# Visual Basic

## [Stockage d'informations dans des variables](javascript:void(0))

L'utilisation d'une variable comprend trois étapes :

1. **Déclarez la variable.** Informez le programme du nom et de la nature de la variable que vous souhaitez utiliser.
2. **Affectez la variable.** Attribuez à la variable une valeur qu'elle stockera.
3. **Utilisez la variable.** pour récupérer la valeur contenue dans la variable et l'utiliser dans votre programme.

Déclarez variable

Dim aNumber As Integer

Dim aDouble As Double

Dim aName As String

Dim YesOrNo As Boolean

## [Affectation de variables](javascript:void(0))

Vous affectez une valeur à votre variable avec le signe =

aNumber = 42

## [Déclaration et affectation de variables avec une valeur par défaut](javascript:void(0))

Dim aDouble As Double = 0

Dim aName As String = "default string"

Dim YesOrNo As Boolean = True

## Concatenation

Dim aString As String = "using string"

Dim bString As String = "variables"

Dim cString As String = ""

cString = aString & " " & bString

## Tableaux : variables représentant plusieurs valeurs

Dim players() As String = {"Dan", "Fred", "Bart", "Carlos", \_

"Ty", "Juan", "Jay", "Sam", "Pedro"}

Dim i As Integer = CInt(Textbox1.Text)

MsgBox(players(i) & " is on first base.")

## Arithmétique : création d'expressions avec des variables et opérateurs

Dim anInteger As Integer = 5 + 4

## [Conversion de variables en texte](javascript:void(0))

Dim anInteger As Integer = 54

MsgBox(CStr(anInteger))

Ce code déclare une variable Integer nommée anInteger, lui affecte la valeur 54, convertit cette valeur en texte et l'affiche dans un message en appelant la fonction CStr.

## Comparaisons : utilisation d'expressions pour comparer des valeurs

Dim A As Double = CDbl(Textbox1.Text)

Dim B As Double = CDbl(Textbox2.Text)

MsgBox(A > B)

MsgBox(A < B)

MsgBox(A = B)

## Procédure

Function GetTheNumberOne() As Integer

Return 1

End Function

(retourne quelquechose)

Sub PlaySound(ByVal SoundFile As String, ByVal Volume As Integer)

My.Computer.Audio.Play(SoundFile, Volume)

End Sub

(Ne retourne pas forcement quelque chose)

## Pour la répétition d'actions : exécution d'une boucle avec la boucle For...Next (Pour…Prochain)

Dim i As Integer = 0

For i = 1 To 10

DisplayNumber(i)

Next

## Utilisation de Do...While et Do...Until (Faire jusqu’à)

Dim sum As Integer = 0

Dim counter As Integer = 0

Do While sum < 100

sum = sum + CInt(Textbox1.Text)

counter = counter + 1

Loop

MsgBox("The loop has run " & CStr(counter) & " times!")

Dim sum As Integer = 0

Do Until sum >= 100

sum = sum + 10

Loop

## If...Then/Else (si… Donc/Sinon)

If My.Computer.Clock.LocalTime.DayOfWeek = DayOfWeek.Monday Then

MsgBox("Today is Monday!")

End If

If My.Computer.Clock.LocalTime.DayOfWeek = DayOfWeek.Friday Then

MsgBox("Today is Friday!")

Else

MsgBox("It isn't Friday yet!")

End If

## [Bloc Try...Catch...Finally](javascript:void(0)) (essai ... sinon Catch (exécute ca)… finalement

Try

' Code here attempts to do something.

Catch

' If an error occurs, code here will run.

Finally

' Code in this block will always run.

End Try

En premier lieu, le code du bloc Try est exécuté. S'il s'exécute sans erreur, le programme passe au bloc Catch et exécute le code dans le bloc Finally. Si une erreur se produit dans le bloc Try, l'exécution passe immédiatement au bloc Catch. Le code présent dans ce bloc est alors exécuté. Ensuite, le code du bloc Finally est exécuté.