

PROGRAMMATION : VISUAL BASIC 6.0

Les Procédures et fonctions:	2
• Structure du programme :	2
• Fonctions particulières:	2
Les Variables :	3
• Structure de tableau:	3
• Chaînes de caractères:	3
Structures Conditionnelles:	4
• If <condition> Then <instruction1> [Else <instruction2>]End If.....	4
• Select Case <Expression> Case ListeValeurs1 [Instructions] Case Else [Instructions] End Select.....	4
• Iif (Condition, ValeurSiVrai, ValeurSiFaux).....	4
• Boucles en nombre défini:.....	4
• Boucles Tant que:.....	5
• Boucles Jusqu'à:	5
• Boucles Pour chacun des:.....	6
Principaux objets et propriétés essentielles	7
• Message Box, Input Box	7
• Options de la fenêtre :	7
• Form:frm (feuille :le conteneur graphique des contrôles de l'application.....	8
• CommandButton: cmd (bouton de commande)	8
• Label: lbl (étiquette)	8
• Image: img (image)	9
• TextBox: txt (zone de texte).....	9
• Options disponibles :	9
• Exemple de code: textbox, bouton et liste.....	10

Les Procédures et fonctions:

- Structure du programme :

La partie supérieure est la zone des **déclarations** (séparée du reste par un trait horizontal). On y place les options et les déclarations de variables publiques.

Option Explicit

Public NomJoueur As String

La partie suivante est la zone des **procédures publiques** (chaque procédure est séparée des autres par un trait horizontal). On y place les procédures publiques utilisables par toutes les procédures de la feuille (événementielles ou pas).

Public Sub SaisieTempérature()

Température = InputBox("Taper une température")

Somme = Somme + Température

End Sub

Les fonctions retournent la valeur de la variable du même nom: Par exemple Carré(7) retournera la valeur 49.

Public Function Carré(x) As Single

Carré = x * x

End Function

Enfin la zone des **procédures événementielles** (chaque procédure est séparée des autres par un trait horizontal).

Sub NomContrôle_Evénement() ' ou aussi Private Sub

< Instructions >

End Sub

- Fonctions particulières:

La fonction **Date** donne la date système: "08/04/98"

La fonction **Time** donne l'heure système: "11:10:00"

La fonction **Day()** donne le numéro du jour dans le mois: 8

La fonction **Month()** donne le numéro du mois dans l'année: 4

La fonction **Year()** donne le numéro de l'année: 1998

La fonction **WeekDay()** donne le numéro du jour dans la semaine sachant que le dimanche porte le numéro 1. Ici **WeekDay(Date)** retournera la valeur 4

Les Variables :

Dim <NomVariable> As <Type>

Exemples :

Dim/Private Taux As Single 'var locales mourant à la fin de la fonction dans la quelle on se situe

Dim/Private Taux As Double ' 300 chiffres significatifs au lieu de 39 (décimaux)

Public/Global Réponse As Integer 'var globales déclarées au début d'un programme et valables jusqu'à fin de la feuille, sauf cas des tableaux (toujours déclarés par dim)

Public/Global Réponse As Double ' de 2 Milliards à l'opposé, au lieu de 32 000

Dim saisipas As Variant ' Prendra son type suivant l'affectation faite (nombre ou string)

- Structure de tableau:

Dim TabTemp(12) As Single ' le premier indice est le n°1

Dim TabMajuscules(65 to 90) As String ' indices de 65 à 90

- Chaines de caractères:

Dim Mot, Lettre As String * 10 ' Ici la variable ne contient qu'une lettre, par défaut des espaces

Len(Phrase) ' retourne la longueur totale (ici 10 même si les dernieres n'ont pas été renseignées)

StrComp(Phrase1,Phrase2) ' compare les deux variables (booléen)

InStr(Chaîne1, Chaîne2) ' recherche la chaîne 2 dans la chaîne 1, retourne la position de la première lettre de chaîne 2 dans chaîne 1 ou 0 si elle ne s'y trouve pas

Ucase(chaine 1) met toute la chaîne 1 en majuscule ' ces fonctions permettent des recherches plus efficaces
en minuscules

Left/Right (chaîne 1,5) ' donne les 5 lettres de gauche/droite de chaîne 1

Mid(chaîne, 4, 2) ' donne les 2 lettres à partir de la 4ème (inclue) de chaîne: la 4 et la 5ème donc

Si l'utilisateur fournit une donnée il faut la stocker dans une **variable** pour pouvoir la réutiliser autant de fois qu'on le veut. Le plus simple est d'utiliser la boîte de dialogue prédéfinie **InputBox** qui est aussi une fonction et qui retourne une valeur de type **Variant**. Cette fonction a pour effet d'affecter une valeur à une variable dûment déclarée.

