

AUTOMATISER SES TÂCHES AVEC DES MACROS EN VBA

Une macro est un ensemble de commandes qui s'exécute autant de fois que nécessaire simplement en les appelant par son nom. Les macros permettent d'automatiser les tâches répétitives ; il suffit d'enregistrer une première fois les commandes voulues — saisie de données, sélection, calculs, mise en forme, impression, etc. — en leur donnant un nom puis, d'exécuter ces commandes en les appelant par leur nom à chaque fois que l'on devra répéter cette même tâche, un peu de la même manière que l'on s'enregistre avec un enregistreur MP3 afin de sortir toujours le même discours quand cela est nécessaire.

Une macro-commande est donc une suite de commandes à exécuter automatiquement alors qu'une macro-fonction est une fonction contenant plusieurs commandes qui s'utilise dans des cellules, de la même manière que n'importe quelle fonction d'Excel.

- ✓ Les macros sont créés dans un langage de programmation appelé VBA (Visual Basic Application). Ce langage est commun à la plupart des applications de Microsoft Office — Word, Excel, PowerPoint, Access, etc. — toutefois il existe des fonctions spécifiques à chaque logiciel et les techniques de programmation peuvent être légèrement différentes (comme avec Access).

Les exemples qui suivent sont, certes, très incomplets mais permettent d'avoir quelques bases pour débiter en VBA. Voici quelques sites pour en savoir plus à ce sujet :

- www.excel-pratique.com/fr/vba
- silkyroad.developpez.com/VBA
- lecompagnon.info/vba-excel
- www.excelabo.net

CRÉATION D'UNE MACRO SIMPLE

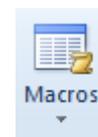
Il est possible de créer une macro de manière très simple, en réalisant une succession de commandes de la même manière que d'habitude.

Une macro se réalise en trois temps :

1. Donner un nom à sa macro
2. Enregistrer les commandes
3. Arrêter la macro
4. Exécuter la macro

PRÉPARATIF

À partir de Word 2007, pour gérer ses macros, il est possible de passer par l'onglet Affichage, puis le bouton Macros.



Toutefois, il est préférable d'afficher l'onglet *Développeur* qui est spécifique aux macros :

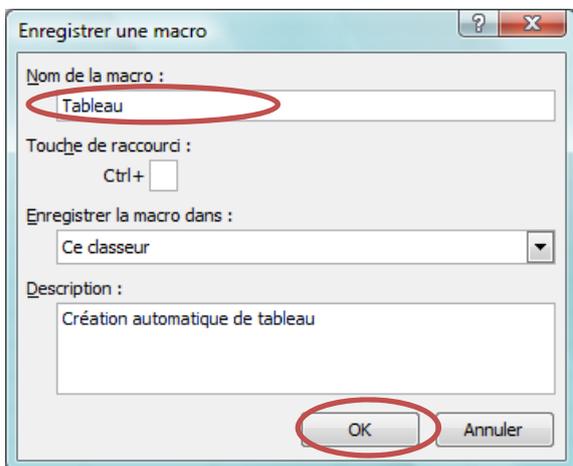
1. Menu Fichier (ou bouton Office dans 2007)
Options
2. Dans 2007 : dans « Standard » Cocher *Afficher l'onglet Développeur dans le ruban*.
Dans 2010 : dans « Personnaliser le ruban », dans la liste des onglets principaux, cocher *Onglet Développeur*
et

CRÉATION D'UNE MACRO SIMPLE

Nous allons créer une macro qui crée automatiquement un tableau sur la feuille active d'Excel.

DONNER UN NOM À SA MACRO

1. Dans Excel, se placer sur la feuille de calcul où va être créé le tableau.
2. Onglet *Développeur*
Cliquer sur le bouton  *Enregistrer une macro* ou sur celui qui est en bas à gauche, sur la barre d'état.
3. Dans la boîte de dialogue qui apparaît :
 - **Nom de la macro** : saisir un nom sans espace pour sa macro (exemple : Tableau)
 - **Touche de raccourci** (facultatif) : saisir une lettre pour exécuter sa macro à partir d'un raccourci clavier. Attention de ne pas utiliser un raccourci clavier existant.
 - **Enregistrer la macro dans** : choisir « Ce classeur » (ou « Nouveau classeur » pour enregistrer sa macro dans un nouveau document ou « Classeur de macros personnelles » pour accéder à sa macro depuis n'importe quel classeur)
 - **Description** : Texte libre donnant des indications sur sa macro (auteur, date, etc.)



et **OK**

- ✓ Le bouton  *Arrêter l'enregistrement* apparaît dans le ruban de l'onglet *Développeur* ainsi que sur la barre d'état. Toutes les commandes effectuées — y compris les aperçus avant impression, les défilements, les sauvegardes — sont désormais enregistrées dans sa macro.

ENREGISTRER SA MACRO

Exécuter toutes les commandes (saisie, sélection, mise en forme, couper-copier-coller, sauvegarde, impression...) nécessaires à la macro de la même manière que lorsqu'on travaille à son document.

Chacune des commandes est automatiquement enregistrée dans la macro.

Exemple de macro pour faire un tableau :

1. Saisir les entêtes de colonnes, exemple : « secteur 1 » en B1 et « Produit 1 » en A2
2. Effectuer la recopie avec la poignée automatique

	A	B	C	D
1		Secteur 1	Secteur 2	Secteur 3
2	Produit 1			
3	Produit 2			
4	Produit 3			
5				

3. En A5 et en E1, écrire « Total »
4. Effectuer les calculs du total :
 - Se placer dans la cellule où doit apparaître le résultat, exemple : B5
 - Cliquer sur le bouton  Somme automatique
 - Sélectionner les cellules à calculer, même si elles sont vides : ici, de B2 à B4 et valider
 - Faire la recopie de la cellule B5 jusqu'à la cellule E5
5. Recommencer le calcul de la somme pour E2 en sélectionnant pour le calcul les cellules de B2 à D2 puis valider et faire une recopie de E2 jusqu'à E4
6. Éventuellement, ajouter d'autres fonctions (Moyenne, Max, min, etc.)
7. Sélectionner les cellules qui vont contenir les chiffres et appliquer un format monétaire 
8. Effectuer la mise en forme du tableau (encadrement des cellules, couleurs, etc.), penser à élargir les lignes et les colonnes.

