



# 3

## Les fonctions VBScript

## 3. Les fonctions VBScript

*VBScript* hérite de ses ancêtres *Basic* et *Visual Basic* d'une multitude de fonctions procurant une grande flexibilité à ce langage. Ces fonctions sont intrinsèques au langage et ne nécessitent donc pas que le script référence un modèle d'objets externe tel que vous apprendrez à les utiliser dans les prochains chapitres.

### Fonctions d'interactivité avec l'utilisateur

*VBScript* prévoit quelques fonctions permettant l'interaction avec l'utilisateur. La première, `MsgBox`, permet d'afficher de l'information à l'utilisateur tandis que la seconde, `InputBox`, permet de saisir de l'information de la part de l'utilisateur.

Fonction	Description	Exemple
<code>Beep</code>	Génère un signal sonore.	<code>Beep</code>
<code>MsgBox</code>	Affiche une boîte de message à l'utilisateur.	<code>MsgBox "Allô la planète"</code>
<code>InputBox</code>	Affiche une boîte d'invite à l'utilisateur et retourne la valeur saisie par celui-ci.	<code>strR = InputBox("Entrez " ▼ &amp; "un nombre entre 1 et 10")</code>

La fonction `MsgBox` permet de spécifier le texte à afficher au sein de la boîte de dialogue mais également les boutons et icônes à afficher ainsi que le titre que devra arborer la boîte de dialogue selon le prototype suivant :

```
MsgBox ( Message [, Boutons] [,Titre] )
```

Le paramètre *Boutons* peut prendre l'une des valeurs constantes suivantes :

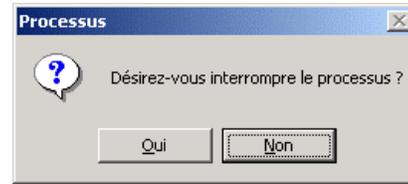
Constante	Valeur	Description
<code>vbOkOnly</code>	0	Affiche le bouton <code>Ok</code> seulement.
<code>vbOkCancel</code>	1	Affiche les boutons <code>Ok</code> et <code>Annuler</code> .
<code>vbAbortRetryIgnore</code>	2	Affiche les boutons <code>Abandonner</code> , <code>Réessayer</code> et <code>Ignorer</code> .
<code>vbYesNoCancel</code>	3	Affiche les boutons <code>Oui</code> , <code>Non</code> et <code>Annuler</code> .
<code>vbYesNo</code>	4	Affiche les boutons <code>Oui</code> et <code>Annuler</code> .
<code>vbRetryCancel</code>	5	Affiche les boutons <code>Réessayer</code> et <code>Annuler</code> .
<code>vbCritical</code>	16	Affiche l'icône correspondant aux erreurs critiques.
<code>vbQuestion</code>	32	Affiche l'icône correspondant aux questions.
<code>vbExclamation</code>	48	Affiche l'icône correspondant aux avertissements.
<code>vbInformation</code>	64	Affiche l'icône correspondant aux informations.
<code>vbDefaultButton1</code>	0	Sélectionne par défaut le premier bouton.
<code>vbDefaultButton2</code>	256	Sélectionne par défaut le second bouton.
<code>vbDefaultButton3</code>	512	Sélectionne par défaut le troisième bouton.
<code>vbDefaultButton4</code>	768	Sélectionne par défaut le quatrième bouton.

Plusieurs valeurs constantes peuvent être combinées à l'aide de l'opérateur + comme le montre l'exemple suivant :

```
MsgBox "Désirez-vous interrompre le processus ?", vbYesNo + vbQuestion + vbDefaultButton2, "Processus"
```

Ce qui produit le résultat ci-contre. Notez le titre de la boîte de dialogue, les deux boutons *Oui* et *Non* mais notez plus particulièrement que le focus a été déposé par défaut sur le second bouton.

On tentera toujours de déposer le focus par défaut sur le bouton dont la réponse est la moins dangereuse pour l'utilisateur.



