



Formation Python : Les fondamentaux

Code : DC05

Durée : 5 jours

Classe : Présentiel / à distance

Public

- Développeurs débutants en Python
- Ingénieurs et data scientists manipulant des données
- Toute personne souhaitant renforcer sa culture en programmation

Prérequis

- Connaissance d'au moins un langage de programmation (bases : variables, boucles, conditions, fonctions)
- La connaissance de la programmation orientée objet est un plus, mais non obligatoire

Objectifs

- Concevoir rapidement des applications Python complètes et évolutives
- Organiser le code avec classes, fonctions, modules et concepts orientés objet
- Utiliser efficacement les modules et packages standards ou tiers
- Développer des interfaces graphiques dynamiques et multiplateformes
- Créer et déployer des applications Web basées sur les données

Programme détaillé

1- Bien commencer avec Python

- Pourquoi utiliser Python plutôt qu'un autre langage ?
- Exemples de projet en Python
- Différences entre les versions de Python
- Installation de Python (pré-requis, fichiers, arborescence détaillée)
- Fonctionnement de l'interpréteur Python
- Les apports de l'interpréteur interactif iPython

2- Les bases de Python (la syntaxe, les types)

- Organisation du code (modules, package, importations)
- Recommandations de nommage
- Les types de base (entier, chaîne, réel, ...)
- Erreurs classiques sur les types
- Boucles (for), opérateurs, tests (if/else)
- Définition et appel de fonction
- Introduction à la syntaxe conditionnelle avec match/case

3- Programmer objet en Python

- Limites de la programmation « simplement » structurée
- Philosophie, concepts, et vocabulaire de la programmation objet
- Représentation des données en programmation objet (classes, propriétés)
- Représentation des traitements (méthodes)
- Contrôler l'accès aux données (encapsulation)
- Exemples de classes prédéfinies en Python permettant de profiter pleinement de l'approche objet
- Cycle de vie d'un objet (constructeur, destructeur)
- Accéder aux attributs et fonctionnalités de l'objet cible (self)
- Comment les objets communiquent entre eux ?
- Classe mère et classe fille (héritage)
- Des fonctions différentes avec ... le même nom (polymorphisme)
- Héritage simple et héritage multiple



Formation Python : Les fondamentaux

Code : DC05

Durée : 5 jours

Classe : Présentiel / à distance

- La fonction super()
- Protection des traitements (public, protected, private)
- Le mécanisme de traitement des exceptions
- Exemple de formalisme UML (diagramme de classe)
- Tour d'horizon des classes fondamentales existantes en Python
- Notion de Design Patterns

4- Types de données évolués

- Savoir choisir les bonnes structures de données
- Tuples, séquences et listes (append, extend, insert, ...)
- Fonctions utiles avec les listes (filter, map, reduce)
- Gestion des piles ou des files
- La puissance des dictionnaires (tableaux associatifs)
- Construire une liste sur la base d'une expression (compréhension)

5- Compléments sur le langage Python

- Générateurs et itérateurs
- Scripts exécutables
- Opérateurs associés aux listes
- Enchaînement de tests par rapport aux listes
- Comparer les séquences

6- Les modules de Python

- Philosophie de Python avec les modules (standard, tiers, ...)
- Tour d'horizon des modules standards
- Module re (expressions rationnelles)
- Modules os et sys (services du système d'exploitation)
- Module csv (fichiers structurés par séparateur)
- Comment trouver les modules ?
- Installer des modules tiers (setuptools, EasyInstall)
- Le dépôt central : Python Package index (Pypi)
- Importer des modules
- Mieux gérer son environnement avec venv, pipx, poetry ou pip-tools

7- Gestion des données (Fichiers, SGBD, XML, JSON, API)

- Accéder aux fichiers (objet File)
- Conformité des modules Python pour accéder à une base de données (DB-API)
- Python et requêtes SQL, ORM avec SQLAlchemy
- Filtrer les données grâce aux expressions régulières (RE)
- Présentation de la manipulation de flux XML en Python (ElementTree)
- Manipuler des fichiers CSV

8- Échange de données via API et formats standards

- Le format JSON : encodage/décodage, gestion de structures complexe
- Sérialisation avec dataclasses et pydantic
- Principe des API et des Web Service
- Requêtes HTTP avec requests (GET, POST, headers, paramètres)
- Récupérer des données via des API externe
- Traitement et validation des données reçues (modèles Pydantic, typage fort, gestion des erreurs)



Formation Python : Les fondamentaux

Code : DC05

Durée : 5 jours

Classe : Présentiel / à distance

9- Création d'interface graphique

- Spécificités du développement d'interface graphique (programmation événementielle)
- Présentation de la bibliothèque Tkinter
- Les principaux conteneurs
- Présentation des widgets disponibles (Button, Radiobutton, Entry, Label, Listbox, Canvas, Menu, Scrollbar, Text...).
- Le gestionnaire de fenêtres
- Le placement des composants, les différents layouts
- La gestion des événements, l'objet event
- Les applications multifenêtres
- Un petit mot sur la bibliothèque Qt
- Les autres librairies d'IHM (PyQt, PySide, ..) les IHM web (flask/streamlit)

10- Qualité et outils

- En quoi consiste l'assurance qualité d'un programme Python (QA) ?
- Vue d'ensemble de la « trousse à outils » proposée au développeur
- Utiliser les plugins des IDEs IntelliJ ou VSCode
- Rechercher des bugs avec PyChecker
- Vérifier le respect des standards avec PyLint
- Utiliser le guide de style PEP 8 (Python Enhancement Proposals)
- Générer de la documentation
- Disposer de statistiques sur l'exécution de son programme (profiling)
- Comprendre l'intérêt des outils de Tests (Doctests, Unit tests)