Exemple :

	A	B	C	D	E
1		Secteur 1	Secteur 2	Secteur 3	Total
2	Produit 1				- €
3	Produit 2				- €
4	Produit 3				- €
5	Total	- €	- €	- €	- €

ARRÊTER L'ENREGISTREMENT DE SA MACRO

Le tableau terminé, arrêter l'enregistrement :

Onglet *Développeur*

Arrêter l'enregistrement

ou cliquer sur le bouton  Arrêter l'enregistrement sur la barre d'état

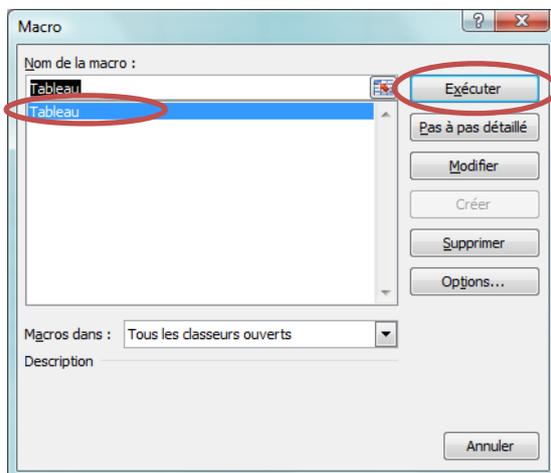
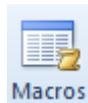
- ✓ Toutes les commandes effectuées ne sont plus enregistrées dans la macro.

EXÉCUTION DE SA MACRO

Se placer sur une feuille vide

Onglet *Développeur*

Macros



Cliquer sur le nom de la macro à exécuter, exemple : Tableau et cliquer sur **Exécuter**

La macro s'exécute de la même manière – mais plus rapidement – que si nous effectuons nous-même ces commandes une par une, créant un nouveau tableau, avec des formules, une mise en forme, élargissant les lignes et les colonnes, etc.

De plus, elle peut être réutilisée autant de fois qu'on le souhaite.

- ✓ Il suffit de saisir des chiffres dans le tableau et les calculs se mettent automatiquement à jour. Toutefois, les modifications dans le tableau — saisie, mise en forme, etc. — ne sont pas enregistrées dans la macro

MODIFICATION DE SA MACRO

Pour visualiser le code de la macro :

1. Onglet *Développeur* ou **Alt | F11**

Macros

Cliquer sur le nom de la macro voulue (exemple : *Tableau*)

et cliquer sur **Modifier**

2. Toutes les commandes enregistrées apparaissent, codées en VBA (Visual Basic Application), langage de programmation des macros dans les logiciels de la suite Microsoft Office. Ces commandes peuvent être modifiées avec une bonne connaissance du VBA.

```
Range("B1").Select
ActiveCell.FormulaR1C1 = "Secteur 1"
Range("A2").Select
ActiveCell.FormulaR1C1 = "Produit 1"
Range("B1").Select
```

Par exemple, dans la liste de code, il suffit de changer « Secteur » par « Région » et « Produit » par « Service » — sans appuyer sur **↵** pour modifier les en-têtes du tableau.

```
Range("B1").Select
ActiveCell.FormulaR1C1 = "Région 1"
Range("A2").Select
ActiveCell.FormulaR1C1 = "Service 1"
Range("B1").Select
```

3. Puis revenir à son tableau Excel (par la barre des tâches par exemple).

- ✓ Ainsi chaque fois que la macro sera exécutée — et non pour les tableaux déjà créés — *Région 1, 2...* apparaîtra sur la ligne des titres et *Service 1, 2...* sur la colonne des titres

POUR EXÉCUTER SA MACRO PAS À PAS

En cas de problème dans l'exécution du code, il est possible d'exécuter sa macro pas à pas afin de vérifier quelle ligne contient une erreur pour la corriger :

1. Onglet *Développeur*

Macros

Cliquer sur le nom de la macro voulue (exemple : *Tableau*)

Cliquer sur **Pas à pas détaillé**

2. Appuyer sur la touche de fonction **F8** pour exécuter les lignes de la macro une par une.

EXÉCUTION DE MACRO À PARTIR D'UN BOUTON

Plutôt que de lancer une macro en passant par des onglets (Onglets *Développeurs*, *Macros...*), pour simplifier la tâche de l'utilisateur, il est possible de lancer une macro à partir d'un bouton dans un tableau.

Voici un exemple avec une macro qui trie automatiquement le tableau suivant par pays, puis par matériel puis par date :

	A	B	C	D
1	Date d'arrivée	Matériel	Pays d'origine	Quantité
2	21/05/2014	Ordinateur	France	12
3	30/04/2010	Logiciel	Danemark	7
4	01/06/2008	Logiciel	Danemark	8
5	06/03/2007	Périphérique	Espagne	17
6	31/07/2015	Ordinateur	Espagne	1
7	01/02/2016	Logiciel	France	9
8	02/03/2011	Logiciel	France	21
9	03/04/2008	Périphérique	France	19
10	04/05/2009	Ordinateur	Danemark	16

CRÉATION DE LA MACRO

NOMMER LA MACRO

1. Onglet *Développeur*

Cliquer sur Enregistrer une macro

ou sur le bouton  *Enregistrer une macro* sur la barre d'état.

2. Nom de la macro : « Tri_Pays » et

✓ Les commandes de la macro vont être enregistrées.

EXÉCUTER LES COMMANDES

1. Cliquer dans l'une des cellules de la colonne des dates

2. Onglet *Données*

Trier 

3. Cliquer dans une des cellules de la colonne du matériel

4. Onglet *Données*

Trier

5. Cliquer dans l'une des cellules de la colonne des pays

6. Onglet *Données*

Trier

✓ Ainsi, le tableau est trié par pays, puis par matériel puis par date.

