

E.S.L.S.C.A.

Thème

La gestion des Interfaces 1

- MsgBox et Inputbox

- Les objets par leurs propriétés

ActiveX

*Démarre Excel, Cliquer sur Développeur, ensuite sur Visual Basic.
L'ensemble des instructions de ce TD sont à mettre dans une feuille spécifique
de type « module ». Pour ce faire, Appuyer sur Insertion, Module.*

1. Les messages

Lors d'une procédure, les messages servent à communiquer avec l'utilisateur.
Il existe des messages qui donnent de l'information et d'autres qui en demandent.

**ATTN : L'ensemble des parties du code donnés ci-dessous est à mettre dans
la bonne syntaxe entre **Sub ()** et **end sub()**.**

Les MsgBox

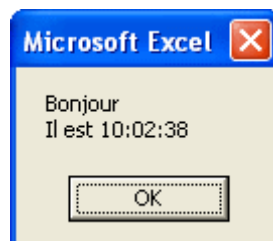
Les MsgBox peuvent simplement donner une information. La procédure est alors stoppée tant
que l'utilisateur n'a pas cliqué sur le bouton.

```
Sub affiche1 ()  
    MsgBox "Bonjour"  
End Sub
```






Le texte peut-être affiché sur plusieurs lignes en utilisant le symbole &. Pour sauter de ligne,
on peut utiliser la fonction chr(10) ou chr(13).

```
Sub Affiche2 ()  
MsgBox "Bonjour" & Chr(10) & "Il est " & Time  
End Sub
```



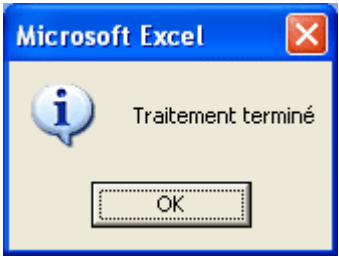
Vous pouvez ajouter une icône concernant le type de message à afficher.
Les types d'attribut icône :

Constante :	Icône :	
vbCritical		Pour une erreur fatale
vbExclamation		Pour une remarque
vbInformation		Pour une information

vbQuestion		Pour une question
------------	---	-------------------

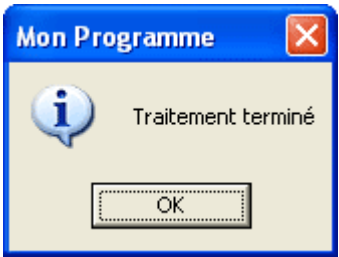
La syntaxe pour ajouter une icône est MsgBox "Message", attribut icône :

```
MsgBox "Traitement terminé", vbInformation
```




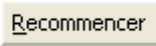





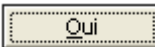
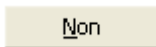
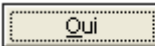
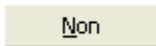

Le titre de la fenêtre (Microsoft Excel) peut être changé. La syntaxe est : MsgBox "Message", attribut icône, "Titre de la fenêtre" :

```
MsgBox "Traitement terminé", vbInformation, "Mon Programme"
```



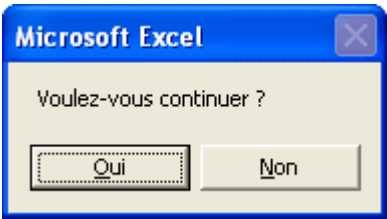
Les MsgBox peuvent également demander une information à l'utilisateur. Dans ce cas, la boîte de message comprend plusieurs boutons

Les types d'attribut Boutons :

Constante :	Boutons :		
vbAbortRetryIgnore			
vbOKCancel			
vbRetryCancel			
vbYesNo			
vbYesNoCancel			

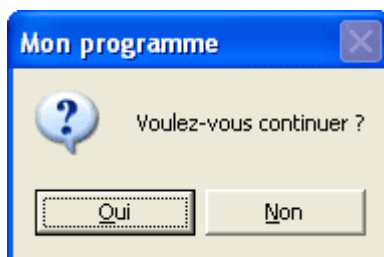
La syntaxe est MsgBox ("Message", attribut bouton):

```
MsgBox "Voulez-vous continuer ?", vbYesNo
```



Vous pouvez également y ajouter les icônes et personnaliser le titre de la fenêtre en utilisant la syntaxe : MsgBox ("Message", attribut bouton + attribut icône, "titre de la fenêtre").

