

DU BON USAGE DE L'ENREGISTREUR DE MACRO

NOTRE MEILLEUR AMI (si nous savons l'appivoiser...)

La première vocation de l'enregistreur de macro EXCEL est la création d'**automates** reproduisant une **succession d'actions** qui pourront être ensuite **reproduites** autant de fois que nécessaire...

C'est également un **outil essentiel en programmation VBA**. Grâce à lui, nous pouvons, par exemple, **explorer les objets complexes** exposés par EXCEL.

Il est nécessaire toutefois d'en bien **comprendre les possibilités et limites** pour l'utiliser **de façon efficace**.

Cette fiche « **Bonnes pratiques VBA** » présente de façon concise ce qu'il est nécessaire de connaître sur l'enregistreur de macro et sur sa bonne utilisation en programmation VBA.

(Les copies-écrans du présent document sont issues d'EXCEL version 365)


L'icône  signale une remarque réclamant une **attention particulière**.

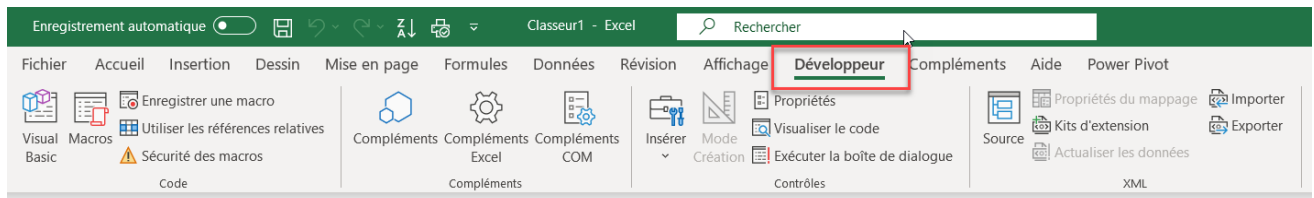
Table des matières

ASSURONS-NOUS DE LA PRESENCE DE L'ONGLET « DEVELOPPEUR »	2
DECLENCHONS L'ENREGISTREMENT D'UNE MACRO	3
STOPPONS L'ENREGISTREMENT D'UNE MACRO	5
REVISONS LE CODE VBA DE LA MACRO	6
ETUDIONS LE CODE VBA DE LA MACRO	7
SACHONS CE QUE L'ENREGISTREUR DE MACRO NE PEUT FAIRE	9
CONCLUONS : LES BONNES PRATIQUES A METTRE EN ŒUVRE	9

BONNES PRATIQUES VBA – L'enregistreur de macro

ASSURONS-NOUS DE LA PRESENCE DE L'ONGLET « DEVELOPPEUR »

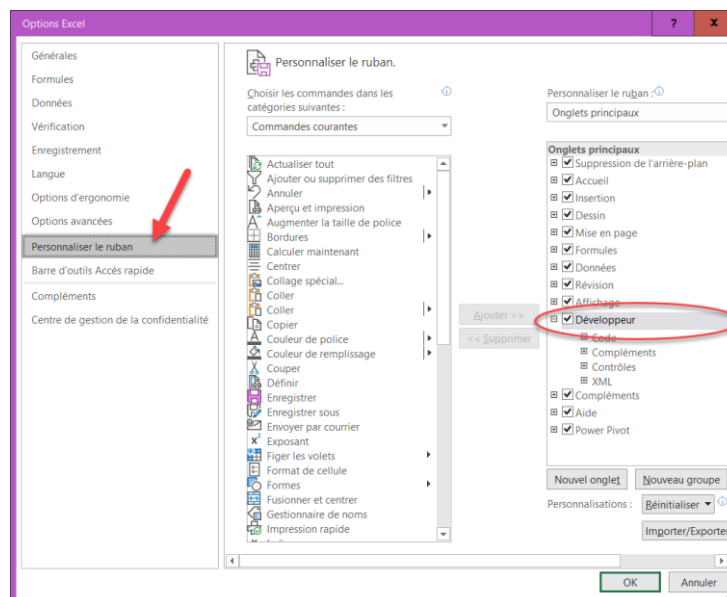
Pour accéder à l'enregistreur de macro, il est nécessaire que l'onglet « **Développeur** » soit présent dans le ruban :



Si tel n'est pas le cas, nous procédons de la façon suivante pour activer cette fonctionnalité :