ARRÊTER LA MACRO

Onglet *Développeur*

Arrêter l'enregistrement

ou cliquer sur le bouton  *Arrêter l'enregistrement* sur la barre d'état

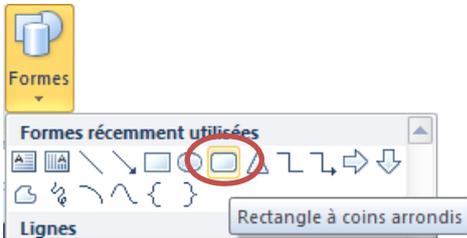
CRÉATION DU BOUTON

DESSIN DU BOUTON

1. Onglet *Insertion*

Formes

Cliquer sur la forme voulue, exemple : *Rectangle à coins arrondis*



2. En maintenant le bouton gauche de la souris enfoncé, dessiner un bouton dans la feuille de calcul
3. Saisir le texte du bouton, exemple : « Tri » et le mettre en forme (onglet Accueil, Gras, Italique, changer la police, la taille des caractères, centrer horizontalement, verticalement, etc.)
4. Effectuer la mise en forme du bouton, changer la couleur (onglet Format, Remplissage, etc.), etc.

	A	B	C	D	E	F
1	Date d'arrivée	Matériel	Pays d'origine	Quantité		
2	01/06/2008	Logiciel	Danemark	8		
3	30/04/2010	Logiciel	Danemark	7		
4	04/05/2009	Ordinateur	Danemark	16		
5	31/07/2015	Ordinateur	Espagne	1		
6	06/03/2007	Périphérique	Espagne	17		
7	02/03/2011	Logiciel	France	21		
8	01/02/2016	Logiciel	France	9		
9	21/05/2014	Ordinateur	France	12		
10	03/04/2008	Périphérique	France	10		

AFFECTATION DE LA MACRO SUR LE BOUTON

Clic droit sur le bouton voulu

Affecter une macro

Cliquer sur la macro voulue, exemple : *Tri_pays*

et **OK**

UTILISATION DU BOUTON

1. Cliquer dans n'importe quelle cellule de la feuille.
 2. Cliquer sur le bouton voulu (le pointeur se transforme en main à l'index pointé)
 3. La macro se lance automatiquement et trie le tableau.
- ✓ Cette macro de tri est très pratique si l'on saisit des nouvelles données dans le tableau et qu'on souhaite retrier son tableau automatiquement.

MODIFICATION DU BOUTON

Si l'on clique sur le bouton pour le modifier, il va automatiquement lancer la macro.

Pour modifier son bouton :

1. En maintenant la touche Contrôle (**Ctrl**), cliquer sur le bouton
2. Modifier la mise en forme de son bouton.

SAUVEGARDER UN CLASSEUR CONTENANT UNE MACRO

Depuis 2007, il existe un format particulier pour sauvegarder un document contenant une macro :

Menu Fichier

Enregistrer sous

1. Saisir un nom
2. Choisir un dossier en cliquant deux fois dedans
3. Dans *Type*, choisir « Classeur Excel (prenant en charge les macros) (*.xslm) »
et **OK**

DÉBUTER LA PROGRAMMATION EN VBA

Au lieu de passer par l'enregistreur de macro, il est également possible de saisir soi-même les commandes en VBA. Cela est plus long et plus laborieux mais il est possible d'intégrer dans ses macros des fonctions plus évoluées.

La feuille contenant les macros est appelé un module.

Un projet — lié à un même classeur — peut contenir plusieurs modules qui eux-mêmes peuvent contenir plusieurs macros.

CRÉATION D'UNE MACRO EN VBA

Donner un nom à sa macro

1. Onglet *Développeur*

Macros

Saisir un nom de macro (exemple : *Test*)

Cliquer sur le bouton **Créer**.

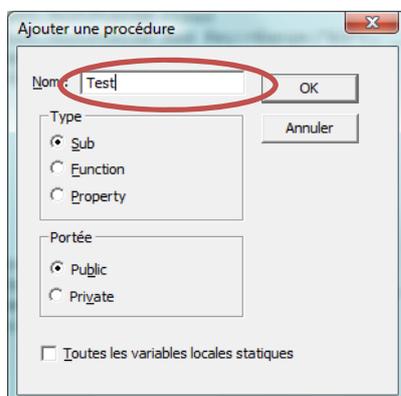
ou

Alt | F11 pour accéder au module (la fenêtre de code VBA)

2. Menu *Insertion*

Procédure

- **Nom** : saisir le nom de sa macro (exemple : *Test*)
- **Type** : cocher *Sub* pour une macro-commande (ou *Function* pour une macro-fonction)
- **Portée** : cocher *Public* pour la rendre accessible à l'ensemble des macros du projet (ou *Private* pour ne la rendre accessible que dans un seul module)



et **OK**

Saisir les commandes voulues

Le module (la fenêtre de code VBA) apparaît :

Saisir les commandes de sa macro entre *Sub* (début de la macro) et *End sub* (fin de la macro) en ne mettant qu'une commande par ligne.

```
Sub Test ()  
|  
End Sub
```

- ✓ L'exécution de sa macro se fait comme précédemment (onglet *Développeur*, *Macros*, etc.)

COMMENTAIRES

Afin de faciliter la lecture de sa macro, il est possible de mettre des commentaires dans son code. Ceux-ci seront précédés d'une apostrophe (') et peuvent être placés après une ligne de code ou sur une ligne indépendante. Dans le code VBA, les commentaires apparaissent en vert.

