

Liste des fonctions VBA

Par Dominique MANIEZ 

Date de publication : 15 octobre 2006

Ce document vous propose la liste des fonctions VBA classées par thèmes.

I - Introduction.....	3
II - Les fonctions de chaîne.....	3
III - Les fonctions de date.....	6
IV - Les fonctions mathématiques.....	7
V - Les fonctions financières.....	8
VI - Les fonctions de gestion de fichiers.....	10
VII - Les fonctions logiques.....	11
VIII - Les fonctions de conversion.....	11
IX - Les fonctions système.....	12
X - Les fonctions de tableau.....	14
XI - Les fonctions de gestion d'objet.....	14
XII - Les fonctions de gestion d'erreur.....	14
XIII - Les fonctions de formatage.....	14
XIV - Les fonctions d'interface utilisateur.....	15
XV - Les fonctions d'impression.....	16

I - Introduction

Il existe un peu plus de 150 fonctions en Visual Basic. Même s'il vous est impossible de toutes les mémoriser, il est important de connaître leur existence ou tout du moins les grandes catégories de fonctions de manière à pouvoir orienter votre recherche.

En effet, il est particulièrement inutile de réinventer la roue et avant de vous lancer dans l'écriture d'un programme, vous devez vous demander si Visual Basic ne possède pas une fonction qui répond à votre besoin.

Pour prendre un exemple, si vous souhaitez extraire la racine carrée d'un nombre, ne tentez pas de rédiger une macro qui accomplisse cette tâche ; la fonction **Sqr** fera très bien l'affaire. Nous allons donc vous présenter différentes catégories de fonctions et quand vous aurez un besoin particulier, il vous faudra consulter ces listes afin de vérifier si une fonction convient à vos exigences.

Les tableaux listent les fonctions par catégorie et par ordre alphabétique, chaque nom de fonction étant accompagné d'une courte description. Si vous repérez une fonction qui est susceptible de vous intéresser, il faudra aller voir sa description complète dans l'aide en ligne.

II - Les fonctions de chaîne

Comme leur nom l'indique, les fonctions de chaîne travaillent sur des chaînes de caractères ; la plupart de ces fonctions s'emploient très souvent dans les programmes et il est par conséquent important de connaître les principales et notamment les fonctions suivantes :

Asc	Renvoie une donnée de type Integer représentant le code de caractère correspondant à la première lettre d'une chaîne.
Chr	Renvoie une valeur de type String contenant le caractère associé au code de caractère indiqué.
InStr	Renvoie une valeur de type Variant (Long) indiquant la position de la première occurrence d'une chaîne à l'intérieur d'une autre chaîne.
InStrRev	Renvoie la position d'une occurrence d'une chaîne dans une autre, à partir de la fin de la chaîne.
LCase	Renvoie une valeur de type String convertie en minuscules.
Left	Renvoie une valeur de type Variant (String) contenant le nombre indiqué de caractères d'une chaîne en partant de la gauche.
Len	Renvoie une valeur de type Long contenant le nombre de caractères d'une chaîne ou

	le nombre d'octets requis pour stocker une variable.
LTrim	Renvoie une valeur de type Variant (String) contenant une copie d'une chaîne en supprimant les espaces de gauche.
Mid	Renvoie une valeur de type Variant (String) contenant un nombre indiqué de caractères extraits d'une chaîne de caractères.
Replace	Renvoie une chaîne dans laquelle une sous-chaîne spécifiée a été remplacée plusieurs fois par une autre sous-chaîne.
Right	Renvoie une valeur de type Variant (String) contenant le nombre indiqué de caractères d'une chaîne en partant de la droite.
RTrim	Renvoie une valeur de type Variant (String) contenant une copie d'une chaîne en supprimant les espaces de droite.
Space	Renvoie une valeur de type Variant (String) comprenant le nombre d'espaces indiqué.
Str	Renvoie une valeur de type Variant (String) représentant un nombre.
StrComp	Renvoie une valeur de type Variant (Integer) indiquant le résultat d'une comparaison de chaînes.
StrConv	Renvoie une valeur de type Variant (String) convertie au format indiqué.
String	Renvoie une valeur de type Variant (String) contenant une chaîne constituée d'un caractère répété sur la longueur indiquée.
StrReverse	Renvoie une chaîne contenant des caractères dont l'ordre a été inversé par rapport à une chaîne donnée.
Trim	Renvoie une valeur de type Variant (String) contenant une copie d'une chaîne en supprimant les espaces de gauche et de droite.
UCase	Renvoie une valeur de type Variant (String) contenant la chaîne indiquée, convertie en majuscules.

