

IDF – Apprendre à programmer en Python

Objectifs	Être capable, après un passage en revue des notions de bases en algorithmique, de mettre en place un environnement de développement afin d'écrire des scripts Python. Utiliser des bibliothèques existantes, implémenter vos propres bibliothèques et créer des interfaces graphiques
Public	Biologistes, chercheurs, ingénieurs ou techniciens, débutants en programmation
Pré-requis	Connaissances de base en informatique
Programme	<p>Les notions de base de la programmation et présentation de Python</p> <ul style="list-style-type: none">▪ Qu'est-ce qu'un programme informatique ?▪ Que signifient les termes algorithmique, langage de programmation ?▪ Où se situe Python dans le domaine de la programmation ? <p>Installation et configuration de l'interpréteur Python</p> <ul style="list-style-type: none">▪ Les versions de Python▪ La gestion de version et de paquets avec Anaconda▪ L'IDE PyCharm/VSCode (Microsoft)▪ Les Notebooks Jupyter <p>Premiers pas en Python</p> <ul style="list-style-type: none">▪ Se familiariser avec l'interpréteur▪ L'indentation et les commentaires▪ Les opérateurs et expressions▪ Les variables▪ Les structures conditionnelles▪ Les boucles <p>La modularité</p> <ul style="list-style-type: none">▪ Les fonctions définies par le mot clef <code>def</code>▪ Les fonctions <code>lambda</code>▪ La bibliothèque standard, les modules et le mot clef <code>import</code>▪ Les autres bibliothèques et le système de gestion <code>pip</code>▪ Sauvegarder, organiser et appeler son code : les fichiers <code>.py</code> <p>Les erreurs</p> <ul style="list-style-type: none">▪ Principes fondamentaux des exceptions▪ Gérer les erreurs simplement avec les blocs <code>try</code> et <code>except</code> <p>Les objets en Python</p> <ul style="list-style-type: none">▪ Les principes du paradigme de la programmation objet▪ Un exemple : la classe <code>str</code> et ses méthodes associées▪ Travailler avec des séquences d'éléments de base : les listes et tuples▪ Travailler avec des séquences d'objets : les dictionnaires▪ Le module <code>os</code> et la gestion des fichiers▪ La portée des variables en Python <p>Les classes</p> <ul style="list-style-type: none">▪ Les conventions de nommage et de gestion▪ La création d'une classe▪ Les attributs▪ Les méthodes de classe et les méthodes statiques

	<ul style="list-style-type: none">▪ La philosophie de Python comparée à celle des autres langages concernant l'encapsulation▪ La classe property▪ Les constructeurs▪ Le paramètre self▪ Utiliser sa classe dans le code▪ Héritage <p>Les mathématiques et les statistiques en Python</p> <ul style="list-style-type: none">▪ Le module <i>math</i>▪ Le module <i>fractions</i>▪ Le module <i>statistics</i> <p>La qualité du code</p> <ul style="list-style-type: none">▪ Vérifier le code : <i>Pylint</i> et <i>Pychecker</i>▪ Déboguer le code avec la librairie <i>ipdb</i>▪ Les tests unitaires avec <i>Unittest</i> <p>Interfaces Homme <-> Machine</p> <ul style="list-style-type: none">▪ Les interfaces graphiques avec le module Tkinter▪ Créer des fenêtres▪ Utiliser les widgets▪ Utiliser les commandes de base pour gérer les évènements		
Dates et lieu	<p>Dates :</p> <ul style="list-style-type: none">• les 02, 03 et 04 décembre 2026 <p>Clôture des inscriptions : le 13 novembre 2026 sur https://www.sirene.inserm.fr/</p> <p>Formation en présentiel</p>		
	<table><tr><td><p>Correspondant.e.s formation :</p><hr/><hr/></td><td><p>INSERM DR Paris IDF Sud Formation Continue 48-50 rue Albert 75013 Paris @ : formation.paris.11@inserm.fr ☎: 01.85.55.38.44 / 01.85.55.38.11</p></td></tr></table>	<p>Correspondant.e.s formation :</p> <hr/> <hr/>	<p>INSERM DR Paris IDF Sud Formation Continue 48-50 rue Albert 75013 Paris @ : formation.paris.11@inserm.fr ☎: 01.85.55.38.44 / 01.85.55.38.11</p>
<p>Correspondant.e.s formation :</p> <hr/> <hr/>	<p>INSERM DR Paris IDF Sud Formation Continue 48-50 rue Albert 75013 Paris @ : formation.paris.11@inserm.fr ☎: 01.85.55.38.44 / 01.85.55.38.11</p>		