MsgBox "Voulez-vous continuer ?", vbYesNo + vbQuestion, "Mon programme"



MsgBox renvoie une valeur différente pour chaque bouton.

Constante :	Valeur :
vbOK	1
vbCancel	2
vbAbort	3
vbRetry	4
vbIgnore	5
vbYes	6
vbNo	7

Ainsi, si l'utilisateur clique sur le bouton "OK", MsgBox renvoie la valeur 1, sur le bouton "Annuler" la valeur 2, sur le bouton "Ignorer" la valeur 5 ...

Cette valeur est récupérée dans une variable. Celle-ci doit être déclarée après Sub et est précédée du mot clé : Dim.

Exemple : Dim *util* as integer. Ici, le nom de la variable est « util », il doit être choisi de manière compréhensible et facilement reconnaissable. *Integer* est le type de la variable, cela veut dire que cette variable est amenée à mémoriser des entiers. L'ensemble des types possibles dans VBA sera vu ultérieurement.

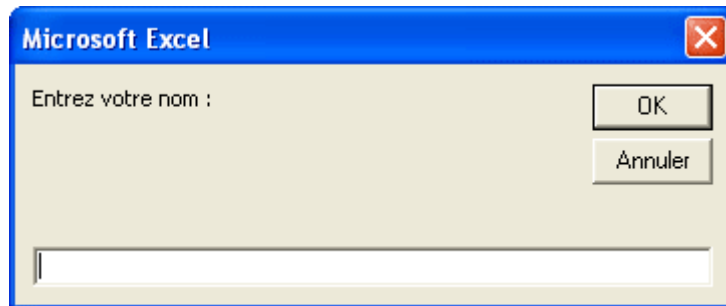
Tout commentaire est précédé du symbole '. VBA considère que la suite est un commentaire et non une instruction VBA. Il est recommandé de mettre beaucoup de commentaires dans vos programmes VBA.

```
Sub Affiche3 ()
'Dans la ligne d'instruction suivante, si l'utilisateur
'clique sur le bouton "Oui", Reponse prendra comme valeur
'6 sinon Reponse prendra comme valeur 7.
Reponse = MsgBox ("Voulez-vous continuer ?", vbYesNo)
'La ligne suivante arrête la procédure si l'utilisateur
'clique sur "Non"
If Reponse = 7 Then Exit Sub
Msgbox "Nous continuons"
End Sub
```

Les InputBox

Les InputBox sont des boîtes de dialogue dans lesquelles l'utilisateur est invité à entrer des données. La syntaxe est : `InputBox ("Message")`.

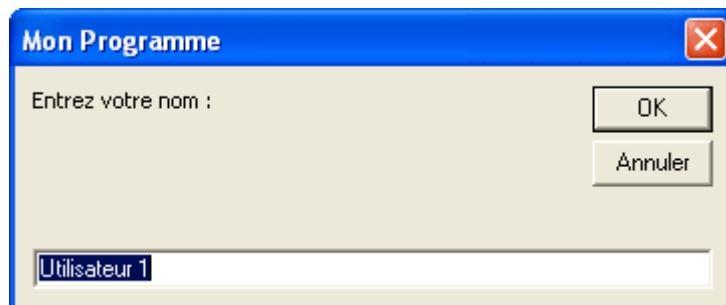
```
InputBox ("Entrez votre nom :")
```



Comme pour les MsgBox, vous pouvez changer le titre de la fenêtre. Vous pouvez également entrer une valeur par défaut dans la zone de saisie. La syntaxe devient : `InputBox ("Message", "Titre de la fenêtre", "Valeur par défaut")`.

La valeur saisie peut être récupérée dans une variable. Si l'utilisateur clique sur le bouton "Annuler", la variable renvoie une chaîne de longueur nulle ("").