Si vous fouillez dans l'aide en ligne, vous serez sans doute surpris de voir que certaines fonctions de chaîne sont doublées, comme **Left** et **Left\$** ou bien encore **Mid** et **Mid\$** ; apparemment, quand on examine les explications sur ces fonctions jumelles, on ne note aucune différence.

Et pourtant, elles existent et nous allons à cette occasion reparler des variables de type **Variant**. Vous vous souvenez peut-être que je vous avais dit que le type de données **Variant** était très souple et qu'il accomplissait une partie du travail à votre place.

Ce travail a bien évidemment un coût et ce que vous n'avez pas à faire, vous le payez en termes de performance. En fait, la différence entre **Mid** et **Mid\$** a trait au type de variable avec lequel la fonction va travailler : **Mid** travaille avec des données **Variant** et **Mid\$** travaille avec des données **String**. La belle affaire me direz-vous puisque cette distinction est totalement transparente pour vous. En réalité, elle ne l'est pas puisque la fonction **Mid\$** est dans certains cas presque deux fois plus rapide que son équivalent qui marche avec des variables **Variant**. Même si la programmation n'est pas votre activité principale et que vous n'avez aucun souci d'optimisation de votre code, il est important de comprendre ces différences et de pouvoir les évaluer. Nous allons voir dans l'exemple suivant qu'il est très facile de mettre en place une procédure de test de performance et vous verrez qu'un petit programme peut fournir des résultats significatifs :

```

Sub testperfl()
Dim boucle As Double
chaîne = " La femme des uns fait le bonheur des autres "
depart = Timer 'déclenche le chrono
For boucle = 1 To 1000000
Dim x
x = Mid(chaîne, 26, 7)
Next boucle
fin = Timer ' arrête le chrono
MsgBox (fin - depart)
' affiche le résultat en secondes
End Sub
    
```

Ce premier programme initialise une chaîne de caractères puis déclenche un compteur à l'aide de la fonction **Timer** qui renvoie une valeur représentant le nombre de secondes écoulées depuis minuit.

Une boucle est exécutée ensuite un million de fois. À l'intérieur de cette boucle, on initialise une variable qui est de type **VARIANT** étant donné qu'aucun type n'est déclaré et on réalise une opération sur une chaîne de caractères. À la fin de la boucle, on déclenche un deuxième chrono et on affiche la différence entre les deux chronos, c'est-à-dire le temps écoulé pendant le déroulement de la boucle.

Le deuxième programme de test est similaire sauf qu'on prend la peine de déclarer les variables chaîne en tant que **String** et qu'on utilise la fonction **Mid\$** :

```

Sub testperf2()
Dim chaîne As String
Dim boucle As Double
chaîne = " La femme des uns fait le bonheur des autres "
depart = Timer
For boucle = 1 To 1000000
Dim x As String
x = Mid$(chaîne, 26, 7)
Next boucle
fin = Timer
MsgBox (fin - depart)
End Sub
    
```

Le tableau ci-dessous vous montre les résultats des tests effectués sur huit fonctions. La deuxième colonne indique le temps avec la fonction renvoyant une variable **VARIANT** et la troisième colonne indique le temps avec la fonction renvoyant une variable **String**. Les valeurs de temps sont exprimées en secondes.

Fonction	VARIANT	String	Différence en %
x = Mid(chaîne, 26, 7)	1.54	1.14	0,35
x = Left(chaîne, 20)	1.30	.92	0,41
x = Right(chaîne, 20)	1.30	.92	0,41
x = LTrim(chaîne)	1.30	.99	0,31
x = RTrim(chaîne)	1.26	.98	0,29
x = Trim(chaîne)	1.38	1.10	0,25
x = LCase(chaîne)	12.14	11.87	0,02
x = UCase(chaîne)	11.32	10.93	0,04

Même s'il faut prendre ces mesures avec des pincettes, elles sont suffisamment significatives sur certaines fonctions pour que cela puisse avoir un impact dans certains programmes qui font un usage intensif du traitement des chaînes de caractères.

