, 3).Font
      .Color = RGB(255, 0, 0)
      .Size = 16
      .Bold = True
      .Underline = xlUnderlineStyleSingle
   End With
   'Centrons le texte horizontalement et verticalement
   oShape.TextFrame.HorizontalAlignment = xlHAlignCenter
   oShape.TextFrame.VerticalAlignment = xlVAlignCenter
End Sub
```

Résultat obtenu :



ASSIGNONS UNE MACRO

Notre possibilité **d'assigner une macro à une forme** (Shape) constitue une caractéristique particulièrement intéressante.

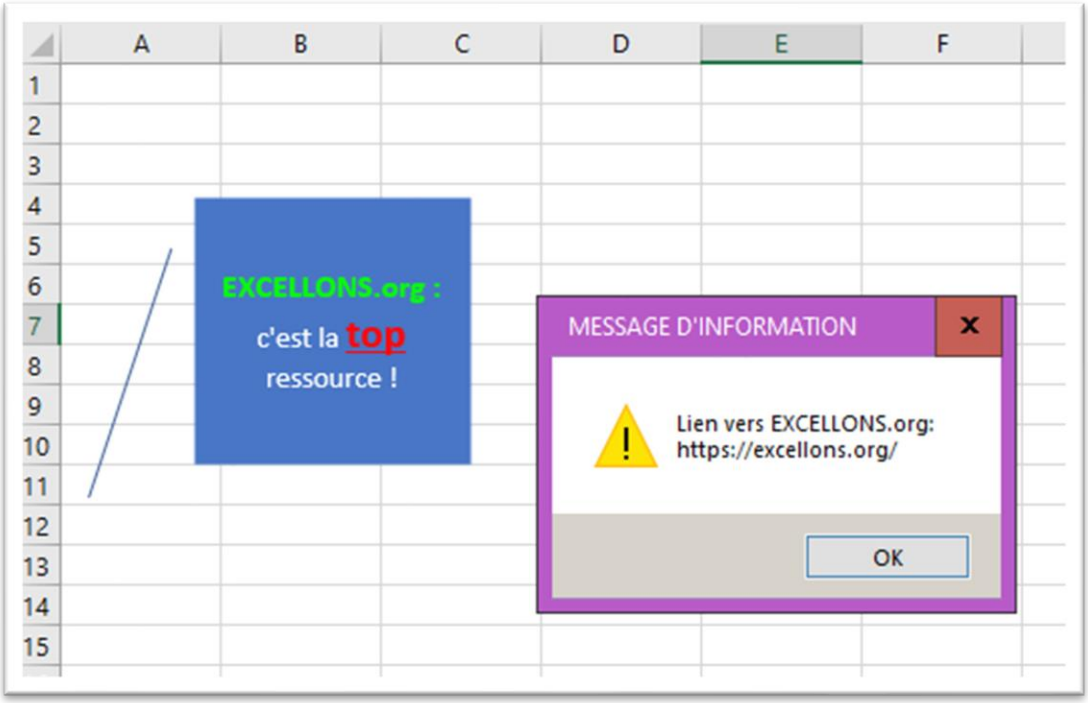
Soit la macro (simplissime) suivante :

```
Sub displayInfo()
   MsgBox "Lien vers EXCELLONS.org: " & vbCrLf _
       & "https://excellons.org/" _
       , vbExclamation, "MESSAGE D'INFORMATION"
End Sub
```

Affectons cette macro à la forme **« Légende »** créée précédemment avec le fragment de code :

```
Sub assignMacro()
   Dim oSheet As Worksheet
   Dim oShape As Shape

   Set oSheet = ThisWorkbook.Worksheets(1)
   Set oShape = oSheet.Shapes("Legende")

