

PYTHON PRÉSENTIEL



DURÉE

7 heures / 14 jours
ou
17 heures / 30 jours
selon formule

HORAIRES DES SESSIONS

À définir selon la
disponibilité du
stagiaire

LIEU DE FORMATION

Sur le lieu de travail
ou à domicile

PRIX (TTC)

À partir de 1790€



OBJECTIFS PÉDAGOGIQUES

- Acquérir une **connaissance approfondie des concepts de base de Python**, y compris la **syntaxe, les structures de contrôle, les types de données, et les fonctions**
- Développer des compétences dans la **programmation orientée objet, y compris la création de classes l'instanciation d'objets, l'héritage, et la polymorphie**
- Apprendre à utiliser efficacement les **structures de données intégrées de Python, telles que les listes, les dictionnaires, les ensembles, et les tuples, pour résoudre des problèmes complexes**
- Se familiariser avec les **bibliothèques et modules Python couramment utilisés, tels que NumPy pour le calcul numérique, pandas pour la manipulation des données, et matplotlib pour la visualisation**
- Appliquer les connaissances acquises à **la résolution de problèmes réels, en développant des projets pratiques pour renforcer la compréhension et les compétences pratiques en programmation Python**



PUBLIC CIBLÉ

Salariés en poste, demandeurs d'emploi, entreprises, particuliers
La certification TOSA s'adresse à des professionnels exerçant dans le domaine informatique mais aussi à d'autres professionnels au contact de ces technologies : biologiste devant agréger des résultats, analyste métier qui souhaite mettre en valeur une mesure de performance, responsable marketing cherchant à trouver des tendances...



PRÉREQUIS

- Connaissances en informatique
- PC
- Webcam
- Micro



A/ CONTENU DU PROGRAMME

Ce programme s'inscrit dans le cadre de la préparation à la certification TOSA Python.

Selon l'audit réalisé auprès du stagiaire en amont de sa formation, un programme adapté à ses besoins pourra alors être mis en place.

Introduction au langage Python

- Introduction à Python
- L'environnement de travail Python
- Créer un fichier et l'exécuter avec Python



Se familiariser avec les **bases du langage de programmation Python – acquérir une base solide en se préparant à explorer des concepts plus avancés et à développer des applications Python plus complexes.**

Notions fondamentales en Python

- Mots-clés et identificateurs en Python
- Instructions, indentation et commentaires en Python
- Les principaux types de données
- Présentation des variables
- Entrée / sortie en Python : fonctions `input()` et `print()`



Développer une **base solide de connaissances et acquérir les compétences et les connaissances de base nécessaires pour développer des applications Python simples et comprendre des concepts plus avancés.**

Les structures de contrôle

- Les opérateurs en Python
- Les instructions if-else en Python
- Les boucles en Python
- Créer des conditions Python complexes
- Convertir une boucle for imbriquée en une boucle for simple



Apprendre à **gérer le flux d'exécution d'un programme en utilisant des structures de contrôle telles que les boucles et les instructions conditionnelles – acquérir les compétences nécessaires pour écrire un code Python robuste et logique en utilisant des structures de contrôle de manière efficace.**

FORMATEXIA

info@formatexia.com

www.formatexia.com

Tél : 01 76 40 35 08



Les fonctions

- Introduction aux fonctions Python
- Définition et syntaxe des fonctions
- Notions avancées sur les paramètres et arguments
- Arguments *args et **kwargs
- Contrôle des valeurs de retour d'une fonction
- La portée des variables en Python
- La fonction yield
- Quelques fonctions utiles



Apprendre à **définir, utiliser et comprendre les fonctions en Python – avoir une solide compréhension des concepts fondamentaux de la programmation fonctionnelle en Python afin de construire des applications plus complexes et modulaires.**

Les différents types de données

- Les chaînes de caractères
- Les listes
- Les tableaux - module array
- Les tuples
- Les ensembles
- Les dictionnaires
- Les matrices
- Récapitulatif



