



## Formation Python Intermédiaire : Bonnes pratiques + Multithreading et Tests

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

■ <b>Durée :</b>	5 jours (35 heures)
■ <b>Tarifs inter-entreprise :</b>	2 995,00 € HT (standard) 2 396,00 € HT (remisé)
■ <b>Public :</b>	Développeurs Python
■ <b>Pré-requis :</b>	Avoir suivi le stage "Python : Initiation + Approfondissement" ou notions équivalentes
■ <b>Objectifs :</b>	Maîtriser les éléments avancés du langage, le multi-threading et l'implémentation de tests
■ <b>Modalités pédagogiques, techniques et d'encadrement :</b>	<ul style="list-style-type: none"> <li>• Formation synchrone en présentiel et distanciel.</li> <li>• Méthodologie basée sur l'Active Learning : 75 % de pratique minimum.</li> <li>• Un PC par participant en présentiel, possibilité de mettre à disposition en bureau à distance un PC et l'environnement adéquat.</li> <li>• Un formateur expert.</li> </ul>
■ <b>Modalités d'évaluation :</b>	<ul style="list-style-type: none"> <li>• Définition des besoins et attentes des apprenants en amont de la formation.</li> <li>• Auto-positionnement à l'entrée et la sortie de la formation.</li> <li>• Suivi continu par les formateurs durant les ateliers pratiques.</li> <li>• Évaluation à chaud de l'adéquation au besoin professionnel des apprenants le dernier jour de formation.</li> </ul>
■ <b>Sanction :</b>	Attestation de fin de formation mentionnant le résultat des acquis
■ <b>Référence :</b>	PYT100589-F
■ <b>Note de satisfaction des participants:</b>	4,69 / 5

■ <b>Contacts :</b>	commercial@dawan.fr - 09 72 37 73 73
■ <b>Modalités d'accès :</b>	Possibilité de faire un devis en ligne (www.dawan.fr, moncompteformation.gouv.fr, maformation.fr, etc.) ou en appelant au standard.
■ <b>Délais d'accès :</b>	Variable selon le type de financement.
■ <b>Accessibilité :</b>	Si vous êtes en situation de handicap, nous sommes en mesure de vous accueillir, n'hésitez pas à nous contacter à referenthandicap@dawan.fr, nous étudierons ensemble vos besoins

## Découvrir les éléments avancés du langage

Expressions Lambda  
Les contextes  
Les itérables  
Les générateurs  
Les décorateurs  
Les métaclases

### 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**