

Déroulé de l'action

Modalités

Session Intra/Inter
En présentiel/Classe virtuelle

Horaires

9H00-12H30 /13H30-17H00

Méthode pédagogique

Alternance exposés théoriques
et exercices pratiques
(80% de pratique)

Suivi et assistance

Support de cours adapté
au logiciel étudié et
au niveau suivi
Assistance téléphonique
gratuite et illimitée

Modalité d'évaluation

Passage de la certification Excel
TOSA en fin de formation
Attestation de stage
Emargement quotidien d'une
feuille de présence

Accessibilité aux personnes handicapées

Pour tout besoin d'adaptation,
retrouver le contact de notre
réfèrent handicap et les
modalités d'accueil sur la page :

[Infos pratiques/Situation de Handicap](#)

Excel VBA - Automatiser des processus dans les applications Microsoft Office avec VBA -TOSA

Public et Objectif : Toute personne qui souhaite maîtriser le langage Visual Basic pour développer des applications sous Excel.

À l'issue de la formation, le stagiaire sera capable de :

- Automatiser des tâches sous Excel
- Développer des procédures sous VBA
- Déboguer les programmes

Pré-requis : Posséder un bon niveau sur les fonctions avancées d'Excel

- 3 jours -

Comprendre le langage Visual Basic

- Généralités sur le langage Visual Basic Application
- Généralités sur la programmation événementielle
- Intérêts d'utiliser VBA dans les applications Office : limites des macros préenregistrées

Créer des macro-commandes simples

- Utiliser l'enregistreur de Macro
- Création d'une macro commandes
- Enregistrement et modification de macros
- Affectation une macro à un bouton ou un objet
- Affecter une macro à un Menu ou une barre d'outils
- Enregistrer une macro dans le classeur de Macros personnelles : perso.xls

Se familiariser avec Visual Basic Editor

- Visualiser et éditer une macro dans l'éditeur de Macros
- Présentation des différentes fenêtres de l'interface
- Lire et écrire du code VBA
- Comprendre la notion d'objets et acquérir les principales méthodes

Utiliser les instructions du langage VBA

- Définition et rôle des fonctions et des procédures
- Déclaration et utilisation de variables
- Constantes
- Les instructions conditionnelles (If...Then, Select Case)
- Les boucles (For...Next, Do...Loop, While et Until)

Comprendre l'utilisation des objets Excel

- Définition d'un objet et intérêt de la programmation orientée objet
- Notion de collections, propriétés, méthodes et événements
- Manipuler les feuilles, les classeurs
- Utilisation des instructions For Each...Next, With...End with

Gérer des erreurs et déboguer

- La structure On Error Go to
- Utilisation de l'objet Err
- Savoir tracer et déboguer un programme

Créer un interface graphique utilisateur : les boîtes de dialogue

- Utilisation des boîtes de dialogue standards (msgbox et inputbox)
- Création de boîtes de dialogue personnalisées avec l'objet Userform
- Utilisation des différents contrôles existants
- Création et manipulation des menus et barres d'outils

Utiliser des modèles objets Excel

- Objets spécifiques Excel

Déroulé de l'action

- Modalités**

Session Intra/Inter
En présentiel/Classe virtuelle

- Horaires**

9H00-12H30 /13H30-17H00

- Méthode pédagogique**

Alternance exposés théoriques
et exercices pratiques
(80% de pratique)

- Suivi et assistance**

Support de cours adapté
au logiciel étudié et
au niveau suivi
Assistance téléphonique
gratuite et illimitée

- Modalité d'évaluation**

Passage de la certification Excel
TOSA en fin de formation
Attestation de stage
Emargement quotidien d'une
feuille de présence

- Accessibilité aux personnes handicapées**

Pour tout besoin d'adaptation,
retrouver le contact de notre
réfèrent handicap et les
modalités d'accueil sur la page :
[Infos pratiques/Situation de Handicap](#)

Excel VBA - Automatiser des processus dans les applications Microsoft Office avec VBA -TOSA

Travaux pratiques

Pour chaque point du programme abordé :

Présentation et démonstration par le formateur, mise en pratique sur les fonctionnalités abordées par le stagiaire avec l'appui du formateur et du groupe, feedbacks du formateur tout au long de l'activité.