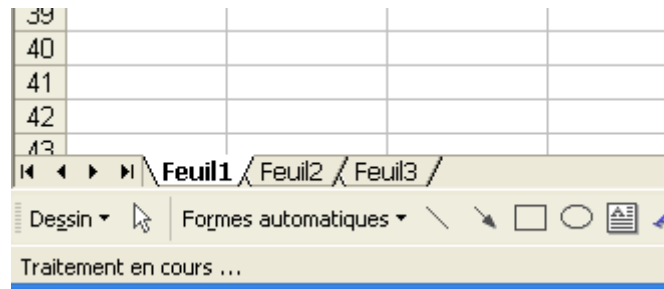
```
Message = InputBox("Entrez votre nom :", "Mon Programme", "Utilisateur 1")
```



```
Message = InputBox("Entrez votre nom :", "Mon Programme", _  
"Utilisateur 1")  
'La ligne suivante arrête la procédure si l'utilisateur  
'clique sur "Annuler"  
If Message = "" Then Exit Sub  
'La ligne suivante place la valeur saisie dans la cellule  
'A1 de la feuille active  
Range("A1").Value = Message
```

Vous pouvez également écrire un message dans la barre d'état de l'application. La syntaxe est:
`Application.StatusBar = "Message"`

```
Application.StatusBar = "Traitement en cours ..."
```



A la fin de la procédure, pensez à supprimer le message de la barre d'état par la ligne d'instruction: `Application.StatusBar = False`.

Exercices :

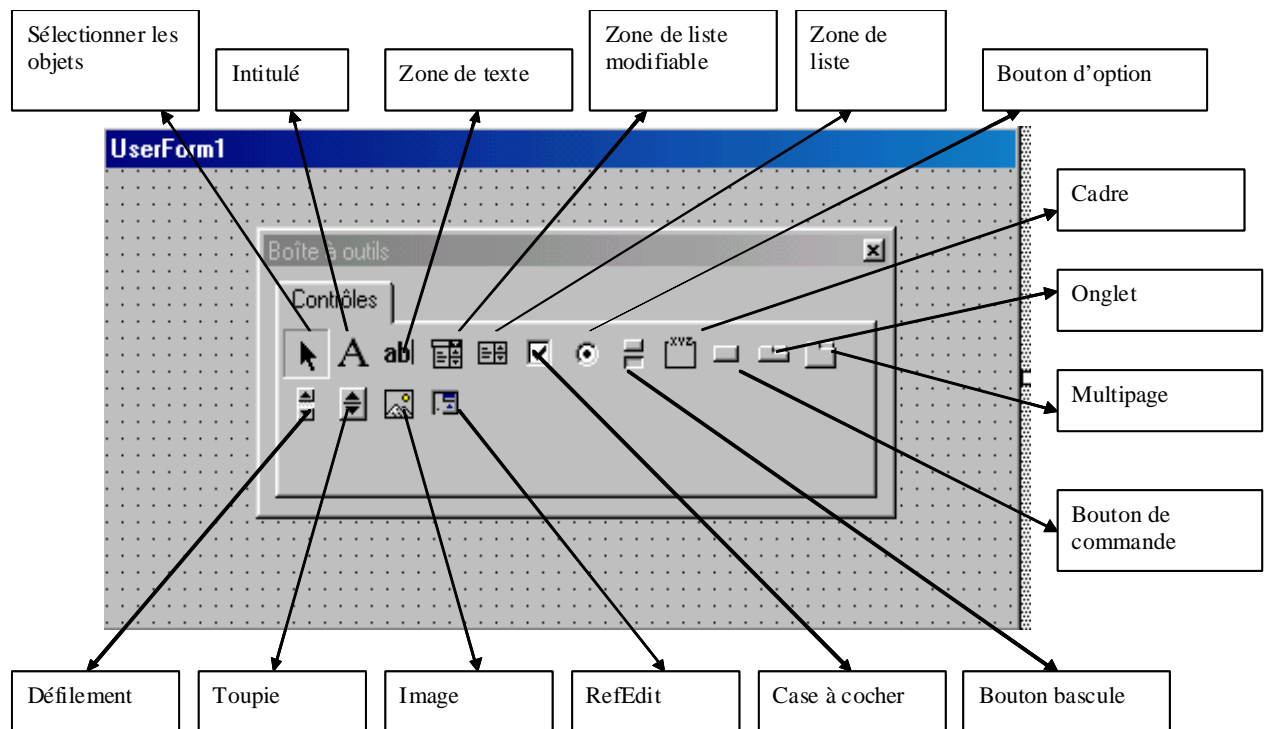
1. Ecrire une procédure VBA qui saisit une le nom et l'âge d'une personne.