Fichier/Options/Personnaliser le ruban

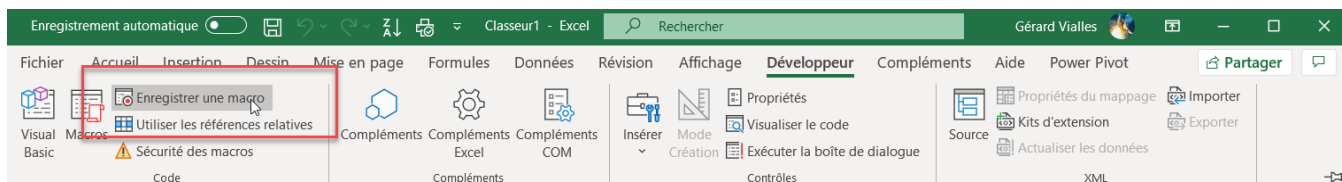
Sélectionnons la **case à cocher** devant la ligne « **Développeur** » :



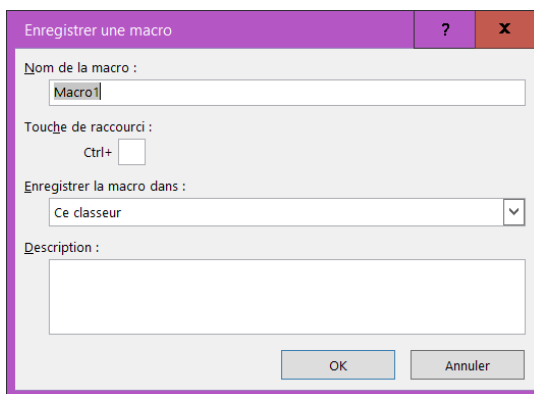
BONNES PRATIQUES VBA – L'enregistreur de macro

DECLENCHONS L'ENREGISTREMENT D'UNE MACRO

Pour déclencher l'enregistrement d'une macro, cliquons sur le bouton « Enregistrer une macro » dans l'onglet « Développeur » ou (Alt+v ;R1):



Nous devons **donner un nom** à notre macro :



EXCEL propose le nom « Macro1 » (1 pour notre première macro, 2 pour la seconde macro, etc.)

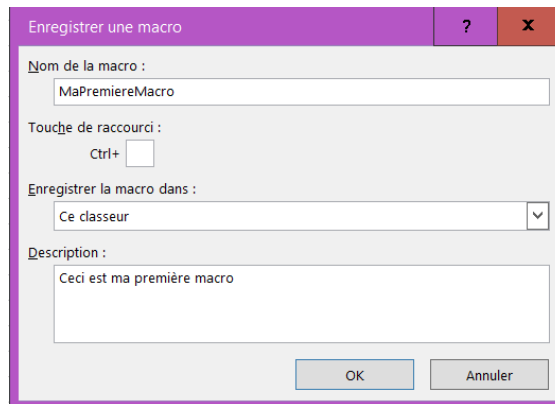
i Donner un nom le plus explicite possible à nos macros afin de les repérer plus facilement par la suite.

Les **règles de nommage** d'une macro sont les suivantes :

- Son nom doit **commencer par une lettre** (majuscule ou minuscule) ou un **trait de soulignement** (_).
- Son nom ne doit comporter **ni espace ni caractères spéciaux** (& * / ...).
- Son **nom doit être unique** dans le classeur.
- Son nom ne doit pas entrer en conflit avec un **mot clé réservé** (comme « Print »...).

BONNES PRATIQUES VBA – L'enregistreur de macro

Enregistrons, par exemple, notre macro de la façon suivante :



S'agissant d'une macro dont nous réviserons ultérieurement le code VBA, nous omettons d'affecter une « Touche de raccourci » et nous l'enregistrons dans « Ce classeur ».

Nous pouvons compléter la zone « **Description** » : elle se retrouvera en **commentaire** dans la procédure VBA générée.

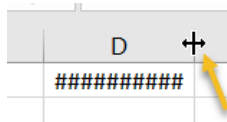


Dès le moment où nous cliquons sur le **bouton « OK »** de la boîte de dialogue « **Enregistrer une macro** », **toutes les actions** réalisées enregistrées dans notre macro : déclenchement d'une action EXCEL, sélection de cellules ou plages de cellules, frappe de caractères, insertion d'objet, etc.