Se familiariser avec les **types de données fondamentaux disponibles dans le langage – acquérir les compétences – nécessaires pour choisir et manipuler efficacement les types de données appropriés dans leurs programmes Python.**

Manipulation de fichiers avec Python

- Introduction à la manipulation de fichiers
- Opérations sur les fichiers
- Échange de données avec le module JSON
- Autres méthodes pour la gestion des fichiers



Apprendre à travailler avec **des fichiers, lire et écrire des données, et effectuer des opérations de gestion de fichiers en utilisant le langage de programmation Python – gérer efficacement les données stockées dans des fichiers, un aspect essentiel de nombreux programmes et applications.**

FORMATEXIA

info@formatexia.com

www.formatexia.com

Tél : 01 76 40 35 08



La gestion des erreurs en Python

- Introduction à la gestion d'erreurs ou d'exceptions
- Gérer les exceptions avec try, except, else et finally
- Liste des exceptions



Apprendre à **identifier, gérer et traiter les erreurs qui peuvent survenir lors de l'exécution d'un programme Python – développer des compétences solides pour anticiper, identifier et gérer efficacement les erreurs dans les programmes Python.**

Modules standards et paquets Python

- Les modules et paquets
- Modules math, random et statistics
- Les modules datetime, time et calendar
- Module re : expressions régulières ou rationnelles



Apprendre à **organiser et structurer les codes en utilisant les modules et les paquets de la bibliothèque standard de Python – développer des compétences organisationnelles et les utiliser efficacement la bibliothèque standard de Python pour créer des applications modulaires et réutilisables.**

POO : Notions fondamentales

- Introduction à la Programmation Orientée Objet en Python
- Constructeurs et destructeurs
- Classes, objets et attributs
- Gérer la visibilité des membres de classes



Apprendre les **principes et les concepts de base de la POO – acquérir les compétences nécessaires pour concevoir et développer des programmes orientés objet de manière efficace et structurée.**



POO : Notions avancées

- Héritage
- Polymorphisme
- Surcharge des opérateurs
- Itérateurs et générateurs
- Listes des méthodes utiles



Approfondir ses **connaissances abordant des concepts plus avancés et des techniques spécifiques à la POO – maîtriser des concepts plus complexes et utiliser la POO de manière avancée dans les projets de développement.**

Sujets avancés

- Les métaclasses
- Les décorateurs
- Les itérateurs avancés
- Instruction assert
- Conclusion



Approfondir les **compétences au-delà des concepts fondamentaux – développer une expertise approfondie dans des aspects spécifiques du langage Python afin d'aborder des défis complexes et travailler sur des projets avancés.**

B/ MOYENS PÉDAGOGIQUES

La formation en présentiel permettra à l'apprenant de développer et renforcer ses compétences en travaillant avec le formateur expert dans son domaine.

L'apprenant peut ainsi bénéficier d'une formation sur mesure adaptée à ses objectifs et son niveau mais également à son secteur d'activité et son métier.

Nature des travaux : Exercices pratiques – évaluations – cas pratiques



C/ MODALITÉS DE FORMATION

Présentiel

D/ MODALITÉS D'ÉVALUATION DES CONNAISSANCES

1. Un audit sera réalisé par le stagiaire **avant le début de sa formation** afin d'évaluer son niveau.
2. **Durant tout son parcours de formation**, le stagiaire sera évalué à travers des exercices et évaluations qui lui seront remises par son formateur.
3. Une attestation de formation sera éditée et envoyée par mail au stagiaire **à la fin de sa formation**.

À l'issue de sa formation, le stagiaire passera la certification TOSA Python enregistrée à France Compétences sous le numéro RS6202. Le stagiaire doit être équipé d'un ordinateur muni d'une webcam, d'un micro et d'une connexion internet pour le passage de sa certification.

L'équipe pédagogique sera en charge de son inscription auprès du certificateur.