Ces deux petits programmes ont surtout un but pédagogique : ils vous montrent qu'il est aisé d'écrire une macro pour tester deux versions différentes d'un même programme. De plus, ils vous enseignent que la facilité fait finalement perdre du temps et qu'on a toujours intérêt à être le plus explicite possible dans un programme, car on gagne en temps et en lisibilité.

III - Les fonctions de date

Les fonctions de date sont nombreuses en Visual Basic et de nombreux exemples sont disponibles dans la **FAQ**. Si vous avez à traiter des dates dans votre programme, il serait bien rare que vous ne trouviez pas dans cette liste une fonction qui réponde à vos besoins.

Date	Renvoie une valeur de type Variant (Date) contenant la date système actuelle.
DateAdd	Renvoie une valeur de type Variant (Date) contenant une date à laquelle un intervalle de temps spécifié a été ajouté.
DateDiff	Renvoie une valeur de type Variant (Long) indiquant le nombre d'intervalles de temps entre deux dates données.
DatePart	Renvoie une valeur de type Variant (Integer) contenant l'élément spécifié d'une date donnée.
DateSerial	Renvoie une valeur de type Variant (Date) correspondant à une année, un mois et un jour déterminés.
DateValue	Renvoie une valeur de type Variant (Date).
Day	Renvoie une valeur de type Variant (Integer) indiquant un nombre entier compris entre 1 et 31, inclus, qui représente le jour du mois.
Hour	Renvoie une valeur de type Variant (Integer) indiquant un nombre entier compris entre 0 et 23 inclus, qui représente l'heure du jour.
Minute	Renvoie une valeur de type Variant (Integer) indiquant un nombre entier compris entre 0 et 59, inclus, qui représente la minute de l'heure en cours.
Month	Renvoie une valeur de type Variant (Integer) indiquant un nombre entier compris entre 1 et 12, inclus, qui représente le mois de l'année.
MonthName	Renvoie une chaîne indiquant le mois spécifié.
Now	Renvoie une valeur de type Variant (Date) indiquant la date et l'heure en cours fournies par la date et l'heure système de votre ordinateur.
Second	Renvoie une valeur de type Variant (Integer) indiquant un nombre entier compris entre 0

	et 59, inclus, qui représente la seconde de la minute en cours.
Time	Renvoie une valeur de type Variant (Date) indiquant l'heure système en cours.
Timer	Renvoie une valeur de type Single représentant le nombre de secondes écoulées depuis minuit.
TimeSerial	Renvoie une valeur de type Variant (Date) contenant une heure précise (heure, minute et seconde).
TimeValue	Renvoie une valeur de type Variant (Date) contenant une heure.
Weekday	Renvoie une valeur de type Variant (Integer) contenant un nombre entier qui représente le jour de la semaine.
WeekdayName	Renvoie une chaîne indiquant le jour de la semaine spécifié.
Year	Renvoie une valeur de type Variant (Integer) contenant un nombre entier qui représente l'année.

IV - Les fonctions mathématiques

Abs	Renvoie une valeur de même type que celle transmise, indiquant la valeur absolue d'un nombre.
Atn	Renvoie une valeur de type Double indiquant l'arctangente d'un nombre.
Cos	Renvoie une valeur de type Double indiquant le cosinus d'un angle.
Exp	Renvoie une valeur de type Double indiquant la valeur de e (base des logarithmes népériens) élevé à une puissance.
Fix	Renvoie la partie entière d'un nombre.
Hex	Renvoie une valeur de type String représentant un nombre sous forme hexadécimale.
Int	Renvoie la partie entière d'un nombre.
Log	Renvoie une valeur de type Double indiquant le logarithme népérien d'un nombre.
Oct	Renvoie une valeur de type Variant (String) représentant la valeur octale d'un nombre.
Partition	Renvoie une chaîne de caractères de type Variant (String) indiquant l'endroit où un

	nombre apparaît au sein d'une série calculée de plages de valeurs.
Rnd	Renvoie une valeur de type Single contenant un nombre aléatoire.
Round	Renvoie un nombre arrondi à un nombre spécifié de positions décimales.
Sgn	Renvoie une valeur de type Variant (Integer) indiquant le signe d'un nombre.
Sin	Renvoie une valeur de type Double indiquant le sinus d'un angle.
Sqr	Renvoie une valeur de type Double indiquant la racine carrée d'un nombre.
Tan	Renvoie une valeur de type Double indiquant la tangente d'un angle.
Val	Renvoie le nombre contenu dans une chaîne de caractère sous la forme d'une valeur numérique d'un type approprié.

