

Formation Python Intermédiaire : Bonnes pratiques + Multithreading et Tests

Formation éligible au CPF, contactez-nous au 02/318.50.01

Durée :	5 jours
Public :	Développeurs Python
Pré-requis :	Avoir suivi le stage "Python : Initiation + Approfondissement" ou notions équivalentes
Objectifs :	Maîtriser les éléments avancés du langage, le multi-threading et l'implémentation de tests
Sanction :	Attestation de fin de stage mentionnant le résultat des acquis
Taux de retour à l'emploi:	Aucune donnée disponible
Référence:	PYT100589-F
Code CPF:	RS6202 - contactez-nous au 02/318.50.01
Note de satisfaction des participants:	4,69 / 5
Certifications :	TOSA 2022-Python score : 620,79 / 1000 calculé le 01/04/2024

Découvrir les éléments avancés du langage

- Expressions Lambda
- Les contextes
- Les itérables
- Les générateurs
- Les décorateurs
- Les métaclasses

Atelier : Multiples créations : itérable, générateur, itérateur

Gérer des modules Python

- Gestion des modules tiers
- Pypi (Python package index)
- Installation, désinstallation d'un package
- Déploiement avancé (virtualenv, buildout)

Atelier : création et installation de modules depuis le dépôt central

Découvrir les design patterns

Origine, catalogue
Structure d'un patron de conception
Application d'un pattern

Gérer des instances de classes

Factory et Abstract Factory pour la création sous condition
Singleton et dérivé : maîtrise des ressources disponibles

Atelier : implémentation d'un singleton et d'une factory en Python

Structurer des données

Le Composite, comment simplifier les listes
Proxy et Adapter, les interfaces de l'accès aux méthodes
La Facade : clarifier un composant

Atelier : utilisation d'un composite dans l'application

Maîtrise du comportement

Strategy : l'usine à méthodes
L'itérateur et ses implémentations existantes
Observer : l'événementiel sans événements
Chaîne de responsabilité et arbres de responsabilité
Visiteur et accès : maîtrise de la collaboration

Atelier : implémentation d'un observer pour la surveillance de variation sur un objet

Apprendre la programmation concurrente avec Python

Processus et threads
Multithreading : introduction à la programmation multi-tâches
Présentation du module Threading
Instanciation et lancement de threads
Cycle de vie
Priorités
GIL : Global Interpreter Lock
Communication inter-processus, synchronisation
Notion de futures
Traitement asynchrone

Atelier : Implémentation de traitements en multi-threading

Comprendre le processus de tests

Métriques de qualité logicielle
Processus de test dans le cycle de vie d'un projet, types, apports
Pratiques XP (eXtreme Programming), Test Driven Development (TDD) et styles
Tests dans une gestion de projet agile
Refactoring de code
Fixtures, Qualités d'un code de test

Couverture de test
Langage Gherkin, génération de script avec Cucumber

Atelier : rédaction de tests dans un projet

Implémenter des tests unitaires avec unittest

Cas de test, suite de tests
Principe d'assertions
Test de retour des fonctions
Court-circuiter les effets de bord avec unittest.mock

Atelier : implémentation de tests unitaires avec unittest

Réaliser des tests d'IHM web

Frameworks de tests fonctionnels : Selenium vs autres
Envoi des requêtes HTTP
Extraction d'information à partir des données reçues
Parsing HTML, JSON

Atelier : Automatisation de tests avec Selenium