Développement VBA Excel

VBA

Principe

- VBA (Visual Basic for Applications) est un langage de programmation adapté aux produits de la suite Microsoft Office.

- Ce langage permet d'étendre les fonctionnalités de base du logiciel

- Le code est directement intégré au classeur Excel auquel il est rattaché et l'éditeur se trouve dans Excel

Langage Objet ?

- Un langage objet est un langage de programmation construit autour d'un concept : l'objet.

- Un objet est défini par:
  - Des propriétés : il s'agit de ses caractéristiques
  - Des méthodes : il s'agit de ses actions

- La création d'un objet se fait à partir d'un modèle commun appelé Classe. On dira alors qu'un objet est une instance d'une Classe. Il est possible de créer vos propres Classes ou d'utiliser celles fournis avec le langage.

Langage Objet ?

BALLE
taille forme couleur
lancer() rebondir()

CELLULE
ligne colonne valeur couleur de fond
evaluer() copier()

Finalement, sur Excel tout est objet : cellule, ligne, colonne, feuille de calcul, classeur…

Macro ?

Définition : Une macro est une portion de code VBA s'exécutant au sein d'un classeur

- Un classeur Excel peut contenir un nombre infini de macros.

- Il est possible de créer des macros avec des visibilités différentes:
  - Au niveau du classeur, pour s'exécuter de n'importe où
  - Au sein de chaque feuille de calcul avec une manipulation locale

Macro ?

Sub bidule()

End Sub

Activer le mode développeur

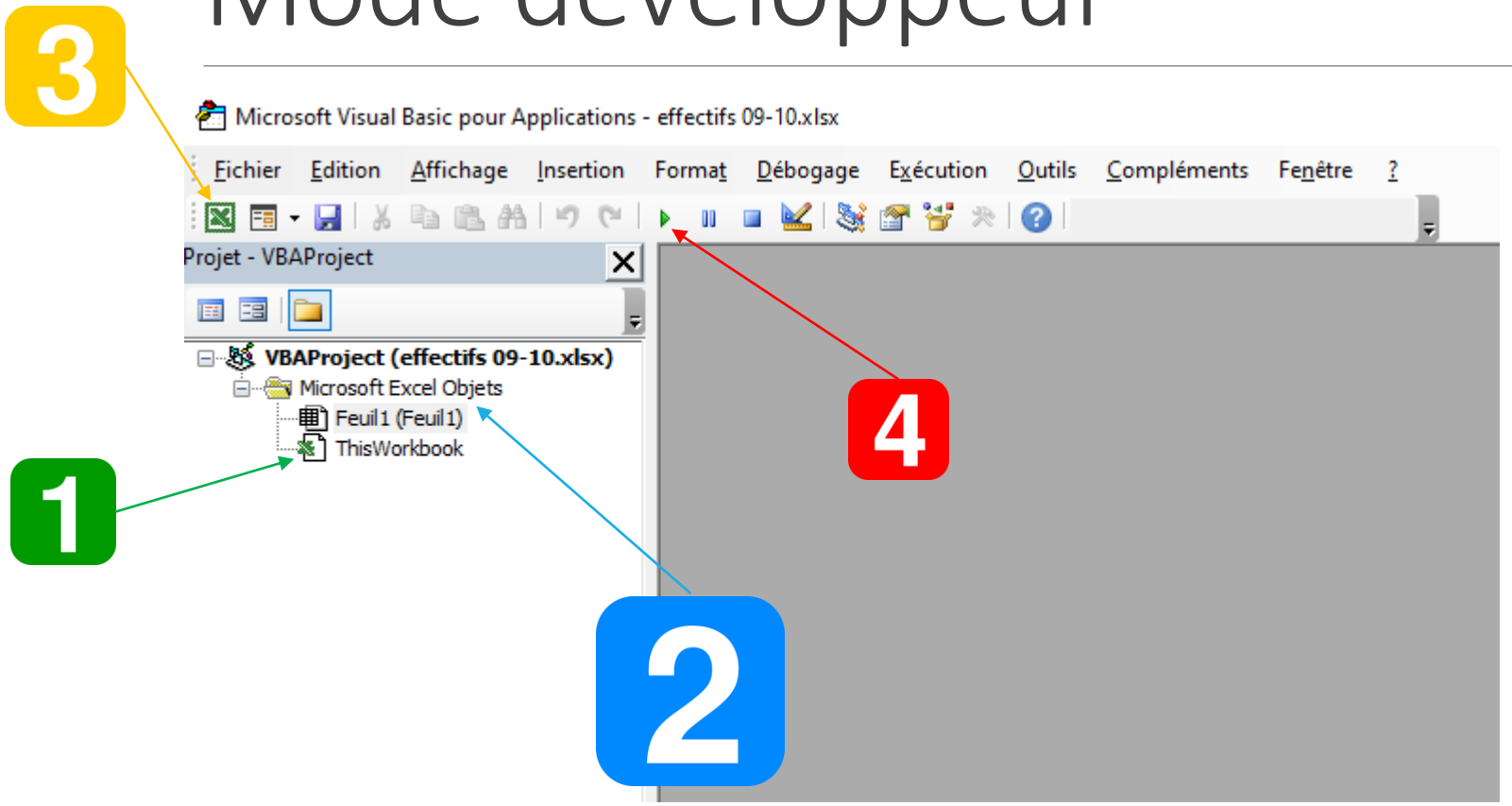
Sur windows:

=> Dans Excel : Fichier / Options / Personnaliser le ruban >> Cocher à droite la case « Développeur »

Sur macos :

=> Aller dans les paramètres d'Excel et saisir « Développeur » puis cocher la case permettant l'activation du mode

Mode développeur



1 : Classeur
2 : Feuille de calcul
3 : Retour au classeur
4 : Stopper Exécution

TP

Afin de tester l'exécution de code VBA au sein de votre classeur, mettons en place une macro permettant de modifier la taille de la police d'une sélection de cellules (sélection réalisée à l'aide de votre souris)

Code :

```
Sub test_macro()


Selection.Font.Size = 24


End Sub
```

Sauvegarde du classeur 

.XLSX

- Valeurs
- Formules

.XLSM

- Valeurs
- Formules
- Code VBA

Accès au contenu des cellules

Range(coordonnées)

Exemples:

Range (« A7 ») ➔ Représente la cellule A7

Accès au contenu d'une cellule via Range en utilisant l'attribut Value

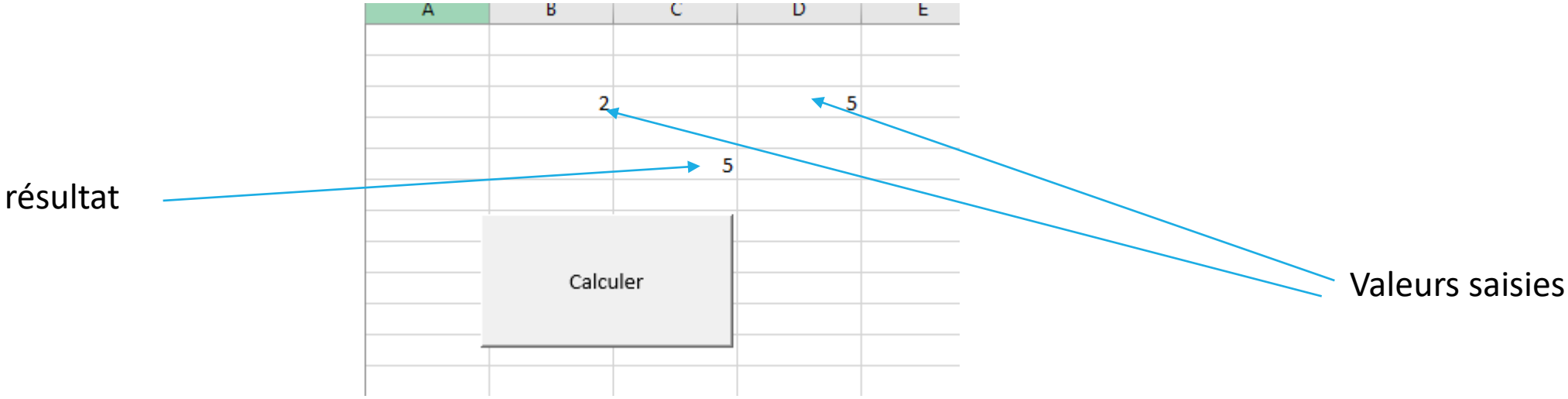
Range (« A12 »).Value = 14 ➔ Affecte la valeur 14 dans la cellule A12

Range(« A34:A47 »).Value = « Hello » ➔ Ecrit *Hello* dans toutes les cellules de la plage A34:A47

TP

Créez une macro permettant de faire la somme du contenu de deux cellules et d'afficher le résultat dans une troisième cellule.