Exemples :

```
' Macro de mise à jour des données  
ActiveCell.FormulaR1C1 = "Région 1" ' Titre de la colonne du tableau
```

SÉLECTION DE CELLULES

SÉLECTION D'UNE CELLULE

Sélection de la cellule C5

```
Range("C5").Select
```

ou

```
Cells(5, 3).Select
```

- ✓ *Select* permet de sélectionner une ou plusieurs cellules alors qu'*Activate* permet de choisir qu'une seule cellule qui sera la cellule active.
Pour une seule cellule *Cells(5, 3).Select* ou *Cells(5, 3).Activate* sont équivalents

Sélection de la cellule nommée « Taux »

```
Range("Taux").Select
```

Sélection d'une cellule en fonction de la cellule active

```
ActiveCell.Offset(7, 3).Select
```

Accès à la feuille nommée « Facture » et sélection de la cellule B2

```
Worksheets("Facture").Activate  
Range("B2").Select
```

Sélection de la feuille « Facture » du classeur ouvert « Gestion »

```
Workbooks("Gestion.xlsm").Worksheets("Facture").Activate
```

Sélectionne la 1^{ère} cellule de la ligne active

```
Cells(ActiveCell.Row, 1).Select
```

Sélectionne la 5^e cellule de la colonne active

```
Cells(5, ActiveCell.Column).Select
```

Sélection de la dernière cellule non vide de la ligne 1

```
Range("XF1").End(xlToLeft).Select
```

Sélection de la cellule qui est après la dernière cellule non vide de la ligne contenant la cellule active

```
Cells(ActiveCell.Row, "XFD").End(xlToLeft)(1, 2).Select
```

Sélection de la dernière cellule non vide de la colonne A

```
Range("A1048576").End(xlUp).Select
```

✓ À noter que la formule suivante — plus simple — peut également fonctionner s'il n'y a pas de cellules vides entre les cellules non vides de la colonne :

```
Range("A1").End(xlDown).Select
```

Sélection de la première cellule vide après la dernière cellule non vide de la colonne A

```
Range("A1048576").End(xlUp)(2, 1).Select
```

SÉLECTION D'UNE ZONE DE CELLULES

Sélection des cellules de B5 à D20

```
Range("B5:D20").Select
```

OU

```
Range(Cells(5, 2), Cells(20, 4)).Select
```

Sélection des cellules de la colonne 1 à la colonne 10 de la ligne active

```
Range(Cells(ActiveCell.Row, 1), Cells(ActiveCell.Row, 10)).Select
```

Sélection des cellules de la ligne 5 à la ligne 12 de la colonne active

```
Range(Cells(5, ActiveCell.Column), Cells(12, ActiveCell.Column)).Select
```

✓ Sélectionne la cellule située 7 lignes en dessous et trois lignes à droite de la cellule active.

Sélection des cellules de la cellule A1 jusqu'à la dernière cellule non vide de la colonne A

```
Range("A1:A" & Range("A1048576").End(xlUp).Row).Select
```

Sélection des cellules de la cellule active jusqu'à la dernière cellule non vide de la ligne

La formule qui suit ne fonctionne que s'il n'y a pas de cellules vides entre les cellules de la ligne

```
Range(ActiveCell, ActiveCell.End(xlToRight)).Select
```

UTILISATION DE VARIABLES DANS DES CELLULES

Les variables servent à stocker et à utiliser des données — texte, nombre, etc. — dans une macro.

En VBA, il est vivement conseillé de définir le type de chaque variable — va-t-elle contenir un texte, une date, un chiffre, etc. — afin d'augmenter la rapidité du programme mais aussi d'éviter des erreurs d'interprétation de certaines variables.

DÉFINITION DU TYPE D'UNE VARIABLE

Toutes les variables sont définies en général au début de la procédure, avant les lignes de commande.

Dim *Nom de la variable* (sans espace ni ponctuation) **As** *Type de la variable*

Exemples :

```
Dim Entreprise As String ' Variable contenant du texte
Dim Naissance As Date ' Variable contenant une date ou une heure
Dim Quantite As Integer ' Nombre entier compris entre -32 768 et 32 767
Dim Distance As Double ' Nombre à virgule compris entre -1,79769313486232 x 10308
    et 1,79769313486232 x 10308
```

Il est également possible de définir une variable avec un symbole, exemple :

Dim *Entreprise*\$ est équivalent à **Dim** *Entreprise* **As** *String*

TYPES DE VARIABLE LES PLUS RÉPANDUS

Nom	Type	Caractéristiques	Symbole
Byte	Numérique	Nombre entier de 0 à 255	
Integer	Numérique	Nombre entier de -32 768 à 32 767	%
Long	Numérique	Nombre entier de - 2 147 483 648 à 2 147 483 647	&
Currency	Numérique	Nombre à décimale fixe de -922 337 203 685 477,5808 à 922 337 203 685 477,5807	@
Single	Numérique	Nombre à virgule flottante de $-3,402823 \times 10^{38}$ à $3,402823 \times 10^{38}$!
Double	Numérique	Nombre à virgule flottante de $-1,79769313486232 \times 10^{308}$ à $1,79769313486232 \times 10^{308}$	#
String	Texte	Texte	\$
Date	Date	Date et heure	
Boolean	Boolean	True (vrai) ou False (faux)	
Object	Objet	Objet Microsoft (Classeur, Feuille, Cellule, etc.)	
Variant	Tous	Tout type de données (type par défaut si la variable n'est pas déclarée)	
Range	Cellule	Zone de cellules	

MODIFICATION OU RÉCUPÉRATION DU CONTENU DES CELLULES

Stockage du contenu de la cellule active dans une variable nommée *Montant*

```
Montant=ActiveCell
```

Stockage dans une variable nommée *Salaire* du contenu de la cellule C5

```
Salaire=Cells(5, 3) ' 5 : numéro de la ligne, 3 : numéro de la colonne (la C)  
' Récupération de la cellule C5 de la feuille « Gestion » :  
Salaire = Sheets("Gestion").Cells(5, 3).Value
```

Récupération de donnée dans un autre classeur

Affiche le contenu de la cellule C3 de la feuille « Facture » dans le classeur « Gestion » :

```
MsgBox Workbooks("Gestion.xlsm").Worksheets("Facture").Cells(2, 2).Value
```

Stockage dans une variable nommée *Prix* du contenu de la cellule située deux lignes et quatre colonnes à partir la cellule active

```
Prix = ActiveCell(3, 5)
```