Maintenant, reste à récupérer la réponse de l'utilisateur à une telle question. La fonction `MsgBox` permet de connaître la réponse de l'utilisateur en testant la valeur retournée par cette fonction :

Constante	Valeur	Description
VbOk	1	L'utilisateur a appuyé sur le bouton Ok.
vbCancel	2	L'utilisateur a appuyé sur le bouton Annuler.
vbAbort	3	L'utilisateur a appuyé sur le bouton Abandonner.
vbRetry	4	L'utilisateur a appuyé sur le bouton Réessayer.
vblgnore	5	L'utilisateur a appuyé sur le bouton Ignorer.
VbYes	6	L'utilisateur a appuyé sur le bouton Oui.
VbNo	7	L'utilisateur a appuyé sur le bouton Non.

Voici un exemple au sein de lequel l'utilisateur est invité à spécifier s'il désire terminer un processus quelconque ou s'il désire continuer à l'aide de la boîte de dialogue ci-contre.

Si l'utilisateur appuie sur le bouton *Oui*, le script s'arrête à l'aide de l'instruction `Exit Do` vue précédemment.



```
Do
  If MsgBox ("Continuer ?", vbYesNo + vbQuestion) = vbNo Then
    Exit Do
  End If
  MsgBox "Alors on recommence...", vbInformation
Loop
MsgBox "C'est ici que cela se termine!", vbInformation
```

CH02\Recommencer.vbs

La fonction `InputBox` permet de spécifier le texte à afficher au sein de la boîte de dialogue mais également le titre que devra arborer la boîte de dialogue et la valeur par défaut inscrite au sein de la zone de saisie selon le prototype suivant :

```
InputBox ( Message [,Titre] [,Defaut] [,xPos] [,yPos] )
```

Le paramètre optionnel `Defaut` permet de préciser la saisie par défaut affichée à l'utilisateur :

```
Dim strReponse
strReponse = InputBox("Entrez votre nom", , "Votre nom")
```

Ce qui produit le résultat ci-contre. Remarquez que du texte est déjà présent dans la zone de saisie et qu'il ne reste à l'utilisateur qu'à le modifier.



Les paramètres optionnels `xPos` et `yPos` permettent de préciser la position exprimée en pixels à laquelle la boîte de saisie devra s'afficher. La boîte s'affiche au centre de l'écran si aucune valeur pour ces paramètres n'est précisée.

Maintenant, reste à récupérer la réponse de l'utilisateur. La fonction `InputBox` permet de connaître la réponse de l'utilisateur en récupérant la valeur retournée par cette fonction. Dans l'exemple suivant, l'utilisateur est invité à préciser sa date de naissance et cette information est ultérieurement stockée dans la variable `dNaissance`. Ensuite, cette variable peut être testée afin d'en assurer la validité. L'exemple suivant recommence la saisie de la date de naissance si l'information saisie par l'utilisateur est invalide :

```
Dim dNaissance

'*****'
'* S'assure de récupérer une date valide. *'
'*****'
Do
    dNaissance = InputBox("Entrez votre date de naissance." & ▼
        & vbCrLf & vbCrLf & "Spécifiez la date dans le format " ▼
        & " yyyy-mm-dd")

Loop Until IsDate(dNaissance)
```

Notez que la fonction `InputBox` retourne une chaîne vide "" lorsque l'utilisateur appuie sur le bouton `Annuler` de la boîte de saisie.

## Fonctions de date et de temps

VBScript prévoit une pluralité de fonctions permettant la manipulation aisée des données de type date. Voici la liste complète des fonctions directement reliées au traitement des dates et heures :

