

XOJO : Les Variables (CPointier)

INTEGER

```
Var x As Integer  
Var y As Integer  
Var Resultat As Integer
```

```
x = 23  
y = 45
```

```
Resultat = y + x
```

```
MessageBox.Show(Resultat.ToString)
```

```
// Resultat = 68 //
```

```
// Autres possibilities //
```

```
Var x As Int 32  
Var y As Int 64
```

```
// Autre Notation possible //
```

```
Var x, y, Resultat As Integer
```

DOUBLE

```
Var x As Double  
Var y As Double  
Var Resultat As Double
```

```
x = 5.3  
y = 10.2
```

```
Resultat = x + y
```

```
// Resultat = 15.5 //
```

```
Var x, y, Resultat As Double
```

```
// A la declaration la variable vaut 0. //
```

```
Var x As Double  
x = 5 / 2
```

```
Var x As Integer  
x = 5 \ 2
```

```
\ x = 2 \ rend la partie entière de la division !
```

```
Var x As Double  
x = 5 Mod 2
```

```
\ x = 1 reste de la division
```

BOOLEAN

Var Flag As Boolean

Flag = True

Flag = False

// A la declaration la variable booléenne vaut False. //

STRING

Var Name As String

Name = "Elvis Presley"

Name = "Elvis ""The King"" Presley"

Var MonAge As String

Var MonAgeNumerique As Integer

MonAge = 18

monAgeNumerique = MonAge.ToInteger

DATE TIME

Var Vacances As DateTime

Vacances = New DateTime(2025, 5, 25)

MessageDialog.Show(Vacances.Year.ToString)

Var Aujourdhui As DateTime = DateTime.Now

Var UnJour As New DateTime(1944, 6, 6)

Var UnJour As New DateTime(1944, 6, 6, 10, 0, 0)

Var Aujourdhui As DateTime = DateTime.Now

Var CeMois As Integer

CeMois = Aujourdhui.Month

Janvier = 1, Février = 2, Mars = 3, Avril = 4, Mai = 5, Juin = 6

Juillet = 7, Aout = 8, Septembre = 9, Octobre = 10, Novembre = 11, Décembre = 12

// Une DateTime possède des propriétés accessibles en lecture seulement : //

DayOfWeek (Integer) : de 1 à 7 pour Dimanche à Samedi.

DayOfYear (Integer) :

WeekOfYear (Integer) :

// DateTime ToString renvoie la DateTime sous String. //

// DateTime peut recevoir plus de 7 paramètres : //

Year (Integer), Month (Integer), Day (Integer), Hour (Integer), Minute (Integer), Second (Integer)

Nanosecond (Integer), TimeZone

COLOR

Var MaCouleur As Color

MaCouleur = Color.RGB(255, 255, 255)

Var MaCouleur As Color

If Color.SelectedFromDialog(MaCouleur, "Choisir une Couleur : ") Then

End If

// La fonction retourne un Boolean si selection et MaCouleur la couleur sélectionnée //

Exemples

Var President As String

President = "Abraham" + " " + "Lincoln"

// President = Abraham Lincoln //

Var Message As String

Message = "Bonjour !"

MessageDialog.Show(Message)

// MessageDialog.Show ne peut pas accepter integer ou Double, il faut utiliser la fonction ToString //

Var MonAgeNumerique As Integer

Var MonAgeChaine As String

MonAgeNumerique = 16

MonAgeChaine = MonAgeNumerique.ToString

MessageDialog.Show(monAgeChaine)

Spécification de format :

Décrit comment le nombre doit être affiché, par exemple pour les % on utilisera #%

Var MonPourcentage As Double

MonPourcentage = 0.25

MessageBox.Show(monPourcentage.ToString("#%"))

Le # représente le nombre à afficher.

Character	Description
#	Espace réservé qui affiche le chiffre de la valeur s'il est présent. Si moins d'espaces réservés sont utilisés que dans le nombre passé, le résultat est arrondi.
0	Espace réservé qui affiche le chiffre de la valeur s'il est présent. Si aucun chiffre n'est présent, 0 (zéro) s'affiche à sa place.
.	Espace réservé pour la position du point décimal.
,	Espace réservé indiquant que le nombre doit être formaté avec des séparateurs de milliers.
%	Affiche le nombre multiplié par 100 suivi du caractère %.
+	Affiche le signe plus à gauche du nombre si le nombre est positif ou un signe moins si le nombre est négatif.
-	Affiche le signe moins à gauche du nombre si celui-ci est négatif. Aucun effet pour les nombres positifs.
E or e	Affiche le nombre en notation scientifique.

Var MonPrix As Double

MonPrix = 24995.9

MessageBox.Show(monPrix.ToString("###,###.00"))

// Le nombre affiché est : 24,995.90 //

Format	Number	Formatted String
###	1,786	1,79
#.0000	1,3	1,3000
0000	5	0005
#%	0,25	25 %
###,###.##	145678,5	145.678,5
###e	145678,5	1.46e+05
-###	-3,7	-3,7
+###	3,7	+3.7

On peut également convertir **String** en **Integer** ou **Double** avec les fonctions **ToInteger** ou **ToDouble**.

Var MonAge **As String**

Var MonAgeNumerique **As Integer**

MonAge = "18"

MonAgeNumerique = MonAge.**ToInteger**