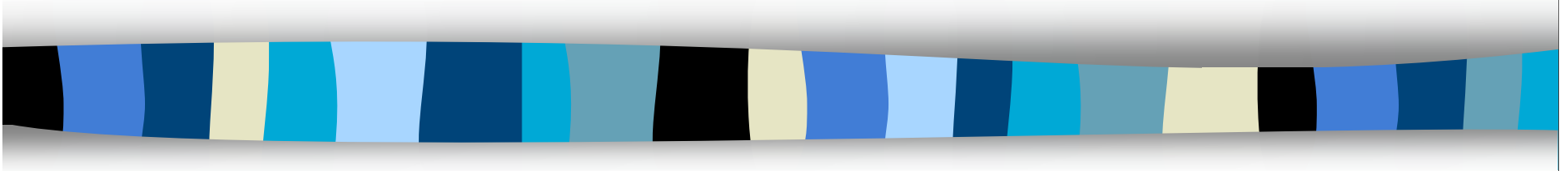


# Cours Visual Basic



URCA

[MCours.com](http://MCours.com)



# Sommaire

- Introduction à la programmation
- VBA et VB
- Syntaxe de base, variables
- Opérateurs de base, boucles



# Introduction à la programmation

- Qu'est-ce que la programmation ?
  - Séquences d'instructions qui permettent de donner des ordres à un ordinateur
- A quoi cela sert-elle ?
  - Développer des applications, automatiser des tâches récurrentes, effectuer des calculs complexes
- Quelques langages de programmation
  - C, C++, JAVA, (Visual) Basic, Perl, PHP,



# Introduction: suite...

- Différences principales entre les langages
  - La syntaxe
  - Langages séquentiels, langages orienté-objet
  - Langages interprétés, langages exécutés
  - Langages spécialement adaptés
    - A Internet (pages web): PHP, Perl, JAVA
    - Aux simulations: Matlab



# VBA ou VB ou Visual Basic

- La terminologie autour de Visual Basic peut mener à une certaine ambiguïté.
- VBA = Visual Basic pour Application
  - Disponible avec MS Office (Excel, Access, Word)
- VB = Visual Basic
  - Langage de programmation contenu dans MS Office mais également intégré dans Visual Studio pour faire des applications indépendantes



# Visual Basic en général

## ■ Avantages

- Simplicité
- Disponible dans MS Office
- Accès à toutes les fonctions de Windows
  - Systèmes de fenêtres, boîtes de dialogue, API

## ■ Inconvénients

- Pas très rapide
- Dépend beaucoup des versions utilisées de Windows et MS Office
- Cher si on veut être en règle



# Visual Basic

- Exemples
- Syntaxe
- Variables



## Pratique: 1er exemple

```
Public Sub afficher_texte()  
    Dim i As Integer  
    For i=1 To 2  
        MsgBox "Hello World !"  
    Next i  
End Sub
```





# Exemples avec d'autres langages

## JAVA

```
class Exemple{
    public static void main(String
args[]){
        int i = 0;
        for(i=0; i<2; i++){
            System.out.println("Hello World !");
        }
    }
}
```

## C

```
int main(int argc, char *
argv[]){
    int i = 0;
    for(i=0; i<2; i++){
        printf("Hello World
!\n");
    }
}
```

# Syntaxe de base

- Il faut toujours indiquer où se trouve le début et la fin du programme que l'on écrit.

```
Public Sub nom_du_programme()  
    ' séquences d'instructions  
End Sub
```

Indique le  
début

Indique la fin



# Variables (1)

- Les variables sont essentielles à tous les programmes et tous les langages
- Elles permettent le stockage à court terme des paramètres du programme. Souvent des valeurs numériques (5, 17.6, -3.121212) ou des caractères
- Plusieurs types de variables
- Correspondent à des zones de mémoire de l'ordinateur



## Variables (2)

- En Visual Basic, on considère env. 12 types de variables. Les plus utilisées:
  - **String** pour stocker des chaînes de caractères
  - **Integer** pour stocker des valeurs entières
  - **Double** pour stocker des valeurs décimales
  - **Long** pour stocker des grandes valeurs entières
  - **Boolean** pour stocker soit un 0 soit un 1 (un bit)