Certaines de ces fonctions sont très spécialisées (comme les fonctions trigonométriques) et ne vous serviront sans doute que très rarement. En revanche vous utiliserez souvent les fonctions **Abs**, **Int**, **Round** et **Val**.

 *D'autres fonctions plus complexes peuvent être dérivées des fonctions mathématiques de base. Vous en retrouverez la liste dans les [pages sources Access](#).*

V - Les fonctions financières

Ces fonctions sont l'équivalent des fonctions financières que l'on trouve dans Excel (la troisième colonne du tableau indique d'ailleurs le nom correspondant Excel).

Si vous développez une macro avec Excel, il est sans doute préférable d'utiliser les fonctions internes d'Excel mais si vous devez faire du calcul financier dans un programme Word ou Access, il sera plus facile d'utiliser les fonctions de Visual Basic.

DDB	Renvoie une valeur de type Double indiquant l'amortissement d'un bien au cours d'une période spécifique en utilisant la méthode d'amortissement dégressif à taux double ou toute autre méthode précisée.	DDB
FV	Renvoie une valeur de type Double indiquant le futur montant d'une annuité basée sur des versements constants et périodiques et sur un taux d'intérêt fixe.	VC
IPmt	Renvoie une valeur de type Double indiquant le montant, sur une période donnée, d'une annuité basée sur des versements constants et	INTPER

	périodiques et sur un taux d'intérêt fixe.	
IRR	Renvoie une valeur de type Double indiquant le taux de rendement interne d'une série de mouvements de trésorerie périodiques (paiements et encaissements).	TRI
MIRR	Renvoie une valeur de type Double indiquant le taux de rendement interne modifié d'une série de mouvements de trésorerie périodiques (paiements et encaissements).	TRIM
Nper	Renvoie une valeur de type Double indiquant le nombre d'échéances d'une annuité basée sur des versements constants et périodiques et sur un taux d'intérêt fixe.	NPM
NPV	Renvoie une valeur de type Double indiquant la valeur nette actuelle d'un investissement, calculée en fonction d'une série de mouvements de trésorerie périodiques (paiements et encaissements) et d'un taux d'escompte.	VAN
Pmt	Renvoie une valeur de type Double indiquant le montant d'une annuité basée sur des versements constants et périodiques et sur un taux d'intérêt fixe.	VPM
PPmt	Renvoie une valeur de type Double indiquant le remboursement du capital, pour une échéance donnée, d'une annuité basée sur des versements constants et périodiques et sur un taux d'intérêt fixe.	PRINCPER
PV	Renvoie une valeur de type Double indiquant le montant actuel d'une annuité basée sur des échéances futures constantes et périodiques, et sur un taux d'intérêt fixe.	VA
Rate	Renvoie une valeur de type Double indiquant le taux d'intérêt par échéance pour une annuité.	TAUX
SLN	Renvoie une valeur de type Double indiquant	AMORLIN

	l'amortissement linéaire d'un bien sur une période donnée.	
SYD	Renvoie une valeur de type Double indiquant l'amortissement global d'un bien sur une période donnée.	SYD

VI - Les fonctions de gestion de fichiers

Ces fonctions permettent de manipuler des fichiers au sens large du terme et vous pouvez aussi bien obtenir le nom d'un fichier, connaître ses attributs, calculer sa longueur que le parcourir octet par octet. Cela étant, on préférera souvent une autre méthode pour manipuler des fichiers et notamment par le biais de l'objet **FileSystemObject** que nous étudierons plus tard.

