

Data Science avec Python, certification API Society RS 6763

Cours Pratique de 4 jours - 28h
Réf : PYS - Prix 2025 : 2 590 HT

À l'issue du programme de formation pour data scientist, les participants maîtriseront l'installation et l'utilisation de modules scientifiques dans un environnement virtuel, la collaboration sur des projets de données, la manipulation et transformation de données pour des analyses complexes, ainsi que la création de visualisations interactives et accessibles adaptées aux besoins des utilisateurs.

OBJECTIFS PÉDAGOGIQUES

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

- Découvrir l'écosystème Python scientifique
- Manipuler et analyser des données avec NumPy et Pandas
- Visualiser les données simples et interactives avec Matplotlib, Seaborn, Plotly
- Mener des projets data science et data visualisation

MÉTHODES PÉDAGOGIQUES

Pédagogie active favorisant l'implication personnelle et les échanges entre participants.

TRAVAUX PRATIQUES

Travaux pratiques individuels et en groupe, réflexion collective

CERTIFICATION

L'examen de certification se déroule en ligne, en différé et en français dans le mois qui suit la formation. Il se compose d'une épreuve théorique d'une durée de 20 minutes - 40 questions type QCM vrai / faux et informations à saisir (24 réponses à valider sur 40), et d'une épreuve pratique de programmation (exercice de code) d'une durée de 120 minutes sur un format de 6 exercices (10 critères à valider sur 21).

LE PROGRAMME

dernière mise à jour : 12/2024

1) L'écosystème Python scientifique

- Présentation des packages Python de data science.
- Installation de bibliothèques en environnement virtuel: pip et le module venv, miniconda, mamba, miniforge, WinPython.
- Environnement de développement.
- Utiliser les environnements IPython, Jupyter Notebook, JupyterLab, IDE : l'exemple de Spyder.
- Découvrir l'éditeur de texte : VS Code.

2) La bibliothèque NumPy

- Introduction et création de tableaux.
- Présentation de la librairie NumPy.
- Avantages des tableaux (performance, manipulation des données).

PARTICIPANTS

Statisticiens, data analysts et data scientists

PRÉREQUIS

Connaissances des bases du langage de programmation Python

COMPÉTENCES DU FORMATEUR

Les experts qui animent la formation sont des spécialistes des matières abordées. 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 qu'ils enseignent. 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

Le formateur évalue la progression pédagogique du participant tout au long de la formation au moyen de QCM, mises en situation, travaux pratiques...

Le participant complète également un test de positionnement en amont et en aval pour valider les compétences acquises.

MOYENS PÉDAGOGIQUES ET TECHNIQUES

- Les moyens pédagogiques et les méthodes d'enseignement utilisés sont principalement : aides audiovisuelles, documentation et support de cours, exercices pratiques d'application et corrigés des exercices pour les stages pratiques, études de cas ou présentation de cas réels pour les séminaires de formation.
- À l'issue de chaque stage ou séminaire, ORSYS fournit aux participants un questionnaire d'évaluation du cours qui est ensuite analysé par nos équipes pédagogiques.
- Une feuille d'émargement par demi-journée de présence est fournie en fin de formation ainsi qu'une attestation de fin de formation si le stagiaire a bien assisté à la totalité de la session.

MODALITÉS ET DÉLAIS D'ACCÈS

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

ACCESSIBILITÉ AUX PERSONNES HANDICAPÉES

Pour toute question ou besoin relatif à l'accessibilité, vous pouvez joindre notre équipe PSH par e-mail à l'adresse psh-accueil@orsys.fr.

- Création de tableaux avec `array()`, `zeros()`, `ones()`, `full()`, `arange()`, `linspace()`, `logspace()`.
- Multiplication matricielle avec `np.dot` et l'opérateur `@`.
- Initialisation avec des données aléatoires (module `random`).
- Manipuler des tableaux et opérations.
- Indexation, slicing, et indexation avancée.
- Transposer et changer de dimensions de tableaux (`transpose()`, `reshape()`).
- Concaténer et découper des tableaux (`concatenate()`, `split()`).
- Manipuler les fonctions classiques et mathématiques (`sum()`, `min()`, `max()`, `median()`).
- Comparer et masquer des données avec des masques booléens.
- Gestion des données et visualisation.
- Charger et sauvegarder des tableaux (`loadtxt()`, `save()`, `load()`).
- Utiliser l'option `axis` dans les fonctions.
- Extraire les informations des données.
- Utiliser les pratiques de visualisation : choix des modules et types de graphiques.
- Générer de graphiques interactifs.

