

Programmer en Python

Tutorat inclus

Formation en ligne - 08h

Réf : 4GZ - Prix 2024 : 940CHF HT

Python est un langage qui prend de plus en plus de place dans le paysage informatique. Accompagné par un formateur expert, ce parcours en ligne vous apprendra les bases du langage Python à travers l'organisation et la structuration de programmes, la programmation orientée objet et l'utilisation des fonctionnalités de la plateforme via la bibliothèque standard.

OBJECTIFS PÉDAGOGIQUES

À l'issue de la formation l'apprenant sera en mesure de :

Maîtriser les différents types de base du langage et les fonctions

Maîtriser les notions d'algorithmique utiles en programmation Python

Effectuer de la programmation orientée objet

Créer des programmes Python performants

Connaître la structure d'un programme

PÉDAGOGIE ET PRATIQUES

Une évaluation tout au long de la formation grâce à une pédagogie active mixant pratique, théorie, gamification et partages. Un chef de projet coordonne les échanges et un service technique est dédié au support de l'apprenant. La formation est diffusée au format SCORM et accessible en illimité pendant 1 an.

ACTIVITÉS DIGITALES

Un questionnaire en début de parcours pour faire connaissance avec le tuteur et fixer ses objectifs de formation ; démonstrations, cours enregistrés, exercices corrigés, classe à distance, quiz, activités sociales et partage de bonnes pratiques.

LE PROGRAMME

dernière mise à jour : 06/2023

1) Fondamentaux du langage Python

- Booléens. Types. Exceptions.
- Bloc conditionnel et conditions avancées. Bloc itératif.
- Généralités. Nombres. Syntaxe.
- N-uplets. Introduction aux fonctions.
- Listes. Chaîne de caractères. Dictionnaires.
- Fonctions et paramètres. Ensembles.
- Algorithmique : itération, générateurs. Programmation fonctionnelle.
- Algorithmique : design patterns. Modules essentiels.
- Représenter des données. Lire et écrire un fichier. Générer et lire les fichiers CSV.
- Sérialiser les données (JSON et Pickle). Utiliser facilement des bases de données SQLAlchemy.
- Programmation orientée objet : classes. Objets. Surcharge. Patron de conception (design patterns).
- Bonnes pratiques : programmation asynchrone. Typage statique. Test. Documentation.

Activités digitales : Démonstration. Cours enregistrés. Exercices corrigés. Quiz.

PARTICIPANTS

Développeurs souhaitant apprendre à utiliser le langage Python.

PRÉREQUIS

Connaissances de base en algorithmie.

COMPÉTENCES DU FORMATEUR

Les experts qui ont conçu la formation et qui accompagnent les apprenants dans le cadre d'un tutorat sont des spécialistes des sujets traités. Ils ont été validés par nos équipes pédagogiques tant sur le plan des connaissances métiers que sur celui de la pédagogie, et ce pour chaque cours. Ils ont au minimum cinq à dix années d'expérience dans leur domaine et occupent ou ont occupé des postes à responsabilité en entreprise.

MODALITÉS D'ÉVALUATION

La progression de l'apprenant est évaluée tout au long de sa formation au moyen de QCM, d'exercices pratiques, de tests ou d'échanges pédagogiques. Sa satisfaction est aussi évaluée à l'issue de sa formation grâce à un questionnaire.

MOYENS PÉDAGOGIQUES ET TECHNIQUES

Les moyens pédagogiques et les méthodes d'enseignement utilisés sont principalement : documentation et support de cours, exercices pratiques d'application et corrigés des exercices, études de cas ou présentation de cas réels. ORSYS fournit aux participants un questionnaire d'évaluation du cours qui est ensuite analysé par nos équipes pédagogiques. Une attestation de fin de formation est fournie si l'apprenant a bien suivi la totalité de la formation.

MODALITÉS ET DÉLAIS D'ACCÈS

L'inscription doit être finalisée 24 heures avant le début de la formation.

ACCESSIBILITÉ AUX PERSONNES HANDICAPÉES

Vous avez un besoin spécifique d'accessibilité ? Contactez Mme FOSSE, référente handicap, à l'adresse suivante psh-accueil@orsys.fr pour étudier au mieux votre demande et sa faisabilité.

2) Python pour la data science

- Python et la data science : NumPy (vecteur, données de tableaux, algèbre linéaire, statistique descriptive).
- Visualisation de données : Anaconda et Jupyter, bibliothèque Pandas et lecture de fichiers, données d'un DataFrame.
- Statistiques inférentielles : méthodes melt et apply, extraction d'information, création de variables, Matplotlib.
- Modélisation multivariée : régression linéaire, Support Vector Machine, algorithme de K-Means.

Activités digitales : Démonstration, cours enregistrés, et quiz.

3) Aller plus loin : Responsive Web Design

- Découvrir le Responsive Web Design (syntaxe des Media Queries, requêtes, taille des écrans, attribut viewport).
- Mise en page avec les grilles responsives (Simple Grid, Columnal, Grid System).
- Mise en page fluide.
- Images responsive (script Respond.js, script picturefill.js).
- Frameworks CSS (Kube, KNACSS, Pure).

Activités digitales : Démonstration, cours enregistrés. Quiz. Mise en pratique : créer une page d'accueil responsive.

4) Aller plus loin : développement d'applications web en Python avec Flask.

- Présentation du microframework web Flask.
- Moteur de templates Jinja.
- Création d'une application web.

Activités digitales : Démonstration, cours enregistrés, exercices corrigés et quiz.

5) Aller plus loin : création d'applications web en Python avec Django

- Introduction.
- Modèle de données.
- Vues.
- Outils.

Activités digitales : Démonstration, cours enregistrés, exercices corrigés et quiz.