Structures Conditionnelles:

- *If <condition> Then <instruction1> [Else <instruction2>] End If*

Exemple :

```
If Moyenne >= 10 Then Décision = "Admis" Else Décision = "Refusé"  
End If
```

- *Select Case <Expression> Case ListeValeurs1 [Instructions] Case Else [Instructions] End Select*

Exemple :

```
Select Case CodeASCIICaractère  
Case 65, 69, 73, 79, 85  
MsgBox(" C'est une voyelle ")  
Case 66 To 90  
MsgBox (" C'est une consonne ")  
Case Else  
MsgBox (" Ce n'est pas une lettre ")  
End Select
```

- *IIf (Condition, ValeurSiVrai, ValeurSiFaux)*

Exemple :

```
Note = InputBox (" Tapez votre note ")  
Réponse = IIf (Note >= 10, " Admis ", " Ajourné ")  
MsgBox (Réponse)
```

- *Boucles en nombre défini:*

```
For Compteur = Début To Fin [Step Incrément]  
Instructions  
[ ... Exit For]  
[Instructions]  
Next [Compteur]
```

Exemple :

For Compteur = 1 To 12

 TabTemp(Compteur)=InputBox("Température N° " & Compteur)

Next Compteur

Exemple :

For Compteur = 1 To 6

 TabTirageLoto (Compteur)=(Rnd * 48 + 1)/1

Next Compteur

- **Boucles Tant que:**

Do While *Condition*

Instructions

[... Exit Do]

[*Instructions*]

Loop ' Ici la condition est testée au début

Do

Instructions

[... Exit Do]

[*Instructions*]

Loop While *Condition* ' Ici la condition est testée à la fin de la boucle

- **Boucles Jusqu'à:**

Do Until *Condition*

Instructions

[... Exit Do]

[*Instructions*]

Loop 'Condition initiale

Do

Instructions

[... Exit Do]

[*Instructions*]

Loop Until *Condition* 'Condition finale

- **Boucles Pour chacun des:**

For Each *Elément* **In** *Ensemble*

Instructions

[... **Exit For**]

[*Instructions*]

Next [*Elément*]

Ensemble est le plus souvent un tableau.

Exemples:

Dim TabHasard(100) **As** Integer

Dim Cellule **As** Integer

Dim Réponse **As** String

Randomize

For Each Cellule **In** TabHasard Cellule = Rnd * 100 + 1

Next

For Each Cellule **In** TabHasard

Réponse = Réponse & Cellule & " "

Next

MsgBox (Réponse)

Principaux objets et propriétés essentielles

APPEL DE FONCTION : NOM_COMMANDE . EVENEMENT [(parametres, si y'a pas on met pas les parenthèses)]

DECLARATION : NOM_COMMANDE . EVENEMENT ()

- Message Box, Input Box

Valeur = `InputBox("Entrez votre donnée ?")` ' Valeur est déjà déclaré (Integer)

`MsgBox("Hello World, la variable est " &var &"texte en plus")`

C'est un simple message affiché dans une boîte agrémentée d'un bouton **OK**.



Syntaxe : `MsgBox` (message , icones et boutons , titre)

`vbCrLf` : retourer a la ligne (ds le message entre valeurs et textes

- Options de la fenêtre :

`vbOKCancel` Affiche les boutons OK et Annuler.

`vbAbortRetryIgnore` Affiche le bouton Abandonner, Réessayer et Ignorer.

`vbYesNoCancel` Affiche les boutons Oui, Non et Annuler.

`vbYesNo` Affiche les boutons Oui et Non.

