

Formation Python Intermédiaire : Bonnes pratiques

Formation éligible au CPF, contactez-nous au 09 72 37 73 73

■ Durée :	3 jours (21 heures)
■ Tarifs inter-entreprise :	1 995,00 € HT (standard - hors certification) 1 596,00 € HT (remisé - hors certification)
■ Public :	Développeurs Python
■ Pré-requis :	Avoir suivi le stage "Python : Initiation + Approfondissement" ou notions équivalentes
■ Objectifs :	Maîtriser les éléments avancés du langage et les design patterns
■ Certification :	La formation prépare à la certification Python détenue par Isograd, enregistrée le 18/12/2024 sous le numéro 6962 au Répertoire Spécifique de France Compétences . Taux de réussite : pas de données disponibles, calculé le 01/05/2025
■ Modalités pédagogiques, techniques et d'encadrement :	• Formation synchrone en présentiel et distanciel. • Méthodologie basée sur l'Active Learning : 75 % de pratique minimum. • Un PC par participant en présentiel, possibilité de mettre à disposition en bureau à distance un PC et l'environnement adéquat. • Un formateur expert.

■ **Modalités d'évaluation :**

- Définition des besoins et attentes des apprenants en amont de la formation.
- Auto-positionnement à l'entrée et la sortie de la formation.
- Suivi continu par les formateurs durant les ateliers pratiques.
- Évaluation à chaud de l'adéquation au besoin professionnel des apprenants le dernier jour de formation.

■ **Sanction :**

Attestation de fin de formation mentionnant le résultat des acquis

■ **Référence :**

PYT100616-F

■ **Note de satisfaction des participants:**

4,95 / 5

■ **Contacts :**

commercial@dawan.fr - 09 72 37 73 73

■ **Modalités d'accès :**

Possibilité de faire un devis en ligne (www.dawan.fr, moncompteformation.gouv.fr, maformation.fr, etc.) ou en appelant au standard.

■ **Délais d'accès :**

11 jours ouvrés en cas de financement CPF.

■ **Accessibilité :**

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