

**OBJECTIFS DE CETTE FORMATION :**

Comprendre les bases de la programmation et la logique algorithmique

Savoir écrire des scripts Python fonctionnels

Manipuler les structures de données essentielles

Lire/écrire des fichiers

Découvrir les bases de l'orienté objet et quelques modules standards

**PARTICIPANT :** Toute personne en charge de créer et de mettre à jour un site web (intranet ou internet).

**PRE-REQUIS :** Savoir utiliser l'environnement Windows et une bonne pratique de internet. Avoir des notions de base de données est un plus.

**FORMATEURS :** spécialiste en développement.

**MODALITES ET PEDAGOGIE :**

- Questionnaire d'évaluation en amont et en fin de formation
- Cas pratiques et études de cas, Quizz
- Assistance téléphonique
- Méthode interactive et intuitive
- Support Stagiaire
- Attestation de fin de formation

**LIVRABLES**

Support de formation (PDF) : notions, exemples, explications.

Cahier d'exercices + corrigés (PDF + fichiers .py).

Scripts Python réalisés en atelier (projets et exemples).

Ressources complémentaires (liens, mémo Python, bonnes pratiques).

**PROGRAMME****JOUR 1 — INTRODUCTION & BASES DU LANGUAGE****1. Découverte de Python**

- Présentation du langage, usages et écosystème
- Installation de Python, IDE (Thonny, VS Code, PyCharm)

**2. Premiers pas**

- Exécuter un script / interprétation
- Variables et types de base (int, float, str, bool)
- Entrées / sorties : input(), print()

**3. Opérations & expressions**

- Opérateurs arithmétiques et logiques
- Conversion de types

**JOUR 2 — CONTROLE DE FLUX ET COLLECTIONS****1. Conditions**

- if, elif, else
- Opérateurs de comparaison

**2. Boucles**

- for et while
- Break / continue

**3. Structures de données**

- Listes, tuples, sets, dictionnaires
- Méthodes courantes d'ajout, suppression, recherche

**JOUR 3 — FONCTIONS & MODULARITE****1. Fonctions**

- Définition, arguments, valeurs de retour
- Portée des variables
- Arguments nommés, valeurs par défaut

**2. Modules & packages**

- Importation et utilisation (math, random, etc.)
- Création de modules simples

**3. Gestion des erreurs**

- Introduction à try / except
- Lever des exceptions

**JOUR 4 — FICHIERS, JSON ET BASES DE DONNEES****1. Manipulation de fichiers**

- Lire / écrire des fichiers texte
- Gestion d'erreurs IO
- Le contexte with

**2. Format JSON**

- Lire / écrire du JSON
- Structurer des données

**3. Introduction aux bases de données**

- Utiliser SQLite
- Exécuter des requêtes simples (SELECT, INSERT, UPDATE)

**JOUR 5 — PROGRAMMATION ORIENTEE OBJET & MINI-PROJET FINAL****1. Introduction à la POO**

- Classes & objets
- Attributs, méthodes
- Notion d'héritage simple

**2. Bibliothèques utiles selon les besoins**

Exemples :

- datetime
- os / pathlib
- requests (si accès internet)

**3. Projet de fin de formation**

Quelques idées :

- Gestionnaire de contacts
- Analyseur de fichier (CSV/TXT)
- Petit jeu en console (pendu, devinette...)
- Mini API client avec requests