

## Chapitre 4 : Objets associés Feuilles de calcul

Emmanuel Hyon  
issus de ceux de Laurent Mesnager

Université de Paris Ouest Nanterre la Défense

2014-2015

**1** La valeur Nothing

**2** Workbook

**3** Application

**4** Documentation

# Le Type Worksheet

Le type `Worksheet` représente une feuille de calcul

Les collections `Worksheets` et `Sheets` contiennent l'ensemble des feuilles de calcul d'un classeur

Pour associer une variable de type `Worksheet` à une des feuilles de calcul, on écrit

```
Dim s1 As WorkSheet , s2 As WorkSheet
```

```
Set s1 = Worksheets(1)
```

```
's1 est associé à la première feuille du classeur
```

```
Set s2 = WorkSheets("Feuil1")
```

```
's2 est associé à la feuille nommée "Feuil1"
```

## Le Type Worksheet

Une fois qu'une variable de type `WorkSheet` est associée à une feuille de calcul, on peut

- ▶ rendre active une feuille de calcul à l'aide de la méthode `Activate`,
- ▶ récupérer le nom d'une feuille à l'aide de la propriété `Name`,
- ▶ récupérer sa position à l'aide de la propriété `Index`,
- ▶ récupérer la collection de toutes les cellules de la feuille à l'aide de la propriété `Cells`,
- ▶ récupérer une plage de cellule de la feuille à l'aide de la méthode `Range`,
- ▶ la déplacer à l'aide de la méthode `Move`,
- ▶ la supprimer à l'aide de la méthode `Delete`,
- ▶ etc ...

# Changer le nom d'une feuille

```
Dim s As Worksheet  
  
Set s = Sheets(1)  
  
s.Name = "MaFeuille"
```

# Effacer le contenu d'une feuille de calcul

```
Dim s As Worksheet
```

```
Set s = Sheets(1)
```

```
s.Cells.ClearContents
```

# Différence entre Sheets et Worksheets

## Différence entre Sheets et Worksheets

- ▶ Sheets
- ▶ Worksheets

## Déplacer une feuille de calcul

```
Dim s1 As Worksheet , s2 As Worksheet
```

```
Set s1 = Sheets(1)
```

```
Set s2 = Sheets(2)
```

Alors, l'instruction

```
s1.Move After:=s2
```

déplace la feuille de calcul associée à s1 après la feuille de calcul associée à s2 tandis que

```
s1.Move Before:=s2
```

déplace la feuille de calcul associée à s1 avant la feuille de calcul associée à s2.



# Supprimer une feuille

Pour supprimer une feuille, on utilise la méthode Delete

```
s.Delete
```

**Remarque.** Lors de la suppression, selon la configuration d'excel, il est possible qu'un message d'alerte soit affiché demandant de confirmer la suppression.

Pour désactiver ce comportement dans un programme VBA, une manière de faire est d'écrire

```
Application.DisplayAlerts = False
```

```
s.Delete
```

```
Application.DisplayAlerts = True
```

## Identifier la feuille sur laquelle se trouve une plage de cellule

La propriété `Worksheet` d'une variable de type `Range` permet de savoir sur quelle feuille de calcul est une plage de cellule.

Cette propriété retourne une variable de type `Worksheet`

```
Dim p As Range, s As Worksheet
```

```
Set s = Sheets(2)
```

```
Set p = s.Range("A1:B6")
```

```
Set s = p.Worksheet
```

La variable `s` est alors associée à la feuille de calcul nommée "Feu11".

## Identifier une plage de cellule d'une feuille de calcul

Une variable de type `WorkSheet` possède une méthode `Range` qui permet d'associer une plage de cellules d'une feuille de calcul à une variable de type `Range`

```
Dim p As Range, s As WorkSheet
```

```
Set s = Sheets(1)
```

```
Set p = s.Range("$A$3:$E$8")
```

La variable `p` est alors associée à la plage de cellules `$A$3:$E$8` de la première feuille de calcul du classeur.

## Parcourir les feuilles de calcul d'un classeur

A l'aide des collections `Worksheets` et `Sheets`, on peut parcourir l'ensemble des feuilles de calcul d'un classeur à l'aide d'une boucle **For Each**

```
Dim s As Worksheet  
  
For Each s In Worksheets  
  
    MsgBox s.Name  
  
Next s
```

# Compter le nombre de feuilles de calcul d'un classeur

La propriété `Count` des collections `Worksheets` et `Sheets` retourne le nombre de feuilles de calcul que contient le classeur

```
MsgBox Sheets.Count & " _feuilles_de_calcul "
```

## Ajouter une feuille de calcul

Pour créer une feuille de calcul, on utilise la méthode `Add` des collections `Worksheets` et `Sheets`.

```
Dim s As Worksheet
```

```
Set s = Worksheets(1)
```

```
Worksheets.Add After:=s
```

```
Sheets.Add Before:=s
```

On indique où doit être inséré la nouvelle feuille de calcul à l'aide des arguments `After` et `Before`.

# La propriété UsedRange

Une propriété très intéressante d'une variable feuille de type `Worksheet` est la propriété `UsedRange`.

La propriété `UsedRange` est une variable de type `Range` associée à la **plus petite** plage rectangulaire contenant l'**ensemble** des cellules non vides de la feuille de calcul.

