

# Python, programmation Objet

## 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 :** AS941

## PRÉSENTATION

### Objectifs & compétences

À l'issue de la formation, le stagiaire sera capable d'utiliser les principales fonctionnalités du langage de programmation Python, pour la conception, le développement et la maintenance d'applications.

Connaître la syntaxe du langage Python

Acquérir les principes de la programmation objet

Mettre en œuvre les fonctionnalités des modules Python et les bonnes pratiques associées

Savoir concevoir des interfaces graphiques

Appréhender l'utilisation des outils de test et d'évaluation d'un programme Python

### Public visé

Développeurs, ingénieurs, chefs de projets proches du développement

### Pré-requis

Avoir des connaissances de base en programmation (souhaitable en langage objet).

## € Tarifs

**Prix public :** 3090 €

### 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)

## 📍 Lieux & Horaires

**Campus :** Ensemble des sites

**Durée :** 35 heures

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

**Distanciel possible :** Oui

## PROGRAMME

### 1 - Syntaxe du langage Python

Les identifiants et les références

Les conventions de codage et les règles de nommage

Les blocs, les commentaires

Les types de données disponibles

Les variables, l'affichage formaté, la portée locale et globale

La manipulation des types numériques, la manipulation de chaînes de caractères

La manipulation des tableaux dynamiques (liste), des tableaux statiques (tuple) et des dictionnaires

L'utilisation des fichiers

La structure conditionnelle if/elif/else

Les opérateurs logiques et les opérateurs de comparaison

Les boucles d'itérations while et for

Interruption d'itérations break/continue

La fonction range

L'écriture et la documentation de fonctions

Les Lambda expression

Les générateurs

La structuration du code en modules

### 2 - Approche orientée objet

Les principes du paradigme Objet

La définition d'un objet (état, comportement, identité)

La notion de classe, d'attributs et de méthodes

L'encapsulation des données

La communication entre les objets

L'héritage, transmission des caractéristiques d'une classe

La notion de polymorphisme

Association entre classes

Les interfaces

## 📅 Prochaines sessions

Cliquez sur la date choisie pour vous inscrire :

■ 23 / 06 / 2025

📍 : Ensemble des sites

✓ : Distanciel possible

🕒 : 35 heures

📅 : 5 jours

■ 07 / 07 / 2025

📍 : Ensemble des sites

✓ : Distanciel possible

🕒 : 35 heures

📅 : 5 jours

■ 21 / 07 / 2025

📍 : Ensemble des sites

✓ : Distanciel possible

🕒 : 35 heures

📅 : 5 jours

■ 01 / 09 / 2025

📍 : Ensemble des sites

✓ : Distanciel possible

🕒 : 35 heures

Présentation d'UML

Les diagrammes de classes, de séquences, d'activités...

Notion de modèle de conception (Design Pattern)

**3 - Programmation objet en Python**

Les particularités du modèle objet de Python

L'écriture de classes et leur instanciation

Les constructeurs et les destructeurs

La protection d'accès des attributs et des méthodes

La nécessité du paramètre Self

L'héritage simple, l'héritage multiple, le polymorphisme

Les notions de visibilité

Les méthodes spéciales

L'introspection

L'implémentation des interfaces

Les bonnes pratiques et les modèles de conception courants

L'utilisation du mécanisme d'exception pour la gestion des erreurs

**4 - Utilisation STDLIB**

Les arguments passés sur la ligne de commande

L'utilisation du moteur d'expressions régulières Python avec le module "re", les caractères spéciaux, les cardinalités

La manipulation du système de fichiers

Présentation de quelques modules importants de la bibliothèque standard : module "sys", "os", "os.path"

Empaquetage et installation d'une bibliothèque Python

Les accès aux bases de données relationnelles, le fonctionnement de la DB API

**5 - Outils QA**

Les outils d'analyse statique de code (Pylint, Pychecker)

L'analyse des comptes rendus d'analyse (types de messages, avertissements, erreurs)

Extraction automatique de documentation

Le débogueur de Python (exécution pas à pas et analyse post-mortem)

Le développement piloté par les tests

Les modules de tests unitaires Python (Unittest...)

L'automatisation des tests, l'agrégation de tests

Les tests de couverture de code, profiling

**6 - Création IHM TKINTER**

Les principes de programmation des interfaces graphiques

Présentation de la bibliothèque Tkinter

Les principaux conteneurs

Présentation des widgets disponibles (Button, Radiobutton, Entry, Label, Listbox, Canvas, Menu, Scrollbar, Text...)

Le gestionnaire de fenêtres

Le placement des composants, les différents layouts

La gestion des événements, l'objet event

Les applications multi-fenêtres

**7 - Interfaçage Python/C**

Présentation du module Ctypes

Le chargement d'une librairie C

Appel d'une fonction

La réécriture d'une fonction Python en C avec l'API Python/C

La création de modules C pour Python avec Pyrex

L'interpréteur Python dans C

L'utilisation du profileur de code

📅 : 5 jours

■ 15 / 09 / 2025

📍 : Ensemble des sites

✓ : Distanciel possible

🕒 : 35 heures

📅 : 5 jours

■ 03 / 11 / 2025

📍 : Ensemble des sites

✓ : Distanciel possible

🕒 : 35 heures

📅 : 5 jours

■ 03 / 11 / 2025

📍 : Ensemble des sites

✓ : Distanciel possible

🕒 : 35 heures

📅 : 5 jours

**MODALITÉS**

**Modalités**

**Modalités :** en présentiel, distanciel ou mixte . Toutes les formations sont en présentiel par défaut mais les salles sont équipées pour faire de l'hybride. – Horaires de 9H à 12H30 et de 14H à 17H30 soit 7H – Intra et Inter entreprise.

**Pédagogie :** essentiellement participative et ludique, centrée sur l'expérience, l'immersion et la mise en pratique. Alternance d'apports théoriques et d'outils pratiques.

**Ressources techniques et pédagogiques :** Support de formation au format PDF ou PPT Ordinateur, vidéoprojecteur, Tableau blanc, Visioconférence : Cisco Webex / Teams / Zoom.

**Pendant la formation :** mises en situation, autodiagnostic, travail individuel ou en sous-groupe sur des cas réels.

**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'evaluation des acquis établie par le formateur en fin de formation.