Formation Python - Développer une librairie

Objectifs : Maîtriser le développement de librairies Python : conception, structuration, tests automatisés, documentation et déploiement.
Appliquer les bonnes pratiques du développement collaboratif en Python pour produire des packages fiables, maintenables et réutilisables.

Compétences visées : - Sélectionner l'outil adapté pour gérer les environnements virtuels
- Créer une librairie
- Comprendre et utiliser efficacement les syntaxes de la commande import
- Écrire des tests unitaires et d'intégration avec pytest
- Documenter sa librairie et générer une documentation professionnelle
- Packager sa librairie et la déployer
- Travailler en équipe avec un outil de versionnement
- Configurer un workflow de qualité avec des pré-commits
- Automatiser les tests et le déploiement avec GitLab CI/CD (ou autre)

Durée : 4 jour(s) (28 heures)

Public : Développeurs Python, data analysts, data scientists et ingénieurs souhaitant structurer, industrialiser et partager leurs développements Python sous forme de librairies réutilisables.

Pré-requis : Pour suivre ce stage dans de bonnes conditions, il est recommandé d'avoir suivi en amont la formation Python – Bases et introduction aux librairies scientifiques

Méthode pédagogique : Nos formations sont majoritairement pratiques (70%), les concepts théoriques sont illustrés d'exemples et exercices. Les supports sont essentiellement construits avec les notebooks Jupyter/Lab et sont suffisamment détaillés pour être repris seul(e) après la formation.

Tarifs :
- Présentiel : 2400 € HT
- Distanciel : 2200 € HT
(-10% pour 2 inscrits, -20% dès 3 inscrits)

Option(s) :
- Forfait déjeuners : 100 € HT

Nos prochaines sessions

Distance
du 29 juin au 2 juil. 2026
du 19 au 22 octobre 2026

Lyon
du 16 au 19 novembre 2026

Toulouse

du 14 au 17 septembre 2026

Programme :

- Choisir son environnement virtuel

- Panaroma des différents outils: virtualenv, poetry, pipenv, conda, uv, pixi, ...
- Le fichier pyproject.toml et les peps qui le définissent
- Créer un environnement virtuel reproductible

Travaux pratiques : Création d'un environnement virtuel avec l'outil retenu par les participants (uv ou pixi sont les plus pertinents aujourd'hui) Création d'un fichier pyproject.toml minimal

- Créer une librairie

- Un peu de terminologie : modules, packages, librairies et framework
- Créer et importer une librairie
- Le fichier __init__.py et les variables spéciales
- L'arborescence standard
- Comprendre et utiliser efficacement les imports relatifs
- Maîtriser la recherche au travers des chemins d'imports
- Créer un binding Python pour une librairie C

Travaux pratiques : Créer le squelette de sa librairie. Utiliser un fichier pyproject.toml pour y ajouter des dépendances et métadonnées. Créer un binding en C ou Cython selon les besoins des participants

- Documenter sa librairie

- Les docstrings et PEPS
- Les formats de documentation Google, NumPy, ou reStructuredText
- Utiliser Sphinx et Jupyter Book
- Publier avec ReadTheDocs

Travaux pratiques : Générer la documentation avec Sphinx et Jupyter Book

- Tester sa librairie

- Le Test Driven Development
- Les bases avec assert, doctest et unittest
- Pytest: les bases, marqueurs et paramètres, fixtures, plugins, ...
- Les autres outils de tests: mocking, hypothesis, ...
- Logger : concepts et bonnes pratiques

Travaux pratiques : Ecrire des tests unitaires, bien traiter les bugs, mise en place de logs

- Packager sa librairie

- Choisir son outil de packaging: setuptools, flit, Hatchling, pdm, ...
- Les formats eggs et wheel
- Pyproject.toml : les bases, dépendances, fichiers non python

Travaux pratiques: Créer un package et l'installer dans un environnement virtuel. Le déposer sur Pypi

- Intégration continue et qualité logicielle

- Conventions de codage : PEP8, PEP257
- Outils qualité: linter, formatter, contrôles de type, ...
- Rappels sur Git : branches, commits, merges, rebases
- Versionner avec git/GitHub/GitLab
- GitLab (ou GitHub) : gestion de dépôt, issues, merge requests
- Bonnes pratiques : convention Conventional Commits, branches courtes et thématiques
- Pré-commit et formattage automatique avec Ruff
- Intégration continue : les concepts et mise en pratique
- Revue de code

Travaux pratiques: Automatisation d'un pipeline CI/CD permettant : - D'exécuter les tests et mesurer la couverture - Builder le package - Générer et déployer la documentation
Sous réserve de contraintes techniques ou de confidentialité, nous vous proposons de personnaliser la formation en réalisant des exercices directement sur vos données ou projets métiers

Modalités d'évaluation : Un formulaire d'auto-évaluation proposé en amont de la formation nous permettra d'évaluer votre niveau et de recueillir vos attentes. Ce même formulaire soumis en aval de la formation fournira une appréciation de votre progression.
Des exercices pratiques seront proposés à la fin de chaque séquence pédagogique pour l'évaluation des acquis.
En fin de formation, vous serez amené(e) à renseigner un questionnaire d'évaluation à chaud.
Une attestation de formation vous sera adressée à l'issue de la session.
Trois mois après votre formation, vous recevrez par email un formulaire d'évaluation à froid sur l'utilisation des acquis de la formation.

Qualification du formateur : Les formations DATA VALUE sont animées par des intervenants experts, alliant expérience opérationnelle et compétences pédagogiques reconnues. Ils disposent de plus de 10 ans d'expérience professionnelle et assurent une veille constante sur les pratiques, outils et normes de leur domaine afin de garantir des contenus actualisés et adaptés aux besoins des stagiaires.

Délais d'accès et modalités de formation : Les inscriptions peuvent être effectuées jusqu'à 48 heures avant le début de la formation.
Vous pourrez trouver des informations plus détaillées concernant les modalités de nos formations à cette adresse : https://www.datavalue.fr/modalites-acces-formation.

Accessibilité : Vous souhaitez suivre notre formation Formation par ville et êtes en situation de handicap ? Merci de nous contacter afin que nous puissions envisager les adaptations nécessaires et vous garantir de bonnes conditions d'apprentissage

Date de dernière modification : 19 avril 2026