

Développer avec Python

Durée : 5 jours

Présentation

Python s'est imposé ces dernières années comme l'un des langages de programmation les plus populaires. Utilisé aussi bien pour le développement, que pour la data science ou l'automatisation, sa syntaxe simple et accessible permet de faire un premier pas dans le monde de la programmation.

À travers cette formation, vous apprendrez des notions essentielles telles que l'approche orientée objet, l'utilisation de quelques bibliothèques utiles, et la mise en place de bonnes pratiques mises préconisées par la communauté Python.

Objectifs de la formation

- Maîtriser la syntaxe Python
- Comprendre les concepts de la programmation orientée objet
- Mettre en place des hiérarchies d'héritage
- Utiliser des bibliothèques populaires
- Mettre en œuvre de bonnes pratiques de programmation Concevoir des interfaces graphiques

Prérequis

- Connaissances de base en algorithmie

Public

- Développeurs
- Chefs de projets
- Data scientists
- Administrateurs système et réseau

Programme de la formation

Présentation du langage

- Historique et dates importantes
- Les différences entre Python 2 et Python 3
- L'interpréteur
- Utiliser Python en mode interactif
- Le site officiel et la documentation
- Bien choisir son environnement de développement
- Atelier : installation de l'interpréteur et de l'IDE

Syntaxe de base du langage

- Les instructions et commentaires
- Les variables

- Les différents types
- Les particularités du typage en Python
- Afficher et récupérer des informations dans la console
- Les séquences
- Le slicing
- L'affectation par référence
- La décomposition

Les blocs d'instructions

- Cimportance de l'indentation en Python
- Imbriquer des blocs d'instructions
- Les conditions
- Les clauses elif et else
- Les différents opérateurs
- La boucle while
- La boucle for
- Les instructions de branchement
- Travaux pratiques : réalisation d'algorithmes (suite de Fibonacci, calcul de PGCD, détermination d'années bissextiles, simulation d'automates, ...)

Les fonctions

- Définition et appel d'une fonction
- Passer des paramètres à une fonction
- Retourner une valeur
- Les étiquettes des paramètres
- Attribuer une valeur par défaut à un paramètre
- Les paramètres à nombre variable *args et **kwargs
- La portée des variables
- Les variables globales
- Les expressions lambda

L'approche objet

- La notion de classes et instances
- Diagramme de classes UML
- Une première classe
- Le constructeur
- La définition des attributs
- L'autoréférence
- Les méthodes
- L'instanciation
- Accéder aux attributs d'un objet
- Appeler une méthode sur un objet
- Les attributs de classes
- Les méthodes de classe et méthodes statiques
- Les méthodes spéciales
- Travaux pratiques : réalisation d'une simulation de moteur de jeu

Héritage et polymorphisme

- Le principe de l'héritage
- Héritage simple
- La redéfinition de méthode
- L'héritage multiple
- Polymorphisme et duck typing

Gestion des exceptions

- Capturer des exceptions
- Les blocs try / except
- Capturer une exception précise
- Les mots-clés else et finally
- Les assertions
- Lever une exception
- Créer ses propres types d'exception
- Travaux pratiques : application directe de l'héritage et de la gestion des exceptions

Utilisation des librairies du langage

- Gestion des fichiers : lecture et écriture
- Les modules de base : math, random, datetime, time, ...
- Manipuler les expressions régulières : le module re
- Opérations sur le système : les modules sys et os
- Traiter des fichiers de données : les modules json et csv
- Communiquer avec une base de données : le module sqlite3
- Les différentes manières d'importer un module
- Créer un package et l'importer
- Le gestionnaire de paquets pip
- Le Python Package Index(PyPI) Installer un module externe

Bonnes pratiques

- La convention PEP 8
- Zen of Python (PEP 20)
- L'encodage en Python 3
- Les tests unitaires
- Le debugger
- Mesurer la qualité du code
- Travaux pratiques : sérialisation et persistance de données, réalisation de tests unitaires et audit du code

Réalisation d'interfaces graphiques

- Présentation des différentes librairies
- Une première fenêtre
- Créer des widgets
- Positionner les widgets dans la fenêtre
- Récupérer des informations d'un formulaire
- Les événements
- Mise en place du pattern MVC

- Travaux pratiques : réalisation d'une IHM simple