

## **Plan de Formation Python - 120 heures**

**Objectif :** Acquérir les compétences pour devenir un développeur Python autonome et mettre en œuvre des projets de complexité moyenne.

**Public cible :** Candidats débutants ou avec des bases à renforcer.

**Durée :** 120 heures réparties sur 8 semaines (environ 15 heures par semaine).

### **Module 1 : Les Fondamentaux de Python (30 heures)**

#### **Semaine 1 : Introduction et bases**

**Jour 1-2 :** Introduction à Python et à la programmation. Installation de Python et de l'environnement de développement (VS Code, PyCharm). Premier programme "Hello World". Types de données (entiers, flottants, chaînes, booléens).

**Jour 3-4 :** Variables, opérateurs et structures de contrôle\*\*. Déclaration de variables, opérateurs arithmétiques et logiques. Utilisation des structures conditionnelles (``if``, ``elif``, ``else``).

**Jour 5 :** Boucles et itérations. Boucles ``for`` et ``while``. Utilisation de ``range()``. ``break`` et ``continue``.

#### **Semaine 2 : Structures de données et fonctions**

**Jour 1-2 :** Listes et tuples. Définition et manipulation des listes (ajout, suppression, indexation). Introduction aux tuples.

**Jour 3 :** Dictionnaires et ensembles. Création et manipulation des dictionnaires. Introduction aux ensembles.

**Jour 4-5 :** Fonctions et gestion d'erreurs. Définition de fonctions, arguments et valeurs de retour. Gestion des exceptions (``try``, ``except``). Mini-projet pour consolider les acquis.

## **Module 2 : Programmation Avancée et POO (30 heures)**

### **Semaine 3 : Programmation Orientée Objet (POO)**

**Jour 1-2** : Concepts de la POO. Classes et objets, constructeur `__init__`, attributs et méthodes.

**Jour 3-4** : Héritage et polymorphisme. Héritage simple et multiple. Polymorphisme et surcharge des méthodes.

**Jour 5** : Modules et packages. Création de modules et installation de packages avec `pip`.

### **Semaine 4 : Manipulation de données et fichiers**

**Jour 1-2** : Lecture et écriture de fichiers. Manipulation de fichiers texte (`.txt`) et de fichiers structurés (CSV, JSON).

**Jour 3-4** : Expressions régulières et dates. Introduction au module `re` et au module `datetime`.

**Jour 5** : Projet pratique : création d'une application de manipulation de données.

## **Module 3 : Développement Web et Bases de Données (30 heures)**

### **Semaine 5 : Web Scraping et APIs**

**Jour 1-2** : Introduction au web scraping. Utilisation de `requests` et de `Beautiful Soup`.

**Jour 3-4** : Utilisation d'APIs. Requêtes GET et POST, traitement de données JSON.

**Jour 5** : Introduction à Flask. Création d'une application web simple.

### **Semaine 6 : Bases de données**

**Jour 1-2** : Introduction aux bases de données\*\*. Concepts de SQL vs NoSQL. Installation et connexion à `SQLite`.

**Jour 3-4** : Langage SQL. Requêtes `SELECT`, `INSERT`, `UPDATE`, `DELETE`.

**Jour 5** : Projet final : application web connectée à une base de données.

## **Module 4 : Approfondissement et Projet Final (30 heures)**

### **Semaine 7 : Tests et bonnes pratiques**

**Jour 1-2** :Tests unitaires avec `unittest` ou `pytest`.

**Jour 3-4** :Git et gestion de versions\*\*. Commandes de base (`commit`, `push`, `pull`).  
Utilisation de GitHub.

**Jour 5** :Bonnes pratiques de codage\*\*. Respect de la convention PEP 8 et documentation du code.

### **Semaine 8 : Projet final**

**Jour 1-5** :Développement d'un projet d'envergure. Définition du cahier des charges, architecture, implémentation et présentation finale.