

# Python Intermédiaire : Bonnes pratiques + Multithreading et Tests

## INFORMATIONS GÉNÉRALES

**Type de formation :** Formation continue

**Éligible au CPF :** Non

**Domaine :** Développement

**Action collective :** Non

**Filière :** Open Source - LAMP : Linux Apache PHP

**Rubrique :** Langages : Python, Scala, GO,...

**Code de formation :** LEDN221

## € Tarifs

**Prix public :** 2995 €

### Tarif & financement :

Nous vous accompagnons pour trouver la meilleure solution de financement parmi les suivantes :

**Le plan de développement des compétences de votre entreprise :** rapprochez-vous de votre service RH.

**Le dispositif FNE-Formation.**

**L'OPCO** (opérateurs de compétences) de votre entreprise.

**France Travail:** sous réserve de l'acceptation de votre dossier par votre conseiller Pôle Emploi.

**CPF -MonCompteFormation**

Contactez nous pour plus d'information : [contact@aston-institut.com](mailto:contact@aston-institut.com)

## PRÉSENTATION

### Objectifs & compétences

A l'issue de la formation, le stagiaire sera capable :

- Maîtriser les éléments avancés du langage, le multi-threading et l'implémentation de tests

### Public visé

Développeurs Python

### Pré-requis

Avoir suivi le stage "Python : Initiation + Approfondissement" ou notions équivalentes

## 📍 Lieux & Horaires

**Durée :** 35 heures

**Délai d'accès :**

**8 jours avant le début de la formation**

## PROGRAMME

### 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

## 📅 Prochaines sessions

Consultez-nous pour les prochaines sessions.

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

## MODALITÉS

### Modalités

Jusqu'à 8 jours avant le début de la formation, sous condition d'un dossier d'inscription complet

### Méthode

**Fin de formation** : entretien individuel.

**Satisfaction des participants** : questionnaire de satisfaction réalisé en fin de formation.

**Assiduité** : certificat de réalisation.

**Validations des acquis** : grille d'évaluation des acquis établie par le formateur en fin de formation.