   oShape.OnAction = "displayInfo"
End Sub
```

Résultat obtenu lors d'un clic sur « Légende » :



RELIONS DES FORMES AVEC UN CONNECTEUR

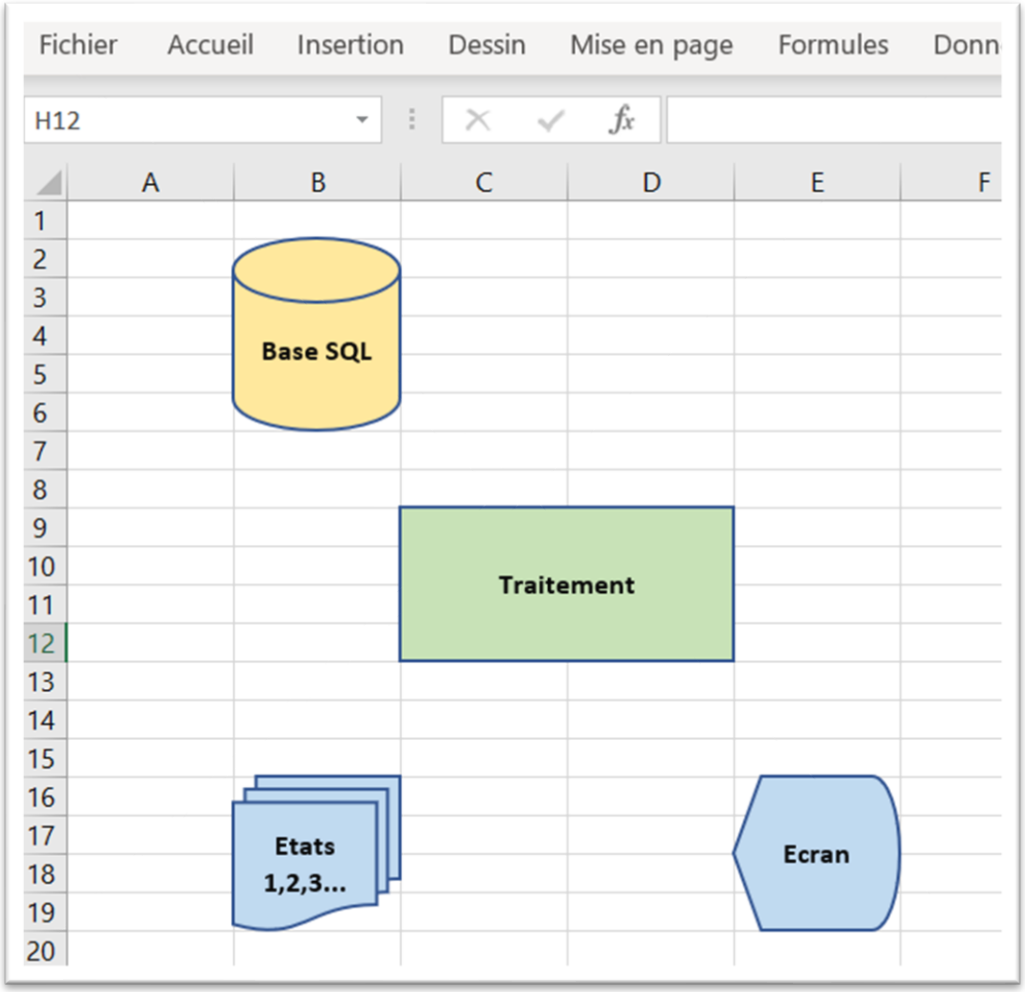
Voyons comment nous pouvons **relier deux formes (shapes)** entre-elles au moyen d'une forme **« Connecteur ».**

Commençons par **tracer un ordinogramme** comportant **4 formes** au moyen du code suivant :

Affectons cette macro à la forme **« Légende »** créée précédemment avec le fragment de code :

```
Sub Add4Shapes()
   Dim oSheet As Worksheet
   Dim oShape As Shape
   Dim oRange As Range

   Set oSheet = ThisWorkbook.Worksheets(3)

   'Insérons la forme 'Database'
   InsertNewShape oSheet, msoShapeFlowchartMagneticDisk, "BaseSQL", "B2:B6",
msoThemeColorAccent4, "Base SQL"

   'Insérons la forme 'Traitement'
   InsertNewShape oSheet, msoShapeFlowchartProcess, "Traitement", "C9:D12",
msoThemeColorAccent6, "Traitement"

   'Insérons la forme 'Etats'
   InsertNewShape oSheet, msoShapeFlowchartMultidocument, "Etats", "B16:B19",
msoThemeColorAccent5, "Etats 1,2,3..."

