



Python – Devenez opérationnel en toute simplicité

Cette formation développement en Python vous permettra de découvrir et de maîtriser les fonctions clés du code afin de développer ses propres applications. À travers des exercices pratiques et des exemples concrets, vous apprendrez à utiliser les outils essentiels pour gagner en efficacité et en productivité.

À la fin de la formation, vous aurez l'opportunité de passer la certification TOSA®, une référence dans l'évaluation des compétences numériques. Obtenez un score sur 1000 et affichez-le fièrement sur votre CV ou vos profils professionnels pour valoriser vos compétences.

Que vous soyez un utilisateur occasionnel ou un professionnel, cette certification est adaptée à tous les niveaux et vous aide à améliorer vos perspectives de carrière, tout en étant éligible au CPF (Compte Personnel de Formation).

Objectifs de la formation

- 👤 Comprendre les bases de Python
- 👤 Maîtriser les fonctions et les modules
- 👤 Utiliser les structures de données
- 👤 Développer des compétences en programmation orientée objet (POO)
- 👤 Gérer les fichiers et les exceptions
- 👤 Appliquer Python à des projets réels

Prérequis

Avoir des connaissances sur l'environnement Microsoft Windows et savoir utiliser un clavier et souris.
Disposer d'une connexion internet fiable

Certification

TOSA Python – [Code RS 6962](#)

Nom du certificateur : ISOGRAD

La certification demeure en option si l'inscription se déroule hors CPF, si nécessaire prévoir son achat en supplément (79€ ttc). L'examen (en français) dure environ 1h et nécessite un minimum de 551 points pour obtenir la certification.

Modalités, méthodes et moyens pédagogiques

La formation est disponible en présentiel ou distanciel (blended-learning, classe virtuelle, présentiel à distance) selon le format choisi. Le formateur alterne différentes méthodes pédagogiques :

- 👤 **Méthode démonstrative** pour illustrer les concepts.
- 👤 **Méthode interrogative** pour favoriser l'échange et la réflexion.

- 🔑 **Méthode active**, comprenant des travaux pratiques et des mises en situation pour ancrer l'apprentissage.
- 🔑 Les moyens pédagogiques dépendent du type de formation et peuvent inclure : Ordinateurs Mac ou PC, Tablettes ou Matériels multimédia (1 poste par personne), connexion internet fibre, tableau blanc ou paperboard, vidéoprojecteur.
- 🔑 Environnements de formation installés sur les postes de travail ou accessibles en ligne.
- 🔑 Supports de cours et exercices pour faciliter l'apprentissage.

Modalités de suivi et d'évaluation des acquis

- 🔑 Les heures effectuées (cours et/ou ateliers) seront émargées
- 🔑 Pour la partie E-learning, le temps passé apparaît sur le relevé de connexion
- 🔑 Un tableau de synthèse de suivi sera utilisé pour optimiser le suivi des apprenants
- 🔑 En cours de formation, par des exercices pratiques corrigés et/ou Quizz
- 🔑 Et, en fin de formation, par un questionnaire d'auto-évaluation et/ou une certification (obligatoire pour le CPF et proposée en option hors CPF), vous évaluerez également le formateur.

Durée de la formation, modalités d'organisation et tarif

- 🔑 Nombre d'heures et/ou jours : 35h ou 5 jours
- 🔑 Présentiel : à définir
- 🔑 E-learning : à définir
- 🔑 Tarif : 2940 € TTC

Modalité d'inscription

Toutes les informations nécessaires à la prise en charge de votre inscription se trouve sur notre espace en ligne www.facplus.fr/sinscrire/

Accessibilité de la formation

Facplus s'engage pour faciliter l'accessibilité de ses formations aux personnes en situation de handicap : les détails de l'accueil sont consultables sur la page www.facplus.fr/handicap/

Programme

Introduction à Python

- 🔑 Comprendre les bases du langage Python et son environnement de développement
- 🔑 Présentation de Python : Histoire, versions, et applications.
- 🔑 Installation de Python et des environnements de développement (IDLE, PyCharm, VS Code)
- 🔑 Écriture et exécution de votre premier script Python
- 🔑 Introduction aux types de données de base (entiers, flottants, chaînes de caractères, booléens)

