

Python

Objectifs pédagogiques

- Mettre en œuvre les aspects avancés de la programmation objet
- Structurer et déployer vos applications
- Créer et appliquer les décorateurs
- Exploiter les itérateurs et les générateurs
- Créer et utiliser les "context manager"
- Mettre en œuvre les mécanismes de la programmation fonctionnelle
- Optimiser les performances en utilisant la programmation asynchrone
- Manipuler vos fichiers de données de façon optimisée avec la librairie « Panda »
- Réaliser des calculs performants à l'aide de la librairie « NumPy »

Prérequis

- Développer en Python, maîtriser les concepts de base du langage

Programme détaillé

Aspects avancés de la programmation objet

- Rappel sur les classes et l'héritage
- Les variables et les méthodes de classes
- Contrôle de la création d'objet (méthode `__new__`)
- Les classes et les méthodes abstraites
- Le module abc
- La notion de méta classe
- Travaux pratiques :
 - Créer et tester le bon fonctionnement d'une classe représentant un Produit.
 - Créer une classe Stock qui va gérer les Produits.

Organisation des projets et distribution des applications

- Les principes de packaging et distribution
- Les différentes API
- La présentation de l'API `setuptools`
- Le déploiement d'application avec `setuptools`
- Travaux pratiques :
 - Créer un package de déploiement pour votre application de gestion de produits.

Les décorateurs

- Le principe de fonctionnement et les usages
- La définition des fonctions de décoration
- L'application des décorateurs
- Les décorateurs standard du langage
- Travaux pratiques :
 - Créer un décorateur qui va gérer des logs sur les méthodes de la classe Produit

Les itérateurs et les générateurs

- Principe de fonctionnement
- Création d'itérateurs
- Implémentation de générateurs
- Le mot-clé yield
- Travaux pratiques :
 - Créer un itérateur et un générateur qui vont parcourir la liste de produits.

La programmation fonctionnelle

- Les principes de base de la programmation fonctionnelle
- La définition de fonctions lambda
- Les fonctions natives filter, map et reduce
- Les listes de compréhensions
- Travaux pratiques :
 - Modifier les différentes méthodes d'accès aux produits en intégrant les expressions lambda pour alléger votre code.

Les « context managers »

- Principe des « context managers »
- L'instruction with
- Déclaration de « context managers »
- Les « context managers » et les décorateurs
- Le Contextlib
- Travaux pratiques :
 - Création d'un « context manager » pour faciliter le parcours de la liste de produits

Manipulation de données avec la librairie Panda

- Introduction à la bibliothèque
- Les séries et les dataframes
- La lecture et l'écriture de fichiers
- Les différentes opérations sur les dataframes
- Travaux pratiques :
 - Stocker les différents Produits dans des fichiers à l'aide de la librairie Panda

Réalisation de calculs scientifiques avec la librairie NumPy

- Présentation de la bibliothèque
- La création de tableaux (ndarray)
- Les différentes opérations disponibles sur les tableaux
- Travaux pratiques :
 - Optimiser les fonctionnalités de votre gestion de stock à l'aide des fonction NumPy

Programmation parallèle et asynchrone

- L'objectif de la programmation asynchrone
- Le modèle de programmation
- Le module thread et les classes associées
- Le module asyncio
- Travaux pratiques :
 - Optimiser l'application de gestion de produits en réalisant des traitements asynchrones pour accéder aux produits.