

|              |                            |
|--------------|----------------------------|
| Référence    | <b>4-PY-PYBA</b>           |
| Durée        | <b>4 jours (28 heures)</b> |
| Éligible CPF | <b>OUI</b>                 |
| Mise à jour  | <b>27/07/2026</b>          |

## Programmer et automatiser des tâches avec Python



### OBJECTIFS PEDAGOGIQUES

- Installer et configurer un environnement de développement Python.
- Écrire des scripts Python en respectant les règles de syntaxe, utiliser les variables, types de données, conditions et boucles.
- Créer des fonctions et organiser un programme en modules, manipuler les principales structures de données.
- Lire, créer et modifier des fichiers texte, CSV et JSON.
- Développer des programmes orientés objet, gérer les erreurs et déboguer un programme.
- Utiliser les principaux outils de gestion des dépendances, développer une application répondant à un besoin professionnel.
- Optimiser la qualité et les performances du code.
- Se préparer à la certification TOSA Python RS6962.



### PUBLIC CONCERNE

Cette formation s'adresse à toute personne souhaitant acquérir ou développer des compétences en programmation Python.



### PREREQUIS

Utilisation courante d'un ordinateur  
Bonne maîtrise de Windows ou macOS  
Notions de logique informatique recommandées



### MOYENS PEDAGOGIQUES

- Réflexion de groupe et apports théoriques du formateur
- Travail d'échange avec les participants sous forme de réunion-discussion
- Utilisation de cas concrets issus de l'expérience professionnelle
- Validation des acquis par des questionnaires, des tests d'évaluation, des mises en situation et des jeux pédagogiques.
- Remise d'un support de cours.



### MODALITES D'EVALUATION

- Feuille de présence signée en demi-journée,
- Evaluation des acquis tout au long de la formation,
- Questionnaire de satisfaction,
- Attestation de stage à chaque apprenant,
- Positionnement préalable oral ou écrit,
- Evaluation formative tout au long de la formation,
- Evaluation sommative faite par le formateur ou à l'aide des certifications disponibles



### MOYENS TECHNIQUES EN PRESENTIEL

Accueil des stagiaires dans une salle dédiée à la formation équipée à minima d'un vidéo projecteur et d'un tableau blanc et/ou paperboard.

Pour les formations nécessitant un ordinateur, un PC est mis à disposition de chaque participant.



### MOYENS TECHNIQUES EN DISTANCIEL

A l'aide d'un logiciel (Teams, Zoom...), d'un micro et éventuellement d'une caméra les apprenants interagissent et communiquent entre eux et avec le formateur.

Sessions organisées en inter comme en intra entreprise.

L'accès à l'environnement d'apprentissage ainsi qu'aux preuves de suivi et d'assiduité (émargement, évaluation) est assuré.

Pour toute question avant et pendant le parcours, assistance technique à disposition au 04 67 13 45 45.



### ORGANISATION

Délai d'accès : 5 jours ouvrés  
(délai variable en fonction du financeur)

Les cours ont lieu de 9h à 12h30 et de 13h30 à 17h



### ACCESSIBILITE

Les personnes en situation d'handicap sont invitées à nous contacter directement, afin d'étudier ensemble les possibilités de suivre la formation.

Pour tout renseignement, notre référent handicap reste à votre disposition : mteyssedou@ait.fr



### PROFIL FORMATEUR

Nos formateurs sont des experts dans leurs domaines d'intervention. Leur expérience de terrain et leurs qualités pédagogiques constituent un gage de qualité.



### CERTIFICATION POSSIBLE

**TOSA – RS6962 - Programmer et automatiser des tâches avec Python** – Enregistré le 18/12/2024

Certificateur : **ISOGRAD**

# Programmer et automatiser des tâches avec Python

## MODULE 1 – DECOUVRIR PYTHON ET SON ENVIRONNEMENT

- ☐ Installation de Python
- ☐ IDE (VS Code)
- ☐ Interpréteur
- ☐ Premiers scripts
- ☐ Variables
- ☐ Commentaires
- ☐ Opérateurs
- ☐ Types de données
- ☐ Travaux pratiques

## MODULE 2 – LANGAGE ET SYNTAXE

- ☐ Utiliser les structures conditionnelles
- ☐ Créer des boucles
- ☐ Gérer les exceptions
- ☐ Ecrire un code lisible
- ☐ Utiliser les opérateurs
- ☐ Manipuler les chaînes de caractères
- ☐ Travaux pratiques

## MODULE 3 – FONCTIONS

- ☐ Créer des fonctions
- ☐ Utiliser les paramètres
- ☐ Retourner des valeurs
- ☐ Comprendre la portée des variables
- ☐ Utiliser les fonctions lambda
- ☐ Documenter son code
- ☐ Travaux pratiques

## MODULE 4 – STRUCTURES DE DONNEES

- ☐ Listes
- ☐ Tuples
- ☐ Dictionnaires
- ☐ Ensembles
- ☐ Compréhensions de listes
- ☐ Optimisation des traitements.
- ☐ Travaux pratiques

## MODULE 5 – PROGRAMMATION ORIENTEE OBJET

- ☐ Créer des classes
- ☐ Créer des objets
- ☐ Utiliser les constructeurs
- ☐ Appliquer l'héritage
- ☐ Utiliser le polymorphisme
- ☐ Organiser une architecture objet
- ☐ Travaux pratiques

## MODULE 6 – MODULES ET PACKAGES

- ☐ Importer des modules
- ☐ Créer ses propres modules
- ☐ Créer des packages
- ☐ Utiliser pip
- ☐ Environnements virtuels
- ☐ Bonnes pratiques de structuration d'un projet
- ☐ Travaux pratiques

## MODULE 7 – MANIPULATION DE DONNEES

- ☐ Lecture et écriture de fichiers
- ☐ Csv
- ☐ Json
- ☐ Découverte de pandas
- ☐ Nettoyage de données
- ☐ Automatisation de traitements
- ☐ Travaux pratiques

## MODULE 8 – OPTIMISATION ET QUALITE DU CODE

- ☐ Déboguer
- ☐ Profiler un programme
- ☐ Améliorer les performances
- ☐ Créer des tests unitaires
- ☐ Utiliser pytest
- ☐ Appliquer les bonnes pratiques de développement
- ☐ Travaux pratiques

## MODULE 9 – PROJET PROFESSIONNEL

- ☐ Fonctions
- ☐ Structures de données
- ☐ Programmation orientée objet
- ☐ Modules
- ☐ Automatisation
- ☐ Optimisation