`vbRetryCancel` Affiche les boutons Réessayer et Annuler.

`vbDefaultButton1` Le premier bouton est le bouton par défaut.

`vbDefaultButton2` Le deuxième bouton est le bouton par défaut.

`vbDefaultButton3` Le troisième bouton est le bouton par défaut.

`vbDefaultButton4` Le quatrième bouton est le bouton par défaut.

`vbApplicationModal` Boîte de dialogue modale. L'utilisateur doit répondre au message affiché dans la zone de message avant de pouvoir continuer de travailler dans l'application en cours.

`vbSystemModal` 4096 Modal système. Toutes les applications sont interrompues jusqu'à ce que l'utilisateur réponde au message affiché dans la zone de message.

`vbOK` ou 1 OK

`vbCancel` ou 2 Annuler

`vbAbort` ou 3 Abandonner

`vbRetry` ou 4 Réessayer

`vbIgnore` ou 5 Ignorer

`vbYes` ou 6 Oui

`vbNo` ou 7 Non

`vbCritical` Affiche l'icône Message critique.

vbQuestion Affiche l'icône Requête d'avertissement.
 vbExclamation Affiche l'icône Message d'avertissement.
 vbInformation Affiche l'icône Message d'information.
 variable = MsgBox ("Oui ou Non ?", vbInformation + vbYesNo, "Choix")
 question = InputBox("Quel est votre nom ?", "Question")
 MsgBox (message , icones et boutons , titre)
 vbCritical Affiche l'icône Message critique.
 vbQuestion Affiche l'icône Requête d'avertissement.
 vbExclamation Affiche l'icône Message d'avertissement.
 vbInformation Affiche l'icône Message d'information.
 variable = MsgBox ("Oui ou Non ?", vbInformation + vbYesNo, "Choix")
 question = InputBox("Quel est votre nom ?", "Question")

Pour ajouter les boutons il faut introduire après la 1er virgule séparant le message et le 2e paramètre ces options

Et si l'on s'applique ça donne :



- **Form:frm (feuille :le conteneur graphique des contrôles de l'application.**

Propriétés : Caption , Name , Font , ForeColor , Icon , Style , Picture , WindowState , BackColor

Unload Me ou Un load formulaire

- **CommandButton: cmd (bouton de commande)**

Utilisation : exécute le code associé à l'événement **click** sur ce bouton.

Propriétés : Caption , Font , ForeColor , BackColor , Name

- **Label: lbl (étiquette)**

Utilisation : affiche une **sortie écran** (texte, nombre, date...) non interactive.

Propriétés Caption , Font , ForeColor , Style , Picture , Name , BackColor



- [Image: img \(image\)](#)

Utilisation : affiche des images en **mode point** (BitMap au format .BMP, .WMF, .ICO) Peut servir de bouton de commande (événement click).

Propriétés : Picture , Name , BackColor , ForeColor

- [TextBox: txt \(zone de texte\)](#)

Utilisation : pour **taper** ou **afficher** du texte.

Propriétés Text , Name , Font , BackColor , ForeColor

Les propriétés sont fixées au départ (valeurs standard).

Modifiées par programmation objet dans la fenêtre propriétés ou par code :

nom_du_contrôle.propriété = valeur

- [Options disponibles :](#)

- Name : Nom du contrôle
- Caption : Contenu du contrôle (Titre d'une feuille , Texte d'un bouton ..etc)
- Font : Police du texte contenu du contrôle
- ForeColor : Couleur du texte contenu du contrôle
- Icon : Icône du contrôle
- Picture : Définit une image à afficher dans le contrôle (Le mode graphique doit être activé)
- Style : Active le mode graphique du contrôle (0 - Mode texte 1 - Graphique)
- BackColor : Couleur de fond
- Text : Contenu d'une zone de texte
- WindowState : Définit si la fenêtre et agrandi , réduite ou normal