CurDir	Renvoie une valeur de type Variant (String) indiquant le chemin en cours.
Dir	Renvoie une valeur de type String représentant le nom d'un fichier, d'un répertoire ou d'un dossier correspondant à une chaîne de recherche, à un attribut de fichier ou au nom de volume d'un lecteur.
EOF	Renvoie une valeur de type Integer contenant la valeur Boolean True lorsque la fin d'un fichier ouvert en mode Random ou Input séquentiel est atteinte.
FileAttr	Renvoie une valeur de type Long représentant le mode d'ouverture des fichiers avec l'instruction Open.
FileDateTime	Renvoie une valeur de type Variant (Date) indiquant la date et l'heure de création ou de dernière modification d'un fichier.
FileLen	Renvoie une valeur de type Long indiquant la longueur en octets d'un fichier.
FreeFile	Renvoie une valeur de type Integer représentant le prochain numéro de fichier pouvant être utilisé par l'instruction Open.
GetAttr	Renvoie une valeur de type Integer indiquant les attributs du fichier ou du dossier.
Input	Renvoie une valeur de type String contenant les caractères lus dans un fichier ouvert en mode Input ou Binary.
Loc	Renvoie une valeur de type Long indiquant la position de lecture/écriture courante dans un fichier ouvert.
LOF	Renvoie une valeur de type Long représentant la taille, exprimée en octets, d'un fichier ouvert à l'aide de l'instruction Open.
Seek	Renvoie une valeur de type Long indiquant la position de lecture/écriture courante dans un fichier ouvert à l'aide de l'instruction Open.

VII - Les fonctions logiques

Choose	Sélectionne et renvoie une valeur à partir d'une liste d'arguments.
IIf	Renvoie l'un ou l'autre de deux arguments selon l'évaluation d'une expression.
IsDate	Renvoie une valeur de type Boolean qui indique si une expression peut être convertie en date.
IsEmpty	Renvoie une valeur de type Boolean indiquant si une variable a été initialisée.
IsError	Renvoie une valeur de type Boolean qui indique si une expression est une valeur d'erreur.
IsMissing	Renvoie une valeur de type Boolean qui indique si un argument facultatif de type Variant a été passé dans une procédure.
IsNull	Renvoie une valeur de type Boolean qui indique si une expression ne contient aucune donnée valide (Null).
IsNumeric	Renvoie une valeur de type Boolean qui indique si une expression peut être interprétée comme un nombre.
IsObject	Renvoie une valeur de type Boolean qui indique si un identificateur représente une variable objet.
Switch	Évalue une liste d'expressions et renvoie une valeur de type Variant ou une expression associée à la première expression de la liste qui a pour valeur True.

Parmi toutes ces fonctions, vous utiliserez surtout la fonction **IIf** qui permet de faire un test conditionnel **If Then** sur une seule ligne et les fonctions **IsEmpty** et **IsNull**.

Exemple :

```
Dim sexe as string
sexe=IIF(civilite="Mr", "Masculin", "Féminin")
```

VIII - Les fonctions de conversion

Les fonctions de conversion sont extrêmement importantes et elles seront bien souvent le seul remède pour ne pas commettre des erreurs de type.

Vous devez vous rappeler qu'il est en effet impossible de mélanger les types de données au sein d'une même expression ; si je souhaite, par exemple, afficher dans une boîte de dialogue le nom d'un étudiant suivi de sa moyenne générale aux examens, il faut que je fasse une conversion de la note puisque je n'ai pas le droit de mettre bout à bout une variable caractère et une variable numérique. Je vais donc employer une fonction de conversion qui va changer le type de la variable numérique en caractère.

Le programme suivant illustre cette technique :

```
Dim nometudiant As String
Dim moyenne As Double
nometudiant = "MARTIN"
moyenne = 12
' Erreur de type
' MsgBox (nometudiant + " : " + moyenne)
```

```
' Conversion donc pas d'erreur
MsgBox (nometudiant + " : " + CStr (moyenne))
```

Les fonctions de conversion convertissent une expression en un type de données spécifique. Leur syntaxe est la suivante :

```
Nom_de_fonction(expression)
```

L'argument *expression* peut être n'importe quelle expression de chaîne ou expression numérique et le nom de la fonction détermine le type renvoyé, comme le montre le tableau suivant

Fonction	Type renvoyé	Plage de valeurs de l'argument expression
CBool	Boolean	Toute chaîne ou expression numérique valide.
CByte	Byte	0 à 255.
CCur	Currency	-922 337 203 685 477,5808 à 922 337 203 685 477,5807.
CDate	Date	Toute expression de date valide.
Cdbl	Double	-1.79769313486231E308 à

Si l'argument *expression* passé à la fonction excède la plage de valeurs du type de données cible, une erreur se produit. Il est donc préférable avant de réaliser une conversion de s'assurer qu'elle soit valide. Vous utiliserez pour ce faire des fonctions logiques comme par exemple la fonction **IsDate** pour déterminer si la valeur de l'argument date peut être convertie en date ou en heure.

