

# Tutoriel 4 : Les tableaux

par Gilbert Miralles ([gilmir.developpez.com](http://gilmir.developpez.com))

Date de publication : Lundi 18 mars 2003

Dernière mise à jour : Lundi 28 janvier 2008



**I - Les tableaux**

I.1 - Définition

I.2 - Comment allons nous déclarer ce tableau ?

**II - Les affectations ou traitements conditionnels****III - Les compteurs de boucles**

III-1 - Qu'est ce qu'une boucle ?

III-2 - Etudions la boucle "For ... Next"

**IV - Les principaux opérateurs arithmétiques et de comparaison**

IV.1 - Les opérateurs arithmétiques

IV.2 - Les opérateurs de comparaison

Prochain tutoriel :

## I - Les tableaux

### I.1 - Définition

Un tableau, c'est dans le cas d'une réalisation graphique, la déclaration de l'espace écran sous forme d'emplacement graphique.


Je m'explique, si nous prenons un écran de configuration graphique de 800x600, cela veut dire que l'écran comprend 800 pixels sur une ligne, et 600 pixels sur une colonne. Un pixel représentant un point sur l'écran.

Ex : Un pixel = ceci [ . ]

Chaque partie de l'écran étant représentée, si nous voulons animer un graphique, nous devons déclarer dans le programme, c'est à dire à Visual Basic, la configuration d'écran que nous allons utiliser.

Dans l'une des applications que j'ai déjà développées ("Patman"), j'ai décidé que le tableau comprendrait 456 cases, comment j'ai fait pour trouver ce chiffre ?

J'ai dessiné plusieurs éléments graphiques devant représenter l'interface de travail, pour l'exemple, je vous reproduis

un de ces éléments , (le voilà), il mesure 23x23 pixels. Pour construire l'interface graphique dans l'espace écran dont je disposais, j'ai calculé que je pouvais mettre dans la longueur de l'écran 19 éléments, et dans la largeur de l'écran 24 éléments. Le calcul est simple 19 que multiplie 24 égale 456.

Ces chiffres peuvent être modifiés en fonction de la configuration d'écran et des dimensions des éléments que vous avez dessinés.

### I.2 - Comment allons nous déclarer ce tableau ?

**Principe :**

**Dim** <nom du tableau><liste de paramètres> **As** <type du tableau>

Généralement nous déclarons par exemple :

```
Dim Mon_tableau (1 To 12) As Integer
```

Dans le cas de l'exemple précité, nous déclarerons notre tableau avec les instructions suivantes :

- Global Const NB\_CASES = 456
- Global Const NB\_LIGNES = 19
- Global Const NB\_COLONNES = 24

Si nous nous référons aux explications précédentes, nous en concluons que nous venons de déclarer un tableau qui comprend 19 lignes, 24 colonnes, 456 cases, qui sont des constantes donc qui ont des valeurs fixes et qui sont

déclarées globalement dans tous le programme, donc qui sera pris en compte par toutes les feuilles de travail s'il y en a plusieurs.

Nous déclarerons ces informations dans un module que nous nommerons par exemple : "Global.bas"

## II - Les affectations ou traitements conditionnels

Nous terminerons ces exercices par les traitements conditionnels :

**Etudions le " IF Then Else " :**


<nom\_de\_la\_variable>=<fonction><nom\_de\_la\_variable><valeur><opération>

**Explication :**

Si (**If**) inclure une condition Alors (**Then**) donner instructions pour appliquer la condition si la condition peut être exécutée, **Else** (nouvelle condition) donner de nouvelles instructions pour exécuter une nouvelle condition si la première n'est pas exécutée. **End If** Fin du traitement conditionnel.

If <condition> Then <instruction> Else <instruction>

```
If tableau(ligne, colonne) = Patman Then
  'Traduction = si le tableau représenté par les lignes et les colonnes est égal à patman(objet)
  alors, Lire les Instructions suivantes</li>
  ligne Patman = Patman 'la ligne du patman devient le patman (objet)
  colonne Patman = Patman 'la colonne du patman devient le patman (objet)<a
  name="contents"></li>
Else
  'Instructions (si la première condition n'est pas remplie)
  'Donner de nouvelles instructions</li>
End if
```

 **Prenez note que lorsque la condition n'est pas remplie, le pointeur de Visual Basic ignore les instructions et saute à l'instruction suivante.**

Nous pouvons utiliser également le terme **Elseif** (ou bien) qui vous permet d'utiliser une autre instruction si celle utilisée précédemment n'est pas exécutée!