BONNES PRATIQUES VBA – L'enregistreur de macro

A titre d'exercice, effectuons les actions suivantes :

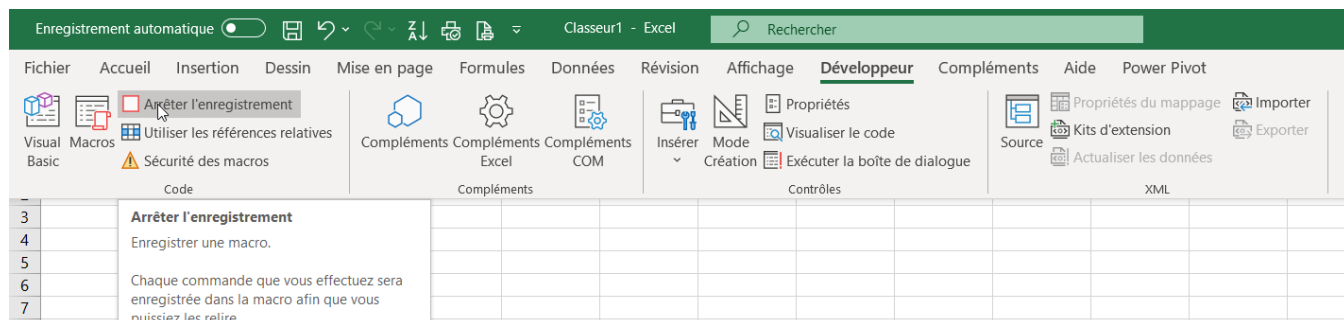
- Dans la cellule A1, entrons la formule permettant d'obtenir la date et l'heure actuelle :
« =MAINTENANT() »
- Copions la cellule dans le presse-papier (Ctrl+C)
- Déplaçons la sélection sur la cellule D1
- Collons le contenu du presse-papier (Ctrl+V)
- Redimensionnons la cellule D1 pour visualiser la date et heure courante par un double-clic sur le quadrillage de la colonne D :



- Stoppons l'enregistrement

STOPPONS L'ENREGISTREMENT D'UNE MACRO

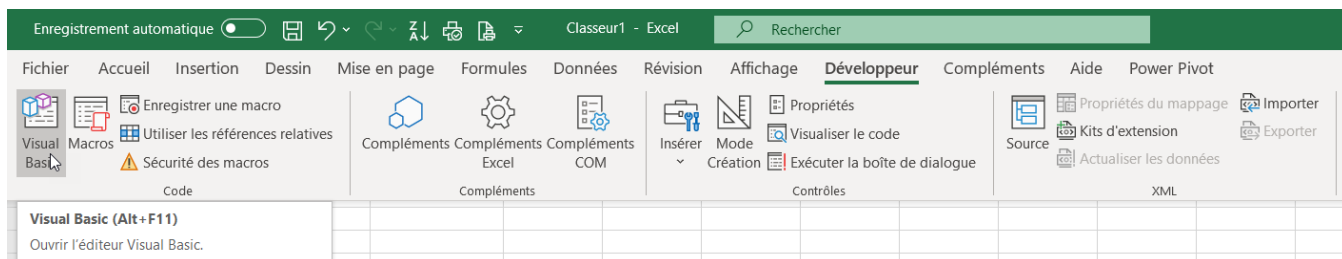
Pour stopper l'enregistrement d'une macro, cliquons sur le bouton « Arrêter l'enregistrement » dans l'onglet « Développeur » ou (Alt+v ;R1):



