Best

Python - Programmation Objet

Utiliser les principales fonctionnalités du langage de programmation Python, pour la conception, le développement et la maintenance d'applications

★★★★★ **4,6/5 (98 avis)**

Présentiel ou en classe à distance

5 jours (35 h)
+ activité à distance
+ certificat optionnel

Prix inter : 3.190,00 € HT
Forfait intra : 9.590,00 € HT

Réf.: LI249

Issu du monde Open Source, Python est un langage réputé facile d'accès, simple et particulièrement portable. Il propose aux développeurs un ensemble d'outils et de fonctionnalités facilitant leurs tâches (piles de modules, pas de nécessité de compilation, débuggeur intégré, Shell de tests, documentation riche...). Mais avant de pouvoir bénéficier de tous ces avantages, les nouveaux développeurs Python devront s'approprier les notions essentielles de programmation objet et apprendre la syntaxe du langage. En suivant cette formation de 5 jours, les participants couvriront tous les sujets leur permettant de concevoir, développer, déployer et maintenir des applications Python.
Cette formation prépare à la certification TOSA Python.

A qui s'adresse cette formation ?

- Développeurs
- Ingénieurs
- Chefs de projets proches du développement

- Disposer de connaissances de base en programmation (idéalement en langage objet)
- Disposez-vous des connaissances nécessaires pour suivre cette formation ? Testez-vous !

Programme

Pendant

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 visibilités
- 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

Après

- Un module e-learning pour revenir sur les points clés de la formation



Les objectifs de la formation

- Connaître la syntaxe du langage Python
- Acquérir les principes de la programmation objet
- Mettre en oeuvre 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



Certificat

Pendant la formation, le formateur évalue la progression pédagogique des participants via des QCM, des mises en situation et des travaux pratiques. Les participants passent un test de positionnement avant et après la formation pour valider leurs compétences acquises.



Les points forts de la formation

- Une formation très pratique : les participants seront amenés à réaliser de nombreux ateliers qui leur permettront d'acquérir une première expérience pratique du développement en Python.
- Les méthodes, bonnes pratiques et astuces pour tirer le meilleur parti de la programmation en Python et développer ainsi des interfaces fiables et performantes.
- 84% des participants à cette formation se sont déclarés satisfaits ou très satisfaits au cours des 12 derniers mois.

Evaluation

Pendant la formation, le formateur évalue la progression pédagogique des participants via des QCM, des mises en situation et des travaux pratiques. Les participants passent un test de positionnement avant et après la formation pour valider leurs compétences acquises.

Dates et villes 2027 - Référence LI249

Dernières places disponibles Session garantie

A distance

du 18 janv. au 22 janv.	du 10 mai au 14 mai	du 20 sept. au 24 sept.
du 8 févr. au 12 févr.	du 14 juin au 18 juin	du 18 oct. au 22 oct.
du 8 mars au 12 mars	du 19 juil. au 23 juil.	du 15 nov. au 19 nov.
du 5 avr. au 9 avr.	du 23 août au 27 août	du 13 déc. au 17 déc.

Tours

du 18 janv. au 22 janv.	du 23 août au 27 août
du 5 avr. au 9 avr.	du 13 déc. au 17 déc.

Toulouse

du 18 janv. au 22 janv.	du 23 août au 27 août
du 5 avr. au 9 avr.	du 13 déc. au 17 déc.

Paris

du 18 janv. au 22 janv.	du 10 mai au 14 mai	du 20 sept. au 24 sept.
du 8 févr. au 12 févr.	du 14 juin au 18 juin	du 18 oct. au 22 oct.
du 8 mars au 12 mars	du 19 juil. au 23 juil.	du 15 nov. au 19 nov.
du 5 avr. au 9 avr.	du 23 août au 27 août	du 13 déc. au 17 déc.

Lille

du 18 janv. au 22 janv.	du 20 sept. au 24 sept.
du 19 juil. au 23 juil.	du 13 déc. au 17 déc.

Nantes

du 8 févr. au 12 févr.	du 23 août au 27 août
du 10 mai au 14 mai	du 15 nov. au 19 nov.

Lyon

du 8 févr. au 12 févr.	du 23 août au 27 août
du 14 juin au 18 juin	du 15 nov. au 19 nov.

Aix-en-Provence

du 8 févr. au 12 févr.	du 20 sept. au 24 sept.
du 10 mai au 14 mai	du 15 nov. au 19 nov.