

# Initiation à la programmation avec Python



Spoc pratique  
**Travaux pratiques continus**



Durée  
**2 jours**



Votre animateur  
**Philippe Brutus**



Code Kertify : **PYT**

**S'inscrire**



[www.kertify.com/inscription](http://www.kertify.com/inscription)

## Apprendre les bases de la programmation avec Python

Cette formation présente les bases de la programmation et met en œuvre les différentes phases de la production d'un programme, en utilisant le langage Python.

### Les objectifs de la formation

- ◆ Savoir décomposer un problème et structurer un programme
- ◆ Maîtriser les éléments du langage Python (mots-réservés, syntaxe) pour écrire un programme
- ◆ Maîtriser le développement Python : Exécuter un programme, Tester et mettre au point un programme

### Participants

- ◆ Toute personne ayant à apprendre à programmer

### Prérequis

- ◆ Aucune connaissance préalable n'est requise

## Le programme de notre formation

### 1. Les bases de la programmation

- ◆ Notions de programme et de langage de programmation
- ◆ Compilation et exécution
- ◆ Responsabilités d'un programmeur
- ◆ Notion d'algorithme et pseudo-langage de programmation
- ◆ Développer une application pour répondre à des besoins

### 2. Production d'une application

- ◆ Ecriture du code (syntaxe et instructions)
- ◆ Exécution du programme
- ◆ Notion de bibliothèque : rôle et usage

### 3. Règles de programmation

- ◆ Conventions de nommage
- ◆ Conventions syntaxiques
- ◆ Utilisation des commentaires
- ◆ Maîtrise du code par découpage en sous-programmes
- ◆ Indentation du code pour définir des blocs d'instructions et améliorer la lisibilité

### 4. Les variables

- ◆ Notion de variable et de constante
- ◆ Types de données : panorama des types, rôle et conséquences du typage
- ◆ Types primitifs : nombres entiers et à virgule, chaînes de caractères...
- ◆ Déclaration, initialisation et affectation d'une variable

- ◆ Instructions de saisie, d’affichage, d’affectation et de conversion de type
- ◆ Gérer plusieurs valeurs de même type dans un tableau

---

## 5. Opérateurs et expressions

- ◆ Opérateur d’affectation
- ◆ Opérateurs arithmétiques
- ◆ Opérateurs de comparaison (sur des nombres, sur des textes)
- ◆ Opérateurs logiques (expressions composées)
- ◆ Expression booléenne

---

## 6. Structures de contrôle

- ◆ Conditionnelles (si-alors, si-alors-sinon, selon) et itérative (pour)
- ◆ Bloc d’instruction (début et contenu indenté)
- ◆ Imbrication des structures de contrôles (boucle dans une conditionnelle, conditionnelle dans boucle...)

---

## 7. Sous-programmes

- ◆ Notion de sous-programme : définition, rôle et importance
- ◆ Différents types de sous-programmes : procédures et fonctions
- ◆ Appel de sous-programme
- ◆ Paramètres de sous-programme, arguments d’appel de sous-programme, mode de passage
- ◆ Résultat d’une fonction : instruction de retour

---

## 8. Maintenance corrective, mise au point et test des programmes

- ◆ Message d’erreur : interprétation du message, localisation de l’erreur
- ◆ Utiliser un outil de mise au point : exécution pas à pas, inspection des variables, points d’arrêt
- ◆ Tests unitaires : définition, conception