Ou

```
Prix = ActiveCell.Offset(2,4)
```

- ✓ Avec ActiveCell, la cellule active est désignée par ActiveCell(**1,1**)
alors qu'avec ActiveCelle.Offset, la cellule active est choisie avec ActiveCell.Offset(**0,0**)

Stockage d'une valeur dans des cellules

```
Cells(5, 3)=1000 ' affiche 1000 dans la cellule C5  
Cells(6, 3)="Bonjour" ' affiche « bonjour » dans la cellule C6  
Cells(7, 3)=Ventes ' Récupère le contenu de la variable Ventes dans la cellule C7  
Cells(8, 3)=Cells(8, 3) * 1.3 ' Augmente de 30% le contenu de la cellule C8
```

☞ Attention : utiliser un point (.) et non la virgule (,) pour les décimales dans les codes VBA

EXEMPLE DE MACRO QUI UTILISE DES VARIABLES ET DES CELLULES

Voici un exemple, à partir du tableau suivant, qui permet d'augmenter de 10% le contenu d'une cellule et affiche le résultat dans une boîte de dialogue avec le nom et le prénom de la personne concernée :

	A	B	C
1	Nom	Prénom	Prime accordée
2	Fontfreyde	Michel	1 000,00 €
3	Fouheyrand	Marie	2 000,00 €
4	Foubert	Simon	3 000,00 €
5	Matthieu	Jacques	500,00 €
6	Montclar	Astorg	1 000,00 €
7	Bridier	Léonard	300,00 €
8	Gaschier	Madeleine	1 000,00 €
9	Lavaur	Suzanne	1 000,00 €
10	Wang	Judith	200,00 €
11	Heulluy	Lucie	2 000,00 €

CRÉER LA MACRO

Onglet *Développeur*
Macros

Nom : *Calcul_Prime* et

SAISIR LE CODE DE LA MACRO

Dans l'exemple qui suit, la commande *MsgBox* affiche un texte dans une boîte de dialogue

- Le signe « & » permet d'assembler des chaînes de caractères
... & " " & ... permet de séparer les variables par un espace vide afin de ne pas coller les données.
- La commande *Chr(10)* permet d'aller à la ligne.
- Les commentaires en vert sont facultatifs

Saisir les lignes suivantes (éventuellement, les récupérer par copier-coller) :

```
Sub Calcul_Prime()
    Dim Taux As Double ' Variable qui contiendra un nombre décimal
    Dim Prime As Double ' Variable qui contiendra un nombre décimal
    Dim Nom As String ' Variable qui contiendra du texte
    Dim Prenom As String ' Variable qui contiendra du texte

    ' Montant de l'augmentation
    Taux = 0.1 ' 0.1 = 10%

    ' Atteindre la première cellule de la ligne
    Cells(ActiveCell.Row, 1).Select ' ActiveCell.Row : numéro de la ligne en cours

    ' Multiplication par (1 + Taux) du contenu de la colonne Prime
    ' (située 2 colonnes après la cellule sélectionnée)
    ActiveCell.Offset(0, 2).Value = ActiveCell.Offset(0, 2).Value * (1 + Taux)

    ' Stockage du contenu des cellules dans des variables
    Nom = ActiveCell ' Cellule active
    Prenom = ActiveCell.Offset(0, 1) ' Une colonne à droite de la cellule active
    Prime = ActiveCell.Offset(0, 2) ' Deux colonnes à droite de la cellule active

    ' Affichage du nom, du prénom et de la prime accordée
    MsgBox Prenom & " " & Nom & Chr(10) & "Prime : " & Prime
End Sub
```

- ✓ Penser à sauvegarder son classeur avec comme Type « Classeur Excel (prenant en charge les macros)... », ces codes sont conservés dans le classeur Excel.

CRÉER UN BOUTON POUR EXÉCUTER LA MACRO

1. Retourner dans son tableau Excel
2. Onglet *Insertion*
Formes
Cliquer sur la forme voulue
3. Dessiner un bouton
4. Saisir un texte dans le bouton, exemple : « Calcul de prime »
5. Mettre en forme le bouton dans l'onglet *Format* (police, couleur, etc.)
6. Affecter une macro au bouton :
Clic droit sur le bouton
Affecter une macro
Cliquer sur la macro voulue : *Calcul_Prime* et **OK**

EXÉCUTER LA MACRO

1. Cliquer sur l'une des cellules de la ligne voulue
 2. Cliquer sur le bouton créé
- ✓ La cellule contenant le montant de la prime est augmentée de 10% et une boîte de dialogue apparaît avec le prénom et le nom du bénéficiaire ainsi que le nouveau montant de la prime.



EN CAS D'ERREUR DANS LA MACRO

1. **Alt | F11** ou onglet *Développeur*
Macros
Cliquer sur le nom de la macro à corriger, *Calcul_Prime*, et **Modifier**
2. Corriger les erreurs dans le code VBA
3. Retourner dans son classeur pour relancer la macro

QUELQUES FONCTIONS DE BASE EN VBA

Les macros qui suivent doivent être placée dans un module VBA (accessible par **Alt | F11**) soit en les saisissant soit en faisant un copier-coller.

Pour les exécuter, il suffit — pour la plus part — de cliquer dans une cellule ayant un nombre puis : onglet *Développeur*

Macros

Cliquer sur le nom de la macro voulue

Cliquer sur **Exécuter**

CONDITION LOGIQUE

La condition logique permet de réaliser des actions différentes suivant la valeur d'une variable ou d'une cellule

Structure

```
If condition Then
    Action si vrai
Else
    Action si faux
End If
```

Exemple

Macro qui n'augmente la valeur de la cellule sélectionnée que si la nouvelle valeur est inférieure à 3000 et affiche dans une boîte de dialogue (*MsgBox*) un message.

```
Sub Limite()
    Dim Montant As Double
    Dim Message As String

    Montant = ActiveCell * 1.1

    If Montant > 3000 Then
        Message = "Valeur de départ trop importante !"
    Else
        ActiveCell = Montant
        Message = "La valeur actuelle est de : " & Montant
    End If

    MsgBox Message
End Sub
```

Pour exécuter la macro : cliquer dans une cellule contenant un nombre et lancer la macro.

