

# L'ESSENTIEL DE PYTHON ET DE LA PROGRAMMATION ORIENTEE OBJET

Durée : 5 jours (35 heures)

## CONNAISSANCES PREALABLES

---

- Connaissances générales de l'informatique.
- Pratique de base d'un système d'exploitation Windows, Linux ou macOS.
- Une première expérience de la programmation constitue un avantage mais n'est pas obligatoire.

## PROFIL DES STAGIAIRES

---

- Développeurs débutants.
- Administrateurs systèmes souhaitant automatiser leurs tâches.
- Ingénieurs DevOps débutants.
- Techniciens informatiques.
- Analystes.
- Chefs de projets techniques souhaitant comprendre le développement.
- Toute personne souhaitant acquérir les bases du langage Python et de la programmation orientée objet.

## OBJECTIFS

---

À l'issue de la formation, les participants seront capables de :

- Comprendre les principes fondamentaux de la programmation avec Python.
- Utiliser les principales structures du langage.
- Manipuler les types de données et les collections.
- Développer des scripts Python structurés.
- Concevoir des fonctions réutilisables.
- Gérer les erreurs et les exceptions.
- Comprendre les concepts fondamentaux de la programmation orientée objet.
- Créer et utiliser des classes et des objets.
- Mettre en œuvre l'encapsulation, l'héritage et le polymorphisme.
- Développer une application Python simple en appliquant les bonnes pratiques de développement.

## CERTIFICATION PREPAREE

---

Aucune

## METHODES PEDAGOGIQUES

---

- Présentations interactives.
- Démonstrations techniques.
- Exercices pratiques progressifs.

- Travaux dirigés.
- Études de cas.
- Développement d'un mini-projet fil rouge.
- Support pédagogique remis aux participants..

## FORMATEUR

---

- Consultant formateur expert Python et développement logiciel, disposant d'une expérience significative dans la conception d'applications Python, l'automatisation des traitements et l'enseignement des concepts de programmation orientée objet.

## METHODE D'EVALUATION DES ACQUIS

---

- Quiz de positionnement.
- Exercices pratiques tout au long de la formation.
- Réalisation d'un projet fil rouge.
- Évaluation finale des acquis.
- Débriefing collectif.

## CONTENU DU COURS

---

### Module 1 : Découvrir Python et les fondamentaux de la programmation (7h)

#### Objectifs

- Installer et utiliser un environnement Python.
- Comprendre les bases du langage.
- Écrire ses premiers programmes.

#### Contenu

##### Présentation de Python

- Historique et caractéristiques.
- Domaines d'utilisation.
- Installation de l'environnement.
- Utilisation de l'interpréteur.

##### Premiers programmes

- Structure d'un programme Python.
- Variables.
- Types de données fondamentaux.
- Opérations et expressions.

##### Entrées et sorties

- Affichage des résultats.
- Saisie utilisateur.
- Formatage des chaînes de caractères.

##### Bonnes pratiques de développement

- Lisibilité du code.
- Indentation.
- Conventions de nommage.

##### Mise en pratique

##### Travaux pratiques :

Développement de scripts simples permettant de manipuler variables, calculs et entrées utilisateur.

## Module 2 : Maîtriser les structures de contrôle et les collections (7h)

### Objectifs

- Contrôler l'exécution des programmes.
- Manipuler les principales structures de données Python.

### Contenu

#### Structures conditionnelles

- if.
- elif.
- else.
- Conditions imbriquées.

#### Boucles

- for.
- while.
- Contrôle des itérations.

#### Les collections Python

- Listes.
- Tuples.
- Ensembles (sets).
- Dictionnaires.

#### Manipulation des données

- Parcours des collections.
- Recherche.
- Tri.
- Filtrage.

#### Mise en pratique

##### Travaux pratiques :

Développement d'applications manipulant des listes d'informations et automatisant des traitements répétitifs.

## Module 3 : Organiser son code avec les fonctions et modules (7h)

### Objectifs

- Structurer les programmes Python.
- Réutiliser efficacement le code.

### Contenu

#### Les fonctions

- Déclaration.
- Paramètres.
- Valeurs de retour.
- Portée des variables.

#### Fonctions avancées

- Paramètres nommés.
- Valeurs par défaut.
- Fonctions lambda.

#### Gestion des modules

- Importation.
- Bibliothèques standard.
- Organisation du code.

#### Gestion des erreurs

- Exceptions.
- Try / Except.

- Finally.
- Bonnes pratiques de gestion des erreurs.

### **Mise en pratique**

#### **Travaux pratiques :**

Création d'une bibliothèque de fonctions réutilisables et gestion robuste des erreurs.

## **Module 4 : Comprendre la programmation orientée objet en Python (7h)**

### **Objectifs**

- Maîtriser les concepts fondamentaux de l'objet.
- Concevoir des classes simples.

### **Contenu**

#### **Introduction à l'approche objet**

- Concepts fondamentaux.
- Classes et objets.
- Attributs et méthodes.

#### **Création de classes**

- Constructeurs.
- Initialisation des objets.
- Manipulation des instances.

#### **Encapsulation**

- Attributs publics et privés.
- Contrôle de l'accès aux données.

#### **Relations entre objets**

- Composition.
- Agrégation.

### **Mise en pratique**

#### **Travaux pratiques :**

Création d'un modèle objet représentant un système métier simple.

## **Module 5 : Approfondir la programmation orientée objet et développer une application complète (7h)**

### **Objectifs**

- Mettre en œuvre les mécanismes avancés de l'objet.
- Développer une application structurée.

### **Contenu**

#### **Héritage**

- Réutilisation du code.
- Spécialisation des classes.
- Redéfinition de méthodes.

#### **Polymorphisme**

- Surcharge comportementale.
- Utilisation de classes parentes.

#### **Bonnes pratiques de conception objet**

- Modularité.
- Maintenabilité.
- Réutilisabilité.

#### **Structuration d'une application Python**

- Organisation des fichiers.
- Séparation des responsabilités.
- Documentation du code.

### **Mise en pratique**

---

**Projet fil rouge :**

Développement d'une application complète intégrant :

- Manipulation de données.
- Fonctions réutilisables.
- Gestion des exceptions.
- Création de classes.
- Héritage et polymorphisme.
- Structuration du code selon les bonnes pratiques Python.

Notre référent handicap se tient à votre disposition au [01.71.19.70.30](tel:01.71.19.70.30) ou par mail à <mailto:referent.handicap@edugroupe.com> pour recueillir vos éventuels besoins d'aménagements, afin de vous offrir la meilleure expérience possible.