Vous constatez que devant le mot *Instruction*, j'ai mis une apostrophe ' ; en effet Visual Basic considère que toute écriture précédée d'une apostrophe n'est pas pris en compte par le compilateur.

Donc toutes les lignes de commentaires sont précédées d'une apostrophe et servent au développeur à expliquer la forme et l'idée de son texte.

Je vous conseille de mettre le plus de commentaires possibles.

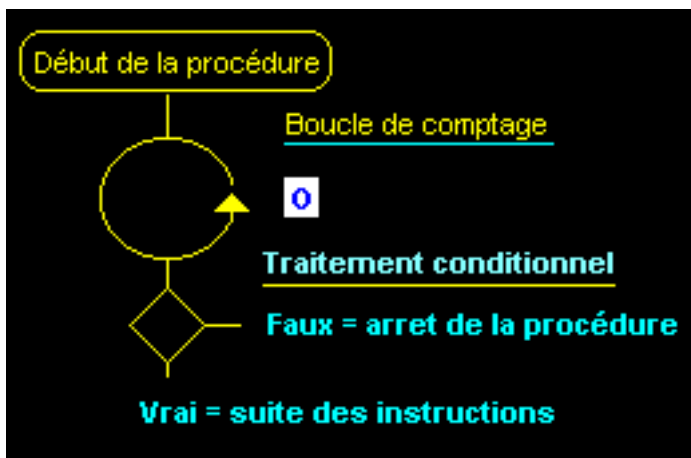
### III - Les compteurs de boucles

#### III-1 - Qu'est ce qu'une boucle ?

On peut comparer une boucle à un point qui circule dans un cercle et qui tourne sans fin.

On peut lui adjoindre un compteur qui incrémentera (ajouter1) le nombre de tours chaque fois que celui-ci passera devant la case départ.

On peut également le faire sortir de la boucle au bout d'un certain nombres de tours.



Démarrage du point

exécuter un tour

Contrôle du nombre de tours

Si la valeur compteur est égale ou supérieure à  
alors le point sort de la boucle

Si la valeur compteur est inférieure à 3

Ajouter 1 à la valeur du compteur et retourner à  
la case départ

#### III-2 - Etudions la boucle "For ... Next"

**For** <compteur> = <debut> **To** <fin> **Step** <pas>

<bloc d'instructions>

**Exit For**

<bloc d'instructions>

**Next** <compteur>

Exemple d'utilisation : On veut comptabiliser les mois de l'année dans une boucle, on écrira :

```

For Mois = 12 To 1 Step -1
  'Instructions
  Total = Total + Montant(Mois)
Next Mois
    
```

Dans cet exemple on décrémente (Comptage négatif) de 12 à 1 la variable **Step** indique le pas de décrémentation, et dans ce cas nous souhaitons décrémente de 1 (mois) si nous avons souhaité effectuer des calculs basés sur un trimestre nous aurions mis comme "valeur de pas" le chiffre 3 ( Step - 3)

La ligne instruction exécute le programme demandé et le **Next** est la variable de fin de procédure.

*Si vous souhaitez comprendre au mieux l'opération de comptage avec les boucles, cliquez sur le lien qui vous ouvre le fichier **[EXERCICE1]***

## IV - Les principaux opérateurs arithmétiques et de comparaison

### IV.1 - Les opérateurs arithmétiques

Opérateur	Signification	Exemple
+	Ajoute deux nombres	$a + b$
-	Soustrait deux nombres ou rend négatif un nombre	$a - b$ ou $-92$
*	Multiplie deux nombres	$a * b$
/	Divise deux nombres	$a / b$

### IV.2 - Les opérateurs de comparaison

Opérateur	Signification	Exemple
<	Inférieur	$a < b$ (a plus petit que b)
<=	Inférieur ou égal	$a <= b$
>	Supérieur	$a > b$
>=	Supérieur ou égal	$a >= b$
=	Egal	$a = b$
<>	Différent	$a <> b$
&	Retourne une chaîne de caractère qui est la concaténation des deux autres	Bonjour & messieurs, retourne "Bonjour messieurs"

D'autres traitements conditionnels seront étudiés en cours de programmation

Prochain tutoriel :

 **Notion d'algorithme**