- Création de diverses Macros exemples (Macros d'extraction, de consolidation, d'ouverture, de fermeture....)

Passage de la certification TOSA

[Certification Automatiser des processus dans les applications Microsoft Office avec VBA \(Tosa\) RS6963](#), certificateur Isograd, certification délivrée le 18/12/2024

En fin de formation, le passage de la certification est organisé autour d'un temps de validation au moyen d'un test à passer.

Compétences attestées :

Domaine 1 : Gestion des objets VBA

1.1 Manipuler les objets Excel : automatiser les processus dans Excel en utilisant et en gérant les principaux objets tels que Workbooks (cahiers de travail), Worksheets (feuilles de calcul), Ranges (pages de cellules), et Charts (graphiques), afin de simplifier les tâches spécifiques et optimiser la gestion des données. 1.2 Gérer les collections d'objets: gérer les groupes d'objets similaires dans Excel, comme les collections Worksheets ou Cells, en parcourant, ajoutant ou supprimant des éléments de la collection, et en utilisant des boucles afin d'automatiser les opérations répétitives sur chaque élément de la collection. 1.3 Interfacer avec d'autres applications: utiliser VBA pour interagir avec d'autres programmes Microsoft 365 tels que Word, Access et Outlook, afin d'automatiser des tâches inter-applications, comme l'exportation de données depuis Excel vers Word ou l'envoi d'emails via Outlook, en exploitant les objets COM (Component Object Model) pour fluidifier les processus.

Domaine 2 : Procédures VAB

2.1 Écrire des procédures : créer des sous-routines et des fonctions en VBA pour exécuter des tâches spécifiques, en utilisant des arguments et des paramètres pour passer des données entre les procédures, afin d'améliorer la modularité et la réutilisabilité du code. 2.2 Traiter les erreurs et exceptions : mettre en place des mécanismes de gestion des erreurs en utilisant des structures et des blocs conditionnels pour anticiper et traiter les erreurs, et en développant des procédures de nettoyage et de récupération après une erreur, afin de rendre le code plus robuste et moins sujet aux plantages. 2.3 Améliorer la performance du code : optimiser les routines VBA en minimisant les accès aux objets Excel, en utilisant des types de données appropriés, en optimisant les boucles et en gérant la mémoire de manière efficace, afin qu'elles s'exécutent plus rapidement et de manière plus performante.

Domaine 3 : Boîtes de dialogue, formulaires et contrôles

3.1 Concevoir des formulaires : créer des UserForms (formulaires utilisateur) en VBA en configurant l'interface utilisateur, en ajoutant des contrôles tels que des boutons, des étiquettes et des zones de texte, et en gérant leur disposition ainsi que leurs propriétés, afin de faciliter l'interaction avec l'utilisateur. 3.2 Implémenter des contrôles ActiveX : ajouter et gérer des contrôles ActiveX dans les formulaires ou directement dans les feuilles de calcul Excel, en insérant des contrôles tels que des boutons, des cases à cocher et des listes déroulantes, afin d'améliorer l'interactivité et la fonctionnalité des applications. 3.3 Gérer les événements de formulaires : écrire du code pour répondre aux actions de l'utilisateur sur les formulaires, en utilisant les événements associés aux contrôles et aux formulaires, afin de déclencher des actions spécifiques et d'améliorer l'interactivité des applications.

Domaine 4 : Environnement VBA et utilisation des outils de débogage

4.1 Utiliser les outils de débogage : connaître et utiliser les outils intégrés dans l'IDE VBA, tels que les points d'arrêt, la fenêtre d'exécution, la surveillance des variables et l'exécution pas à pas du code, afin d'identifier et de corriger les erreurs efficacement. 4.2 Analyser et suivre le code : utiliser les fonctionnalités de traçage et de journalisation en employant des instructions Debug.Print pour afficher des messages dans la fenêtre Exécution et en analysant les flux d'exécution, afin de comprendre le comportement du code, identifier les problèmes potentiels et optimiser les routines. 4.3 Optimiser l'environnement de développement : configurer l'IDE VBA en personnalisant l'éditeur, en gérant les références de projet, en utilisant des modèles de code et en mettant en place un environnement structuré, afin de maximiser l'efficacité du développement et faciliter la gestion des projets VBA.