# La propriété UsedRange

Supposons associée à la feuille ci-dessous une variable feuille de type Worksheet

Microsoft Excel - Classeur1

Fichier Edition Affichage Insertion Format Outils Données Fenêtre ?

A1

	A	B	C	D	E	F	G	H	I
1									
2									
3									
4			1	2		3			
5			4		6	6			
6			8	8		9			
7									
8			6	7	8	9		0	
9									
10									
11									
12									



## La propriété UsedRange

```
Dim p As Range, feuille As Worksheet
```

```
Set feuille = Worksheets("nom_de_la_feuille")
```

```
Set p = feuille.UsedRange
```

La référence absolue de la plage p est \$C\$4:\$H\$8.

Microsoft Excel - Classeur1

Echier Edition Affichage Insertion Format Outils Données Fenêtre ?

A1

	A	B	C	D	E	F	G	H	I
1									
2									
3									
4			1	2		3			
5			4		6	6			
6			8	8		9			
7									
8			6	7	8	9		0	
9									
10									
11									
12									

Plage retournée par UsedRange = \$C\$4:\$H\$8

## La propriété UsedRange

Pour connaître les coordonnées des cellules limites de la plage, il y a la propriété `.End`

## La feuille active

La feuille active est la feuille du classeur actif mis au premier plan.

Si l'on ne mentionne pas la feuille en associant une variable de type Range a une plage de cellules, c'est la feuille active qui est considérée.

```
Dim p As Range
```

```
Set p = Range("A1:H8")
```

La variable p est associée à la plage de cellules A1:H8 de la **feuille active**

## Activer une feuille

Il est fortement conseillé lorsqu'on associe une variable de type Range d'indiquer explicitement la feuille de calcul ou d'activer préalablement la feuille de calcul contenant la plage.

### Exemple.

```
Dim p As Range  
Set p = Sheets(1).Range("A1:H8")
```

La variable feuille est alors associée à plage A1:H8 de la première feuille du classeur actif.

## Activer une feuille

Il est fortement conseillé lorsqu'on associe une variable de type Range d'indiquer explicitement la feuille de calcul ou d'activer préalablement la feuille de calcul contenant la plage.

### Exemple.

```
Sheets(1). Activate
```

```
Set p = Range("A1:H8")
```

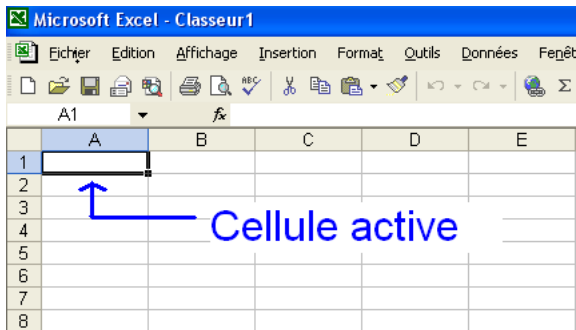
La variable `feuille` est alors associée à la A1:H8 de la feuille active (qui est la première feuille du classeur actif).

## La variable `ActiveSheet`

Le langage VBA fournit une variable de type `Worksheet` contenant la référence de la feuille active : la variable `ActiveSheet`.

## Cellule active

La cellule active est la cellule de feuille active mise en surbrillance.



## La variable `ActiveCell`

La variable `ActiveCell` est une variable de type `Range` automatiquement associée à la cellule active.

On peut activer une cellule à l'aide de la méthode `Activate`

```
Set p = Range("F6")
```

```
p.Activate
```

C'est la cellule F6 de la feuille active qui est désormais la cellule active



# Sommaire

**1** La valeur Nothing

2 Workbook

3 Application

4 Documentation

## La valeur Nothing

Il existe beaucoup d'autres propriétés et méthodes pour les types Range, Worksheet.

**Attention.** A chaque fois qu'on associe une variable à une plage de cellule, feuille de calcul ou classeur, il est conseillé, lorsqu'on n'a plus besoin de cette association, de libérer la mémoire en écrivant

```
Dim p As Range
```

```
Set p = Range("A1:R123") 'on réserve de la mémoire pour p  
  
'traitements
```

```
Set p = Nothing 'on libère la mémoire allouée à p
```

# Sommaire

1 La valeur Nothing

2 Workbook

3 Application

4 Documentation

# L'objet Workbook

Le type `Worksheet` représente un objet classeur

La collection `Workbooks` contient l'ensemble des des classeurs ouverts dans excel (dans une instance d'excel)

Pour associer une variable de type `Workbook` à une des feuilles de calcul, on écrit

```
Dim s1 As Workbook, dim s2 As Worksheet
```

```
Set s1 = Workbooks(1)
```

```
's1 est associé au premier classeur ouvert
```

```
Set s2 = Workbooks("classeur.xls").Worksheets("feuille1")  
associe s2 à la feuille 1 du classeur
```

# Sommaire

1 La valeur Nothing

2 Workbook

**3 Application**

4 Documentation

# L'objet Application

L'objet Application représente une instance d'excel.

# Sommaire

1 La valeur Nothing

2 Workbook

3 Application

4 Documentation

## Quelques liens

- ▶ [http ://msdn.microsoft.com/en-us/library/office/dn467914%28v=office.15%29.aspx](http://msdn.microsoft.com/en-us/library/office/dn467914%28v=office.15%29.aspx)
- ▶ Excel et VBA de Mikaël Bidault