

## Formation Python Intermédiaire : Bonnes pratiques

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

<b>Durée :</b>	3 jours
<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 et les design patterns
<b>Sanction :</b>	Attestation de fin de stage mentionnant le résultat des acquis
<b>Taux de retour à l'emploi:</b>	Aucune donnée disponible
<b>Référence:</b>	PYT100616-F
<b>Code CPF:</b>	RS6202 - contactez-nous au 02/318.50.01
<b>Note de satisfaction des participants:</b>	4,95 / 5
<b>Certifications :</b>	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**