La fonction **CDate** reconnaît les littéraux date et heure ainsi que certains nombres appartenant à la plage de dates autorisées. Lors de la conversion d'un nombre en date, la partie entière du nombre est convertie en date. Si le nombre comprend une partie décimale, celle-ci est convertie en heures exprimées en partant de minuit.

La fonction **CDate** reconnaît les formats de date définis dans les paramètres régionaux de votre système. L'ordre des jours, mois et années risque de ne pouvoir être défini si les données sont fournies dans un format différent des paramètres de date reconnus. De plus, les formats de date complets précisant le jour de la semaine ne sont pas reconnus. Il existe d'autres fonctions de conversion comme **Str** ou **Val** mais il vaut mieux utiliser les fonctions normalisées. Par exemple, la fonction **Val** ne respecte pas les conventions étrangères alors que la fonction **CCur** reconnaît divers types de séparateurs décimaux, de séparateurs des milliers et diverses options monétaires, selon les paramètres régionaux de votre ordinateur.

IX - Les fonctions système

Command	Renvoie la partie argument de la ligne de commande utilisée pour lancer Microsoft Visual Basic ou un programme exécutable développé avec Visual Basic.
DoEvents	Arrête momentanément l'exécution afin que le système d'exploitation puisse traiter d'autres événements.
Environ	Renvoie la valeur de type String associée à une variable d'environnement du système d'exploitation. Non disponible sur le Macintosh.
GetAllSettings	Renvoie une liste des clés et leurs valeurs respectives (créées à l'origine à l'aide de l'instruction SaveSetting), figurant dans une

	entrée d'application de la base de registres de Windows.
GetSetting	Renvoie une valeur de clé d'une entrée d'application de la base de registres de Windows.
IMEStatus	Renvoie une valeur de type Integer indiquant le mode IME (Input Method Editor) en cours de Microsoft Windows ; disponible uniquement dans les versions destinées aux pays asiatiques.
MacID	Utilisée sur Macintosh pour convertir une constante à quatre caractères en une valeur pouvant être exploitée par les fonctions Dir, Kill, Shell et AppActivate.
MacScript	Exécute un script AppleScript et retourne une valeur renvoyée par le script, le cas échéant.
QBColor	Renvoie une valeur de type Long indiquant le code de couleur RGB correspondant au numéro de couleur indiqué.
RGB	Renvoie un entier de type Long représentant le code RGB.
Shell	Lance un programme exécutable et renvoie une valeur de type Variant (Double) représentant l'identificateur (ID) de la tâche exécutée en cas de succès, ou un zéro en cas d'échec.
TypeName	Renvoie une valeur de type String qui fournit des informations sur une variable.
VarType	Renvoie une valeur de type Integer qui indique le sous-type d'une variable.

Ces fonctions donnent des renseignements sur l'état du système et vous en aurez rarement besoin, car certaines sont vraiment très spécialisées. La fonction **VarType** peut être intéressante pour bien comprendre le rôle des types de données.

X - Les fonctions de tableau

Array	Renvoie une variable de type Variant contenant un tableau.
Filter	Renvoie un tableau de base zéro contenant un sous-ensemble d'un tableau de chaîne basé sur des critères de filtre spécifiés.
IsArray	Renvoie une valeur de type Boolean qui indique si une variable est un tableau.
Join	Renvoie une chaîne créée par la jonction de plusieurs sous-chaînes contenues dans un tableau.
LBound	Renvoie une valeur de type Long contenant le plus petit indice disponible pour la dimension indiquée d'un tableau.
Split	Renvoie un tableau de base zéro à une dimension contenant le nombre spécifié de sous-chaînes.
UBound	Renvoie une valeur de type Long contenant le plus grand indice disponible pour la dimension indiquée d'un tableau.

Rappel : un tableau est une variable particulière qui peut contenir plusieurs valeurs ; chaque valeur du tableau s'appelle un élément et on accède à chaque élément par un numéro nommé indice.

