

## Bootcamp Python Pratique – 2 semaines (15 jours)

### Objectif général du bootcamp

À l'issue de ce bootcamp intensif, chaque participant sera capable de :

- Comprendre les bases solides du langage Python et les appliquer dans des situations concrètes
- Écrire des scripts Python clairs, efficaces et réutilisables
- Automatiser des tâches courantes et répétitives (gestion de fichiers, rapports, emails, scripts planifiés)
- Manipuler, analyser et visualiser des données réelles à l'aide de bibliothèques clés comme *pandas* et *matplotlib*
- Interagir avec des APIs et des bases de données (JSON, HTTP, SQLite)
- Créer de petites applications (scripts, outils, interfaces graphiques ou mini-apps web avec Flask)
- Appliquer les bonnes pratiques de développement Python (fonctions, programmation orientée objet, organisation du code, bases de Git)
- Réaliser un mini-projet fonctionnel de bout en bout, mettant en pratique l'ensemble des compétences acquises (*jeu simple, outil d'automatisation, application graphique ou mini-application web*)

### Semaine 1 – Bases solides & automatisation

#### Jour 1 – Découverte & premiers scripts

- Qu'est-ce que le Python et à quoi ça sert ?
- Installation (Python, VS Code / Jupyter)
- Premier programme
- Variables et types de base

#### Jour 2 – Conditions & logique

- `if / else / elif`
- Opérateurs logiques
- Prise de décision dans un programme

#### Jour 3 – Boucles & répétition

- `for, while`
- Compteurs
- Boucles avec conditions

#### Jour 4 – Structures de données

- Listes
- Dictionnaires

- Tuples & sets (simplement)

#### Jour 5 – Fonctions & mini-projet

- Créer des fonctions
- Paramètres & retour de valeur
- Organisation du code

#### Jour 6 – Fichiers & données

- Lire / écrire dans un fichier (`.txt`, `.csv`)
- Sauvegarder des données
- Introduction à `pandas`

#### Jour 7 – Automatisation

- Scripts utiles
- `os`, `datetime`
- Renommer des fichiers automatiquement
- Générer des rapports simples

### Semaine 2 – Python appliqué & projets concrets

#### Jour 8 – Visualisation & analyse

- Introduction à `pandas`
- Graphiques avec `matplotlib`
- Comprendre des données réelles

#### Jour 9 – Python pour le Web

- Introduction au Web (très simple)
- Flask (micro-framework Python)
- Routes & pages HTML
- Formulaires simples

#### Jour 10 – Python & APIs

- Qu'est-ce qu'une API ?
- Requêtes HTTP avec `requests`
- JSON
- Exploiter des APIs publiques (météo, pays, crypto...)

#### Jour 11 – Bases de données

- Pourquoi une base de données ?
- SQLite (simple, local)
- CRUD : Create / Read / Update / Delete
- Connexion Python ↔ base de données

## Jour 12 – Python orienté objets

- Classes et objets (par analogie réelle)
- Attributs et méthodes
- Héritage (simple)
- Organisation d'un projet en classes

## Jour 13 – Interfaces graphiques

- Introduction à `Tkinter`
- Boutons, champs, formulaires
- Gestion des événements

## Jour 14 – Python pour la Data & l'IA

- Rappels `pandas`
- Nettoyage de données
- Introduction à `scikit-learn`
- Prédiction simple (classification / régression)

## Jour 15 – Automatisation avancée & productivité

- Scripts planifiés
- Emails automatiques
- Web scraping (bases)
- Génération automatique de rapports

## Temps additionnel (optionnel)

- Bonnes pratiques Python
- Organisation d'un projet
- Git & GitHub (bases)