Fonction	Description	Exemple
Date	Retourne la date du jour courant selon la date système.	MsgBox "Nous sommes le " & <b>Date()</b>
DateAdd	Retourne une date résultant de l'addition d'un nombre de jour, de mois ou d'années spécifié. La fonction soustrait si la valeur spécifiée est négative.	Dim dt1, dt2 dt1 = Today() dt2 = DateAdd("d",15, dt1)
DateDiff	Retourne le nombre de secondes, minutes, heures, jours, mois ou d'années espaçant deux dates spécifiées.	Dim dt1, dt2 dt1 = #15-03-2002# dt2 = <b>DateDiff</b> ("m",dt1,Date())
DatePart	Retourne une partie spécifique de la date spécifiée. Peut être le jour, le mois, etc.	Dim nMois, dA dA = Today() nMois =DatePart(dA, "m") 'Retourne le mois
DateSerial	Retourne une date en format Date à partir d'informations en format sériel.	MsgBox <b>DateSerial</b> (1975, 11, 3) 'Donne "3 novembre 1975"
DateValue	Retourne une date en format Date à partir d'informations spécifiées en format String.	Dim dtA dtA = <b>DateValue</b> ("1/31/2000")
Day	Retourne le numéro du jour contenu au sein d'une expression spécifiée.	Dim dDate, dJour dDate = Date() dJour = <b>Day</b> (dDate)
Hour	Retourne les heures contenues au sein d'une expression spécifiée.	Dim dHre, nHeure dHre = Time() nHeure = <b>Hour</b> (dHre)
Minute	Retourne les minutes contenues au sein d'une expression spécifiée.	Dim dHre, nMinute dHre = Time() nMinute = <b>Hour</b> (dHre)
Month	Retourne le numéro du mois (entre 1 et 12) contenu au sein d'une expression spécifiée.	Dim dDate, dMois dDate = Date() dMois = <b>Month</b> (dDate)
MonthName	Retourne le nom du mois correspondant au numéro de mois spécifié.	Dim dDate, dMois dDate = Date() dMois = Month(dDate) MsgBox "Nous sommes au mois " & <b>MonthName</b> (dMois)
Now	Retourne la date du jour courant et l'heure courante selon les date et heure systèmes.	MsgBox "Date et heure: " & <b>Now()</b>
Second	Retourne les secondes contenues au sein d'une expression spécifiée.	Dim dHre, nSeconde dHre = Time() nSeconde = <b>Second</b> (dHre)
Time	Retourne l'heure courante selon l'heure système.	MsgBox "Il est " & <b>Time()</b>
TimeSerial	Retourne une date en format Date à partir d'informations en format sériel.	MsgBox <b>TimeSerial</b> (23, 75, 0) '"1/2/0001 12:15:00 AM"

TimeValue	Retourne une date en format <code>Date</code> à partir d'informations spécifiées en format <code>String</code> .	Dim dtA dtA = <b>TimeValue</b> ("4:35:17 PM")
Weekday	Retourne le numéro du jour de la semaine (entre 1 et 7) contenu au sein d'une expression spécifiée.	Dim dDate, dJour dDate = Date() dJour = <b>Weekday</b> (dDate)
WeekdayName	Retourne le nom du jour de la semaine correspondant au numéro de jour spécifié.	Dim dDate, dJour dDate = Date() dJour = Weekday(dDate) MsgBox "Nous sommes " ▼ & <b>WeekdayName</b> (dJour)
Year	Retourne l'année contenu au sein d'une expression spécifiée.	Dim dDate, nAnnee dDate = Date() MsgBox "Nous sommes en " ▼ & "l'an " & <b>Year</b> (dDate)

Le code suivant permet de préciser le temps écoulé depuis la naissance de l'utilisateur et la date courante.

```
Option Explicit

Dim dNaissance, dDiff, nJours, strDate

'*****
'* S'assure d'abord de récupérer une date valide.          *'
'*****
Do
    dNaissance = InputBox("Entrez votre date de naissance." & ▼
        & vbCrLf & vbCrLf & "Spécifiez la date dans le format " ▼
        & " yyyy-mm-dd")

Loop Until IsDate(dNaissance)

'*****
'* Calcule le nombre de mois séparant la date de naissance *'
'*****
dDiff = DateDiff("m", dNaissance, Date())

'*****
'* Calcule le nombre de jours restants                       *'
'*****
nJours = DateDiff("d", DateAdd("m", dDiff, dNaissance), Date())

'*****
'* Affiche le résultat à l'utilisateur                       *'
'*****
strDate = WeekdayName(Weekday(dNaissance)) & " le " & Day(dNaissance) ▼
    & " " & MonthName(Month(dNaissance)) & " " & Year(dNaissance)

MsgBox "Si vous êtes né(e) " & strDate & " vous avez " & dDiff \ 12 ▼
    & " ans, " & dDiff Mod 12 & " mois et " & nJours & " jours."
```

CH03\Naissance.vbs

## Fonctions de manipulation des chaînes de caractères

VBScript dispose de plusieurs fonctions permettant de manipuler les chaînes de caractères. Cependant, avant de commencer, il peut être important de distinguer les caractères ASCII des caractères Unicode.