   'Insérons la forme 'Ecran'
   InsertNewShape oSheet, msoShapeFlowchartDisplay, "Display", "E16:E19", msoThemeColorAccent5,
"Ecran"
End Sub
```

Résultat obtenu :



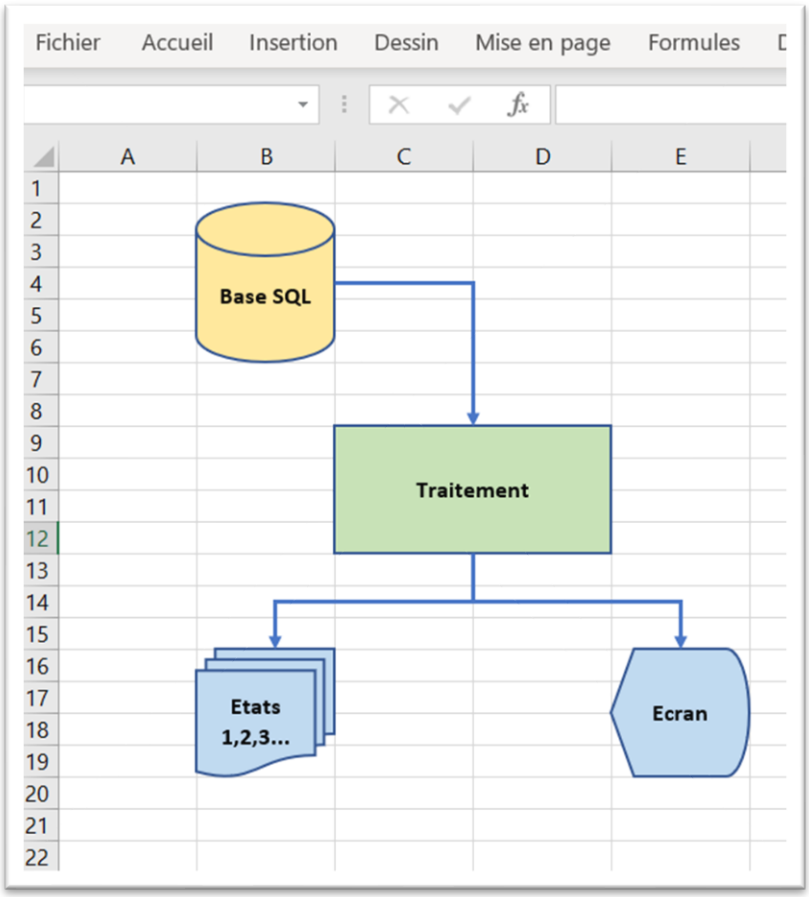
Pour relier 2 formes entres elles, **créons une forme 'Connecteur'** au moyen de la méthode **« AddConnector »** (voir plus haut) puis utilisons les méthodes de l'objet **'ConnectorFormat'** :

BeginConnect et **EndConnect**

Par exemple, au moyen du code suivant, **nous relions les formes** de notre ordinogramme deux à deux :

```
Sub AddConnectors()
   Dim oSheet As Worksheet
   Dim oShape1 As Shape, oShape2 As Shape
   Dim oRange As Range

   Set oSheet = ThisWorkbook.Worksheets(1)

   'Connectons la forme 'Database' à la forme "Traitement"
   Set oShape1 = oSheet.Shapes("BaseSQL")
   Set oShape2 = oSheet.Shapes("Traitement")
   InsertConnector oSheet, oShape1, oShape2, 5, 1

   'Connectons la forme 'Traitement' à la forme "Etats"
   Set oShape1 = oSheet.Shapes("Traitement")
   Set oShape2 = oSheet.Shapes("Etats")
   InsertConnector oSheet, oShape1, oShape2, 3, 1

   'Connectons la forme 'Traitement' à la forme "Display"
   Set oShape1 = oSheet.Shapes("Traitement")
   Set oShape2 = oSheet.Shapes("Display")
   InsertConnector oSheet, oShape1, oShape2, 3, 1

End Sub
Sub InsertConnector(zSheet As Worksheet, zShape1 As Shape, zShape2 As Shape, zPoint1 As Long,
zPoint2 As Long)
   Dim oConnector As Shape
   Dim oRange As Range

   'Insérons le connecteur
   Set oConnector = zSheet.Shapes.AddConnector(msoConnectorElbow, 120, 50, 200, 100)


   With oConnector
      'Formatons le connecteur
      .Line.Weight = 1.75
      .Line.ForeColor.ObjectThemeColor = msoThemeColorAccent1
      .Line.EndArrowheadStyle = msoArrowheadTriangle
      'Relions les 2 formes avec le connecteur
      .ConnectorFormat.BeginConnect zShape1, zPoint1
      .ConnectorFormat.EndConnect zShape2, zPoint2
      '.RerouteConnections
   End With
End Sub
```

Résultat obtenu :



Les méthodes **BeginConnect** et **EndConnect** réclament le paramètre obligatoire **'SiteConnection'** indiquant le **point d'ancrage** du connecteur sur la forme à relier.

Ce point d'ancrage est à choisir parmi les **points d'ancrage existants pour la forme à relier**. Chaque type de forme possède **sa propre collection** de points d'ancrage.

Du fait qu'EXCEL **n'expose pas** la collection des points d'ancrage d'une forme donnée, il n'est pas aisé de déterminer le point nous convenant pour le tracé désiré.

Si, comme dans notre exemple, nous désirons positionner les connecteurs à **des points bien spécifiques** des formes, le plus simple consiste à utiliser **l'enregistreur de macro** afin de déterminer les numéros de ces points.

Une autre option consiste à indiquer le point d'ancrage '1' et d'utiliser la méthode **'RerouteConnections'** de l'objet 'Shape' qui a pour effet **de prendre le chemin le plus court** entre les deux formes reliées. Par exemple :

```
Set oConnector = zSheet.Shapes.AddConnector(msoConnectorElbow, 120, 50, 200, 100)
With oConnector
   'Relions les 2 formes avec le connecteur
   .ConnectorFormat.BeginConnect zShape1, 1
   .ConnectorFormat.EndConnect zShape2, 1
   .RerouteConnections
End With
```