

Annexe – Cours Visual Basic Manipulation des Types de base

I) Introduction

On s'intéressera ici principalement à la manipulation des types de base principaux, à savoir : les Strings, les nombres (Integer, Double...) et les tableaux.

II) Les Strings

Lorsque l'on parle d'IHM, il est fréquent d'avoir à manipuler des strings pour modifier cette IHM, avoir des notions de base de manipulation de String s'avère donc essentielle.

1) Sous Visual Basic.net

Sous Visual Basic.net, un String est un objet, il est donc aisé de le modifier, voici les principaux outils de modification :

Méthodes statiques de la classe String		
Type	Membre	Description
Propriétés	Length	Nombre de caractères d'une chaîne Exemple : Dim s As String = "Visual Basic" s.Length ' sera égal à 12
Méthodes	Compare	Compare deux chaînes de caractères en respectant la casse. Exemple : String.Compare("Visual Basic", "VISUAL BASIC") Cette méthode renverra -1 (ou 1 si on inverse les arguments), elle renvoie 0 en cas d'égalité parfaite. Cette fonction accepte un troisième argument, True : la casse est respectée (défaut), False la casse n'est pas respectée.
	IndexOf, LastIndexOf	Indique l'index de la première occurrence (ou de la dernière occurrence) d'une sous-chaîne dans une chaîne : Dim s1 As String = "Visual Basic .NET, Visual Studio .NET" Dim s2 As Char() = {"."} s1.IndexOf(s2) 'Renverra 13 s1.LastIndexOf(s2) 'Renverra 33
	Insert	Insère une chaîne dans une autre chaîne. Par exemple, insérer le mot « Basic » Dans « Visual .NET » Dim s1 As String = "Visual .NET" s1.Insert(8, "Basic ") 's1 vaut "Visual Basic .NET"
	Remove	Supprime un nombre de caractères dans une chaîne à partir d'une position donnée. Par exemple, supprimer « Basic » dans « Visual Basic

		.NET » Dim s1 As String = "Visual Basic .NET" s1.Remove(7, 6)
	Replace	Remplace toutes les occurrences d'une chaîne dans une autre chaîne. Par exemple, remplacer « Studio » par « Basic » dans « Visual Studio .NET » Dim s1 As String = "Visual Studio .NET" s1.Replace("Studio", "Basic") 'Renvoi "Visual Basic .NET"
	Split	Les sous-chaînes délimitées par un ou plusieurs caractères dans une chaîne sont copiées dans un tableau de chaînes. Par exemple extraire les mots de « Visual.Basic.NET » et « Visual.Studio.NET » Dim s1 As String = _ "Visual.Basic.NET;Visual.Studio.NET" Dim s2 As Char() = {".", ";"} Dim s3 As String() 'Tableau qui va récupérer tous les mots s3 = s1.Split(s2) 's3 = {"Visual", "Basic", "NET", "Visual", ' "Studio", "NET"} '
	SubString	Extraire une sous-chaîne d'une chaîne. Par exemple extraire la sous-chaîne « Basic » de « Visual Basic .NET » Dim s1 As String = "Visual Basic .NET" s1.Substring(7, 5) 'renvoie « Basic » Il est possible d'utiliser Char : s1.Chars(7) 'renvoie « B »
	ToLower, ToUpper	Renvoi une chaîne en minuscules (ToLower) ou en majuscules (ToUpper). Dim s1 As String = "Un Texte !" s1.ToLower() 'Renvoi "un texte !"
	Trim, TrimStart, TrimEnd	Supprime toutes les occurrences d'un ou plusieurs caractères, aux deux, ou à l'une des extrémités d'une chaîne Dim s1 As String = "#@Visual#@" Dim s2 As Char() = {"#", "@"} s1.Trim(s2) 'Renvoi "Visual" s1.TrimStart(s2) 'Renvoi "Visual#@" s1.TrimEnd(s2) 'Renvoi "#@Visual"

En plus de toutes ces fonctions, il est possible d'utiliser les fonctions Visual Basic sous Visual Basic.net, cela est permis grâce à une volonté de Microsoft d'adaptation des codes Visual Basic vers Visual Basic.net. Il est donc possible d'utiliser sous Visual Basic.net les fonctions issues de Visual Basic 6, mais ceci est vivement déconseillé et ne devrait être utilisé uniquement dans le cas

d'importation de codes existant vers la plate forme .net.

Dernier point important, les chaines vides :

Il est possible sous Visual Basic de faire des comparaisons directes de chaines vides, on utilise pour cela les mots clef `Nothing` et `Empty`, exemple :

```
Dim s As String
If s = String.Empty Then 'ou : if s = Nothing
    '...
End If
```

Il est préférable cependant d'utiliser `String.Empty`, `Nothing` étant plutôt à réserver aux objets vides.

2) Sous Visual Basic

Sous Visual Basic, diverses fonctions apportent de quoi manipuler des `Strings`, on retrouve globalement les mêmes utilités que sous Visual Basic.net :

Membre	Description
Len	Retourne la longueur (nombre de caractères) d'une chaine <pre>Dim s As String s = "Test" Len(s) 'Retourne 4</pre>
InStr	Cette fonction cherche et retourne la première occurrence d'une sous-chaine de caractères dans une chaine. <pre>Dim s As String s = "Visual Basic" InStr(s, "Basic") 'Retourne 7</pre>
Mid	Renvoie une sous-chaine à partir des positions spécifiées <pre>Dim s As String s = "Visual Basic" Mid(s, 1, 3) 'Retourne "Vis" Mid(s, 5) 'Retourne "al Basic"</pre>
StrComp	Compare deux chaines de caractères en respectant la casse. Exemple : <pre>StrComp("Visual Basic", "VISUAL BASIC")</pre> Cette méthode renverra -1 (ou 1 si on inverse les arguments), elle renvoi 0 en cas d'égalité parfaite. Un troisième paramètre (optionnel) permet de choisir entre une comparaison texte et une comparaison respectant la casse.
StrConv	Fonction générique de conversion, il est possible de l'utiliser pour : -Upper Case (texte en majuscules) -Lower Case (en minuscules) -Proper Case (Première lettre de chaque mot en majuscule)

