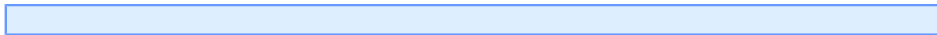


Tutoriel 26 : Les éléments de programmation "Les Variables"

par Gilbert Miralles (gilmir.developpez.com)

Date de publication : Lundi 18 mars 2003

Dernière mise à jour : Lundi 4 février 2008



- I - Introduction : Les variables en VB6
 - II - Définition des variables
 - III - Les types fondamentaux de données
 - IV - Etendue d'une variable
 - V - Déclarations de variables
 - VI - Déclarer des variables sans DIM
 - VII - Modifier la valeur d'une variable
 - VIII - Travailler avec des types de données spécifiques
 - IX - Type de données personnalisées
- Prochain tutoriel :

I - Introduction : Les variables en VB6

Comme dans n'importe quel langage de programmation, la notion de variable est bien établie.

Nous trouvons deux grands types de variables dans Visual basic 6.

Les variables classiques et les variables objet.

Les premières permettent de manipuler toutes sortes de données, comme par exemple, les valeurs numériques, les valeurs logiques, les dates, le texte, etc...

Le deuxième type est un peu plus complexe car il permet de pointer sur les objets du projet(les feuilles) ainsi que sur des applications externes à Visual Basic comme Excel ou Word.

Une variable objet permet ainsi de créer une *instance* d'une application afin de travailler directement à l'intérieur de celle-ci.

II - Définition des variables

Les variables peuvent s'apparenter à des cases dans lesquelles on stocke des informations.

Ces cases sont situées dans la mémoire de l'ordinateur.

Elles peuvent contenir une grande variété de données.

Pour stocker une donnée dans une variable, nous lui assignerons une valeur en utilisant l'opérateur "égale" (=).

Chaque variable reçoit un nom qui lui est propre et qui peut comporter jusqu'à 254 caractères.

Le nom d'une variable doit commencer par une lettre et il ne doit être identique à aucun mot réservé du Visual basic.

III - Les types fondamentaux de données

Les entiers(Integer), flottants (Real), chiffres monétaires (Currency) et les chaînes de caractères (String) sont gérés dans des variables de type bien définis.

Chaque type peut couvrir une échelle de valeurs qui lui est propre.

Les débordements des limites de valeur sont la cause d'erreurs pendant l'exécution du programme.

La déclaration d'une variable, qui définit son type, est effectuée à l'aide d'un suffixe(le% désigne le type entier par exemple) ou de l'instruction Dim (exemple : Dim Nombre As Integer) Le suffixe permet de reconnaître le type d'une variable du premier coup d'œil.

Vous pouvez lire dans le tableau décrit plus bas, une description des types fondamentaux de données de Visual basic.

Depuis la version 6 de Visual Basic, le programme vous oblige à déclarer les variables par la commande "Option Explicite"

IV - Etendue d'une variable

L'étendue, ou durée de vie, d'une variable détermine le type de commandes pouvant accéder à cette variable. Une variable déclarée à l'intérieur d'une procédure a une étendue locale : la variable est créée et détruite chaque fois que la procédure est exécutée. Il n'existe aucun moyen d'y accéder en dehors de la procédure.

Une variable déclarée en dehors d'une procédure a une étendue globale : sa valeur est accessible et modifiable par n'importe quelle commande.

Lorsque vous déclarez des variables, une variable locale et une variable globale peuvent avoir le même nom. La modification de la valeur de l'une des variables ne modifiera pas la valeur de l'autre. Néanmoins, si vous ne déclarez pas de variables, vous risquez de modifier par inadvertance la valeur d'une variable globale.

Pour éviter tout problème, prenez l'habitude de déclarer explicitement toutes les variables.

V - Déclarations de variables

Pour déclarer explicitement une variable avant de l'utiliser, (en principe au début d'une procédure d'évènement), il suffit d'indiquer le nom de la variable derrière l'instruction "Dim" (Dim est l'abréviation de Dimension).

Cette action réserve en mémoire l'espace nécessaire pour stocker la variable et indique à Visual basic le type de variable déclarée.

Si vous ne précisez pas le type de la variable, Visual Basic la déclare par défaut comme étant de type "Variant".

Après avoir déclaré la variable, vous pouvez faire appel à celle-ci à partir de votre code, par exemple, l'instruction suivante : `Nom = "Developpez"` affecte "Developpez" à la variable "Nom" `Label1.Caption = Nom` affichera "Developpez" dans l'étiquette (Label1) du programme.

VI - Déclarer des variables sans DIM

Il est également possible de déclarer une variable sans le mot "Dim".

Cette technique est nommée "déclaration implicite".

