

Python par la pratique

Référence : **DEPYT001**

Durée : **4 jours (28 heures)**

Certification : **TOSA®**

CPF

TOSA

Connaissances préalables

- 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 préparée

Cette formation prépare le TOSA® Python. Le TOSA® est la référence dans le domaine de la certification des compétences informatiques et digitales. Les certifications sont basées sur des méthodologies d'évaluation scientifiques. Cette formation prépare au passage de la certification Programmer et automatiser des tâches avec Python (Tosa) du certificateur ISOGRAD, enregistrée au répertoire spécifique en date du 18/12/2024. Toutes les personnes présentées par EduGroupe ont obtenu 1 certificat précisant leur niveau d'utilisation du logiciel. Pour en savoir plus sur cette certification [cliquez ici](#) et accédez aux informations complètes fournies par France Compétences. Pour connaître tous les détails concernant les prérequis relatifs au passage de l'examen de certification en ligne, nous vous invitons à [cliquer ici](#) pour accéder à la documen

Méthodes pédagogiques

- 6 à 12 personnes maximum par cours, 1 poste de travail par stagiaire
- Remise d'une documentation pédagogique papier ou numérique pendant le stage
- La formation est constituée d'apports théoriques, d'exercices pratiques et de réflexions

Formateur·rice

- Consultant-Formateur expert Développement Internet

Méthodes d'évaluation 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

1. JOUR 1

-

2. 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
- Exemples de travaux pratiques : Acquisition de l'environnement de développement

3. 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
- Console Python
- Encoding : utf-8
- Exemples de travaux pratiques : Ecriture d'un premier programme de base Python

4. Types de bases

- Typage faible
- Booléens, numériques
- Les chaînes de caractères
- Conversions de types, casting
- Types agrégés : Tuples, List, Set et dictionnaires
- Exemples de travaux pratiques : Sur le programme Python démarré, travail sur les types de variable en Python

5. JOUR 2

-

6. Les instructions de base

- Commentaires sur une ligne # ou plus """
- Notion de blocs et indentations
- Les opérateurs : Affectation
- 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
- L'import de modules
- Exemples de travaux pratiques : Suite du travail sur le programme Python

7. Procédures et fonctions

- def(), arguments et valeur de retour
- Variables globales et l'instruction globale
- Arguments par défaut, *args et **kwargs -
- Fonctions lambda
- Exemples de travaux pratiques : Définition et exploitation de nouvelles fonctions avec passage de paramètres et types de retour

8. Gestion des fichiers

- Gestion de fichiers et répertoires
- Accès séquentiel, aléatoire
- Le module os, os.path, shutil, zlib
- Exemples de travaux pratiques : Création d'un nouveau programme qui accède aux fichiers et répertoires, différents types d'accès étudiés

9. JOUR 3

-

10. 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
- Exemples de travaux pratiques : Utilisation de modules systèmes

11. Les classes

- Approche objets
- Classes et instances / objets : self
- Constructeur : __init__
- Données et méthodes membres
- Exemples de travaux pratiques : Dans un nouveau programme, création de classes simples, instanciation de celles-ci pour en récupérer des objets

12. JOUR 4

-

13. Les librairies Python

- Accès aux bases de données relationnelles
- Programmation graphique UI avec Tkinter
- Exemples de travaux pratiques : Création d'un programme d'interface utilisateur utilisant la librairie Tkinter

14. Mise au point de programme

- Débogage : exécution pas à pas

15. Python et le Web

- Scripts Python en tant que Common Gateway

16. Certification TOSA® Python

- Cette formation prépare au passage de la certification TOSA® Python
- La certification TOSA® délivre un niveau de compétences : Niveau Opérationnel (score de 551 à 725), Niveau Avancé (Score de 726 à 875), Niveau Expert (Score de 876 à 1000)

Notre référent handicap se tient à votre disposition au [01.71.19.70.30](tel: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.