



# PYTHON Initiation

**Durée : 3 jours soit 21h**

**Objectifs :**

S'initier au langage Python et avoir les bases de la programmation et de l'algorithmique.

L'apprenant sera capable de :

- Structurer des programmes selon un algorithme
- Maîtriser les éléments de lexique et de syntaxe d'un langage pour écrire un programme
- Exécuter un programme
- Déboguer et tester un programme

**Prérequis :**

Aucun

**Réalisation :**

Présentiel et/ou distanciel

**Date(s) et lieu :**

A définir

**Public :**

Salariés, demandeurs d'emploi, agents collectivité, enseignants

**Accessibilité :**

Pour les personnes atteintes d'un handicap, nous contacter

**Débouchés :**

Tous métiers

**Contact :**

Edwige CILIONE, formatrice bureautique expérimentée depuis 25 ans et entourée d'une équipe de formateurs professionnels

**Modalités d'accès :**

A la signature de la convention et/ou de la prise en charge

**Délai d'accès :**

De 2 jours à 4 semaines

**Type action:**

Actions d'acquisition, d'entretien ou de perfectionnement des connaissances

**Méthode pédagogique :**

Méthode active, démonstrative et participative  
Exercices autonomes réguliers pour s'assurer de l'assimilation

**Méthode d'évaluation :**

**1. des acquis :**

Évaluation de l'atteinte des objectifs par grille critériée et/ou possibilité de passer un test TOSA.

**2. à chaud :**

Questionnaire évaluation de formation pour mesurer le niveau de satisfaction

Python



**Tarif : à partir de 2.370 € net de taxes**

*Exonéré de TVA*



# PYTHON Initiation

## CONTENU DU MODULE

### Introduction à Python

- Qu'est-ce que le langage Python ?
- Comment fonctionne Python ?
- Les environnements Windows, Linux, et Mac/OS

### Les bases du Python

- Présentation de l'interpréteur Python
- Les variables, Les commentaires, Les types
- Les opérateurs, précedence et associativité
- Conversion implicite et casting
- Les objets mutables et les objets immuables
- Les opérateurs de comparaison, is et id

### Les structures de contrôle

- Expressions, déclarations et types d'exécution
- Les instructions if, elseif et else, Les opérateurs logiques
- L'opérateur match, Opérateur ternaire, Les itérables
- La boucle for, La boucle while, Le type range()
- La fonction enumerate(), Les instructions break

### Les nombres

- Les nombres entiers, Les nombres à virgule flottante
- Opérateurs pour les nombres, Les fonctions pour les nombres

### Les chaînes de caractères

- Introduction aux chaînes de caractères
- Formater les chaînes de caractères
- Les index des chaînes de caractères
- len() et les méthodes pour les chaînes de caractères
- Autres méthodes pour les chaînes de caractères

### Les listes

- Introduction aux listes, Listes et index, liste et décompactage
- Les opérations sur les listes et les copies
- Fonctions natives et ajouter des éléments aux listes
- Supprimer des éléments d'une liste
- Rechercher dans une liste, Trier une liste

### Les dictionnaires

- Introduction aux dictionnaires
- Opérateur in et méthodes get(), copy() et clear()
- Parcourir les dictionnaires, Supprimer des éléments d'un dictionnaire

### Les tuples et les sets

- Introduction aux tuples, introduction aux sets
- Recherche et méthodes pour les tuples, méthodes pour les sets
- Comparaison de sets

### Les fonctions

- Introduction aux fonctions
- Paramètres et arguments
- Passer des valeurs dynamiques
- Passer un nombre indéfini d'arguments
- Contexte d'exécution et environnement lexical
- Exemples de résolution des variables
- Les fermetures, Les instructions global et nonlocal
- Les annotations de fonction
- Les chaînes de documentation

### Les classes et les objets

- Introduction aux classes
- La syntaxe des classes
- La méthode \_\_init\_\_()
- Les méthodes
- Variables de classe, méthodes de classe et statiques
- Propriétés publiques et privées, L'héritage
- Les fonctions internes isinstance() et isinstance()
- L'héritage multiple
- Méthodes spéciales

### La programmation fonctionnelle

- Présentation de la programmation fonctionnelle
- Exemple de programmation fonctionnelle
- Les fonctions natives map(), filter(), zip() et reduce()
- Les expressions lambda
- La compréhension de listes de sets et de dictionnaires

### La gestion d'erreurs

- Introduction à la gestion d'erreurs
- Les classes des exceptions
- Les instructions try, except et finally
- L'instruction except en détail
- Exceptions personnalisées

### Les modules et pip

- Introduction au système de modules
- Les modules, Nom des modules et paquets
- Présentation de PyPI
- Utilisation de pip
- Environnements virtuels

### Le module datetime

- Introduction au module datetime
- Les classe time, date, datetime et timedelta
- Les informations sur le fuseau horaire

### Le module regex

- Introduction aux expressions régulières
- Syntaxes de base
- Les méthodes findall(), finditer() et match()
- La syntaxe des expressions régulières
- Classes et quantification
- Les groupes et les lookahead

### Introduction aux tests

- Introduction aux tests
- L'instruction assert, Le module doctest
- Introduction au module unittest
- Les méthodes skip() et assertNotEqual()
- Les méthodes assertIs(), assertTrue(), assertNone(), assertIn() et assertIsInstance()
- Tester des cas d'erreur, Initialisation et nettoyage

### Système de fichiers

- Introduction au système de fichiers
- Lire des fichiers
- Écrire dans des fichiers

**L'apprenant pourra, par la suite, suivre une formation correspondant au niveau suivant.**

Le nombre de participants est de **5 maximum** par stage.

Un support et les documents de cours sont remis à chaque participant.

Chaque stagiaire dispose d'un micro-ordinateur.

L'apprentissage est basé sur des exercices pratiques entre chaque module de cours.