Autrefois, les chaînes de caractères étaient constituées de caractères codés sur 8 bits regroupés arbitrairement au sein d'une table dite ASCII. Selon cette table, les lettres majuscules occupaient les positions 65 à 90 (A à Z), les lettres minuscules occupaient les positions 97 à 122 (a à z), les nombres occupaient les positions 48 à 57 (0 à 9) et les autres symboles occupaient diverses autres positions. La valeur ordinaire des caractères était et est toujours reconnue internationalement pour les caractères de 0 à 127 dont la table d'association est inscrite ci-dessous. La valeur des caractères supplémentaires étaient alors attribuée selon le système d'exploitation, DOS ou Windows.

Ctrl	Dec	Hex	Char	Code	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
^@	0	00		NUL	32	20	sp	64	40	Ⓔ	96	60	'
^A	1	01	☐	SOH	33	21	!	65	41	A	97	61	a
^B	2	02	Ⓚ	STX	34	22	"	66	42	B	98	62	b
^C	3	03	♥	ETX	35	23	#	67	43	C	99	63	c
^D	4	04	♦	EOI	36	24	\$	68	44	D	100	64	d
^E	5	05	♣	ENQ	37	25	%	69	45	E	101	65	e
^F	6	06	♠	ACK	38	26	&	70	46	F	102	66	f
^G	7	07	•	BEL	39	27	'	71	47	G	103	67	g
^H	8	08	◼	BS	40	28	(	72	48	H	104	68	h
^I	9	09	◊	HI	41	29	)	73	49	I	105	69	i
^J	10	0A	◻	LF	42	2A	*	74	4A	J	106	6A	j
^K	11	0B	♂	VI	43	2B	+	75	4B	K	107	6B	k
^L	12	0C	♀	FF	44	2C	,	76	4C	L	108	6C	l
^M	13	0D	♯	CR	45	2D	-	77	4D	M	109	6D	m
^N	14	0E	♯	SO	46	2E	.	78	4E	N	110	6E	n
^O	15	0F	⌘	SI	47	2F	/	79	4F	O	111	6F	o
^P	16	10	◀	SLE	48	30	0	80	50	P	112	70	p
^Q	17	11	▶	CS1	49	31	1	81	51	Q	113	71	q
^R	18	12	↕	DC2	50	32	2	82	52	R	114	72	r
^S	19	13	!!	DC3	51	33	3	83	53	S	115	73	s
^T	20	14	¶	DC4	52	34	4	84	54	T	116	74	t
^U	21	15	§	NAK	53	35	5	85	55	U	117	75	u
^V	22	16	■	SYN	54	36	6	86	56	V	118	76	v
^W	23	17	‡	EIB	55	37	7	87	57	W	119	77	w
^X	24	18	↑	CAN	56	38	8	88	58	X	120	78	x
^Y	25	19	↓	EM	57	39	9	89	59	Y	121	79	y
^Z	26	1A	→	SIB	58	3A	:	90	5A	Z	122	7A	z
^[	27	1B	←	ESC	59	3B	;	91	5B	[	123	7B	{
^\	28	1C	└	FS	60	3C	<	92	5C	\	124	7C	
]`	29	1D	┘	GS	61	3D	=	93	5D	]	125	7D	}
^^	30	1E	▲	RS	62	3E	>	94	5E	^	126	7E	~
_	31	1F	▼	US	63	3F	?	95	5F	_	127	7F	Δ <sup>†</sup>

Avec les années 90 et l'avènement de l'internet et l'internationalisation des communications informatiques, la codification ASCII des caractères s'est avérée désuète puisqu'elle ne savait pas intégrer les caractères des différents langages internationaux tels l'arabe et le chinois. Un consortium de fabricants mis alors au point la codification Unicode maintenant reconnue internationalement. Cette nouvelle codification code les caractères sur 16 bits et permet jusqu'à 65536 caractères différents. Visual Basic s'est adapté à cette nouvelle codification et certaines de ses fonctions natives de traitement des chaînes de caractères ont été adaptées afin de traiter cette nouvelle approche de la codification des caractères. Notez donc que l'ensemble des fonctions de traitement des chaînes de caractères de *VBScript* ont été prévues pour traiter des caractères ASCII codés sur 8 bits mais que certaines fonctions prévoient l'utilisation de caractères Unicode codés sur 16 bits. Ces dernières sont explicitement identifiées à l'aide d'un `w` majuscule (*pour Wide Unicode*).