BONNES PRATIQUES VBA – L'enregistreur de macro

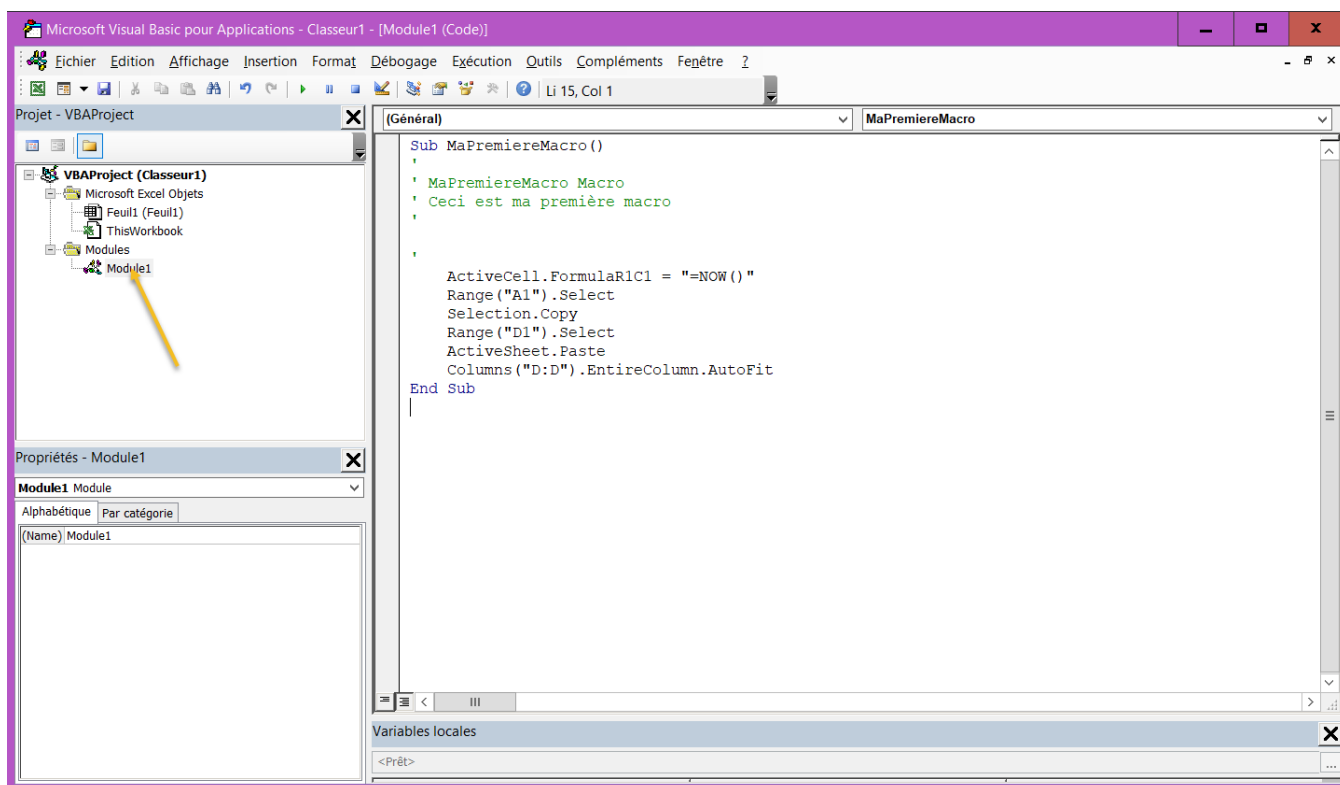
REVISONS LE CODE VBA DE LA MACRO

Pour réviser le code VBA d'une macro enregistrée, cliquons sur le bouton « Visual Basic » dans l'onglet « Développeur » ou (Alt+F11):



L'éditeur de code VBA (VBE) s'ouvre.

En sélectionnant « Modules/Module1 » nous visualisons le **code VBA généré pour notre exercice** :



Pour **une formation à la programmation VBA**, nous nous reporterons utilement au cours :

<https://www.excel-pratique.com/fr/vba>

BONNES PRATIQUES VBA – L'enregistreur de macro

ETUDIONS LE CODE VBA DE LA MACRO

Observons le code VBA généré lors de notre exercice :

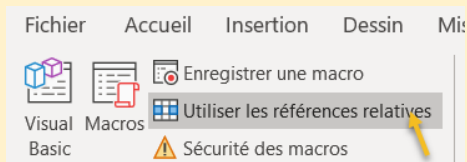
```
Sub MaPremiereMacro()  
'  
' MaPremiereMacro Macro  
' Ceci est ma première macro  
'  
'  
1 ActiveCell.FormulaR1C1 = "=NOW()"  
2 Range("A1").Select  
3 Selection.Copy  
4 Range("D1").Select  
5 ActiveSheet.Paste  
6 Columns("D:D").EntireColumn.AutoFit  
End Sub
```

- La **description** de la macro a été reproduite dans l'entête
- Ligne 1 : Le code se réfère à « **ActiveCell** » pour entrer la formule.
- Ligne 1 : La formule « MAINTENANT() » a été **traduite en sa forme anglaise**.
(Un outil de traduction français/anglais des formules EXCEL est disponible ici : https://www.excel-pratique.com/fr/index_des_fonctions/traduction_formule).
- Ligne 3 : La copie de la cellule est réalisée au moyen de **la propriété « Selection »**.
- Ligne 6 : Le redimensionnement de la colonne « D » est réalisé au moyen de **la fonction « Autofit »**.



Par défaut, l'enregistreur de macro, réfère les cellules et les plages de cellules par de leur **références absolues**.