BOUCLE AVEC COMPTEUR

Macro qui recommence une boucle en incrémentant une valeur jusqu'à ce que la valeur maximale soit atteinte.

Structure

```
For i=Valeur minimale to Valeur maximale step Incrément
    Actions
Next i
```

Exemple

Affiche les nombres de 1 à 10 dans les cellules de G1 à G10 (colonne 7)

```
Sub Compteur()  
    Dim i As Integer  
  
    For i = 1 To 10  
        Cells(i, 7) = i  
    Next i  
End Sub
```

- ✓ L'instruction `Exit For` permet de sortir de la boucle, exemple :

```
If Cells(i, 1) = "Fin" Then Exit For
```

BOUCLES RÉPÉTITIVES TANT QU'UNE CONDITION EST VRAIE

Boucles qui recommencent une procédure jusqu'à ce qu'une condition soit remplie.

WHILE... WEND

While... Wend est la méthode la plus simple pour réaliser ce genre de boucle.

Structure

```
While condition  
    Action  
Wend
```

Exemple

Augmente le contenu de la cellule sélectionnée de 10% (variable *Augmentation*) tant que son contenu est inférieur à 5000.

Pour exécuter la macro, cliquer dans une cellule contenant un nombre avant de lancer la macro.

```
Sub Boucle()  
    Dim Montant As Double  
    Dim Augmentation As Double  
  
    Montant = ActiveCell  
    Augmentation = 0.1  
    While Montant < 5000  
        Montant = Montant * (1 + Augmentation)  
    Wend  
    ActiveCell = Montant  
End Sub
```

DO WHILE... LOOP

Do While... Loop s'utilise de la même manière que *While... Wend*

Structure

```
Do While condition  
    Action  
Loop
```

Exemple

```
Sub Boucle()  
    Dim Montant As Double  
    Dim Augmentation As Double  
  
    Montant = ActiveCell  
    Augmentation = 0.1  
    Do While Montant < 5000  
        Montant = Montant * (1 + Augmentation)  
    Loop  
    ActiveCell = Montant  
End Sub
```

Do... LOOP WHILE

À la différence de *Do While... Loop*, la boucle *Do... Loop While* permet de mettre la condition d'arrêt à la fin

Structure

```
Do
    Action
Loop While condition
```

Exemple

```
Sub Boucle()
    Dim Montant As Double
    Dim Augmentation As Double

    Montant = ActiveCell
    Augmentation = 0.1
    Do
        Montant = Montant * (1 + Augmentation)
    Loop While Montant < 5000
    ActiveCell = Montant
End Sub
```

- ✓ Contrairement à *Do While... Loop*, avec *Do... Loop While* la macro sera exécutée une première fois — donc la variable *Montant* sera augmenté de 10% même si son contenu est supérieur à 5000 — avant de vérifier s'il faut arrêter ou non la boucle.

DO UNTIL... LOOP

À la différence de *Do While... Loop* qui recommence la boucle tant que la condition est vraie, *Do Until... Loop* s'arrête dès que la condition est vraie.

Structure

```
Do Until condition
    Action
Loop
```

Exemple

Arrête d'augmenter le contenu de la cellule sélectionnée de 10% dès que son contenu est supérieur ou égal à 5000.

```
Sub Boucle()
    Dim Montant As Double
    Dim Augmentation As Double

    Montant = ActiveCell
    Augmentation = 0.1
    Do Until Montant >= 5000
        Montant = Montant * (1 + Augmentation)
    Loop
    ActiveCell = Montant
End Sub
```

- ✓ Comme pour la fonction *Do... Loop While*, il est possible de mettre la fonction *Until* à la fin de la boucle — *Do... Loop Until* — pour que celle-ci s'exécute au moins une fois

EXIT DO

Dans les boucles *Do... Loop*, il est possible d'ajouter une instruction *Exit Do* pour arrêter la boucle avant que la condition de la boucle soit remplie

Exemple

Si la variable *Augmentation* dépasse les 5%, les instructions jusqu'à la fin de la boucle sont ignorées et le programme se poursuit à partir de la fin de la boucle

```
Sub Boucle()  
    Dim Montant As Double  
    Dim Augmentation As Double  
  
    Montant = ActiveCell  
    Augmentation = 0.1  
    Do Until Montant >= 5000  
        Augmentation = Augmentation + 0.1  
        If Augmentation > 0.5 Then Exit Do  
        Montant = Montant * (1 + Augmentation)  
    Loop  
    ActiveCell = Montant  
    MsgBox Augmentation  
End Sub
```

CAS D'UNE VARIABLE

Déclenche une action en fonction de la valeur d'une variable.

Structure

```
Select Case Variable  
Case Valeur 1  
    Action 1  
Case Valeur 2  
    Action 2  
Case Valeur 3  
    Action 3  
...  
Case Else  
    Autre action  
End Select
```

Exemple

Macro qui affiche un message indiquant dans quelle tranche se situe la quantité commandée.

```
Sub Tranche()  
    Dim Quantite As Double  
  
    Quantite = ActiveCell  
    Select Case Quantite  
    Case 1  
        MsgBox "Commande unique"  
    Case 2  
        MsgBox "Double commande"  
    Case 3  
        MsgBox "Triple commande"  
    Case 4, 5, 6  
        MsgBox "Petite quantité commandée"  
    Case 7 To 10  
        MsgBox "Grande quantité commandée"  
    Case Is > 10  
        MsgBox "Très grande quantité commandée"  
    Case Else  
        MsgBox "Veuillez cliquer dans une cellule ayant un nombre !"  
    End Select  
End Sub
```

Pour exécuter la macro : cliquer dans une cellule avec un nombre entre 1 et 20 et lancer la macro.

MODIFICATION D'UN ENSEMBLE DE CELLULES SÉLECTIONNÉES

Répète une action sur l'ensemble des cellules sélectionnées.

