

PYTHON – Opérationnel

Objectifs :

- Connaître les pratiques courantes du langage
- Faire le scripting en Python
- Organiser votre code en fonction, classes et modules
- Manipuler les modules existants
- Tester la programmation objet en Python

Public : Ce programme s'adresse à des développeurs

Prérequis : Avoir des connaissances en langage de programmation

Durée : 4 jours – 28 heures

PROGRAMME

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, Arithmétique, Logiques, Relationnels
- Saisie et affichage : input, print et formatage
- Structure conditionnelle
- Les structures de boucle : While, For, Range()
- Break et continue
- La gestion des 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 bibliothèques Python

- Accès aux bases de données relationnelles
- Programmation graphique UI avec Tkinter
- Exemples de travaux pratiques (à titre indicatif)
- Création d'un programme d'interface utilisateur utilisant la bibliothèque Tkinter
- Ajout dans ce programme d'un accès à une base de données relationnelle puis exploitation de cet accès

Mise au point de programme

- Débogage : exécution pas à pas

Python et le Web

- Scripts Python en tant que Common Gateway Interface CGI
- Introduction au Framework Django

Moyens pédagogiques :

- Alternance d'exposés théoriques et pratiques
- Prédominance de la pratique

Modalités de validation des acquis :

Attestation de stage

Evaluation de fin de formation

Passage d'une certification TOSA