

Python par la pratique

Référence : **DEPYT001**

Durée : **4 jours**

Certification : **Aucune**

CONNAISSANCES PREALABLES

- Connaître un langage de programmation.

PROFIL DES STAGIAIRES

- Développeurs.

OBJECTIFS

- Identifier les usages courants du langage.
- Mettre en pratique le scripting en Python.
- Structurer votre code en fonction, classes et modules.
- Utiliser des modules existants.
- Décrire la programmation réseau avec Python.
- Expérimenter la programmation objet en Python.

CERTIFICATION PREPAREE

- Aucune

METHODES PEDAGOGIQUES

- Mise à disposition d'un poste de travail par stagiaire
- Remise d'une documentation pédagogique numérique pendant le stage
- La formation est constituée d'apports théoriques, d'exercices pratiques, de réflexions et de retours d'expérience
- Le suivi de cette formation donne lieu à la signature d'une feuille d'émargement

FORMATEUR

Consultant-Formateur expert Développement Internet

METHODE D'EVALUATION DES ACQUIS

- Auto-évaluation des acquis par le stagiaire via un questionnaire
- Attestation des compétences acquises envoyée au stagiaire
- Attestation de fin de stage adressée avec la facture

CONTENU DU COURS

Introduction

- Historique
- Environnement de travail
- Champs d'application
- Scripts, fonctions et classes
- Versions architectures et plateformes pour Python
- Editeurs et environnements de développement (IDE)
- Installation et configuration de Python
- Conformité du codage à PEP8

Programme type

- Programme minimal et point d'entrée
- Extension basée sur les packages : import
- Lancement de programmes et passage d'arguments
- Versions compilées de Python : .py, .pyc
- Console Python
- Encoding : utf-8

Types de bases

- Typage faible
- Booléens, numériques
- Les chaînes de caractères
- Conversions de types, casting
- Types agrégés : Tuples, List, Set et dictionnaires

Les instructions de base

- Commentaires sur une ligne # ou plus """
- Notion de blocs et indentations
- Les opérateurs : Affectation
- Les opérateurs : Arithmétique
- Les opérateurs : Logiques
- Les opérateurs : Relationnels
- Saisie et affichage : input, print et formatage
- Structure conditionnelle : la directive "Match"
- Les structures de boucle : While, For, Range()
- Break et continue

- La gestion des exceptions : Groupes d'exceptions
- La gestion des exceptions : Notes d'exceptions
- L'import de modules

Procédures et fonctions

- def(), arguments et valeur de retour
- Variables globales et l'instruction globale
- Arguments par défaut, *args et **kwargs
- Fonctions lambda

Gestion des fichiers

- Gestion de fichiers et répertoires
- Accès séquentiel, aléatoire
- Le module os, os.path, shutil, zlib

Les modules

- Définition de modules et instruction import
- Modules en tant que fichier py et en tant que répertoire
- Résolution des modules
- Module et programme : __main__
- Installation de modules : pip, easy_install

Les classes

- Approche objets
- Classes et instances / objets : self
- Constructeur : __init__
- Données et méthodes membres
- Héritage multiple

Les librairies Python

- Accès aux bases de données relationnelles
- Programmation graphique UI avec Tkinter

Mise au point de programme

- Débogage : exécution pas à pas

Python et le Web

- Scripts Python en tant que Common Gateway

Notre **référent handicap** se tient à votre disposition au 01.71.19.70.30 ou par mail à referent.handicap@edugroupe.com pour recueillir vos éventuels besoins d'aménagements, afin de vous offrir la meilleure expérience possible.