

Python, programmation Objet

Python est un langage de programmation puissant et facile à apprendre. Il dispose de structures de données de haut niveau et permet une approche simple mais efficace de la programmation orientée objet. Sa syntaxe est élégante, son typage est dynamique et il est interprété ; Python est un langage idéal pour l'écriture de scripts et le développement rapide d'applications dans de nombreux domaines et sur la plupart des plateformes.

Présentiel - Synchrone

Public Visé

Développeurs, ingénieurs, chefs de projets proches du développement.

✓ Pré Requis

Connaissances de base en programmation.

Objectifs pédagogiques et d'évaluation

À l'issue de la formation l'apprenant sera capable de :
 Maîtriser la syntaxe du langage Python
 Acquérir les notions essentielles de la programmation objet
 Connaître et mettre en œuvre les différents modules Python
 Concevoir des interfaces graphiques
 Mettre en œuvre les outils de test et d'évaluation de la qualité d'un programme Python

Méthodes pédagogiques

Beaucoup de tests et d'exemples concrets en lien direct avec les besoins des participants
 Des techniques directement mobilisables dans le cadre de leurs fonctions
 Formation axée sur la mise en pratique
 Méthode participative
 Mise en situation professionnelle
 Alternance de cours et d'exercices dirigés ou en autonomie
 Travaux individuels et corrections collectives
 Evaluation croisée et partage de bonnes pratiques
 Quizz

Parcours pédagogique

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
 Travaux pratiques : Installation et prise en main de l'interpréteur Python

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)
 Travaux pratiques : Modélisation en UML d'un cas d'étude simple

Programmation Objet en Python

Les particularités du modèle Objet de Python

L'écriture de classes et leur instantiation
 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
 Travaux pratiques : Pratique des différents concepts Objet au travers de l'implantation de l'étude de cas

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
 Travaux pratiques : Mise en œuvre de modules Python : expressions régulières, accès à une base de données

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
 Travaux pratiques : Utilisation des outils pylint et pychecker pour la vérification d'un code Python. Mise en œuvre de tests unitaires

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
 Travaux pratiques : Conception d'une interface graphique avec la bibliothèque Tkinter

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
 L'interpréteur Python dans C
 L'utilisation du profileur de code
 Travaux pratiques : Appel de fonctions écrites en C depuis Python. Création de modules C pour Python avec Pyrex

1 ordinateur par stagiaire - Salle de formation claire, climatisée et spacieuse - Tableau blanc - Vidéo projecteur - Support de cours - Logiciel d'assistance des stagiaires à distance

Qualification Intervenant-e-s

Formateur expert en développement et programmation

Méthodes et modalités d'évaluation

Evaluation diagnostique en amont : questionnaire d'auto-positionnement et recueil des attentes & besoins – Tour de table -
Evaluation formative en cours de formation : exercices - Evaluation sommative en fin de formation : exercice/QCM - Bilan individuel des compétences acquises – Attestation de fin de formation -
Questionnaire de satisfaction à chaud - Questionnaire de satisfaction à froid envoyé 15 jours après la formation pour assurer le suivi post formation

Modalités d'accessibilité handicap

Nos formations sont accessibles aux personnes en situation de handicap et aux besoins spécifiques. Une étude personnalisée avec notre référente handicap, permettra d'adapter les moyens pédagogiques, techniques ou organisationnels.



Durée

35.00 Heures

5 Jours

Effectif

De 3 à 8 Personnes