

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

€ Tarifs

Prix public : 2960 €

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

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.

Connaitre 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).

📍 Lieux & Horaires

Durée : 35 heures

Délai d'accès :

Jusqu'à 8 jours avant le début de la formation

PROGRAMME

Chapitre 1 : Algorithmie et découverte de la programmation avec Python

- Historique, alternatives
- A propos de Python 2 et Python 3
- Les variables
- Lecture / Ecriture
- Les boucles
- Les Tableaux
- Les fonctions
- Les fonctions personnalisées

Atelier pratique :

Mise en place de notre Notebook et ensemble d'exercices pas à pas pour découvrir le Python.

Calculatrice, tiktok, etc.

Chapitre 2 : 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

📅 Prochaines sessions

Consultez-nous pour les prochaines sessions.

- La fonction range
- L'écriture et la documentation de fonctions
- Les lambda expression
- Les générateurs
- La structuration du code en modules

Atelier pratique :

Toujours dans le Notebook, nous suivons un ensemble de 15 exercices commentés.
Récupération de données sur internet, sauvegarde dans un fichier, etc.

Chapitre 3 : Approche orientée objet et bonnes pratiques

- 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)

Atelier pratique :

Fabrication d'une application de gestion de stock : Clients, commandes, produits.

Chapitre 4 : 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

Atelier pratique :

En reprenant l'application précédente, nous vérifions les bonnes pratiques et mettons en place les nouveautés de Python liés à la POO.

Chapitre 5 : Interface graphique avec la bibliothèque PYQT

- Introduction
- Présentation du binding PyQt
- Installation des librairies Qt et PyQt
- Principes généraux de PyQt
- Notion d'application Qt
- Intégration des technologies HTML et CSS dans Qt
- Interconnexion des différentes barres (barres de menus, d'outils et de status)
- Gestion des menus contextuels
- Gestion des stratégies de placement (Layout)
- Quelques composants : QLineEdit, QComboBox, QSpinner...

Atelier pratique :

En reprenant l'application de gestion des stocks, nous fabriquons une interface pour afficher et modifier les valeurs (produits, commandes et clients).

Chapitre 6 : Utilisation SRDLIB

- 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

Atelier pratique :

Analyse des paramètres fournis par l'utilisateur avec la mise en place d'expressions régulières.

Chapitre 7 : 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

Atelier pratique :

En reprenant l'application « gestion des stocks », nous sauvegardons les résultats dans un fichier CSV.

Chapitre 8 : Qualité du code et les tests unitaires

- Les outils d'analyse statique de code (Pylint, Pychecker).
- Le débogueur de Python (exécution pas à pas et analyse post-mortem).
- Les modules de tests unitaires Python (unittest...).

Atelier pratique :

Analyse de la qualité du code de « gestion des stocks » et mise en place des tests unitaires dans une approche dite « TDD »

MODALITÉS**Modalités**

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

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