3) La bibliothèque Pandas

- Introduction et structures de données.
- Présentation de la bibliothèque Pandas.
- Création de séries avec la classe `Series`.
- Création de tableaux 2D ou `DataFrame` avec la classe `DataFrame`.
- Extraction des indices de ligne et de colonne (attributs `index` et `columns`).
- Lire et exporter des données dans différents formats (`csv`, `xls`).
- Mettre en œuvre les méthodes de base : `head()` et `tail()`.
- Indexation et slicing : implicite, explicite, et utilisation des indexeurs `loc` et `iloc`.
- Sélectionner des données et utiliser des expressions booléennes.
- Manipulation et transformation des données.
- Insérer et modifier des données.
- Renommer des colonnes avec `rename()`.
- Concaténer des données avec `concat()` et fusion/jointure avec `merge()` et `join()`.
- Copier des données : copie superficielle ou profonde (`copy()`).
- Traiter des données manquantes (`isna()`, `isnull()`, `notna()`, `notnull()`, `dropna()`, `fillna()`, `interpolate()`).
- Manipuler des indices : `set_index()`, `sort_index()`.
- Trier les valeurs avec `sort_values()`.
- Transposer des données avec `transpose()`.

4) Analyse et agrégation des données

- Agrégation des données : `sum()`, `cumsum()`, `min()`, `max()`, `count()`, `mean()`, `median()`, `var()`, `std()`, `quantile()`, `describe()`
- Groupement et analyse avec `groupby()`.
- Utiliser des fonctions `aggregate()`, `apply()`, `filter()`, `transform()`.
- Créer de tableaux croisés dynamiques avec `pivot_table()`.
- Segmenter les données avec `qcut()` et `cut()`.
- Calculer des moyennes glissantes avec `rolling()`, `expanding()`, `ewm()`.
- Traiter des données temporelles à travers `to_datetime()`, `to_timedelta()`, `date_range()`, `period_range()`...

5) La bibliothèque Matplotlib

- Introduction et création de graphiques.
- Présentation de la bibliothèque.
- Afficher des graphiques depuis un script Python (`plt.show()`) ou depuis un notebook.
- Utiliser le style `MATLAB` ou le style orienté objet pour afficher les graphiques.
- Modifier le style des graphiques.
- Objets `figure` et axes.
- Tracer des courbes avec `plot()`.
- Types de graphiques et interactions.

- Afficher des nuages de points avec `scatter()`.
- Afficher des barres d'erreurs avec `error_bar()`.
- Remplir la surface entre deux lignes avec `fill_between()`.
- Tracer des histogrammes avec `hist()`.
- Tracer des graphiques en 3D avec `mplot3d`.
- Interagir avec les graphiques dans Jupyter notebook avec le widget `interact`.
- Utiliser `pandas plot` pour créer des tracés rapidement : `plot()`, `bar()`, `barh()`, `hist()`, `box()`, `scatter()`, `pie()`.

6) La bibliothèque Seaborn

- Introduction à Seaborn et fonctionnalités de base.
- Fonctionnement de l'API Seaborn : distinction entre les plots de niveau figure et de niveau axes.
- Relational Plots : utiliser des fonctions pour tracer des relations entre variables.
- Tracer des distributions : utiliser des fonctions pour visualiser des distributions de données.
- Données qualitatives : tracer des données catégorielles.
- Cartes thermiques : utiliser la fonction `heatmap()` pour tracer des cartes thermiques.
- Modèles de régression linéaire : tracer des modèles de régression avec `lmlplot()`.
- Personnalisation des graphiques : changer le rendu de la figure avec les fonctions.

7) La bibliothèque Plotly

- Présentation de la librairie Plotly et de Kaleido : introduction et exploration de Plotly Express.
- Tracer des courbes avec `line()` : modification de la figure avec les options `title`, `width`, `height`, `marker`, `labels`, etc..
- Créer des graphiques en aires avec `area()` : ajout de motifs avec `pattern_shape`.
- Créer des nuages de points avec `scatter()` : utilisation des options `size`, `size_max`, `opacity`, `symbol`, `color_continuous_`
- Graphiques en 3D : utilisation de `scatter_3d()` et `line_3d()`.
- Mettre en forme des diagrammes en barres avec `bar()` et des histogrammes avec `histogram()`.
- Tracer des cartes avec `line_map()`, `scatter_map()`, `line_geo()`, `scatter_geo()`, et `choropleth()`.

LES DATES

CLASSE À DISTANCE

2025 : 09 sept., 16 déc.