1.1. Elle doit ensuite afficher, en fonction de l'âge rentré, le nom de la personne et un message en conséquence (ex. « bébé », « enfant », « adolescent », « jeune », « mûr », « moins jeune », etc.) et doit proposer en même temps à l'utilisateur s'il souhaite afficher les données rentrées ou non.

1.2. Si la réponse est non, elle affiche un message « good bye » et se termine. Sinon, elle affiche dans un message les valeurs rentrées (nom, âge, message).

2. Les objets et les propriétés ActiveX

La boîte à outils contient un certain nombre d'éléments prédéfinis dont nous allons vous présenter ici les plus courants. Exception faite de l'outil **Sélectionner les objets**, qui correspond à celui de la barre d'outils *Dessin*, tous ces outils s'utilisent de la même façon.



- **Intitulé (Label) :** les intitulés peuvent être utilisés à n'importe quel endroit de la boîte de dialogue en guise d'invites, de commentaires, d'incitation, etc. Ils n'ont pas d'autre fonction dans le cadre de l'utilisation ultérieure de la boîte de dialogue, mais ils peuvent être modifiés par le programme. Vous pouvez aussi les utiliser pour afficher du texte qui ne peut pas être modifié.
- **Zone de texte (Textbox) :** pour la saisie des contenus (textes, nombres, références ou formules), utilisez des zones de modification. Elles peuvent avoir n'importe quelle taille et, par conséquent, également contenir plusieurs lignes de texte.
- **Zone de liste modifiable (Combobox) :** zone de liste combinée avec une zone de texte dans laquelle l'entrée de liste sélectionnée s'inscrit et peut être éditée.
- **Zone de liste (Listbox) :** c'est une liste munie d'une barre de défilement et dans laquelle vous pourrez par la suite sélectionner une entrée.
- **Case à cocher (CheckBox) :** les cases à cocher, contrairement aux cases d'option, ne s'excluent pas mutuellement. Vous pouvez aussi les réunir dans des zones de groupe pour montrer quelles sont celles qui appartiennent à une même catégorie ou qui se rapportent à un même domaine.

- **Bouton d'option (OptionButton) :** Ils se rapportent à une alternative donnée, doivent toujours être réunis dans une zone de groupe. Il s'agit d'options qui s'excluent mutuellement ; une seule peut être sélectionnée à la fois.
- **Bouton bascule :** bouton permettant de basculer entre deux états.
- **Cadre (Frame):** si vous avez besoin de cases d'options, autrement dit de plusieurs boutons ronds dont un seul peut être sélectionné à la fois, vous devez les regrouper à l'intérieur d'un cadre. Vous pouvez également rassembler des cases à cocher et d'autres éléments dans une telle zone. Cela n'a cependant aucune répercussion sur leur fonction dans ce cas.
- **Bouton de commande (CommandButton) :** Avec cet outil, vous pouvez créer autant de boutons que vous voulez. Il lance, termine ou interrompt une action ou une série d'actions
- **Défilement (ScrollBar):** lorsqu'il est actionné dans la boîte de dialogue, ce contrôle retourne une valeur correspondant approximativement à la position du curseur dans la barre (par défaut, ces valeurs sont comprises entre 0 et 100). Il peut être combiné avec d'autres éléments qui réagissent à la valeur retournée.
- **Toupie (SpinButton):** cet élément fonctionne comme une barre de défilement à cette différence près que vous ne disposez ici que des deux flèches et pas de curseur.
- **Onglet :** contrôle permettant d'insérer des onglets dans un formulaire.
- **Multipage :** permet d'insérer des onglets dans un formulaire, ces onglets pouvant à leur tour contenir des contrôles.
- **Image :** permet d'insérer une image ou un dessin dans le formulaire
- **RefEdit :** cet outil permet de sélectionner des références de plages de cellules dans un formulaire.