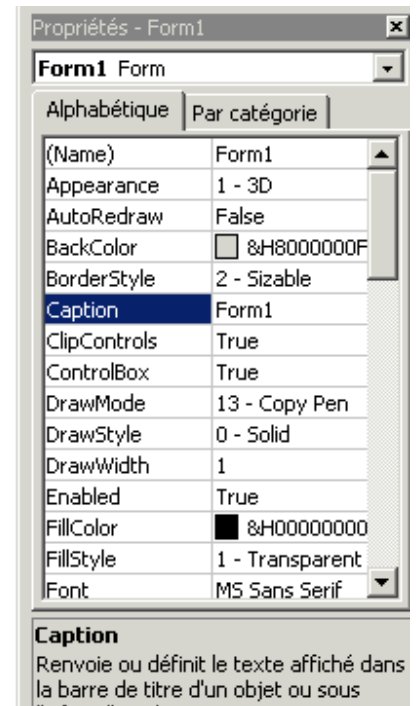
- c'est pareil pour les cases à cocher (avec un toto As Intégrer comme paramètre d'entrée) :

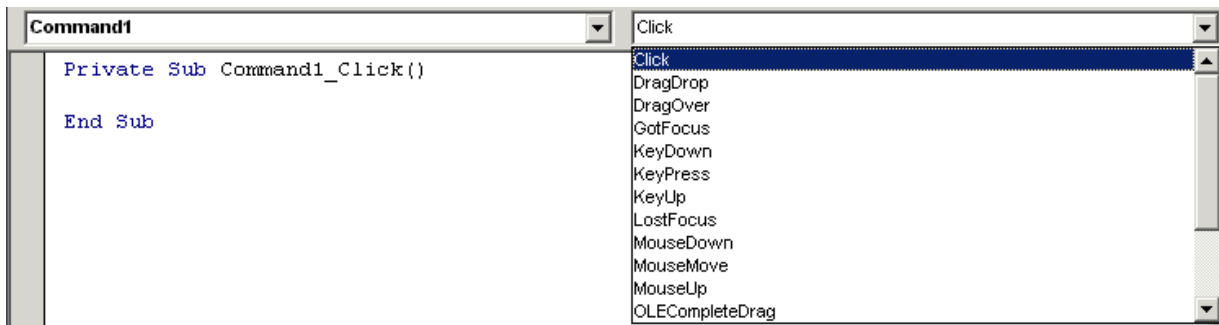
If ckcCouleur.Value = 1 **Then** infos = infos & vbCrLf & "Couleur : Oui" **Else** ...

- les boutons d'option (ou exclusif, un seul du bloc ne peut être pris à la fois) :

If optFrançais.Value = True **Then** infos = infos & vbCrLf & vbCrLf & "Français"

- pour introduire une procédure relative à un contrôle on double clique sur celui-ci, et on choisit l'évènement :





- [Exemple de code: textbox, bouton et liste](#)



```
Private Sub amdAffvalue_Click()
```

```
MsgBox "La valeur de cette objet est " & lstList.Text
```

```
End Sub
```

```
Private Sub cmdAdd_Click(Index As Integer)
```

```
On Error GoTo 1
```

```
lstList.AddItem txtText.Text
```

```
1 'Rien
```

```
End Sub
```

```
Private Sub cmdAffnbr_Click(Index As Integer)
```

```
MsgBox "Il y a " & lstList.ListCount & " objets dans la liste"
```

```
End Sub
```

```
Private Sub cmdAffnum_Click()
```

```
MsgBox "Vous avez sélectionner l'objet numéro " & lstList.ListIndex + 1
```

```
End Sub
```

```
Private Sub cmdAll_Click()
```

```
MsgBox "L'Objet " & lstList.Text & vbCrLf & vbCrLf & "est l'objet " & _
```

```
lstList.ListIndex & " de la liste" & vbCrLf & vbCrLf & "Il y a " & _
```

```
lstList.ListCount & " objets dans cette liste" ' les _ st là pour faire du code de plusieurs lignes
```

```
End Sub
```

```
Private Sub cmdDel_Click() ' cmdDel: nom de la commande
```

```
On Error GoTo 1 ' Si erreur aller à l'étiquette 1
```

```
lstList.RemoveItem lstList.ListIndex ' Eleve l'enregistrement de lstListe qui est pointé par ListIndex
```

```
Exit Sub ' termine si on est pas dans le cas 1
```

```
1 MsgBox "Vous devez sélectionner un objet" ' Etiquette 1
```

```
End Sub
```