XI - Les fonctions de gestion d'objet

CallByName	Exécute une méthode d'un objet ou définit ou renvoie une propriété d'un objet.
CreateObject	Crée et renvoie une référence à un objet ActiveX.
GetObject	Renvoie une référence à un objet fourni par un composant ActiveX.

De plus amples détails sur la programmation orientée objet sous VB sont disponibles dans le **tutoriel** de Xavier VLIEGHE.

XII - Les fonctions de gestion d'erreur

Si on veut bien programmer, il faut être paranoïaque et prévoir le pire. Les fonctions de gestion d'erreur vous permettent de traiter préventivement les erreurs éventuelles qui peuvent se produire dans votre programme.

CVErr	Renvoie une donnée de type Variant et de sous-type Error contenant un numéro d'erreur spécifié par l'utilisateur.
Error	Renvoie le message d'erreur correspondant à un numéro d'erreur donné.

XIII - Les fonctions de formatage

Format	Renvoie une valeur de type Variant (String) contenant une expression formatée en
---------------	--

	fonction des instructions contenues dans l'expression de mise en forme.
FormatCurrency	Renvoie une expression formatée sous forme de valeur de type Currency utilisant le symbole monétaire défini dans le Panneau de configuration du système.
FormatDateTime	Renvoie une expression formatée sous forme de date ou d'heure.
FormatNumber	Renvoie une expression formatée sous forme de nombre.
FormatPercent	Renvoie une expression formatée sous forme de pourcentage (multiplié par 100) avec un caractère % de fin.

XIV - Les fonctions d'interface utilisateur

InputBox	Affiche une invite dans une boîte de dialogue, attend que l'utilisateur tape du texte ou clique sur un bouton, puis renvoie le contenu de la zone de texte sous la forme d'une valeur de type String.
MsgBox	Affiche un message dans une boîte de dialogue, attend que l'utilisateur clique sur un bouton, puis renvoie une valeur de type Integer qui indique le bouton choisi par l'utilisateur.

La fonction **MsgBox** est très simple d'utilisation. En revanche, il nous faut dire un mot de **InputBox** qui est aussi une fonction importante dans le cadre du dialogue homme-machine puisqu'elle permet de récupérer des informations saisies par l'utilisateur.

Il est très important de noter que la valeur renvoyée par **InputBox** est une variable **String** ce qui signifie que si vous faites saisir un nombre à un utilisateur, il faudra en principe le convertir, grâce à une fonction de conversion telle que **Cdbl**, **Cint** ou **CLng**, en une variable numérique, même si Visual Basic n'est parfois pas très regardant.

La syntaxe de **InputBox** ressemble à celle de **MsgBox** :

```
InputBox(prompt[, title] [, default] [, xpos] [, ypos] [, helpfile, context])
```

Avec les arguments :

default	Facultatif. Expression de chaîne affichée par défaut dans la zone de texte en l'absence de toute autre valeur. Si l'argument default est omis, la zone de texte qui s'affiche est vide.
xpos	Facultatif. Expression numérique indiquant, en twips, la distance horizontale qui sépare le bord gauche de l'écran de la bordure gauche de la boîte de dialogue. Si l'argument xpos est omis, la boîte de dialogue est centrée horizontalement.
ypos	Facultatif. Expression numérique indiquant, en twips, la distance verticale qui sépare le haut de l'écran de la bordure supérieure de la boîte de dialogue. Si l'argument ypos est omis, la boîte de dialogue est positionnée verticalement, à environ un tiers de l'écran en partant du haut.

Pour votre gouverne, vous pouvez retenir qu'un twip est un 20^e de point et qu'un point est approximativement un 7^e de pouce qui, comme chacun le sait, mesure 2,54 centimètres. Visual Basic utilise les twips en interne pour la résolution de l'affichage et certains paramètres de fonctions doivent être exprimés dans cette mesure. Il n'y a pas malheureusement de correspondance directe entre les twips et les pixels, car les twips dépendent de la taille de la police qui peut varier d'un système à l'autre.

XV - Les fonctions d'impression

Spc	Utilisée avec l'instruction Print # ou la méthode Print pour positionner la sortie.
Tab	Utilisée avec l'instruction Print # ou la méthode Print pour positionner la sortie.