Pour déclarer une variable sous cette forme, il suffit simplement d'employer son nom, par exemple :

```
Nom = "Oscar"
```

Néanmoins nous utiliserons l'option explicite pour la réalisation de toutes les applications développées dans cette rubrique.

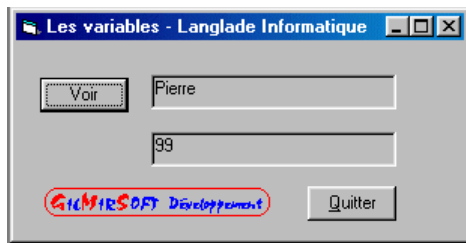
VII - Modifier la valeur d'une variable

Les instructions sont organisés en trois groupes. La première instruction déclare la variable Nom à l'aide de Dim. Aucun type n'étant spécifié, la variable prend le type "Variant" et peut contenir du texte ou des nombres.

Les secondes et troisième lignes affectent le nom de "Pierre" à la variable "Nom", puis affichent le contenu de la variable dans l'étiquette "Label" de la feuille.

Il s'agit là d'un des emplois les plus courants des variables : le transfert d'informations vers une propriété d'objets.

Exemple :



Modifier la valeur d'une variable

Veuillez saisir les instructions suivantes
pour déclarer et utiliser la variable "Nom:"

```
Dim Nom  
Nom = "Pierre"  
Label1.Caption = Nom  
Nom = 99  
Label2.Caption = Nom
```

Cliquez sur le bouton "Voir"

VIII - Travailler avec des types de données spécifiques

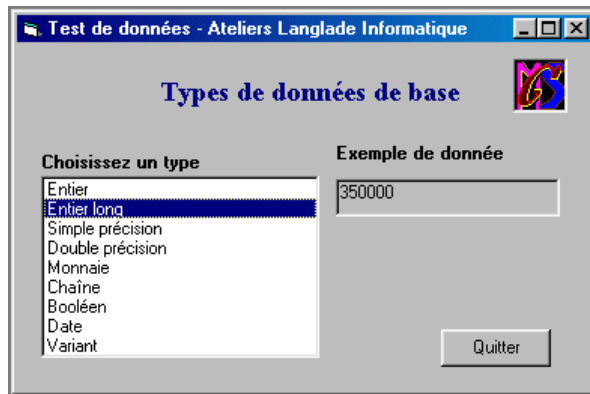
Type de données	Taille	Plage de valeurs	Exemples
Entier (Integer)	2 octets	-32768 à +32767	Dim Oiseaux% Calendrier% = 12
Entier long	4 octets	-2.147.483.648 à + 2.147.483.647	Dim Charge& Charge& = 35000
Simple précision virgule flottants	4 octets	-3.042.823.E38 à 3.402823E38	Dim Prix! Prix! = 895.50
Double précision virgule flottants	8 octets	-1.7976931348623 à 1.7976931348623	Dim Pi # 2D3308 Pi#=3.1415926535 2D3308
Monnaie et (Currency)	8 octets		Dim Dette@ Dette@ = 760010010.50
Chaîne de caractères (String)	1 octet par caractère	0 à 65 535 caractères	Dim Chien\$ Chien\$ = "Berger allemand"
Booléen (boolean)	2 octets	True (Vrai) ou False (Faux)	Dim Drapeau As Boolean Drapeau = True
Date	8 octets	1 janvier 100 à 31 décembre 9999	Dim anniversaire As Date Anniversaire = #3-1-56#
Variante	16 octets pour les types et 22 octets + 1 octet par caractère pour les chaînes de caractères	Les plages de tous les autres types	Dim Prix Prix = 289.135

De ce tableau, il ressort qu'il vaut mieux (si possible) utiliser un type de données autre que celui de variante.

Nous avons concocté un petit programme pour argumenter le tableau précédent.

Télécharger l'exemple ici!

Cet exemple est livré avec les sources à titre pédagogique, vous pourrez voir le code de la procédure "List" ainsi que le code de l'instruction "Select Case"



le programme permet d'expérimenter 9 types de données, dont le type "Entier", Virgule flottante en simple précision et Date.

Le programme affiche un exemple de chaque type lorsque vous cliquez sur son nom dans la liste.

IX - Type de données personnalisées

Visual Basic permet également de créer vos propres types de données.

Cette fonction est particulièrement utile si vous avez à traiter un groupe de données formant un ensemble incluant différentes catégories de types de données.

La déclaration suivante est un exemple de création d'un type de données personnalisé.

```
Type Entreprise
    Nom As String
    DateDinscription As Date
End Type
```

Vous faites appel à ce type de données par l'instruction suivante :

```
Dim ChefEntreprise As Entreprise
ChefEntrepriseName = "Jacques Dupont"
```

Prochain tutoriel :

 ***Gérer des contrôles comme des variables objets***