## Variables (3)

- En VB, il n'est pas obligatoire de déclarer les variables, mais vivement conseillé

- Syntaxe de déclaration:

```
Public Sub prog_qui_fait_rien()
```

```
    Dim age As Integer
```

```
    Dim nom As String
```

```
    Dim revenu As Long
```

```
End Sub
```



## Variables (4)

- Les variables sont très importantes et bien comprendre leur utilité
- Ne jamais hésiter à déclarer une nouvelle variable
- Autre syntaxe possible:

```
Public Sub prog_qui_fait_rien()  
    Dim age As Integer, nom As  
        String  
End Sub
```

# Opérateurs et variables (1)

- Nombreux opérateurs: + \* / - & Mod \ ^
- Les opérateurs font des opérations avec des variables

```
Public Sub mon_prog ()
```

```
Dim a As Integer, b As Integer, c As Integer, d As Integer
```

```
a = 5
```

```
b = 7
```

```
c = a + b
```

```
d = a * b
```

```
End Sub
```

*c vaudra 12*

*d vaudra 35*

# Opérateurs et variables (2)

## ■ Autres exemples avec des opérateurs:

```
Public Sub mon_prog ()  
Dim a As Integer, b As Integer, c As Integer, d As Integer  
a = 5  
b = 7  
a ← a + b  
d = a * b  
End Sub
```

*a sera modifié et vaudra 12*

*d vaudra 84*



# Opérateurs et variables (3)

## ■ Importance des types

```
Public Sub mon_prog ()  
  Dim a As Integer, b As Integer, d As  
  Integer  
  a = 9  
  b = 2  
  d = a / b  
End Sub
```

*d vaudra 4*

```
Public Sub mon_prog ()  
  Dim a As Integer, b As Integer, d As  
  Double  
  a = 9  
  b = 2  
  d = a / b  
End Sub
```

*d vaudra 4.5*



# Opérateurs et variables (4)

- Opérateurs Mod (division entière) et & (concaténation)
- Exemples avec Mod:
  - $34 \text{ Mod } 12 = 10$  car  $2 \times 12 + 10 = 34$
  - $5 \text{ Mod } 2 = 1$  car  $2 \times 2 + 1 = 5$
- Exemples avec &
  - "Jean" & "Von Duren" = "Jean Von Duren"



# Instructions conditionnelles (1)

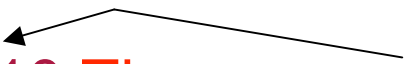
- Un programme doit pouvoir s'adapter à ses paramètres.
- En fonction de certaines valeurs, plutôt faire ça, ou plutôt faire autre chose...
- On utilise le **If** (en français Si)
- Sémantique:
  - **Si** « condition est vérifiée » **Alors**
    - Faire qqch de particulier
  - **Autrement**
    - Faire autre chose...

# Instructions conditionnelles (2)

## ■ Syntaxe en VB:

```
Dim a As Integer, b As Integer
a = 5
If a < 10 Then
    b = 1
Else
    b = 2
End If
```

Condition





# Instructions conditionnelles (3)

## ■ Autres syntaxes:

```
Dim a As Integer, b As Integer  
a = 5  
If a < 10 Then  
    b = 1  
Else If a < 20 Then  
    b = 2  
Else  
    b = 3  
End If
```



# Boucles (1)

- Une boucle est un ensemble d'instructions qui est répété un certain nombre de fois
- S'il l'on veut faire un compteur qui compte jusqu'à 10000, alors c'est la seule solution
- Plusieurs syntaxes possibles
  - For, While, Loop, Do While, ...
- Les plus utilisées: **For** et **While**



## Boucles (2) - For

### ■ Compteur jusqu'à 10000:

Dim **a** As Integer

**For** a = 0 **To** 10000

... Tout ce qui se trouve ici est  
répété 10001 fois

**Next** a

- Sémantique:

**Pour** a valant **De** 0 **A** 10000

Fais qqch



## Boucles (3) - While

- Différence avec **For**: on ne connaît pas toujours la fin de la boucle à l'avance

```
Dim a As Integer
```

```
a = 0
```

```
While a < 10000
```

```
    ... Tout ce qui se trouve ici
```

```
est
```

```
    répété 10000 fois
```

```
    a = a + 1
```

```
Wend
```