Ex : Vous pouvez créer un contrôle '**CommandButton**' qui ouvre une autre feuille.

Ex : Vous pouvez aussi afficher un texte, une image ou les deux sur un contrôle '**CommandButton**'.

Les contrôles sont d'abord désignés par des noms et des numéros par défaut

La propriété par défaut d'un contrôle CommandButton est la propriété Value.

L'événement par défaut d'un contrôle CommandButton est l'événement Click.

Propriétés des contrôles ActiveX :

Name : Définir le nom du contrôle.

Min : Spécifie la valeur (*Value*) minimale acceptable pour la propriété d'un contrôle.

Max : Spécifie la valeur (*Value*) maximale acceptable pour la propriété d'un contrôle.

LinkedCell : Spécifie à quelle **cellule** de la feuille le contrôle est-il **lié**

SmallChange : Spécifie le déplacement (valeur du pas) se produisant lorsque l'utilisateur clique sur une flèche de défilement dans un contrôle SpinButton.

LargeChange : Spécifie la distance de déplacement (valeur du pas) intervenant lorsque l'utilisateur clique entre le curseur de défilement et la flèche de défilement.

Style : Pour un contrôle ComboBox, spécifie de quelle façon l'utilisateur va utiliser le contrôle.

ListFillRange : Indiquer le nom de la base où se trouve les informations de la zone de liste modifiable.

ColumnCount : Cette propriété indique à la zone de liste modifiable sur le nombre de colonnes de la base d'infos.

BoundColumn : Cette propriété indique à la zone de liste quelle valeur de colonne mettre dans la cellule lié.

Caption : Texte descriptif apparaissant sur un objet pour l'identifier ou le décrire.

NB : A ne pas Confondre avec Name : Nom par défaut donné aux objets par VB à leur création.

Font : Définit les caractéristiques (Gras, Italique, Police, Taille) du texte utilisé

SpécialEffect : Spécifie l'aspect à l'écran d'un objet.

RowSource : Permet de relié une zone de liste modifiable à une plage de cellules d'une feuille.

Syntaxe : Nom de feuille!cellules à lier

ControlSource : Permet de relier une liste à la cellule correspondante de la feuille cible.

Syntaxe : Nom de feuille!cellule à lier

Exercice : « Une calculatrice pour calculer un remboursement d'un prêt de voiture »

Le but de cet exercice est de construire une calculatrice qui vous permettra de prévoir le montant des remboursements d'un prêt de voiture. Votre outil vous permettra de réaliser plusieurs simulations, en modifiant le montant de l'emprunt, les taux d'intérêt, le nombre de mensualités et la marque de la voiture que vous désirez acquérir, tout en éliminant tout risque de valeurs erronées.

Cet énoncé vous guidera dans votre travail :

Notre but est de créer une interface utilisateur graphique qui aura pour avantage de limiter les choix à des valeurs plausibles, ce qui réduit ou élimine les erreurs de l'utilisateur et rend le modèle facile à utiliser. De plus, nous allons pouvoir protéger notre feuille de calcul en cas de valeurs saisies erronées.

NB :

La barre d'outils Commandes contient toutes les sortes de contrôles ActiveX que vous pouvez mettre sur une feuille de calcul : zones de liste, toupies, zones de liste modifiables, etc.

