

# Formation Python Initiation

**Formation éligible au CPF, contactez-nous au 09.72.37.73.73 pour en savoir plus**

<b>Durée :</b>	3 jours
<b>Public :</b>	Développeurs
<b>Pré-requis :</b>	Aucun
<b>Objectifs :</b>	Connaître les possibilités et limites de Python - Maîtriser la syntaxe essentielle de Python
<b>Référence :</b>	PYT252-F
<b>Code CPF :</b>	Nous contacter
<b>Demandeurs d'emploi :</b>	Financement CPF possible, contactez-nous au 09.72.37.73.73

## Découvrir le langage

Historique, alternatives

A propos de Python 2 et Python 3

Présentation et installation de Python dans un environnement Windows ou Linux

Mise en oeuvre de Python : accès au terminal

Environnements de développement assistés ; association à des outils connus et éditeurs pratiques

### Atelier : Mise en place d'un environnement de développement

## Maîtriser la syntaxe de base

Syntaxe de base, en interprétation directe et en script

Déclarer des variables en Python, types internes et leur utilisation

Utiliser les opérateurs et expressions

Présentation et détail des structures de contrôle : les tests, les boucles

Utiliser les boucles et tests pour réaliser des algorithmes

Se servir des tableaux, algorithmes divers de traitement de données massives

Types de données et opérations avancées (ensembles, dictionnaires, ordonnancement...)

Gestion des exceptions

### Atelier : Multiples algorithmes pour maîtriser la syntaxe de base.

## Structurer le code avec des fonctions

Intérêt des structurations du code dans le projet et pour le groupe de projets

Créer et organiser les fonctions

Retour des fonctions, absence de retour, multiples retours et intérêt dans un algorithme complexe

Espaces de noms, manipulation

Créations de bibliothèques de fonctions, documentation

Opérations les chaînes de caractères

Calculer avec Python

Gestion des dates, heures, intervalles de temps avec Python

Fonctions diverses nécessaires

## **Atelier : Construction d'une bibliothèque de fonctions**

### **Maîtriser l'Objet et modéliser une implémentation**

Différents styles de programmation, apports de l'objet

Objet et classes

Gestion des attributs et méthodes

Instanciation, destruction

Encapsuler les données et utiliser des propriétés

Agrégation d'objets

Héritage simple et multiples

Polymorphisme

Introspection

Utilisation d'interfaces

Modélisation UML : diagrammes de classes, d'objets

**Atelier : Conception UML d'une application et implémentation des classes métiers**