Structure

```
For Each Variable In Selection  
    Actions  
Next
```

Exemple

Augmente les valeurs de 10% des cellules sélectionnées

```
Sub Augmentation()  
    Dim Cellule As Range  
    Dim Taux As Double  
  
    Taux = 0.1  
    For Each Cellule In Selection  
        Cellule = Cellule * (1 + Taux)  
    Next  
End Sub
```

Pour exécuter la macro, sélectionner des cellules contenant des nombres et lancer la macro.

LES MACROS-FONCTIONS

Une macro-fonction est un ensemble de commandes qui affichent un résultat en fonction de variables données. Une fois la fonction créée, elle s'utilise comme les fonctions de l'onglet Formules.

Dans le tableau suivant, nous souhaitons connaître le montant de la remise.

- Si les ventes sont inférieures à 10 000 €, le taux sera de deux fois le taux indiqué
- Si les ventes sont supérieures ou égales à 10 000 € le taux indiqué ne change pas.

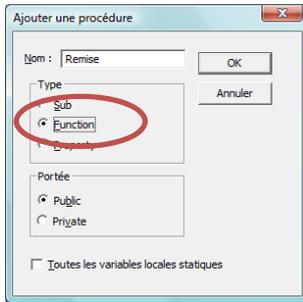
	A	B	C
1	Ventes	Taux	Remise
2	1 000,00 €	5,00%	
3	8 000,00 €	11,00%	
4	10 000,00 €	2,00%	
5	9 000,00 €	6,00%	
6	7 000,00 €	4,00%	
7	12 000,00 €	9,00%	
8	14 000,00 €	8,00%	

CRÉATION D'UNE MACRO-FONCTION

PRÉPARATION DE LA MACRO FONCTION

1. **Alt | F11** pour accéder aux modules de VBA
2. Éventuellement, insérer un nouveau module si nécessaire :
Menu *Insertion*,
Module
3. Menu *Insertion*
Procédure

- **Nom** : « Remise »
- **Type** : cocher Fonction
- **Portée** : Public



et **OK**

SAISIE DES COMMANDES

Dans Remise, saisir les variables nécessaires au calcul :

`Public Function Remise(Vente, Taux)`

- ✓ Le nom des variables importe peu, — `Remise(x,y)`, etc. — l'essentiel est de se souvenir à quoi correspond chaque variable pour la suite du calcul, d'où l'intérêt de donner des noms parlant.

Saisir les commandes du code :

```
Public Function Remise(Vente, Taux)
    If Vente >= 10000 Then
        Taux = Taux * 2
    End If
    Remise = Vente * Taux
End Function
```

- ✓ La variable *Remise* est celle qui donnera le résultat à afficher dans le tableau.

UTILISATION D'UNE MACRO-FONCTION

1. Retourner à la feuille de calcul
2. Dans la cellule C2, saisir la formule voulue :
 - =remise(
 - Cliquer dans la cellule contenant le montant des ventes (A2)
 - Point-virgule (;)
 - Cliquer dans la cellule contenant le taux (B2)

Exemple : =remise(A2;B2)

3. Valider, faire la recopie vers le bas, effectuer la mise en forme, etc.

	A	B	C	D
1	Ventes	Taux	Remise	
2	1 000,00 €	5,00%	50,00 €	
3	8 000,00 €	11,00%	880,00 €	
4	10 000,00 €	2,00%	400,00 €	
5	9 000,00 €	6,00%	540,00 €	
6	7 000,00 €	4,00%	280,00 €	
7	12 000,00 €	9,00%	2 160,00 €	
8	14 000,00 €	8,00%	2 240,00 €	

EXERCICE SUR LES MACROS-COMMANDES

OBJECTIF : Automatiser un tableau qui modifie les cellules dont les valeurs dépassent un maximum.

CRÉATION DU TABLEAU

1. Créer le tableau suivant :

	A	B
1	Maximum autorisé :	4 000
2		
3	Quantité	Remarque
4	1 500	
5	2 200	
6	3 200	
7	4 200	
8	1 800	
9	4 600	
10	2 700	
11	3 600	
12	1 100	

2. Nommer « Maximum » la cellule B1
3. Dans la cellule B4, afficher « Correct » si la quantité est inférieure ou égale à la cellule du maximum, sinon, afficher « Erreur », puis faire une recopie vers le bas.

CRÉATION DES MACROS

Création d'une macro simple qui remplace un nombre par 3000 et le met en gras.

1. Cliquer dans une cellule contenant un nombre.
2. Créer une macro appelée « Securite »
3. Saisir 3000 — ne pas valider avec la touche  mais avec le bouton de validation dans la barre des formules : 
4. Mettre le contenu de la cellule en gras.
5. Arrêter sa macro et la tester.

Création d'une macro qui ne modifie les valeurs que si celles-ci sont supérieures au maximum autorisé

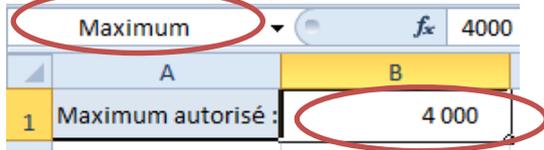
1. Remettre les chiffres d'origine, enlever le gras.
2. Modifier le code VBA précédemment créé :
Stocker le maximum autorisé dans une variable appelé *MaximumPermis* qui sera définie en *Integer*.
3. Faire une boucle avec compteur (For i=1...) qui parcourt toutes les cellules de B4 à B12 afin d'appliquer la condition suivante à chaque cellule.
4. Créer une condition logique (avec *If... Then... Else...*) : si le contenu de la cellule est supérieur au maximum autorisé, alors le contenu de la cellule est égal au maximum autorisé et le chiffre est mis en gras (pour cela il est possible de récupérer les commandes précédemment créées avec la macro) sinon, ne rien faire.
5. Créer un bouton pour lancer la macro.
6. Saisir 3000 comme maximum autorisé et lancer la macro à partir du bouton.

CORRECTION DE L'EXERCICE SUR LES MACROS-COMMANDES

CRÉATION DU TABLEAU

Pour nommer « Maximum » la cellule B1 :

1. Cliquer dans la cellule voulue B1
2. Cliquer dans la zone de nom



3. Saisir le nom voulu (exemple : *Maximum*) et

Condition logique

1. Cliquer dans la cellule B4
2. Saisir la condition logique :
=SI(A4<=Maximum;"Correct";"Erreur")
3. Valider et faire la recopie vers le bas.