Vous choisirez lors de l'écriture de votre macro les 3 cellules dans votre feuille de calcul



Variables

Dim nom_variable [As type]

nom_variable :

Unique pour chaque variable dans une macro

[a-z,A-Z,_, 0-9 (pas en première position)

type: (quelques exemples)

Integer : nombre entier

Double : nombre réel

String : chaine de caractères

Boolean : True / False

Variables

Exemple :

Dim my_var As Integer
my_var = 8

~~Dim my_var As Integer = 8~~
~~Dim my_var = 8~~

Dim bob,bryan As Integer

bryan est de type Integer

bob n'a pas de type, on dit qu'il est de type Variant

Équivalent:

Dim bob
Dim bryan As Integer

Type Variant: vigilance

Dim variable

Si ……..

variable = « hello »

Sinon

variable = 8

Fin SI

variable = variable +4

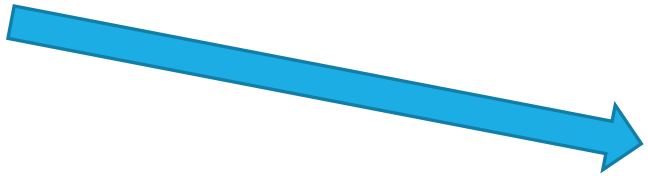
MessageBox

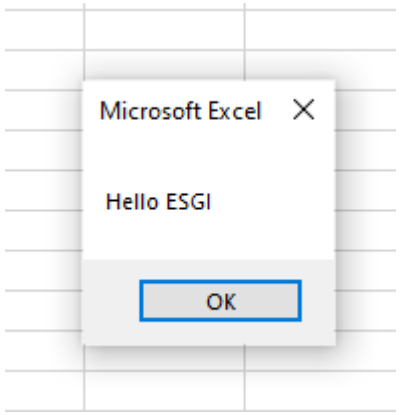
- Un MessageBox est une popup permettant d'afficher une valeur au format chaine de caractères.

```
Sub msgtest()

MsgBox "Hello ESGI"

End Sub
```





MsgBox *chaine_a_afficher*

MessageBox

- Permet d'afficher le contenu d'une variable (si type String)

```
Dim test As String
test = « coucou »
MsgBox test
```

- Peut permettre d'afficher des valeurs numériques

```
Dim nb As Integer
nb = 7
MsgBox Cstr(nb)
```

- Mélanger texte et variable, Opérateur de concaténation : **&**

```
MsgBox « Valeur de nb : » & nb
```

Conditions

```
If ……..  Then

Else

End If
```

```
If ……..  Then

End If
```

```
If ……..  Then

ElseIf …….. Then

Else

End If
```

Opérateurs

Comparaison		Logique	
>	Supérieur	and	ET logique
>=	Supérieur ou égal	or	OU logique
<	Inférieur	not	NEGATION
<=	Inférieur ou égal		
=	Egalité		
<>	Différend		

Boucles

Boucle for :

- Equivalent boucle POUR en algo
- Toujours bornes incluses
- S'utilise quand on connait le nombre d'itérations :

```
For var = 0 To 10
        Corps de la boucle
Next var
```

```
For var = 10 To 0 Step -1
        Corps de la boucle
Next var
```

Boucles

Boucle while :

- Equivalent boucle TANT QUE en algo
- S'utilise quand on ne connait pas le nombre d'itérations :

```
While condition
        Corps de la boucle
Wend
```

```
Dim a As Integer
a=6
While a <= 10

        Corps de la boucle
        a++   a=a+1
Wend
```

Boucles

Quitter une boucle à tout instant:

Boucle For :

Exit For

Boucle While:

Exit While

TP

Sur les 3000 premières lignes d'une colonne de votre choix, construire une macro permettant d'écrire le numéro de la ligne dans chacune des cellules

	A	B
1		1
2		2
3		3
4		4
5		5
6		6
7		7
8		8
9		9
10		10

...

2998		2998
2999		2999
3000		3000
3001		