

Ascent Formation

22 rue Jean Marie David, Le Spot
35740 PACÉ
Email : contact@ascent-formation.fr
Tel : +33 9 75 75 37 81



Programmation Python

Lot 4 - Développement - Développement logiciel - Référence 4-060

Objectif général: À l'issue de la formation, les participants seront capables de développer des applications Python robustes et performantes en utilisant les principes de la programmation objet, les modules standards et les outils d'évaluation de qualité.

Durée : 30.00 heures (5.00 jours)

Modalité : Toutes les modalités (présentiel, classe virtuelle et hybride) sont possibles

Lieu possible de réalisation : Locaux du bénéficiaire ou locaux d'Ascent Formation, en France métropolitaine et dans les DOM-COM

Nombre maximal de participants : 20

Profils des stagiaires

- Développeurs débutants ou expérimentés souhaitant maîtriser le langage Python.
- Ingénieurs souhaitant développer des applications logicielles.
- Toute personne souhaitant approfondir ses connaissances en Python et ses usages professionnels.

Prérequis

- Connaissances de base en programmation (variables, boucles, conditions).
- Notions élémentaires sur les systèmes d'exploitation et la manipulation de fichiers.

Niveau SAME visé : Application (A)

Compétences visées

- Écrire des programmes Python en respectant les bonnes pratiques de codage.
- Appliquer les concepts de la programmation objet dans des projets Python.
- Utiliser les modules standards et tiers pour enrichir les fonctionnalités d'une application.
- Concevoir des interfaces graphiques simples et intuitives.
- Tester et évaluer la qualité des programmes Python.

Objectifs pédagogiques

- Comprendre et maîtriser la syntaxe de Python pour développer des programmes fiables.
- Implémenter des solutions modulaires et réutilisables grâce à la programmation objet.
- Intégrer des interfaces graphiques et des outils de visualisation.
- Automatiser les tests et évaluer la performance des applications.

Contenu de la formation

- **Jour 1 : Introduction et bases du langage Python (6 heures)**

- Introduction à Python (1,5 heure)
Historique, caractéristiques et avantages de Python. Installation et configuration de l'environnement de développement (PyCharm, VS Code).
Travaux pratiques : Configurer un projet Python.
- Syntaxe et structures de base (2,5 heures)
Variables, types de données, opérateurs, structures de contrôle (if, for, while). Manipulation des chaînes de caractères et des fichiers.
Travaux pratiques : Écrire un programme Python simple.
- Gestion des erreurs et exceptions (2 heures)
Principes de gestion des exceptions (try, except, finally).
Travaux pratiques : Gérer des erreurs dans un programme Python.
- Jour 2 : Programmation objet avec Python (6 heures)
 - Concepts de base de la programmation objet (2 heures)
Classes, objets, méthodes, attributs et constructeurs. Héritage, polymorphisme et encapsulation.
Travaux pratiques : Modéliser et implémenter une classe Python.
 - Création de modules et packages (2 heures)
Organisation du code en modules et packages. Importation et réutilisation de modules.
Travaux pratiques : Créer un projet structuré avec plusieurs modules.
 - Gestion des fichiers et bases de données (2 heures)
Lecture et écriture dans des fichiers. Introduction à SQLite et manipulation des bases de données avec Python.
Travaux pratiques : Développer une application avec persistance des données.
- Jour 3 : Modules Python et bibliothèques tierces (6 heures)
 - Utilisation des modules standards (2 heures)
Modules courants : os, sys, datetime, math, random.
Travaux pratiques : Résoudre des problèmes simples en utilisant les modules standards.
 - Introduction aux bibliothèques tierces (2 heures)
Découverte de Pandas, NumPy et Matplotlib.
Travaux pratiques : Manipuler des données et les visualiser.
 - Introduction aux APIs avec Python (2 heures)
Appels d'API avec la bibliothèque Requests.
Travaux pratiques : Consommer une API REST et traiter les données reçues.
- Jour 4 : Interfaces graphiques et tests (6 heures)
 - Conception d'interfaces graphiques avec Tkinter (3 heures)
Création de fenêtres, gestion des événements, widgets (boutons, entrées, listes).
Travaux pratiques : Concevoir une application graphique simple.
 - Tests et évaluation de la qualité (3 heures)
Introduction à unittest et pytest pour les tests unitaires. Automatisation des tests et couverture de code.
Travaux pratiques : Mettre en place une suite de tests pour un projet Python.

• Jour 5 : Projet final et révision des acquis (6 heures)

• Développement d'un projet final (4 heures)

Scénario : Développer une application Python complète (ex. gestionnaire de tâches, calculatrice avancée). Étapes : conception, développement, intégration des tests et évaluation.

Travaux pratiques : Réalisation du projet sous supervision du formateur.

• Révision et validation des acquis (2 heures)

Quiz interactif et récapitulatif des notions clés abordées. Débriefing et retours personnalisés sur le projet final.

Organisation de la formation

Equipe pédagogique

Professionnel expert technique et pédagogique.

Moyens pédagogiques et techniques, outils

- Espace numérique de travail
- Documents supports de formation projetés
- Exposés théoriques
- Etude de cas concrets
- Quiz en salle
- Mise à disposition en ligne de documents supports à la suite de la formation
- Travaux pratiques (Minimum 60 % du temps de formation)
- Outils utilisés (liste non-exhaustive)
 - Logiciels (liste non-exhaustive) : Python, PyCharm, VS Code, Tkinter, Pandas, Matplotlib.
 - Matériel : PC ou Mac avec environnement Python configuré.
 - Plateformes : Documentation officielle Python, API publiques pour les travaux pratiques.

Dispositif de suivi de l'exécution de l'évaluation des résultats de la formation

- Feuilles de présence
- Questions orales ou écrites (QCM)
- Mises en situation
- Formulaires d'évaluation de la formation
- Certificat de réalisation de l'action de formation

Indicateurs de résultats :

- Taux de satisfaction des apprenants
- Nombre d'apprenants
- Taux et causes des abandons
- Taux de retour des enquêtes
- Taux d'interruption en cours de prestation