À l'issue du test, le candidat se voit attribuer un score (0 à 1000), correspondant à un niveau (Initial, basique, opérationnel, avancé ou expert), ce qui lui permettra de faire valoir ses compétences sur le marché du travail.

Résumé de la certification :

Test adaptatif TOSA : QCM, manipulation de logiciels, Scoring mathématique IRT (Item Response Theory)

- 35 questions
- Durée de 90 minutes
- Passage en conditions d'examen en ligne
- Score sur une échelle de 1 à 1000
- Délivrance de la certification si le score est supérieur à 551 sous 5 jours ouvrés

Compétences attestées :

Niveau opérationnel (score Tosa 551 à 725)

- Utiliser les fonctionnalités de contrôles de flux afin d'automatiser un traitement de données conditionnels et/ou répétitifs
- Créer un programme pour des traitements complets de données, en utilisant les fonctionnalités avancées des contrôles de flux et des opérations arithmétiques
- Reconnaître et créer des objets simples de type intégrés (built-in), en vue de créer des programmes manipulant des données



- Reconnaître les objets structurés de type intégré, afin d'effectuer des traitements de multiples variables simples
- Manipuler les objets intégrés simples et structurés et les assembler dans le but de créer des fonctions simples et réutilisables
- Maîtriser les concepts liés aux modules en vue d'inclure des fonctionnalités d'un module à un programme
- Sélectionner et importer des fonctions spécifiques d'un package de l'API afin de les réutiliser dans un traitement de données
- Créer un package simple complet pour créer des fonctionnalités partageables et réutilisables / Stocker et traiter simultanément plusieurs données, en vue de créer un flux de traitement de données simples adapté
- Modéliser un problème et automatiser des manipulations de données, afin de traiter un volume important ou complexe de données.

Niveau avancé (score Tosa 726 à 875)

- Connaître les règles syntaxiques et sémantiques de Python afin de créer des programmes clairs et réutilisables
- Maîtriser les outils d'entrée-sortie, dans le but de lire et modifier des documents externes au programme
- Maîtriser les spécificités des types intégrés afin de traiter efficacement et de manière claire des grands volumes de données
- Créer des fonctions documentées, des classes et leurs méthodes associées, dans le but de créer des fonctionnalités intégrables dans d'autres programmes
- Importer des packages composés pour exploiter l'ensemble des fonctionnalités incluses
- Utiliser les packages fondamentaux de la bibliothèque standard afin de manipuler efficacement des données externes habituelles
- Implémenter des structures de données adaptées et choisir les fonctions adaptées afin de concevoir des programmes à la complexité algorithmique maîtrisée

Niveau expert (score Tosa 876 à 1000)

- Intégrer la gestion d'erreur afin de créer des programmes couvrant tous les cas d'entrée et de gestion de données
- Utiliser les règles et conventions de nommage afin de créer des programmes pouvant s'intégrer dans un environnement complet
- Mettre en œuvre l'intégralité des outils fonctionnels, afin de créer des fonctions adaptées et efficaces
- Utiliser et implémenter les concepts de la programmation orientée objet, afin de créer des structures spécifiques à un projet
- Maîtriser l'ensemble des concepts liés aux modules pour intégrer des packages de la manière la plus performante possible
- Utiliser l'intégralité de la bibliothèque standard pour traiter des données et documents externes, en local ou sur internet
- Mettre en place des outils algorithmiques avancés afin de drastiquement améliorer la performance d'un programme existant



Le passage de la certification est obligatoire pour les formations financées par le CPF



E/ MODALITÉS D'ÉVALUATION DE L'APPRÉCIATION DES PARTICIPANTS

Questionnaire de satisfaction téléphonique du service qualité et/ou questionnaire de satisfaction remis en fin de formation

NOTE : Si vous, ou l'un(e) de vos salarié(e)s, est une personne en situation de handicap, merci de contacter notre référent handicap, Mr David Allouche à l'adresse mail : referenthandicap@formatexia.com