Examinons ces fonctions de traitement des chaînes de caractères :

Fonction	Description	Exemple
Asc	Retourne la valeur ASCII d'un caractère spécifié. Si plusieurs caractères sont spécifiés, le premier seulement sera considéré.	Dim C C = Asc("A") ' C vaut 65
Chr	Retourne le caractère correspondant à la valeur ASCII spécifiée.	Dim C C = Chr(65) ' C vaut "A"
Filter	Retourne un tableau indexé à zéro contenant les sous-éléments de la chaîne de caractères spécifiée selon un critère de filtre.	Dim C1() As String Dim C2() As String C2 = Filter(C1, "Allo", True)
FormatCurrency	Retourne en format monétaire l'expression spécifiée.	Dim stArgent stArgent = FormatCurrency(12.25)
FormatDateTime	Retourne en format date l'expression spécifiée.	Dim stDate stDate = FormatDateTime(Now(), vbLongDate)
FormatNumber	Retourne en format numérique l'expression spécifiée.	Dim Prix, N N = 5.6 Prix = FormatNumber(N, 2) 'Retourne 5.60
FormatPercent	Retourne en pourcentage l'expression spécifiée où 1 vaut 100%.	Dim stPcnt stPcnt = FormatPercent(.85)
InStr	Retourne la position à laquelle une expression spécifiée est trouvée au sein d'une chaîne de caractère spécifiée. La fonction retourne 0 si l'expression n'a pu être trouvée.	Dim nPosition, St St = "Allo" nPosition = InStr(st, "o")
InStrRev	Retourne la position à laquelle une expression spécifiée est trouvée au sein d'une chaîne de caractère spécifiée en partant de la fin de celle-ci. La fonction retourne 0 si l'expression n'a pu être trouvée.	Dim nPos, St St = "Allo" nPos = InStrRev(st, "o")
LCase	Retourne en minuscules la chaîne de caractères spécifiée.	Dim St1, St2 St1 = "Une phrase" St2 = LCase(St1)



Left	Retourne les <i>n</i> premiers caractères d'une chaîne de caractères selon le nombre de caractères spécifié.	Dim St1, St2 St1 = "Une phrase" St2 = Left(St1, 3)
Len	Retourne le nombre de caractères constituant la chaîne de caractères spécifiée.	Dim nLength nLength = Len("Une phrase")
LTrim	Retourne une chaîne de caractères constituée d'une copie de la chaîne de caractères spécifiée sans espaces avant le premier caractère imprimable.	Dim St1, St2 St1 = " Une phrase" St2 = LTrim(St1)
Mid	Retourne la chaîne de caractères contenue à un endroit spécifié au sein d'une autre chaîne de caractères.	Dim St1, St2 As String St1 = "Bonjour" St2 = Mid(St1, 3, 4) 'Donne "njou"
Replace	Retourne une chaîne de caractères copiée d'une autre au sein de laquelle une expression est remplacée par une expression différente.	Dim St1, St2 As String St1 = "Bonjour" St2 = Replace(St1, "o", "?") 'Donne "B?nj?ur"
Right	Retourne les <i>n</i> derniers caractères d'une chaîne de caractères selon le nombre de caractères spécifié.	Dim St1, St2 St1 = "Une phrase" St2 = Right(St1, 3)
RTrim	Retourne une chaîne de caractères constituée d'une copie de la chaîne de caractères spécifiée sans espaces après le premier caractère imprimable.	Dim St1, St2 St1 = "Une phrase " St2 = RTrim(St1)
Space	Retourne une chaîne de caractères consistant au nombre d'espaces spécifiées.	Dim St As String St = Space(10) 'Donne " "
Split	Retourne un tableau uni-dimensionnel constitué des mots d'une chaîne de caractères spécifiée séparés par le caractère délimiteur spécifié (par défaut le caractère d'espacement)	Dim St1 As String Dim St2() As String St1 = "Look at these!" St2 = Split(St1) 'Donne ["Look", "at", "these!"]
StrComp	Retourne -1, 0, ou 1 selon le résultat de la comparaison des chaînes de caractères. Les chaînes sont comparées selon un tri alphanumérique.	Dim St1, St2 As String Dim C As Integer St1 = "ABCD" St2 = "abcd" C = StrComp(St1, St2) 'C donne 0
StrReverse	Retourne une chaîne de caractères résultant de l'inversion de l'ordre de tous les caractères de la chaîne spécifiée.	Dim St1, St2 As String St1 = "Allô" St2 = StrReverse(St1)
Trim	Retourne une chaîne de caractères constituée d'une copie de la chaîne de caractères spécifiée sans espaces avant et après le premier caractère imprimable.	Dim St1, St2 St1 = " Une phrase " St2 = Trim(St1)
UCase	Retourne en majuscules la chaîne de caractères spécifiée.	Dim St1, St2 St1 = "Une phrase" St2 = UCase(St1)

