

Ecole	**JAH INFORMATIQUE PLUS**
Filière	**Technicien En Gestion Informatisée 1er Année**
Cours	**Programmation VB**

Microsoft Visual Basic 2008 Professionnelle

Chapitre II : VB.Net (V2008)

Variables :

a- Définition :

Une variable est une case de mémoire identifier par un nom. Ce nom doit commencer au moins par une lettre alphabétique. Le langage VB n'est pas sensible à la casse. (Min, Maj.).

b- Déclaration des variables :

Dim nom variable as type de données

Exemple :

Dim somme as float (‘réel simple)
Dim produit as Doble (‘réel long)
Dim I as Integer (‘entier)
Dim exist as Boolean (‘valeur logique) true or false

c- Types des variables :

On distingue deux types principaux des variables :

- Variables locales:

Ce sont des variables qui sont déclaré dans des procédures ou dans des fonctions.

Une variable locale ne peut être utilisé que dans la procédure ou dans la fonction créatrice de cette variable.

La case mémoire réservée pour une variable locale sera libéré dés de le micro processeur termine l'exécution de la procédure ou de la fonction créatrice de cette variable.

- Variables globales:

Ce sont des variables qui sont déclaré dans le début de la class ou dehors de la procédure et des fonction.

Une variable globale peut être utilisé dans toutes les procédures et les fonctions de la forme créatrice ou la class créatrice de cette variable. [La portée des variables].

Une case mémoire réservée pour une variable globale sera libérée dés qu'ou ferme la forme créatrice a cette variable [durée de vie d'une variable].

d- Modes d'accès a une variable globale :

On peut utiliser deux modes principaux d'accès à une variable :

- Variable déclarer privée : elle ne peut être utiliser que dans la forme créatrice a cette variable.

  Exemple :

  ```
  Dim S as float
  Private S as float
  ```

- Variable déclaré public : elles peuvent être utiliser a partir de d'autre forme qui ne sont pas créatrice a cette variable.

  Exemple :

  ```
  Public S as float
  ```

-Pour afficher l'heure = Time of day
-Pour afficher la date actuelle => date.today ou date.now

❖ Structure de contrôle : Condition

Syntaxe :

✓ Forme 1 :

```
If      condition(s) then
        Bloc des instructions
EndIf
```

✓ Forme 2 :

```
If      condition     then
        Bloc 1
Else
        Bloc 2
EndIf
```

✓ Forme 3 :

```
If      Condition 1 then
        Bloc 1
ElseIf  Condition    2        then
        Bloc 2
ElseIf  Condition 3 then
        Bloc 3
Else
        Bloc 4
EndIf
```

✓ Forme 4 :

```
If    Condition          then
      Bloc 1
Else
If    Condition 2 then
      Bloc 2
EndIf
EndIf
```

❖ Contrôle de type radio-buton et checkBox possèdent une propriété .checked qui vérifie si le contrôle est caché ou non.

- Checked = true => caché
- Checked = false => non caché

✓ Forme 5 :

```
If    Condition          then
      Bloc 1
Else
If    Condition          then
      Bloc 2
Else
If    Condition          then
      Bloc 3
Else
      Condition          bloc 4
EndIf
EndIf
EndIf
```

➢ Les opérateurs :

❖ Opérateurs de comparaison :

> : Supérieur strictement
< : Inférieur
>= : Supérieur ou égale
<= : Inférieur ou égale
= : Egale à
<> : Différent de

❖ Opérateurs logiques (Connecteurs) :

And : « et » logique (somme <= 0 and produit = 10)
Or : « ou » logique (somme >= or produit >=10)
Not : non logique Not (somme >=0)

➢ Les fonctions de conversion :

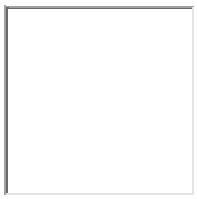
Val (texte) convertie en double
Cdbl (texte) convertie en double aux parties décimales

Cint (texte) convertie en entier
Csng (texte) convertie en single
Cdate (texte) convertie en date

- Les contrôles : ComboBox et ListBox
  - ComboBox : c'est une liste déroulante

Ce contrôle vous permet de crée : 
-une liste modifiable
-une liste non modifiable
Cette liste ne permet pas le multi choix

- ListBox : c'est une liste de choix non modifiable qui permet le multi choix



Les deux contrôles possèdent plusieurs propriétés communes :

⇒ .Texte : présente le texte de l'élément sélectionné.
⇒ .Selected Index : l'indice de l'élément sélectionné (le numéro)
⇒ .Item (i) : l'élément numéro i à base 0.

Les deux contrôles possèdent aussi des méthodes communes :

⇒ .addItem (texte) => ajout d'une nouvelle élément à la liste.
⇒ .RemouveItem (texte) => supprime un élément s'il existe.
⇒ .RemouveItemat (i) => supprime un élément par son indice « i ».
⇒