

— Public	Développeurs, ingénieurs, chefs de projets proches du développement.
— Durée	5 jours - 35 heures
— Pré-requis	Connaissances de base en programmation.
— Objectifs	Maîtriser la syntaxe du langage Python Acquérir les notions essentielles de la programmation objet Connaître et mettre en oeuvre les différents modules Python Concevoir des interfaces graphiques
— Méthodes pédagogiques	Pour bien préparer la formation, le stagiaire effectue une auto-évaluation de positionnement avec un questionnaire complété par un entretien
— Moyens techniques	1 poste de travail complet par personne De nombreux exercices d'application Mise en place d'ateliers pratiques Remise d'un support de cours Passage de certification(s) dans le cadre du CPF Modalité d'évaluation des acquis : Evaluation des besoins et objectifs en pré et post formation Evaluation technique des connaissances en pré et post formation Evaluation générale du stage Remise d'une attestation de stage
— Délai d'accès	L'inscription à cette formation est possible jusqu'à 5 jours ouvrés avant le début de la session
— Accessibilité handicapés	Au centre d'affaires ELITE partenaire d'ACF à 20 m. Guide d'accessibilité à l'accueil.

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.
- 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. 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 multifenêtres.

6. 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.

7. CONCLUSION

- Analyse critique de Python.
- L'évolution du langage.
- Éléments de Webographie et de bibliographie.

NOUS CONTACTER

Siège social

16, ALLÉE FRANÇOIS VILLON
38130 ÉCHIROLLES

Téléphone

04 76 23 20 50 - 06 81 73 19 35

Centre de formation

87, RUE GÉNÉRAL MANGIN
38000 GRENOBLE

E-mail

contact@audit-conseil-formation.com

Suivez-nous sur les réseaux sociaux, rejoignez la communauté !



ACF Audit Conseil Formation



@ACF_Formation



ACFauditconseilformation