Saisissez les libellés suivants dans une feuille de calcul que vous nommerez Prêt d'un classeur que vous appellerez PrêtVoiture:

	A	B	C
1			
2		Prix	
3		Apport	
4		Prêt	
5		Intérêts	
6		Années	
7		Montant	

1/ Limiter les années à des valeurs plausibles :

Pour l'achat d'une voiture, la durée maximale d'un emprunt est de 5ans. En prenant un peu de marge, on autorisera, pour le nombre possible des années, les valeurs comprises entre 1 et 6. Un contrôle Toupie est une manière efficace de spécifier de telles valeurs entières.

- 1- Activez la boîte à outils contrôles,
- 2- Cliquez sur le bouton Toupie dans cette boîte à outils,
- 3- Gardez la touche Alt enfoncée (pour aligner le contrôle sur la grille) et cliquez sur Fermer dans l'angle supérieur gauche de la cellule E6.
- 4- Relâchez Alt et placez le bouton Toupie horizontalement
- 5- Cliquez sur le bouton « Propriétés »
- 6- Saisissez 1 comme valeur Min et 6 comme valeur Max,
- 7- Pour la propriété LinkedCell (Cellule liée), saisissez C6 et appuyez sur entrée.
- 8- Cliquez sur le bouton « Désactiver le mode de création » et essayer de cliquer sur le bouton Toupie.

Décrivez ce que vous remarquez.

.....
.....
.....
.....

2/ Limiter le montant de l'apport à des valeurs plausibles :

Une plage raisonnable de l'apport serait des valeurs comprises entre 0% et 100%, en incrémentant par tranche de 5%. Le bouton Toupie ne spécifiant que des valeurs entières, vous pouvez procéder comme suit :

- 1- Cliquez sur le bouton « Mode création »,
- 2- Maintenez les touches Alt + Ctrl enfoncées et déplacez le bouton Toupie de la cellule E6 à la cellule E3. relâchez le bouton,
- 3- Dans la fenêtre Propriétés, saisissez 100 comme valeur Max ,0 comme valeur Min et saisissez H3 comme valeur de LinkedCell,
- 4- Entrez 5 comme valeur de SmallChange et appuyez sur Entrée,
- 5- Cliquez sur le bouton « Désactiver le mode de création »,
- 6- Tapez $=H3/100$ dans la cellule C3 et essayer de cliquer sur le bouton Toupie.

Décrivez ce que vous remarquez.

.....
.....
.....
.....

3/ Limiter le taux d'intérêt à des valeurs plausibles :

Il est préférable d'autoriser l'incrément des taux d'intérêt par petits pas de 0.25% dans une plage comprise entre 0% et à peu près 20%. On utilisera cette fois-ci un contrôle de barre de défilement au lieu d'un bouton Toupie.

- 1- Cliquez sur le bouton « Défilement » dans la barre d'outils Contrôles,
- 2- Placez la barre de défilement dans la cellule E5,
- 3- Dans la fenêtre propriétés, saisissez 2000 comme valeur Max, 25 comme valeur de SmallChange, 100 comme valeur de LargeChange et H5 comme valeur de LinkedCell et appuyez sur entrée.
- 4- Cliquez sur le bouton « Désactiver le mode de création »,
- 5- Tapez $=H5/10000$ dans la cellule C5 et essayer de cliquer sur les flèches des bornes finales et entre le curseur et une extrémité.

Décrivez ce que vous remarquez.

.....
.....
.....
.....

4/ Retrouver une valeur à partir d'une liste :

Le prix d'une voiture étant fonction de la voiture que vous voulez acheter. Vous avez la possibilité de sélectionner la description de la voiture et faire apparaître automatiquement le prix dans la cellule Prix. Ceci grâce à l'utilisation d'une zone de liste.

a/ Préparer une liste de voitures :

Commencez par saisir la liste de voitures suivante :

	J	K	L
1			
2		Renault 24	21750 €
3		Peugeot 308	18000 €
4		Rover 602	19500 €
5		Peugeot 407	20250 €
6		BMW 317	21300 €
7		Citroen KX	19050 €
8		Opel Vectrum	18300 €
9			