	... La liste est longue, une version complète de son utilisation est disponible ici : http://msdn.microsoft.com/fr-fr/library/cd7w43ec%28VS.80%29.aspx
Ucase, Lcase	Renvoie le texte en majuscules/minuscules
Space	Peu utile, renvoie une chaîne d'espaces Par exemple : Dim s As String s = Space(5) ' donc s=" "
String	Proche de Space, permet de créer une chaîne de caractères définie Par exemple : Dim s As String s = String(5, "-") ' donc s="-----"
Left	Retourne les x premiers caractères d'une chaîne Dim s As String s = "Test" Left(s, 2) ' Renvoie "Te"
Right	Comme Left, mais en partant de la fin
Trim, Ltrim, Rtrim	Ltrim : supprime les espaces en début de chaîne Rtrim : supprime les espaces en fin de chaîne Trim : supprime les espaces en début et fin de chaîne Voir http://msdn.microsoft.com/fr-fr/library/h9wz3dez%28VS.80%29.aspx

Sous Visual Basic, pour tester qu'une chaîne est nulle il suffit de tester si elle est égale à ""

```
Dim s As String
If s="" Then
    MsgBox "La chaîne est vide !"
End If
```

III) Les Nombres

Les deux langages (Visual Basic et Visual Basic.net) sont très proches ici, aussi il sera question de Visual Basic.net, mais sachez que Visual Basic pour la très large majorité est en tout point identique.

Pour utiliser les fonctions mathématiques de Visual Basic.net, il faut réaliser l'import suivant :

```
Imports System.Math
```

Pour utiliser les fonctions mathématiques de Visual Basic, il n'y a rien de spécifique à faire, pensez à utiliser en cas de problème l'objet Math mais celui-ci normalement est pré-utilisé (c'est à dire qu'il n'y a pas besoin de faire Math.Sqrt par exemple, Sqrt marche directement)

Visual Basic a cependant un gros défaut : les fonctions arc, hyperbolique (de Sin, Cos...), Min et Max n'existent pas, il faut donc les programmer à la main, cependant vu la facilité de la chose (2/3

lignes par fonction), et la facilité à trouver des implémentations existantes, elles ne seront pas présentées ici...

Sous Excel, bien que VBA soit issu de Visual Basic 6, Microsoft à rajouté ces fonctions, nul besoin de les reprogrammer.

Visual Basic	Visual Basic.net	Description
- (inexistant)	PI	Nombre π (de type Double)
-	E	Nombre e (de type Double)
Abs	Abs	Retourne la valeur absolue d'un nombre. Ex : Abs (-12, 4)
-, -	Acos, Asin	Retourne l'angle dont le cosinus/sinus est spécifié
-, -	Atan, Atan2	Retourne l'angle dont la tangente est le nombre spécifié (Atan), ou le quotient de deux nombres spécifiés (Atan2)
Fix	Ceiling	Retourne le plus petit entier supérieur ou égal au nombre spécifié
Sin, Cos, -, -	Sin, Cos, Sinh, Cosh	Retourne le sinus ou le cosinus. Retourne le sinus hyperbolique ou le cosinus hyperbolique
Exp	Exp	Retourne le nombre e élevé à la puissance spécifiée
Tan, -	Tan, Tanh	Tangente et tangente hyperbolique
Int	Floor	Retourne le plus grand nombre entier inférieur ou égal au nombre spécifié.
Log, -	Log, Log10	Retourne l'entier le plus proche du nombre spécifié (le premier entier inférieur si <0,5, le premier entier supérieur sinon)
-, -	Min, Max	Retournent la plus petit ou la plus grande des deux valeurs numériques. Les deux valeurs comparées doivent avoir le même type.
^	Pow	Fonction puissance
Round, Cint	Round	Retourne le nombre entier le plus proche de la valeur spécifiée
Sgn	Sign	Retourne une valeur indiquant le signe d'un nombre (0 si le nombre est nul)
Sqr	Sqrt	Retourne la racine carrée d'un nombre spécifié

Pour palier aux soucis de l' inexistence de π et de e sous Visual Basic, il est possible de faire appel aux constantes :

```
Const Pi = 3.14159265358979
```

IV) Les Tableaux

	GetUpperBound	Limite supérieure de la dimension spécifiée du tableau
	IndexOf	Index de la première occurrence d'une valeur dans un tableau : Dim Semaine() As String = {"Lundi", " "Mardi", "Mercredi", "Jeudi", "Vendredi", " "Samedi", "Dimanche"}
	LastIndexOf	Comme IndexOf, renvoie la dernière occurrence d'une valeur dans un tableau
	Reverse	Inverse l'ordre des éléments d'un tableau. En reprenant l'exemple du tableau Semaine : Array.Reverse(Semaine) For i=0 To Semaine.GetUpperBound(10) Console.WriteLine(Semaine(i)) Next 'Affiche: "Dimanche", "Samedi", "Vendredi"...
	Sort	Trie les éléments d'un tableau. Toujours avec Semaine : Array.sort(Semaine) ' (voir la boucle ci dessus) 'Affiche : "Dimanche", "Jeudi", "Lundi", ' "Mardi", "Mercredi", "Samedi", "Vendredi"