## Fonctions mathématiques

En plus des opérateurs arithmétiques de base *VBScript* dispose de plusieurs fonctions permettant d'effectuer des calculs mathématiques.

Fonction	Description	Exemple
Abs	Retourne la valeur absolue de l'expression numérique (sans le signe).	Abs (-2.5) ' Retourne 2.5
Atn	Retourne l'arc tangente en radians d'un angle exprimé par le ratio de deux côtés d'un triangle rectangle. Le ratio est la longueur du côté opposé à l'angle divisé par la longueur du côté adjacent à l'angle. Pour convertir les radians en degrés, multipliez les radians par 180/pi.	Dim PI As Double PI = 4 * Atn(1)
Cos	Retourne le cosinus en radians d'un angle exprimé par le ratio de deux côtés d'un triangle rectangle. Le ratio est la longueur du côté adjacent à l'angle divisé par la longueur de l'hypoténuse. Pour convertir les radians en degrés, multipliez les radians par 180/pi.	Dim Sect As Double Sect = 1 / Cos(1.3)
Exp	Retourne la valeur de $e^x$ pour un $x$ donné.	Exp(1) ' Retourne 2.718282
Int	Retourne la partie entière d'un nombre sans les décimales après arrondissement.	Int(-5.7) ' Retourne -6
Fix	Retourne la partie entière d'un nombre sans les décimales et sans arrondissement.	Fix(-5.7) ' Retourne -5
Log	Retourne le logarithme naturel d'un nombre. Le logarithme naturel, ou népérien, est le logarithme sur la base $e$ valant approximativement 2.718282.	Lg10 = Log(x) / Log(10#) ' Calcule le logarithme de 10
Randomize	Initialise le générateur de nombres aléatoires.	Randomize
Rnd	Retourne un nombre aléatoire situé entre 0 inclusivement et 1 exclusivement.	(Max - Min + 1) * Rnd() + Min
Round	Retourne un nombre arrondi à la précision spécifiée.	Dim Nbr As Single Nbr = Round(0.27182, 2) ' Retourne 0.27
Sgn	Retourne une valeur indiquant le signe du nombre spécifié. Cette valeur vaut -1 lorsque le nombre est négatif, 1 lorsque le nombre est positif et 0 lorsque le nombre est nul.	Sgn(8)      Retourne 1 Sgn(-3)     Retourne -1 Sgn(0)      Retourne 0
Sin	Retourne le sinus en radians d'un angle exprimé par le ratio de deux côtés d'un triangle rectangle. Le ratio est la longueur du côté opposé à l'angle divisé par la longueur de l'hypoténuse. Pour convertir les radians en degrés, multipliez les radians par 180/pi.	Dim CoSec As Double CoSec = 1 / Sin(1.3)
Sqr	Retourne la racine carrée d'un nombre.	Sqr(25) ' Retourne 5
Tan	Retourne la tangente en radians d'un angle exprimé par le ratio de deux côtés d'un triangle rectangle. Le ratio est la longueur du côté opposé à l'angle divisé par la longueur du côté adjacent à l'angle. Pour convertir les radians en degrés, multipliez les radians par 180/pi.	Dim CoTan As Double CoTan = 1 / Tan(1.3)

### Fonctions mathématiques dérivées

Plusieurs fonctions mathématiques ne sont pas intrinsèques à *VBScript* telles le calcul de la sécante ou le calcul d'un sinus hyperbolique. Cependant, ces fonctions mathématiques peuvent être connues à l'aide des fonctions mathématiques intrinsèques à *VBScript* :

