

# LES STRUCTURES CONDITIONNELLES

## 1. La structure If

Les instructions à exécuter peuvent dépendre d'une (ou plusieurs) condition (s). Il faut alors utiliser une structure décisionnelle qui oriente le déroulement du programme vers des blocs d'instructions déterminés. C'est la structure **If ... Then ... Else ... End**.

- Syntaxe sur une seule ligne : **If** condition **Then** instruction1 **Else** instruction2  
condition : expression dont le résultat est vrai ou faux (**True** ou **False**) Else instruction2 :  
facultatif

### Exemple :

```
If Moyenne >=10 Then Résultat = "Admis" Else Résultat = "Ajourné" If  
Moyenne >=10 Then Résultat = "Admis"
```

- Syntaxe sous forme de bloc  
**If** condition **Then**  
Bloc d'instruction1  
**Else**  
Bloc d'instruction2 **End If**

### Exemples :

- **If** Moyenne >=10 **Then**  
Admis = Admis + 1  
MsgBox ("Etudiant Admis")  
**Else**  
Ajourné = Ajourné + 1  
MsgBox ("Etudiant Ajourné")  
**End If**
- **If** Moyenne >=10 **Then**  
Admis = Admis + 1
  - **If** Moyenne > 14 **Then**  
MsgBox ("Etudiant Admis avec mention")

**Else**

MsgBox ("Etudiant Admis")

**End If**

**Else**

Ajourné = Ajourné + 1

MsgBox ("Etudiant Ajourné")

**End If**

## 2. La structure Case

Au-delà de 3 possibilités on a besoin d'une autre structure qui peut gérer plusieurs cas. C'est la structure **Select ... Case ...**

C'est une extension du **If**. Elle permet une programmation plus claire en évitant une trop grande imbrication de If successifs.

### Syntaxe :

**Select Case** Expression

**Case** ListeValeurs1

Bloc d'instruction1

**Case** ListeValeurs2 Bloc d'instruction2

.....

**[Case Else**

Bloc d'instruction N]

**End Select**

Expression : ou variable ListeValeurs : peut-être :

- une suite de valeurs : 1, 3, 5, 7, 9
- une fourchette : 0 To 9
- une plage : Is >= 10

Case Else

Bloc d'instruction N : facultatif

Une seule Expression (ou Variable) est testée au début, puis est comparée avec les listes de valeurs. A la première concordance, le bloc d'instruction correspondant est exécuté, puis le programme sort de la structure.

Si aucune concordance n'est trouvée les instructions placées après le Else sont exécutées.

Exemples :

**Select Case** CodeASCIICaractère

**Case** 65, 69, 73, 79, 85

MsgBox(" C'est une voyelle ")

**Case** 66 To 90

MsgBox(" C'est une consonne ")

**Case Else**

MsgBox(" Ce n'est pas une lettre ")

**End Select**

### 3. La structure If

C'est exactement la fonction IF d'EXCEL.

Syntaxe :

If (Condition, ValeurSiVrai, ValeurSiFaux)

Exemple :

Dim Note As Single

Dim Réponse As String

Note = InputBox (" Tapez votre note ")

Réponse = If (Note >= 10, " Admis ", " Ajourné ")

MsgBox (Réponse)

## LES COMPLEMENTS SUR MSGBOX

Lorsque plusieurs boutons de commande sont affichés par la MsgBox, il est possible de faire un traitement en fonction du bouton de commande choisi. Pour cela, il est nécessaire d'utiliser une variable qui recevra le résultat du clic.

### Exemples :

```
Dim REP As String
REP = MsgBox("Confirmation ?", vbYesNo)

If REP = vbYes Then
    MsgBox ("Traitement OUI")
Else
    MsgBox ("Traitement NON")
End If

Dim REP As String
REP = MsgBox("Confirmation ?", vbYesNoCancel)

Select Case REP
    Case vbYes
        MsgBox ("Traitement OUI")
    Case vbNo
        MsgBox ("Traitement NON")
    Case vbCancel
        MsgBox ("Traitement ANNULE")
End
```

## **LES BOUCLES**

### **1. La structure itérative**

Le programme va exécuter un bloc d'instructions de façon indéfinie. L'arrêt se fera lorsque la condition ne sera plus vérifiée.

#### **Syntaxe :**

**While** condition

Bloc d'instructions

**Wend**

La condition est testée au début, c'est à dire à l'entrée de la boucle.

*Cours VB.Net*

*Option : Génie Logiciel, Niveau 2*

*Formateur : **TSIAZE KENFACK** Zephirin*

La boucle sera répétée tant que la condition est vraie.

Si la condition n'est pas vraie au départ, le bloc d'instructions ne sera pas exécutée.

Exemple : Mot = InputBox("Donnez votre mot de passe")

**While** Mot <> MotDePasse

Mot = InputBox("Donnez votre mot de passe")

**Wend**

## 2. La structure répétitive

Le programme va exécuter un bloc d'instructions de façon indéfinie. L'arrêt se fera lorsque la condition sera vérifiée.

**Syntaxe :**

**Do**

Bloc d'instructions

**Loop** Until condition

La condition est testée après avoir exécuté une fois le bloc d'instruction.

La boucle sera répétée tant que la condition est fausse.

Exemple :

**Do**

Mot = InputBox("Donnez votre mot de passe")

**Loop** Until Mot = MotDePasse

## 3. La structure Pour

Le programme va exécuter un bloc d'instructions de façon définie. L'arrêt se fera lorsque la valeur finale sera atteinte.

**Syntaxe :**

**For** variable = valeur\_initial To valeur\_finale [Step incrémentation] Bloc d'instructions

**Next**

La variable est incrémentée à chaque fin de boucle du nombre indiqué par l'incrément. Si l'incrément n'est pas spécifié, il est fixé à 1. Si la valeur finale est inférieure à la valeur initiale, l'incrément est négatif.

Exemple :

**For** Ind = 1 to 25

    MsgBox(Ind)

**Next**

## **Procédures**

Ne retourne pas de valeur

**Syntaxe :**

**Sub** Nom\_procedure([BYVAL] variable as Type,[BYREF] variable As Type)

    déclarations

    instructions

**End sub**

**BYVAL** : paramètre est passé par valeur

**BYREF** : paramètre est passé par adresse ou référence

## **Fonctions**

La fonction renvoie une valeur utilisable

**Syntaxe :**

**Function** Nom\_fonction([BYVAL] variable as Type,[BYREF] variable As Type)

    déclarations

    instructions

    Return variable\_calculée

**End Function**

## Appel des procédures et fonctions

L'appel se fait simplement par son nom suivi de la liste des paramètres effectifs

- Nom\_procedure (valeur1,valeur2,..)
- Variable = Nom\_fonction(valeur1,valeur2,..)

## Portée des variables

La portée des variables diffère selon le lieu de déclaration.

On distingue trois possibilités :

- Niveau Procédure
- Niveau Formulaire
- Niveau module