

## Python - Par la pratique

**A l'issue de ce stage les participants seront en mesure de :** - Connaître les usages courants du langage - Maîtriser le scripting en Python - Structurer son code en fonction, classes et modules - Utiliser des modules existants - Vous initier à la programmation réseau avec Python - Programmer objet en Python.

**Référence :** PYT

**Durée :** 4 jours - 28 h

**Certification :** M2i Langages de Programmation (non incluse)

**Appréciation des résultats :** Evaluation qualitative de fin de stage

**Modalités et moyens pédagogiques :**

- Démonstrations - Cas pratiques - Synthèse et évaluation des acquis

**Prérequis :** Connaître un langage de programmation.

**Public concerné :** Développeurs.

Cette formation:

- est animée par un consultant-formateur dont les compétences techniques, professionnelles et pédagogiques ont été validées par des diplômes et/ou testées et approuvées par l'éditeur et/ou par M2i Formation
- bénéficie d'un suivi de son exécution par une feuille de présence émarginée par demi-journée par les stagiaires et le formateur.

## PROGRAMME

### ▪ Introduction

Historique  
Environnement de travail  
Champs d'application  
Scripts, fonctions et classes  
Versions architectures et plateformes pour Python  
Editeurs et environnements de Développement (IDE)  
Installation et configuration de Python  
Conformité du codage à PEP8

### ▪ Programme type

Programme minimal et point d'entrée  
Extension basée sur les packages : import  
Lancement de programmes, passage d'arguments  
Version compilée de Python : .py, .pyc  
Console Python  
Encoding : utf-8

### ▪ Types de bases

Typage faible  
Booléens, numériques  
Les chaînes de caractères  
Conversions de types, casting  
Type agrégés : Tuples, List, Set et dictionnaires

### ▪ Les instructions de base

Commentaires sur une ligne # ou plus ""  
Notion de bloc et indentations  
Les opérateurs :  
Affectation  
Arithmétique  
Logiques  
Relationnels  
Saisie et affichage : input, print et formatage  
Structure conditionnelle  
Les structures de boucle : while, for, range()

## Python - Par la pratique

---

Break et continue  
La gestion des exceptions  
L'import de modules

### ▪ Procédures et fonctions

def(), arguments et valeur de retour  
Variables globales et l'instruction globale  
Arguments par défaut, \*args et \*\*kwargs -  
Fonctions lambda

### ▪ Gestion des fichiers

Gestion de fichiers et répertoires  
Accès séquentiel, aléatoire  
Le module os, os.path, shutil, zlib

### ▪ Les modules

Définition de modules et instruction import  
Modules en tant que fichier py, en tant que répertoire  
Résolution des modules  
Module et programme : \_\_main\_\_  
Installation de modules : pip, easy\_install

### ▪ Les classes

Approche objets  
Classes et instances / objets : self

Constructeur : \_\_init\_\_  
Données et méthodes membres  
Héritage multiple

### ▪ Les bibliothèques Python

Accès aux bases de données relationnelles  
Programmation graphique UI avec Tkinter

### ▪ Mise au point de programme

Débogage : exécution pas à pas

### ▪ Python et le Web

Scripts Python en tant que CGI  
Introduction au framework Django

### ▪ Certification (en option)

Nos tests de validation des compétences font partie intégrante du processus d'apprentissage car ils permettent de développer différents niveaux d'abstractions.

Solliciter l'apprenant à l'aide de nos QCM, c'est lui permettre d'étayer sa réflexion en mobilisant sa mémoire pour choisir la bonne réponse. Nous sommes bien dans une technique d'ancrage mémoriel. L'examen sera passé à la fin de la formation.