

# Learning the VBA language by practicing

## Part A: Variables, functions and instructions

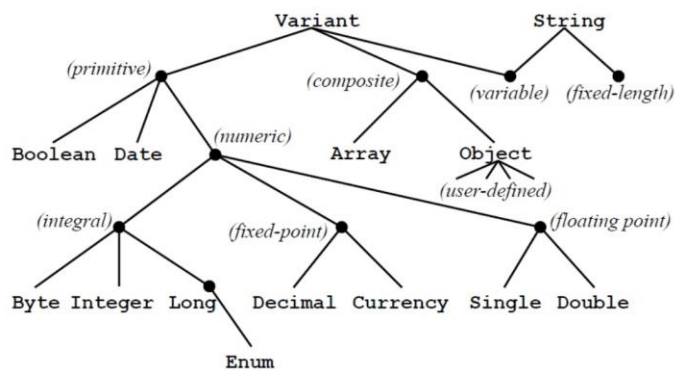
### 1. Declaration of variables

To declare a variable, you use:

```
Dim nomVariable As type
```

The following tree summarizes the types supported by the vba language:

#### Types



Examples:

a. `Dim v As double`

b. `Dim x As Integer`

```
x = 5.5
```

```
MsgBox "value is " & x
```

c. `Dim x As Double`

```
x = 5.5
```

```
MsgBox "value is " & x
```

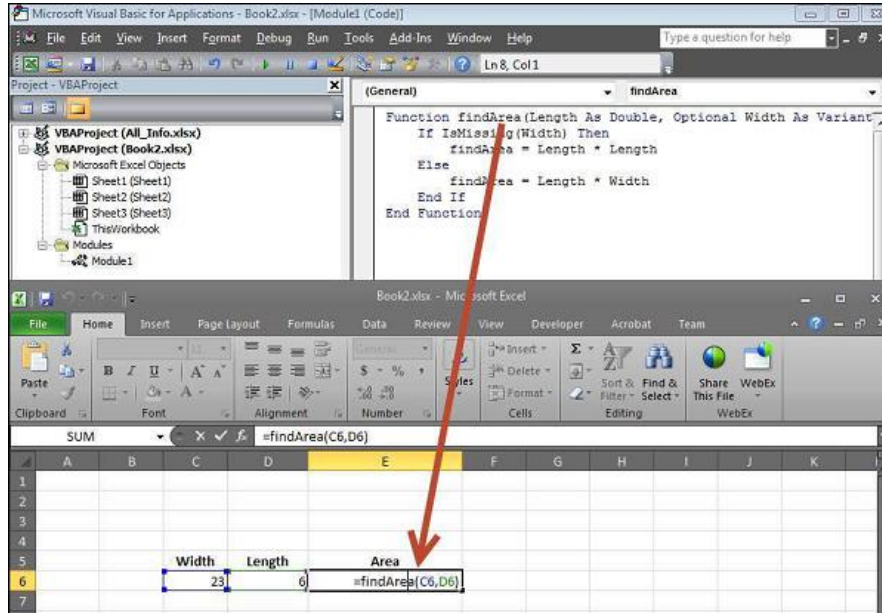
d. `Dim continue As Boolean`

```
continue = True
```

```
If continue = True Then MsgBox "Boolean variables are cool"
```

## 2. Functions

The below figure shows how to define a VBA function. Observe that a VBA function can have an optional return statement. This is required if you want to return a value from a function.



### Exercise:

Write a VBA function `priceTot` which computes the total price by considering the price and tax. Consider a tax rate equals to 1.22.

## 3. Decision

### 3.1. Statement IF

```
If condition Then
    bloc d'instructions
    si la condition est vraie
Else
    bloc d'instructions
    si la condition est fausse
End If
```

### Exercise:

Consider again the `priceTot` function and suppose that it accepts two parameters: price and category. The category may be either "ordinary" or "luxury". For the former, the tax rate is 1.22 and for the latter is 1.60.

### 3.2. Statement Select

```
b.1.  
Select Case variable  
  Case valeur 1  
    bloc d'instructions  
  Case valeur 2  
    bloc d'instructions  
  ...  
  Case Else  
    bloc d'instructions  
End Select
```

#### Exercise:

Consider again the priceTot function and use the *select statement* instead of *if statement*.

### 3.3. Statement Select with ranges of values

It is possible to introduce ranges of values in the box section of the Select Case structure. The comparison becomes more sophisticated. Variable is a numerical in this case, integer or even real.

```
Select Case variable  
  Case Is op.de.comparaison valeur  
    bloc d'instructions  
  
  Case valeur de départ To valeur de fin  
    bloc d'instructions  
  
  Case Else  
    bloc d'instructions  
End Select
```

#### Example:

```
'calcul du prix unitaire en fonction de la quantité  
Public Function MonPU(quantite As Long) As Double  
  'variable intermédiaire  
  Dim pu As Double  
  'selon les valeurs de quantité  
  Select Case quantite  
    Case Is < 100  
      pu = 0.5  
    Case 100 To 200  
      pu = 0.3  
    Case Is > 200 'Case Else aurait fait l'affaire aussi  
      pu = 0.2  
  End Select  
  MonPU = pu  
End Function
```

#### 4. For statement

```
For indice = val.départ to val.fin step pas  
    bloc d'instructions  
    ...  
Next indice
```

#### Exercise:

Write a function which computes:

$$s = \sum_{i=1}^n i * i$$

#### 5. While statement

```
Do While condition  
    Bloc d'instructions...  
    ...  
Loop
```

#### Exercise:

Consider again the previous function, replace for statement by while statement.

#### 6. Do statement

```
Do  
    Bloc d'instructions  
    ...  
    ...  
Loop While condition
```

### Exercise:

Consider again the previous function, replace for statement by do statement.

### Exercise:

Write a text corresponding to the code given below.

Entrée : plage (range)  
Sortie : S (réel)  
Calcul : Somme des valeurs

```
'Travail sur le type Range
Public Function MaSommeRange(plage As Range) As Double
'variables intermédiaires
Dim s As Double, i As Long, j As Long
'initialisation de la somme
s = 0
'parcours de la plage de cellules
For i = 1 To plage.Rows.Count Step 1 'lignes
  For j = 1 To plage.Columns.Count Step 1 'colonnes
    'lecture des valeurs et somme
    s = s + plage.Cells(i, j).Value
  Next j
Next i
'renvoyer le résultat
MaSommeRange = s
End Function
```

Lignes et colonnes commencent à l'indice 1, quelle que soit la position de la plage dans la feuille.

Nombre de lignes de la plage de cellules.

Nombre de colonnes.

Accès à la valeur (Value) de la cellule : ligne n°i, colonne n°j

### Exercise:

Write a text corresponding to the code given below.

```
'Travail sur le type Range avec un For Each
Public Function MaSommeRangeEach(plage As Range) As Double
'variables intermédiaires
Dim s As Double, cellule As Range
'initialisation de la somme
s = 0
'parcours de la plage de cellules
For Each cellule In plage
  s = s + cellule.Value
Next cellule
'renvoyer le résultat
MaSommeRangeEach = s
End Function
```

Une cellule est une plage de cellules avec une seule cellule.

plage fait figure de collection à traiter. Qu'importe le sens du parcours ici (ligne par ligne, ou colonne par colonne).

C'est bien la valeur contenue dans la cellule qui est exploitée pour la somme.