



Axée principalement sur le développement de l'autonomie, la formation est extrêmement pratique afin de vous permettre de bien assimiler les concepts de ce fabuleux langage. Un support complet avec les principaux points clefs ainsi que les exercices corrigés est fourni. Enfin la dernière demi-journée est consacrée au codage d'un cas pratique proposé (par le client lors d'une formation intra entreprise) testé et documenté en pair-programming (ou individuellement) avec revue de code collective.

## Public visé

- Développeurs
- Ingénieurs et chercheurs
- Utilisateurs de Matlab voulant se former aux librairies Python scientifique

## Pré-requis

- Connaissances de base en algorithmie

## Durée et tarif

- **3 jours** soit 21 heures
- **1800 € HT** par personne

## Formation conventionnée

- Financement possible via OPCO

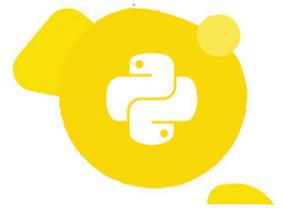


## Objectifs pédagogiques

- Connaître les possibilités du langage Python
- Être capable d'aborder des frameworks Python tels que Django, Flask ou Pyramid
- Être capable de réaliser des scripts d'administration système
- Être capable de reprendre et modifier un programme écrit par une autre personne
- Être capable d'installer et d'utiliser des librairies tierces comme PyQT, ElementTree (XML), Numpy, Scipy
- Pandas



# Programme



## Python en ligne de commande

- Interpréteur Python
- Différences avec le mode script
- Interpréteurs iPython et bPython

## Maîtriser les types et objets de bases

- Chaînes
- Tableaux indicés
- Tableaux associatifs
- Nombres Python et particularités
- Booléens
- Autres types avancés de Python
- Introspection
- Gestion de la mémoire, objets muables et immuables

## Structuration du code

- If, for, while, émuler un switch\_case
- Les fonctions
- Visibilité des variables

## Gérer les exceptions et les erreurs

- Try except finally
- Raise
- Exception
- assert
- traceback
- pdb

## Mots clefs et fonctions utiles

- with, map, lambda, filter, any, all etc.

## Conventions de codage et de nommage

- pep-0008
- pep-0257
- pylint

## Développer des bibliothèques

- Packages et modules
- Techniques et règles d'importation des Bibliothèques tierces
- Ecrire ses tests sans gêner les imports
- Installer une bibliothèque avec pip

## P.O.O.

- Présentation des concepts de la POO
- Attributs et méthodes de classe/instance
- Les méthodes spéciales
- Héritage, surcharge, polymorphisme, la classe object
- Héritage multiple et le MRO
- Bonnes pratiques de la programmation objet avec Python

## La stdlib

Parcours de quelques modules phares

- Gestion de fichiers
- Argparse
- Logging
- Configparser
- Os
- Subprocess
- re

