

# Programmation Impérative

## Structures itératives

# Contenu

- Introduction
- Structure For .... Next
- Structure While ... Wend
- Structure Do ... Loop While
- Structure Do ... Loop Until
- Structure Do While .... Loop
- Structure Do Until ... Loop

# Introduction

- Ce document introduit les concepts suivants :
  - Boucle `For ... Next` pour exécuter des instructions en les répétant un certain nombre de fois ;
  - Boucle `Do ... Loop` pour exécuter des instructions jusqu'à ce (Tant) qu'une condition spécifique soit remplie ;

# Boucle For...Next

- Elle permet d'exécuter de manière répétitive et un nombre de fois défini à l'avance un groupe spécifique d'instructions de programmation.
- Elle évite d'avoir à rédiger une longue liste d'instructions de programmation.

# Syntaxe de la boucle For...Next

For variable = debut To fin

Liste d'instructions à exécuter de manière répétitive

Next [variable]

- variable : variable numérique (de compteur) qui enregistre le nombre d'itérations; elle doit être déclarée avant de l'utiliser.

# Exemple

```
Dim i As Integer  
For i = 1 To 4  
    Beep()  
Next i
```

Cette boucle est équivalente aux instructions suivants:

```
Beep()  
Beep()  
Beep()  
Beep()
```

À chaque exécution de la boucle, la variable de compteur *i* est incrémentée de 1. Au début *i* est égale à 1 et à la fin elle est égale à 4.

## Exemple 2 : Instruction Exit For

```
Dim i As Integer
```

```
Dim InpName As String
```

```
For i = 1 to 5
```

```
    InpName = InputBox(``Entrer un nom ou saisissez Fin pour arrêter``)
```

```
    If InpName = ``Fin`` Then Exit For
```

```
    MsgBox(``Vous avez saisi`` & InpName)
```

```
Next i
```

- ➡ Si l'utilisateur entre `Fin`, l'instruction `Exit For` termine la boucle et l'exécution reprend après l'instruction `Next`.

# Boucles For...Next complexes

Il est possible d'indiquer l'intervalle d'incrémentation en utilisant le mot-clé `Step`

```
Dim i As integer
```

```
For i = 5 to 25 Step 5
```

```
    Console.WriteLine(" la valeur de i est : " & i )
```

```
Next i
```



# Utilisation de `Step` avec des valeurs décimales

```
Dim i As Single
```

```
For i = 1 to 2.5 Step 0.5
```

```
    If i <> 2.5 then MsgBox (" la valeur de i est : " & Cstr(i))
```

```
    Else MsgBox ("La boucle se termine")
```

```
End If
```

```
Next i
```

# Boucle Do...Loop

- La boucle Do...Loop est intéressante lorsqu'on ne sait pas à l'avance combien de fois la boucle doit être répétée.
- Elle exécute un groupe d'instructions jusqu'à ce qu'une condition donnée spécifique soit remplie (Vraie).

# Syntaxe: boucles Do...Loop

- Les boucles Do...Loop peuvent se présenter de plusieurs façons, en fonction de la manière dont la condition est évaluée et de l'endroit où elle apparaît.

- La syntaxe la plus courante est :

Do While condition

*Bloc d'instructions à exécuter*

Loop

# Exemple 1

```
Dim InpName As String
Do While InpName <> ``Fin``
    InpName = InputBox(``Entrer un nom ou saisissez Fin pour arrêter``)
    If InpName <> ``Fin`` Then
        MsgBox(``Vous avez saisi`` & InpName)
    Loop
```

- Boucler tant que la variable InpName ne contient pas le mot Fin
- Si la condition placée en tête de la boucle n'a pas la valeur True quand cette expression logique (condition) est évaluée pour la première fois, la boucle Do...Loop n'est jamais exécutée.

## Exemple 2

```
Dim InpName As String
```

```
Do
```

```
    InpName = InputBox(``Entrer un nom ou saisissez Fin pour arrêter``)
```

```
    If InpName <> ``Fin`` Then
```

```
        MsgBox(``Vous avez saisi : `` & InpName)
```

```
Loop While InpName <> ``Fin``
```

- La boucle s'exécute au moins une fois. La condition est testée après l'entrée d'un nom à l'aide de la fonction InputBox.

# Boucle infini

```
Dim Number as Double
```

```
Do
```

```
    Number = InputBox("`Entrez un nombre à passer au carré. Saisissez -1  
pour arrêter.`")
```

```
    Number = Number ^ 2
```

```
    MsgBox("`Le carré est : `" & Cstr(Number))
```

```
Loop While Number >= 0
```

# Exemple : Convertisseur des températures

```
Dim FTemp, Celsius As Single
```

```
Dim StrTemp As String
```

```
Dim Prompt As String = "Saisissez une température en degrés  
Fahrenheit."
```

```
Do
```

```
    StrFTemp = InputBox(Prompt, "Fahrenheit vers Celsius ")
```

```
    If strFTemp <> " " Then                `Caractère vide
```

```
        FTemp = CSng(strFTemp)
```

```
        Celsius = (FTemp - 32) * 5 / 9
```

```
        MsgBox(CStr(Celsius), , "Température en degrés Celsius ")
```

```
    End If
```

```
Loop While strFTemp <> " "
```

# Utilisation du mot clé **Until** dans les boucles **Do...Loop**

```
Dim InpName As String
```

```
Do
```

```
    InpName = InputBox(``Entrer un nom ou saisissez Fin pour arrêter``)
```

```
    If InpName <> ``Fin`` Then
```

```
        MsgBox(``Vous avez saisi : `` & InpName)
```

```
Loop Until InpName = ``Fin``
```

➡ La boucle s'exécute jusqu'à ce que la condition donnée soit True.



# Utilisation du mot clé `Until` dans les boucles `Do...Loop`

Un `Do Until...Loop` est une boucle qui itère tant que sa condition de sortie est fausse (comme `Do Until...Loop`), cette dernière est testée à chaque début de boucle, donc le programme n'y passe pas forcément.

```
Dim Entier1 As Integer = 4
```

```
Do Until Entier1 > 3
```

```
    Entier1 = Entier1 + 1
```

```
Loop
```

```
MsgBox(Entier1) ' 4
```