The screenshot shows the Excel interface. The formula bar (top right) displays '=SI(A4<=Maximum;"Correct";"Erreur")'. Below it, a table is shown with columns 'Quantité' and 'Remarque'. The table data is as follows:

	A	B	C	D	E
1	Maximum autorisé :	4 000			
2					
3	Quantité	Remarque			
4	1 500	Correct			
5	2 200	Correct			
6	3 200	Correct			
7	4 200	Erreur			
8	1 800	Correct			
9	4 600	Erreur			
10	2 700	Correct			
11	3 600	Correct			
12	1 100	Correct			
13					

A black arrow points downwards from row 4 to row 12, indicating the copy operation.

CRÉATION DES MACROS

Macro qui remplace une valeur par 3000 et met la cellule en gras

1. Cliquer dans une cellule contenant un nombre
(à faire en dehors de la macro, sinon la macro ira chercher toujours la même cellule).
 2. Onglet *Développeur*
Enregistrer une macro
ou cliquer sur  sur la barre d'état
Nom : *Securite* et
 3. Saisir le chiffre 3000 et valider avec le bouton de validation dans la barre des formules et non avec (afin d'éviter que la macro enregistre un changement de cellule).
 4. Cliquer sur le bouton **G** pour mettre en gras.
- 👉 Surtout ne pas changer de cellule pendant l'enregistrement de la macro.

5. Cliquer sur le bouton  Arrêter l'enregistrement, dans l'onglet *Développeur* ou sur la barre d'état.
 6. Pour tester sa macro :
Cliquer sur un des nombres du tableau
Onglet *Développeur*
Macros
Cliquer sur *Securite*
Cliquer sur **Exécuter**
- ✓ Dans la cellule sélectionnée apparaît le chiffre 3000 qui est mis en gras.
Penser à enlever le gras et à remettre les chiffres d'origine pour la suite...

Boucle qui remplace les nombres qui dépassent le maximum par la valeur maximale autorisée et les laisse en gras

1. Modifier la macro existante :

Onglet *Développeur*

Macros

Cliquer sur *Securite*

Cliquer sur **Modifier**

ou

Alt | F11

Le code VBA de la macro *Invite* apparaît à l'écran

(les commentaires ajoutés ci-dessous en verts sont des indications facultatives) :

```
ActiveCell.FormulaR1C1 = "3000" ' Affiche 3000 dans la cellule sélectionnée  
Selection.Font.Bold = True ' Met la cellule sélectionnée en gras
```

2. Définition des variables (au-dessus du code existant) :

```
Dim MaximumPermis As Integer  
Dim i As Integer  
MaximumPermis = Range("Maximum")  
' MaximumPermis est égal au contenu de la cellule nommée Maximum
```

3. Création de la boucle :

```
For i = 4 To 12 ' De 4 à 12 pour aller de la ligne 4 à la ligne 12  
ActiveCell.FormulaR1C1 = "3000" ← supprimer cette ligne  
Cells(i, 1) = MaximumPermis  
' Sélectionne la cellule de la ligne i et de la colonne 1  
' pour remplacer son contenu par le maximum permis  
Cells(i, 1).Font.Bold = True  
' Remplacer « Selection » par « Cells(i,1) »  
' pour mettre en gras les cellules concernées.  
Next i
```

4. Création de la condition logique :

```
For i = 4 To 12  
If Cells(i, 1) > MaximumPermis Then  
Cells(i, 1) = MaximumPermis  
Cells(i, 1).Font.Bold = True  
End If  
Next i
```

- ✓ Explication du code VBA :

For i = 4 To 12 : De i est égal à 4 jusqu'à 12

If Cells(i, 1) > MaximumPermis Then

Si la cellule de la ligne i, colonne 1, est supérieur au maximum permis

Cells(i, 1) = MaximumPermis La cellule de la ligne i, colonne 12 est égal au maximum permis

Cells(i, 1).Font.Bold = True la cellule est mise en gras

End If Fin de la condition logique

Next i Retour au début de la boucle i

Modèle de la macro terminée :

```
Sub Securite()  
'  
' Securite Macro  
'  
    Dim MaximumPermis As Integer  
    Dim i As Integer  
  
    MaximumPermis = Range("Maximum")  
  
    For i = 4 To 12  
        If Cells(i, 1) > MaximumPermis Then  
            Cells(i, 1) = MaximumPermis  
            Cells(i, 1).Font.Bold = True  
        End If  
    Next i  
End Sub
```

Création du bouton de macro

1. Retourner dans sa feuille de calcul
2. Onglet *Insertion*
Formes
Cliquer sur la forme voulue, exemple : *Rectangle à coins arrondis*
3. En maintenant le bouton gauche de la souris enfoncé, dessiner un bouton dans la feuille de calcul
4. Saisir le texte du bouton, exemple : « Maximum Permis » et le mettre en forme (onglet Format, etc.)
5. Affecter une macro au bouton :
Clic droit sur le bouton
Affecter une macro
Cliquer sur *Securite* et **OK**

POUR VÉRIFIER :

1. Modifier la valeur maximale, exemple : 3000
 2. Cliquer dans n'importe quelle cellule de la feuille
 3. Cliquer sur le bouton de macro « Maximum permis »
- ✓ Les chiffres supérieurs au maximum sont remplacés par le maximum autorisé et mis en gras (ce qui permet de vérifier quelles cellules ont été modifiées).
Dans la colonne *remarque*, « Correct » apparaît dans toutes les cellules. ■

	A	B	C	D
1	Maximum autorisé :	3 000		
2				
3	Quantité	Remarque		
4	1 500	Correct		
5	2 200	Correct		
6	3 000	Correct		
7	3 000	Correct		
8	1 800	Correct		
9	3 000	Correct		
10	2 700	Correct		
11	3 000	Correct		
12	1 100	Correct		