Fonction		Équivalent
Cosécante	Csc(x)	1 / Sin(x)
Cotangente	Ctan(x)	1 / Tan(x)
Sécante	Sec(x)	1 / Cos(x)
Sinus inverse	Asin(x)	Atn(x / Sqr(-x * x + 1))
Cosinus inverse	Acos(x)	Atn(-x / Sqr(-x * x + 1)) + 2 * Atn(1)
Sécante inverse	Asec(x)	2 * Atn(1) - Atn(Sgn(x) / Sqr(x * x - 1))
Cosécante inverse	Acsc(x)	Atn(Sgn(x) / Sqr(x * x - 1))
Cotangente inverse	Acot(x)	2 * Atn(1) - Atn(x)
Sinus hyperbole	Sinh(x)	(Exp(x) - Exp(-x)) / 2
Cosinus hyperbole	Cosh(x)	(Exp(x) + Exp(-x)) / 2
Tangente hyperbole	Tanh(x)	(Exp(x) - Exp(-x)) / (Exp(x) + Exp(-x))
Sécante hyperbole	Sech(x)	2 / (Exp(x) + Exp(-x))
Cosécante hyperbole	Csch(x)	2 / (Exp(x) - Exp(-x))
Cotangente hyperbole	Coth(x)	(Exp(x) + Exp(-x)) / (Exp(x) - Exp(-x))

## Fonctions de manipulation des types de données

VBScript prévoit plusieurs fonctions permettant de manipuler les variables et leur type de données. Les fonctions dont le nom est préfixé de `Is` permettent toujours de retourner une valeur booléenne (`true` ou `false`) selon si l'expression spécifiée respecte les critères vérifiés par la fonction.

Fonction	Description	Exemple
Hex	Retourne la valeur hexadécimale de l'expression.	<code>N = Hex(32)</code> ' Retourne 20
IsArray	Retourne <code>True</code> si l'expression spécifiée est un tableau.	<code>If IsArray(x) Then</code>
IsDate	Retourne <code>True</code> si l'expression spécifiée est une date valide.	<code>If IsDate(x) Then</code>
IsEmpty	Retourne <code>True</code> si l'expression spécifiée est un type variant vide.	<code>If IsEmpty(v) Then</code>
IsNull	Retourne <code>True</code> si l'expression spécifiée est une valeur Nulle.	<code>If IsNull(RS("x")) Then</code>
IsNumeric	Retourne <code>True</code> si l'expression spécifiée est une valeur numérique valide.	<code>If IsNumeric(x) Then</code>
IsObject	Retourne <code>True</code> si l'expression spécifiée est un objet.	<code>If IsObject(x) Then</code>
Oct	Retourne la valeur octale de l'expression.	<code>N = Oct(32)</code> ' Retourne 40
TypeName	Retourne sous forme de chaîne de caractères le nom du type de données contenue au sein de l'expression spécifiée.	<code>MsgBox TypeName(4)</code> ' Affiche "Integer" <code>MsgBox TypeName("Allô")</code> ' Affiche "String"
VarType	Retourne une valeur numérique identifiant le type de données de l'expression spécifiée.	<code>Z = "Allô"</code> <code>If VarType(Z) =</code> ▼ <code>vbString Then</code>

La dernière fonction de la liste - `VarType` - identifiera le type de données de l'expression spécifiée en retournant l'une des constantes suivantes :

Constante	Valeur	Constante	Valeur
<code>vbEmpty</code>	0	<code>vbString</code>	8
<code>vbNull</code>	1	<code>vbObject</code>	9
<code>vbInteger</code>	2	<code>vbError</code>	10
<code>vbLong</code>	3	<code>vbBoolean</code>	11
<code>vbSingle</code>	4	<code>vbVariant</code>	12
<code>vbDouble</code>	5	<code>vbDataObject</code>	13
<code>vbCurrency</code>	6	<code>vbByte</code>	17
<code>vbDate</code>	7	<code>vbArray</code>	8192

## Fonctions de gestion des paramètres régionaux du système

VBScript sous *Windows Scripting Host* prévoit deux fonctions permettant de connaître et modifier les paramètres régionaux utilisés par l'utilisateur. Les différentes configurations possibles sont référencées par un identificateur unique (ID) dont les principaux sont listés plus bas.