- 1- Sélectionnez la cellule K2 et appuyez sur Ctrl+* pour sélectionner tout le bloc de cellules,
- 2- Nommez cette plage ListeVoitures (dans le menu Insertion>>Nom>>Définir),
- 3- dans la barre d'outils contrôles, cliquez sur la zone de liste modifiable placez ce contrôle dans la cellule E2,
- 4- Dans la fenêtre propriétés, sélectionnez 2- fmStyleDropDownList pour la propriété style, saisissez C2 comme valeur de LinkedCell et appuyez sur entrée,
- 5- Saisissez ListeVoiture comme liste de la propriété ListFillRange et appuyez sur entrée
- 6- Cliquez sur le bouton « Désactiver le mode de création »,
- 7- Cliquez sur la flèche de la zone de liste modifiable, et sélectionnez une marque de voiture donnée. Que remarquez-vous ?

.....

b/ Retrouver le prix à partir de la liste :

- 1- Cliquez sur le bouton « Mode Création » puis sur la zone de liste modifiable,
- 2- Dans la fenêtre Propriétés, saisissez 2 comme valeur de la propriété ColumnCount, (cette propriété informe la zone de liste modifiable qu'il y a bien deux colonnes de valeurs dans la ListFillRange),
- 3- Saisissez 2 comme valeur de la propriété BoundColumn et appuyez sur entrée (la propriété BoundColumn indique à la zone de liste quelle valeur de colonne mettre dans la cellule liée),
- 4- Quittez le mode création et cliquez sur la flèche de la zone de liste,
- 5- Sélectionnez une voiture donnée, que remarquez-vous ?

.....

c/ Définir les largeurs de colonnes :

Jusqu'à maintenant, quelque soit la place disponible pour le prix, la zone de liste modifiable utilise la même largeur pour chaque colonne. Vous pouvez contrôler manuellement les largeurs de colonne.

- 1- Activez le Mode Création et sélectionnez la zone de liste modifiable,
- 2- Saisissez 2.2 cm ; 2 cm comme valeur de la propriété ColumnWidths (pour la première et la seconde colonne),
- 3- Remarquez que la valeur affichée de la propriété se change en 62.35 pt ; 56.7 pt (C'est l'équivalent en points (1 cm=28.35 pt)), vous pouvez donc saisir des valeurs en cm ou en pt Excel les convertira en pt,
- 4- Quittez le mode Création et cliquez sur la zone de liste modifiable

Décrivez ce que vous remarquez.

.....
.....
.....
.....

NB :

Une zone de liste modifiable a deux styles possibles : zone de liste déroulante (c'est le type qu'on a utilisé ici, Propriété : Style 2-...), vous permettant de ne sélectionner que les rubriques à partir de la liste, ou être une zone de liste combinée à une boîte d'édition (Propriété : Style 1-...), vous permettant d'entrer de nouvelles valeurs aussi bien que de les sélectionner à partir de la liste. Vous pouvez explorer cette possibilité en sélectionnant le Style 1-... dans Propriétés.

5/ Protection de la feuille de calcul :

Faire tourner le modèle sur un exemple. Pour ce faire, il vous est demandé de rentrer la bonne formule pour calculer le montant du remboursement.

6/ Protection de la feuille de calcul :

Dans ce modèle, nous avons créé une calculatrice de prêt qui va nous permettre de saisir directement les valeurs dans les cellules ceci sans utiliser des macros. Cependant, rien n'empêchera la destruction d'une formule (en rentrant « Chat » dans la cellule C2 comme prix de voiture on détruira de ce fait notre formule).

Vous pouvez protéger la feuille de calcul. Ceci évitera la saisie de valeurs non plausibles dans le modèle et empêcherait aussi les contrôles ActiveX de changer les valeurs des cellules liées. Vous pouvez toutefois positionner la protection de feuille de calcul de tel manière que Visual Basic puisse toujours changer les cellules.

Vous avez besoin des étapes qui suivent pour générer une procédure de gestion d'événement afin de protéger effectivement le modèle.

a/ Créer un gestionnaire d'évènement pour tous les boutons de contrôle :

La première étape est de convertir les contrôles ActiveX à partir de la liaison aux cellules, ceci en utilisant un gestionnaire d'évènement pour permettre une nouvelle valeur dans une cellule.

