

# Essentiel du Python

## Bases de l'algorithmie

### Objectif :

- Savoir décomposer un problème complexe de manière simple
- Savoir concevoir un algorithme rigoureux
- Savoir traduire un algorithme en Python
- Savoir lire et comprendre un programme conçu par autrui
- Savoir détecter des bugs et les corriger
- Savoir utiliser des bibliothèques externes
- Savoir utiliser Python comme outil d'analyse numérique

### Public visé :

Toute personne souhaitant s'initier à Python.

### Pré-requis :

Aucun.

### Méthode pédagogique :

Exposés & ateliers de code.

### Matériel nécessaire :

Poste informatique avec un accès à internet haut débit.

### Durée :

Modulable sur demande de 1 à 5 jours.

## Programme détaillé :

### Présentation

- Modalités
- Objectifs du module

### Introduction et généralités

- Définitions
- Description d'algorithmes, pseudocode
- Les erreurs de programmation

### Les types de variables

### Prise en main de Python et de l'environnement de travail (pyCharm)

- Création d'un nouveau projet
- Création d'un nouveau fichier Python
- Exécution d'un script Python

### Les opérateurs arithmétiques

### Instruction conditionnelle

- Utilité des instructions conditionnelles
- Syntaxe d'une instruction conditionnelle (en pseudocode et en Python)
- Syntaxe du bloc d'instructions qui suit

### Différents types de condition

- Conditions simples et multiples
- Conditions sur les types des objets
  - Instructions conditionnelles imbriquées

### Les boucles

- Quest ce que c'est ?
- Quand les utilise-t'on ?
- Qu'est ce qui les différencie ?

### Les fonctions

### Debugger

- Quelle est son utilité et comment l'utiliser ?
- Les points d'arrêt
- Principe de « Step into », « Step over »

### Les types avancés de variables

- Listes et tuples
- Les dictionnaires
- A quoi ça sert ? Quand les utiliser ?

### Les exceptions

- Comment écrire du code plus sûr ?
- Comment intercepter les bugs et éviter le crash du programme ?

### Les packages

- Utilisation et importation de package
- introduction à Numpy, Scipy et Pandas

### Bibliothèques graphiques

- comment générer rapidement des graphiques séduisants et convaincants ?
- rapide tour d'horizon de quelques solutions