# Titre

Initiation à la programmation avec Python

# Contenu validé

Oui

# Catch phrase (2 lignes max)

Découvrez le développement et la programmation informatique en pratiquant avec l’un des langages les plus utilisés de ces dernières années.

# Description (10 lignes max)

Vous découvrirez les concepts clés de la programmation (algorithmique, variables, fonctions) petit à petit, tout en vous familiarisant avec la syntaxe de Python. Ainsi durant ce cours, nous découvrirons l’état d’esprit et les étapes clés nécessaires pour créer un programme. Nous utiliserons Python, car ce langage est à la fois simple et puissant, mais les concepts clés que nous aborderons peuvent être réutilisés quelles que soient les technologies choisies.

Cette formation commence par une présentation simple des concepts informatiques (matériel, logiciels), pour arriver par cas pratiques (écritures de fonctions, études de codes simples, recherches d’anomalies) à la compréhension et la création de programmes en Python.

# 

# Identifiant technique

pydebutant

# Domaine

développement

# Sous-domaine

Python

# Url

/formations/developpement/python-debutant

# Ordre dans la page

1

# Prochaines sessions

## Lyon

21-22 juin 2021

## Lille

14-15 juin 2021

## Paris

28-29 juin 2021

# Équilibre théorie / pratique

25% théorie / 75% pratique

# Public visé / participants

* Experts métiers
* Chefs de projets

# Prérequis

Aucun. Une grande curiosité est conseillée.

# Durée

3j soit 21 h

# Tarifs

1650 € HT (pauses et repas inclus)

# Objectifs pédagogiques

* Décomposer un problème en sous-problèmes indépendants
* Établir un algorithme permettant de résoudre un problème
* Maîtriser les éléments de lexique et de syntaxe de Python pour traduire un algorithme et écrire un programme
* Utiliser Python (écriture et exécution) pour résoudre un problème
* Tester un programme et définir un jeu d’essai

# 

# Plan de formation / programme

## **Rappels sur le fonctionnement d’un ordinateur**

### Présentation du matériel présent dans un ordinateur de bureau classique

* Disque dur
* RAM
* Processeur
* Carte graphique
* ...

### Explication des spécificités de chacun des périphériques (en lien avec le développement logiciel)

## **Qu’est-ce qu’un programme ?**

### Description succincte des différents langages de programmation et des courants de pensée (orienté objet, fonctionnel, …)

### Présentation de l’approche analytique des problèmes (décomposition) pour arriver à créer un algorithme

### Présentation des outils nécessaires à l’écriture de programmes (éditeurs de textes, IDE, …)

TP : trouver l’algorithme permettant de résoudre un problème simple, installation des outils de développement utilisés pendant le cours

## **Initiation à Python**

### De l’algorithme au programme

* Explication des différentes opérations possibles sur les données (conditions, boucles, assignation) et leur représentation en python

### Les variables

* Qu'est-ce qu'une variable ?
* Les types de variables et leurs caractéristiques
* Création et modification des variables
* Affichage de variables
* Conversion des différents types de variables
* Variables de types conteneurs (listes, dictionnaires…)

### Opérateurs et compositions d’opérateurs

* Les différents opérateurs (logiques, affectations, comparaison...)
* Les opérations (addition, division…)

### Les blocs pour l’exécution conditionnelle et la répétition d’instructions

* Découverte des mécanismes d’exécution conditionnelle (si / sinon si / sinon)
* Présentation des blocs et de leur syntaxe (début, fin)
* Présentation des instructions d’itération (while, for) et de leurs spécificités en Python.
* Imbrication de différents blocs

TP : exécution de notre premier programme python, traduire des algorithmes simples en python et réciproquement, écriture de codes Python simples permettant de manipuler des conditions et des itérations

## **Structuration d’un code Python**

### Présentation des conventions de nommages et de structuration du code

### Présentation du rôle et de l’utilisation des commentaires.

### Quelques techniques pour améliorer la lisibilité de ses programmes : découpage du code, structuration

### Introduction du concept de fonction et de leurs intérêts

### Les fonctions en Python

* Concepts de signature, paramètres et valeurs renvoyées.
* Bonnes pratiques de nommage

### Comment résoudre ses erreurs ?

* Compréhension et interprétation des messages d'erreur.
* Découverte du concept de tests unitaires pour faciliter le développement
* Présentation et utilisation d’un débugueur graphique (point d’arrêt, affichage de variables)

TP : mise en pratique sur un projet nécessitant d’organiser correctement son code

## **Utiliser Python dans des cas métiers**

### Lire et écrire des fichiers textes pour en extraire des informations

### Identifier les bibliothèques incluses de base dans Python

### Profiter du très riche écosystème de Python

* Trouver une bibliothèque
* Installer une bibliothèque

TP : mise en pratique sur des problématiques classiques des stagiaires (envois de mails, lecture / écriture de JSON et CSV, création automatique de rapports à partir de fichiers Office…)