Activez le mode Création et sélectionnez la zone de liste modifiable, Dans propriétés changez (Name) en cboPrix (cbo veut dire zone de liste modifiable). Effacez la propriété LinkedCell et cliquez sur le bouton « Visualiser le code » dans la barre d'outils contrôles,

Une nouvelle procédure de gestion d'évènement nommée cboPrix_Change apparaît (c'est l'évènement par défaut pour une zone de liste modifiable).

Dans le corps de la macro, saisissez :

```
Private Sub cboPrix_Change()  
    Range("C2").Value = cboPrix.Value  
End Sub
```

Cette procédure changera la cellule C2 pour qu'elle corresponde à la nouvelle valeur de la zone de liste modifiable, chaque fois que cette valeur change.

Activez Excel, quittez le mode Création et essayez la zone de liste modifiable. Que remarquez-vous ?

.....
.....
.....

Répétez les étapes 1 à 5 pour le bouton Toupie qui positionne le pourcentage de l'apport. Nommez-le spnApport, effacez la propriété LinkedCell et dans sa procédure d'évènement, saisissez :

```
Private Sub spnApport_Change()  
    Range("C3").Value = spnApport.Value / 100  
End Sub
```

Répétez les étapes 1 à 5 pour la barre de défilement. Nommez-la scrTaux, effacez la propriété LinkedCell et dans sa procédure d'évènement, saisissez :

```
Private Sub scrTaux_Change()  
    Range("C5").Value = scrTaux.Value / 10000  
End Sub
```

Répétez les étapes 1 à 5 pour le bouton Toupie qui positionne le nombre d'années. Nommez-le spnAnnées, effacez la propriété LinkedCell et dans sa procédure d'évènement, saisissez :

```
Private Sub spnAnnées_Change()  
    Range("C6").Value = spnAnnées.Value  
End Sub
```

Effacez les cellules H3 et H5, vous n'avez plus besoin des valeurs s'y trouvant.

Pour chaque contrôle, vous avez ainsi créé une procédure de gestion d'événement et aucun contrôle n'est lié à une cellule. Vous êtes maintenant prêts à protéger votre feuille.

b/ Protéger la feuille de calcul :

Vous pouvez directement protéger notre feuille de calcul en cliquant sur la commande Protection du menu Outils, puis en cliquant sur la commande Protéger la feuille et dans ce cas, ni l'utilisateur, ni les contrôles ActiveX, ni les macros, ne peuvent changer la valeur des cellules verrouillées.

Une Macro, peut protéger une feuille de calcul de manière à pouvoir modifier les cellules verrouillées. Or, cette protection spéciale cesse d'être efficace lorsque vous fermez et rouvrez le classeur.

Vous allez dans cette partie générer un événement qui s'exécutera chaque fois que vous ouvrez un classeur :

- 1- Activez Visual Basic, cliquez sur le bouton «Explorateur de projet » et double-cliquez sur l'objet ThisWorkbook.
- 2- A partir de la liste Objet (au-dessus de la fenêtre code), sélectionnez Workbook.
- 3- Insérez cette instruction dans le corps de procédure Workbook_open :

```
Private Sub Workbook_Open()  
    Sheets("Prêt").Protect UserInterfaceOnly:=True  
End Sub
```

L'argument UserInterfaceOnly de la méthode Protect permet à une macro de faire des modifications même si un utilisateur ou un contrôle ne le peut pas.

- 4- Sauvegardez et fermez le classeur PrêtVoiture, puis rouvrez-le
- 5- Essayez de saisir des nombres dans le modèle. Que remarquez-vous
- 6- Essayez de changer le modèle en utilisant les contrôles ActiveX.

Décrivez ce que vous remarquez

.....
.....
.....
.....

Exercice :

Appliquer les interfaces vues dans cet exercice sur un problème de gestion d'un tableau de cours boursiers.