Langage et syntaxe

- 🔑 Identifier des types d'objets simples
- 🔑 Connaître les principaux cas d'usage de Python
- 🔑 Utiliser les opérateurs arithmétiques
- 🔑 Créer un contrôle de flux conditionnel
- 🔑 Créer des boucles simples

Structures de Contrôle

- 🔑 Maîtriser les structures de contrôle de flux en Python
- 🔑 Instructions conditionnelles : if, elif, else

- 📌 Boucles : for, while
- 📌 Utilisation de break, continue et else avec les boucles
- 📌 Compréhension des listes (list comprehensions)

Fonctions et Modules

- 📌 Apprendre à créer et utiliser des fonctions et modules en Python
- 📌 Définition et appel de fonctions
- 📌 Paramètres et valeurs de retour
- 📌 Portée des variables (locales et globales).
- 📌 Introduction aux modules et bibliothèques standard
- 📌 Importation et utilisation de modules

Structures de Données

- 📌 Comprendre et utiliser les structures de données en Python
- 📌 Listes : création, manipulation, méthodes
- 📌 Tuples : caractéristiques et utilisation
- 📌 Dictionnaires : création, manipulation, méthodes
- 📌 Ensembles : création, manipulation, méthodes
- 📌 Documenter une fonction
- 📌 Effectuer une itération sur un objet de type structuré
- 📌 Manipuler les fonctions associées aux objets structurés
- 📌 Créer des fonctions

Gestion des Fichiers

- 📌 Apprendre à lire et écrire des fichiers en Python
- 📌 Ouverture et fermeture de fichiers
- 📌 Lecture et écriture de fichiers texte
- 📌 Gestion des fichiers CSV
- 📌 Utilisation des context managers (with)
- 📌 Programmation Orientée Objet (POO)
- 📌 Maîtriser les concepts de la programmation orientée objet en Python
- 📌 Classes et objets : définition et utilisation
- 📌 Attributs et méthodes
- 📌 Héritage et polymorphisme
- 📌 Encapsulation et abstraction
- 📌 Méthodes spéciales (constructeurs, destructeurs, etc.)

Gestion des Exceptions

- 📌 Comprendre et gérer les exceptions en Python
- 📌 Introduction aux exceptions
- 📌 Utilisation des blocs try, except, else et finally
- 📌 Création d'exceptions personnalisées
- 📌 Bonnes pratiques de gestion des erreurs

Optimisation du code

- 📌 Lire une donnée en entrée
- 📌 Lire et prétraiter plusieurs données d'entrée
- 📌 Choisir les types de variables adaptés au problème
- 📌 Combiner des types de données différents
- 📌 Trier des données
- 📌 Modéliser un graphe et l'explorer
- 📌 Créer des combinaisons de boucles et conditions adaptées à différents problèmes
- 📌 Extraire automatiquement une information d'un volume important de données

Bibliothèques et Frameworks

- 🔗 Découvrir et utiliser des bibliothèques et frameworks populaires en Python
- 🔗 Introduction à NumPy pour les calculs numériques
- 🔗 Utilisation de Pandas pour la manipulation de données
- 🔗 Introduction à Matplotlib pour la visualisation de données
- 🔗 Présentation de Flask et Django pour le développement web

Projets Pratiques

- 🔗 Appliquer les connaissances acquises à travers des projets pratiques
- 🔗 **Projet 1** : Analyse de données avec Pandas
- 🔗 **Projet 2** : Création d'une application web simple avec Flask
- 🔗 **Projet 3** : Développement d'un jeu simple avec Pygame
- 🔗 **Projet 4** : Automatisation de tâches avec des scripts Python

Bonnes Pratiques et Outils Avancés

- 🔗 Adopter les bonnes pratiques de développement et utiliser des outils avancés
- 🔗 Bonnes pratiques de codage (PEP 8)
- 🔗 Utilisation de Git pour le contrôle de version
- 🔗 Introduction aux tests unitaires avec unittest
- 🔗 Déploiement d'applications Python

Évaluation

Évaluation continue : Observation et feedback pendant les ateliers pratiques

Projets finaux : Réalisation de projets pratiques évalués sur la base de critères prédéfinis (qualité du code, respect des bonnes pratiques, fonctionnalité)