

# Automatiser des tâches dans les applications Microsoft Office VBA (Visual Basic for Applications) et Office Script

Samir DELIMI

# Office Script et VBA (Visual Basic for Applications)

- **Office Script et VBA (Visual Basic for Applications)** sont deux langages utilisés pour automatiser des tâches dans les applications Microsoft Office (comme Excel), mais ils diffèrent à plusieurs niveaux.
- **Passage au cloud** : Avec l'augmentation des services basés sur le cloud, comme Microsoft 365 et Excel Online, il y avait un besoin croissant de solutions d'automatisation qui fonctionnent directement dans un environnement cloud. VBA, étant une technologie de bureau, n'était pas compatible avec les versions web d'Office.

# Office Script et VBA (Visual Basic for Applications)

- **VBA** est basé sur Visual Basic, un langage de programmation plus ancien, qui existe depuis **les années 1990**.
- **Office Script** a été lancé pour Excel en ligne en **décembre 2019**, en tant qu'outil d'automatisation moderne, principalement pour les utilisateurs de **Microsoft 365**.
- Office Script a été développé par Microsoft pour répondre aux **nouveaux besoins** liés à l'évolution des outils de bureautique vers des **solutions (applications) en ligne** et **basées sur le cloud**.
- Office Script est une alternative moderne à VBA, mieux adaptée aux besoins actuels d'automatisation dans des environnements distribués et collaboratifs.

# Office Script :

- **Langage** : Office Script utilise **TypeScript**, un sur-ensemble de **JavaScript**, pour écrire des scripts.
- **Exécution** : Office Script fonctionne principalement **dans la version web d'Excel (Excel Online)** et est accessible via l'interface **Power Automate** pour l'automatisation.
- **Utilisation cloud** : Les scripts Office Script sont adaptés pour travailler dans des **environnements cloud (comme Microsoft 365)**. Ils peuvent être exécutés de **manière asynchrone**, c'est-à-dire qu'ils sont bien intégrés aux flux de travail automatisés dans le cloud.
- **Accessibilité** : Office Script est plus récent et conçu pour répondre aux besoins modernes d'automatisation dans un environnement cloud.

# VBA (Visual Basic for Applications) :

- **Langage** : VBA est basé sur Visual Basic, un langage de programmation plus ancien, qui existe depuis les années 1990.
- **Exécution** : VBA s'exécute principalement dans les **versions de bureau des applications Office** (Excel, Word, Access, etc.). Il ne fonctionne pas dans la version web.
- **Capacités locales** : VBA **est plus puissant pour des tâches complexes sur un ordinateur local**, car il peut interagir directement avec le système d'exploitation et d'autres applications installées sur l'ordinateur.
- **Popularité et support** : VBA a été largement utilisé pendant des décennies, et bien qu'il soit encore supporté dans les applications de bureau d'Office, **il est en quelque sorte en train d'être remplacé par des solutions plus modernes comme Office Script pour les environnements en ligne.**

# Différences clés :

- **Langage de programmation** : Office Script utilise TypeScript (ou JavaScript), tandis que VBA utilise Visual Basic.
- **Plateforme** : Office Script est conçu pour fonctionner dans le cloud et les versions en ligne d'Office, alors que VBA est destiné aux applications de bureau.
- **Utilisation future** : Microsoft pousse Office Script pour les nouvelles automatisations, surtout dans les environnements de collaboration en ligne, tandis que VBA reste pertinent pour les automatisations de bureau complexes.

# Pourquoi Office Script a été créé :

- **1- Passage au cloud :** Avec l'augmentation des services basés sur le cloud, comme Microsoft 365 et Excel Online, il y avait un besoin croissant de solutions d'automatisation qui fonctionnent directement dans un environnement cloud. VBA, étant une technologie de bureau, n'était pas compatible avec les versions web d'Office.

# Pourquoi Office Script a été créé :

- **2- Automatisation moderne** : Les entreprises modernes utilisent de plus en plus des flux de travail automatisés via des plateformes comme Power Automate et PowerApps. Office Script a été conçu pour être intégré à ces plateformes, permettant aux utilisateurs d'automatiser des tâches à grande échelle et de manière asynchrone, en s'appuyant sur le cloud.



# Pourquoi Office Script a été créé :

- **3- Langage moderne et flexible** : Office Script utilise TypeScript, un langage moderne qui est largement adopté par les développeurs. Contrairement à VBA (basé sur Visual Basic), TypeScript/Javascript est plus courant dans les environnements de développement actuels, ce qui facilite l'intégration d'Office Script dans les workflows des entreprises déjà familières avec ces technologies.