Il est possible de demander à l'enregistreur de macro de procéder par **références relatives** en activant le choix « **Utiliser les références relatives** » de l'onglet « **Développeur** » :



BONNES PRATIQUES VBA – L'enregistreur de macro

Tirons les leçons de nos observations :

- Ligne 1 : dans notre exercice, nous n'avons pas positionné explicitement la sélection sur la cellule A1. Du fait que l'enregistreur utilise ActiveCell, dans la **révision du code VBA**, nous devons nous assurer que **la cellule voulue est sélectionnée**.
- Ligne 3 : l'enregistreur met en œuvre les **méthodes** de la **propriété « Selection »**. Cette méthode n'est pas optimale ni sur le plan de la **fiabilité** ni **des performances**. Nous préférons faire appel aux **méthodes équivalentes des objets « Range »**.

Ainsi, il serait préférable de **modifier le code VBA** généré par l'enregistreur de macro. Par exemple, de la façon suivante :

```
Sub MaPremiereMacro()  
'  
' MaPremiereMacro Macro  
' Ceci est ma première macro  
"  
    Set oRange = ActiveSheet.Range("A1")  
    oRange.FormulaR1C1 = "=NOW()"  
    oRange.Copy  
  
    Range("D1").Select  
    ActiveSheet.Paste  
    Columns("D:D").EntireColumn.AutoFit  
End Sub
```

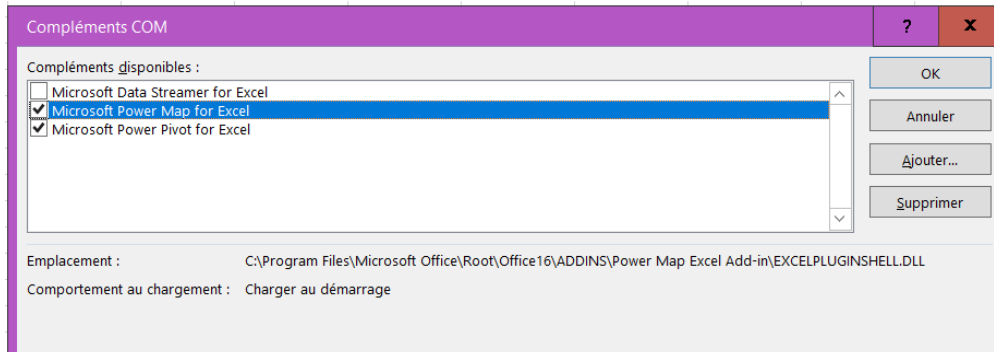
Comme nous le voyons, même sur une séquence extrêmement simple, nous pouvons **apporter des améliorations** au code généré par l'enregistreur de macro. L'expérience prouve que, lors de l'enregistrement de macro, nous réalisons **des actions inutiles, nous tâtonnons** et il nous arrive de **ne pas stopper à temps l'enregistreur**. Tout ceci a pour effet de générer du **code VBA inutile** (voire néfaste) que **nous éliminerons** lors de notre révision de code.

BONNES PRATIQUES VBA – L'enregistreur de macro

SACHONS CE QUE L'ENREGISTREUR DE MACRO NE PEUT FAIRE

L'enregistreur de macro ne nous sera pas d'un grand secours dans les situations suivantes :

- Si nous utilisons les « **Compléments COM** » :



L'enregistreur n'est **pas en mesure de générer de code**.

- Nous ne pouvons pas **passer d'arguments** à la procédure générée.
- Nous ne pouvons pas concevoir les **fonctions personnalisées**.
- Nous ne pouvons pas créer de code se déclenchant **sur des évènements**.
- Nous ne pouvons pas **créer de boucles** (For Next, Do While, Do Until, For each Next).
- Nous ne pouvons pas **gérer des conditions** (If-Then-Else, Select Case).
- L'enregistreur de macro ne génère aucune **déclarations explicites**.

CONCLUONS : LES BONNES PRATIQUES A METTRE EN ŒUVRE

- Utilisons l'enregistreur de macro **aussi souvent que nécessaire**. C'est un moyen rapide pour explorer les objets EXCEL, leur collections, méthodes, propriétés.
(Un autre moyen, encore plus rapide : le pur copier/coller de code. Mais, dans ce cas aussi, il est indispensable de se l'approprier : « science sans conscience n'est que ruine de l'âme » nous rappelle Rabelais).
- N'enregistrons **pas de séquences trop longues** et privilégions des sujets ciblés.
- **Révisons systématiquement** le code généré.