

Plan de cours N° : 618

Durée : 5 jours (35h)

PARTICIPANTS / PRE-REQUIS

Développeurs, administrateurs systèmes, ou ingénieurs scientifiques désireux d'apprendre la programmation Python.
La connaissance d'un langage de script ou de programmation quelconque sera fortement appréciée.

OBJECTIFS PEDAGOGIQUES

Installer Python et choisir un environnement de développement. Expérimenter la syntaxe Python et les types de données. Identifier et appliquer la programmation orientée objet. Utiliser des modules et packages Python standard. Gérer les entrées/sorties de fichiers. Manipuler des données XML en Python. Interagir avec des bases de données. Implémenter la programmation parallèle. Créer des interfaces graphiques simples. Appliquer les bonnes pratiques de développement. Intégrer du code C dans des projets Python.

MOYENS PEDAGOGIQUES

Réflexion de groupe et apports théoriques du formateur
Travail d'échange avec les participants sous forme de réunion-discussion
Utilisation de cas concrets issus de l'expérience professionnelle
Validation des acquis par des exercices de synthèse
Alternance entre apports théoriques et exercices pratiques (en moyenne 30 et 70%)
Remise d'un support de cours.
Assistance post-formation d'une durée de 1 an sur le contenu de la formation via notre adresse mail dédiée formateurs@atp-formation.com

MOYENS PERMETTANT LE SUIVI DE L'EXECUTION ET DES RESULTATS

Feuille de présence signée en demi-journée,
Evaluation des acquis tout au long de la formation,
Questionnaire de satisfaction,
Attestation de stage à chaque apprenant,
Positionnement préalable oral ou écrit,
Evaluation formative tout au long de la formation,
Evaluation sommative faite par le formateur ou à l'aide des certifications disponibles.

MOYENS TECHNIQUES EN PRESENTIEL

Accueil des stagiaires dans une salle dédiée à la formation, équipée d'ordinateurs, d'un vidéo projecteur d'un tableau blanc.

MOYENS TECHNIQUES DES CLASSES A DISTANCE

A l'aide d'un logiciel comme Teams, Zoom etc... un micro et éventuellement une caméra pour l'apprenant, suivez une formation en temps réel et entièrement à distance. Lors de la classe en ligne, les apprenants interagissent et communiquent entre eux et avec le formateur.

Les formations en distanciel sont organisées en Inter-Entreprise comme en Intra-Entreprise. L'accès à l'environnement d'apprentissage (support de cours, labs) ainsi qu'aux preuves de suivi et d'assiduité (émargement, évaluation) est assuré.

Les participants recevront une convocation avec lien de connexion

Pour toute question avant et pendant le parcours, une assistance technique et pédagogique est à disposition par mail et par téléphone auprès de notre équipe par téléphone au 04.76.41.14.20 ou par mail à contact@atp-formation.com

ORGANISATION

Les cours ont lieu de 9h00-12h30 13h30-17h00.

PROFIL FORMATEUR

Nos formateurs sont des experts dans leurs domaines d'intervention
Leur expérience de terrain et leurs qualités pédagogiques constituent un gage de qualité.

ACCESSIBILITE

Les personnes atteintes de handicap souhaitant suivre cette formation sont invitées à nous contacter directement, afin d'étudier ensemble les possibilités de suivre la formation.

MISE A JOUR

28/12/2023

Plan de cours N° : 618

Durée : 5 jours (35h)

Installation et prise en main

- Installation et prise en main
- Connaître les différentes versions de Python
- Détails de l'arborescence de fichiers de l'installation Python
- Interpréteur Python
- Interpréteur interactif Idle Python
- Les différents outils de développement
PyCharm, Pydev, Spider

Initiation à la syntaxe

- Utilisation de l'indentation
- Les types de données élémentaires en Python
Manipuler les nombres
Manipuler les chaînes
Manipuler les booléens
- Les types de données complexes
Utilisation de listes/tableaux
Les listes et l'utilisation des "List Comprehensions"
Utilisation de dictionnaires
Utilisation de tuples
Utilisation de sets
- Les opérateurs
+, in, and, or, ...
- Les différents types d'instructions
if/else
for/while
break/continue
assert
- Mise en oeuvre de fonctions
Gestion des paramètres
Mise en oeuvre de fonction lambda
Mise en oeuvre d'un générateur
- Scripts exécutables

Introduction à la programmation orientée objet en Python

- Pourquoi programmer objet
Concept d'encapsulation
Concept d'héritage
- Mise en oeuvre de classes
Définition et instanciation d'une classe
Définition d'attributs de classe
Définition de méthodes de classe
Constructeurs et destructeurs
Mise en oeuvre de propriétés
Redéfinitions d'opérateurs
- Mise en oeuvre de l'héritage
Surcharge VS redéfinition
- Gestion des exceptions
try/except/raise
Hiérarchie de classes d'exceptions
- Mise en oeuvre d'un itérateur

Utilisation de modules et de packages Python

- Concepts fondamentaux
Mise en oeuvre et utilisation de modules
Mise en oeuvre et utilisation de packages
- La bibliothèque de modules standards
Module datetime
Module math
Module re : Regular Expressions
Modules os, sys et shutil : services du système d'exploitation
La documentation des différents modules

Gestion des entrées/sorties

- Le module io
- Module csv : traitement de fichiers structurés par séparateurs
- Utilisation de fichier JSON
- La sérialisation avec le module Pickle
- La sérialisation avec le module Shelve

Manipulation de données XML

- Concepts fondamentaux relatifs à XML
Le W3C
Les principales technologies XML
- Parsing XML : les différentes techniques
Parser un document XML avec le modèle SAX
Parser un document XML avec le modèle DOM
La librairie lxml

Plan de cours N° : 618

Durée : 5 jours (35h)

Manipulation d'une base de données en Python

- La spécification DB-API
- Accès à une base de données
 - Connexion à la base de données
 - Exécution de requêtes SQL
 - Manipulation des résultats
 - Utilisation de procédures stockées
 - Gestion de transactions
- Moteur de persistance relationnel/objet
 - Concepts fondamentaux
 - Introduction à SQLAlchemy

Introduction à la mise en oeuvre d'interface graphique en Python

- Les différents APIs graphiques : Tkinter, PyQt5
- Mise en oeuvre d'une interface graphique simple

Bonnes pratiques de développement

- Documenter vos codes avec les Docstrings
- Mise en oeuvre de tests unitaires : DocTest et unittest
- Débugger du code python
- Utilisation de dépôts de codes sources
 - PyPI : Python Package Index
 - Utilisation de l'outil PIP