# Pourquoi Office Script a été créé :

- **4- Accessibilité accrue** : VBA est puissant mais limité aux utilisateurs des versions de bureau d'Office. Avec Office Script, Microsoft visait à offrir une solution d'automatisation accessible à tous ceux qui utilisent Excel Online, que ce soit pour des tâches quotidiennes ou des workflows plus complexes dans un environnement collaboratif.

# Convertir des scripts VBA (Visual Basic for Applications) en Office Scripts pour Excel

Est-ce que je peux convertir des scripts vba vers office script?

- Il est possible de convertir des scripts VBA (Visual Basic for Applications) en Office Scripts pour Excel, mais ce processus nécessite quelques ajustements manuels, car les deux langages sont différents :
  - **VBA** est basé sur **Visual Basic** , utilisé historiquement dans les applications Office pour automatiser les tâches.
  - **Office Scripts** utilise **TypeScript** (une version de **JavaScript**) et est principalement destiné à **Excel Online** dans **Microsoft 365**

.

# Processus de conversion :

- 1. Analyse du code VBA :** Examinez chaque section du code pour comprendre sa logique et les fonctionnalités utilisées.
- 2. Traduction en TypeScript :**
  1. Les structures de contrôle (boucles, conditions) doivent être adaptées à la syntaxe TypeScript.
  2. Les objets Excel (comme les cellules, les feuilles et les plages) sont manipulés différemment.
- 3. Utilisation de l'API Office Scripts :** Office Scripts dispose de son propre objet **ExcelScript** pour interagir avec Excel.

# Exemple de conversion simple :

## Code VBA :

- Sub ChangeCellValue()
- Dim ws As Worksheet
- Set ws = ThisWorkbook.Sheets("Feuil1")
- ws.Range("A1").Value = "Hello World"
- End Sub

## Script de bureau équivalent :

- function main(workbook: ExcelScript.Workbook) {
- let sheet = workbook.getWorksheet("Feuil1");
- sheet.getRange("A1").setValue("Hello World");
- }

# Explication de l'exemple: Script TypeScript pour Excel utilisant l'API d'Office Scripts

- **Fonctionnement du script :**

1. **Déclaration de la fonction principale :**

```
function main(workbook: ExcelScript.Workbook)
```

- La fonction main est le point d'entrée pour les scripts d'Office Scripts dans Excel.
- Elle prend un paramètre workbook qui représente le classeur Excel actif sur lequel le script s'exécute.

1. **Accéder à une feuille spécifique :**

```
let sheet = workbook.getWorksheet("Feuil1");
```

- La méthode getWorksheet est utilisée pour obtenir une feuille appelée "Feuil1" dans le classeur.
- Si cette feuille n'existe pas, une erreur sera générée.

1. **Définir une valeur dans une cellule :**

2. 

```
sheet.getRange("A1").setValue("Hello World");
```

- La méthode getRange("A1") cible une cellule spécifique, ici la cellule A1.
- La méthode setValue("Hello World") définit la valeur de cette cellule comme étant "Hello World".

- **Résultat :**

- **Lorsque le script est exécuté : Il ouvre la feuille "Feuil1" dans le classeur. Il écrit le texte "Hello World" dans la cellule A1.**

- **Conditions à vérifier :**

- Si "Feuil1" n'existe pas dans le classeur, une erreur sera levée. Pour éviter cela, vous pouvez d'abord vérifier si la feuille existe ou la créer.
- Souhaitez-vous une adaptation ou des améliorations à ce script ?

# Points importants :

## 1. Structures syntaxiques :

- VBA utilise des commandes comme If...Then, For Each, etc.
- Office Scripts utilise une syntaxe **JavaScript/TypeScript** (if, for, etc.).

## 2. Objets Excel :

- VBA utilise des objets comme Range, Cells, Sheets.
- Office Scripts utilise des méthodes comme `getRange()`, `getCell()`, `getWorksheet()`.

## 3. Compatibilité :

- Office Scripts fonctionne uniquement sur **Excel Online** (Microsoft 365) et n'est pas pris en charge dans Excel Desktop.
- Les fonctionnalités VBA avancées ne sont pas toutes disponibles dans Office Scripts.



- **En résumé,** Office Script est une réponse à l'évolution des outils de bureautique vers des solutions en ligne et basées sur le cloud, en offrant une alternative moderne à VBA, mieux adaptée aux besoins actuels d'automatisation dans des environnements distribués et collaboratifs.