Fonction	Description	Exemple
GetLocale	Retourne l'ID des paramètres régionaux courants.	Dim Org Org = GetLocale()
SetLocale	Précise le nouvel ID des paramètres régionaux et retourne l'ID des paramètres régionaux courants avant le changement.	Dim Org Org = SetLocale("fr-ca")

Voici une liste des identificateurs de paramètres régionaux les plus importants et certains beaucoup moins importants.

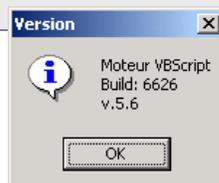
Abrv.	Valeur	Description	Abrv.	Valeur	Description
af	1078	Africain	fr-fr	1036	Français – France
ar-ae	14337	Arabe – Émirats unis	fr-be	2060	Français – Belgique
ar-sa	1025	Arabe – Arabie	fr-ca	3084	Français – Canada
eu	1069	Basque	fr-ch	4108	Français – Suisse
zh-cn	2052	Chinois – Chine	de-de	1031	Allemand – Allemagne
zh-hk	3076	Chinois – Hong Kong	de-at	3079	Allemand – Autriche
zh-sg	4100	Chinois – Singapour	de-ch	2055	Allemand – Suisse
zh-tw	1028	Chinois – Taiwan	el	1032	Grec
hr	1050	Croate	he	1037	Hébreu
cs	1029	Tchèque	hi	1081	Indou
en-au	3081	Anglais – Australie	it-it	1040	Italien – Italie
en-ca	4105	Anglais – Canada	ja	1041	Japonais
en-cb	9225	Anglais – Caraïbes	ko	1042	Coréen
en-ie	6153	Anglais – Irlande	pl	1045	Polonais
en-za	7177	Anglais – Afrique sud	pt-pt	2070	Portugais – Portugal
en-gb	2057	Anglais – Britannique	ru	1049	Russe
en-us	1033	Anglais – Etats-Unis	es-es	1034	Espagnol – Espagne

## Fonctions de contrôle du script et de son exécution

Fonction	Description	Exemple
Array	Génère un tableau à partir d'expressions spécifiées sous forme de liste.	Dim A, B A = Array(10, 15, 20) B = A(2) ' B vaut 20
CreateObject	Procède à la création d'un objet COM à l'aide du <i>ProgID</i> associé et retourne un pointeur sur l'objet ainsi créé.	Set Pres = CreateObject ▼ ("Powerpoint.Application") Pres.Visible = True
Eval	Évalue une expression sous forme de chaîne de caractères et en retourne le résultat.	If Eval("x = y") Then
Erase	Supprime tous les index du tableau spécifié.	Dim A A = Array(10, 15, 20) Erase A
GetObject	Retourne un pointeur sur un objet existant correspondant au <i>ProgID</i> ou au fichier spécifié.	Set Obj = ▼ GetObject("C:\feuil.xls") Obj.Visible = True
LBound	Retourne l'index inférieur du tableau spécifié.	A = Array(10, 15, 20) For X =LBound(A) To UBound(A) MsgBox A(X) Next
ScriptEngine	Retourne le nom du moteur de script utilisé au sein du script.	MsgBox ScriptEngine() ' Retourne VBScript
ScriptEngineBuildVersion	Retourne le numéro d'assemblage du moteur de script utilisé.	MsgBox ▼ ScriptEngineBuildVersion()
ScriptEngineMajorVersion	Retourne le numéro principal de la version du moteur de script utilisé.	MsgBox ▼ ScriptEngineMajorVersion()
ScriptEngineMinorVersion	Retourne le numéro secondaire de la version du moteur de script utilisé.	MsgBox ▼ ScriptEngineMinorVersion()
UBound	Retourne l'index supérieur du tableau spécifié.	Dim A A = Array(10, 15, 20) For X =LBound(A) To UBound(A) MsgBox A(X) Next

L'exemple suivant récupère et affiche différentes informations concernant le moteur de script actuel (*nom du moteur et numéro de version principalement*).

```
MsgBox "Moteur " & ScriptEngine() & vbCrLf & ▼  
"Build: " & ScriptEngineBuildVersion() & vbCrLf & ▼  
"v." & ScriptEngineMajorVersion() & "." & ▼  
ScriptEngineMinorVersion(), vbInformation, "Version"
```



CH03\Versions.vbs