PYTHON INITIATION

Formation proposée par AMINE BAQUA

Python

Cette formation permet de maîtriser la programmation orientée objet en Python afin de structurer des applications, comprendre des architectures existantes et refactoriser du code de manière professionnelle.

◷DURÉE
20 Heure(s)
▣MODALITÉS
Présentiel / Distanciel
♙PARTICIPANTS
Jusqu'à 11 participants
★LANGUE
Français
LIEU
France entière
♙CERTIFICATION
Attestation de formation
▦MISE À JOUR
04/11/2026

VOTRE FORMATEUR

AMINE BAQUA
Formateur professionnel
Conseiller Pédagogique / Formateur

TARIFS INDICATIFS

Inter-entreprise	**Pas de dates planifiées**
Intra-entreprise	**Sur devis**
Individuelle	**Sur devis**
Durée	**20 Heure(s)**

Tarifs indicatifs, sous réserve d'analyse du besoin et de devis.

◎Objectifs pédagogiques

✓ - Maîtriser les bases essentielles du langage Python
✓ - Comprendre et appliquer les concepts de la programmation orientée objet
✓ - Gérer les erreurs et sécuriser ses traitements
✓ - Structurer un projet Python de manière professionnelle
✓ - Exploiter les modules standards les plus utiles

⌘Compétences visées

Python

♙Public cible et prérequis

Public cible

- Développeurs débutants/intermédiaires, personnes ayant déjà des bases en Python ou équipes techniques travaillant sur du code existant

Prérequis

- Connaissances de base en Python ou dans un autre langage de programmation.

Programme détaillé

Jour 1 – Maîtriser les bases essentielles de Python

TYPES DE DONNEES ET VARIABLES
- Types simples (int, float, str, bool)
- Conversions et opérations

STRUCTURES DE CONTROLES
- Conditions (if / elif / else)
- Boucles (for / while)

STRUCTURES DE DONNEES
- Listes, tuples, dictionnaires
- Parcours et manipulation

LES FONCTIONS
- Définition et utilisation
- Paramètres et valeurs de retour
- Portée des variables (scope)

Jour 2 – Programmation Orientée Objet (fondamentaux)

INTRIDUCTION A LA POO
- Classes et objets
- Attributs et méthodes
- Constructeur (__init__)

ENCAPSULATION
- Organisation interne d’une classe
- Attributs publics / privés

CONCEPTION SIMPLE
- Responsabilité d’une classe
- Modélisation de cas simples

Mise en pratique sur un cas concret

Jour 3 – POO avancée et gestion des erreurs

CONCEPTS AVANCES
- Héritage
- Polymorphisme
- Méthodes spéciales

GESTION DES EXCEPTIONS
- try / except / finally
- Lever des exceptions personnalisées
- Sécuriser les traitements

REFACTORISATION
- Identifier les faiblesses d’un code
- Améliorer structure et robustesse

Cas pratique sur code existant

Jour 4 – Structuration de projet et modules Python

ORGANISTION D'UN PROJET
- Fichiers, modules et packages
- Imports et dépendances

MODULES STANDARD UTILES

- Paramétrage et scripting avec sys / os / pathlib
- Sérialisation avec json
- Logging avec logging

ACCÈS AUX DONNÉES
- Introduction aux bases de données
- Connexion avec psycopg2
- Présentation de SQLAlchemy (ORM)

ENVIRONNEMENT DE TRAVAIL
- Installation de dépendances avec pip
- Environnements virtuels (venv)

MISE EN PRATIQUE
- Structuration complète d’un projet Python
- Amélioration d’un code existant

▣Méthodes pédagogiques & évaluation

Méthodes pédagogiques

- ✓ Apports théoriques et retours d'expérience
- ✓ Exercices pratiques et mises en situation
- ✓ Travaux de groupe et échanges entre pairs
- ✓ Études de cas concrets

Modalités d'évaluation

- ✓ Quiz de validation des acquis
- ✓ Mises en situation évaluées
- ✓ Projet fil rouge
- ✓ Feuille de route post-formation

♿Accessibilité

Une formation accessible à tous

Ce programme peut être adapté pour les personnes en situation de handicap. Contactez-nous pour étudier vos besoins spécifiques.