

Python – Perfectionnement

3 jours (21 heures)

Perfectionnement

Présentiel ou classe à distance

à partir de

1905 € HT par personne (TVA en vigueur) Repas inclus en présentiel

Contenu mis à jour le 30/09/2026. Document téléchargé le 02/10/2026.

Objectifs

A l'issue de cette formation, vous serez capable de :

- Décrire les subtilités du langage Python et en tirer parti pour écrire des programmes bien structurés, robustes et efficaces
- Gérer le développement en langage Python, de façon approfondie.

Les compétences visées par cette formation font l'objet d'une évaluation.

À qui s'adresse cette formation ?

Développeurs, administrateurs et architectes.

Prérequis

Avoir suivi le cours PYT "Python – Par la pratique" ou avoir les compétences équivalentes.

Programme

1 Pendant votre formation

Python – Perfectionnement

Jour 1

Langage

- Fonctions : filter, map, reduce (rappel : lambda, *args, **kwargs vus en initiation)
- Modèle de données avancé : module collections (deque, Counter, defaultdict, OrderedDict)
- Littéraux : bytes, r-strings, hexadécimal, octal, binaire
- Mutabilité et immutabilité
- Générateurs : yield, yield from, expressions génératrices, compréhensions imbriquées
- Décorateurs : fonctions et classes
- Introspection : dir, help, type, getattr, setattr
- Modules standards : math, random, itertools, functools

Exemples de travaux pratiques (à titre indicatif)

- Création d'un programme avec exploitation avancée des collections de Python
- Changement des types de passage d'argument

POO avancée

- Héritage multiple
- Métaclasses
- Classes abstraites (abc)
- Descripteurs

Exemples de travaux pratiques (à titre indicatif)

- Implémentation d'une classe abstraite avec abc.ABC et @abstractmethod
- Création d'une métaclasse simple
- Mise en oeuvre d'un descripteur de propriété
- Résolution d'une diamond inheritance avec MRO

Programmation multithread

- Concepts de bases : programme, thread, synchronisation
- Gestion de threads : modules thread, threading
- Threads et la Programmation Orientée Objets
- Gestion des aspects concurrentiels : lock, mutex, sémaphores
- Threads et échanges de données
- Notion de pool de threads

Exemples de travaux pratiques (à titre indicatif)

- Création d'un programme lançant plusieurs threads
- Synchronisation de ces threads pour obtenir un résultat dépendant de traitements parallèles
- Protection des données globales du programme écrit grâce aux mutex et lock
- Ajout d'un sémaphore pour l'accès à des ressources en nombre restreint

- *Echange de données entre threads lancés*
- *Travail sur la notion de pool de threads disponibles*

Jour 2

Programmation réseau avec les sockets

- Rappels sur TCP / IP et concepts de base de l'API socket
- Utilisation du module socket
- Socket en mode connecté : TCP ou stream
- Socket en mode non connecté : UDP ou datagram
- Les sockets et la Programmation Orientée Objets
- Combinaison des sockets et des threads

Exemples de travaux pratiques (à titre indicatif)

- *Création d'un programme serveur puis client échangeant des données via les sockets en TCP puis UDP*
- *Démonstration de l'avantage du multithreading dans ce cas pour les serveurs*

Persistance de données

- Concepts de base : sérialisation / désérialisation
- Différents modèles de persistance : Pickle
- Persistance texte avec JSON

Python et XML

- Concepts de base :
 - DOM (Document Object Model)
 - SAX (Simple API for XML)
 - Parser...
- Gestion de fichiers XML selon SAX et selon DOM
- Requêtage Xpath et transformation avec XSL

Exemples de travaux pratiques (à titre indicatif)

- *Création d'un programme de lecture d'un flux de données de taille importante via SAX*
- *Mise à jour d'une structure via DOM*

Les bases de données

- Concepts de base : SQL, tables, curseur
- Gestion de la base de données SQLite

Exemples de travaux pratiques (à titre indicatif)

- *Création d'un programme qui sérialise un flux JSON*
- *Ecriture d'un programme d'accès à une base de données MySQL*
- *Mise au point de requêtes de lecture, insertion, mise à jour depuis Python*

Jour 3

Intégration Python / C et Python / Java

- Présentation générale
- Python / C et les packages : ctypes, cffi, pybind11
- Natifs C
- Natifs Java

Exemple de travaux pratiques (à titre indicatif)

- Création d'un programme interfaçant avec des API écrites en C et en Java

Mise au point de programme

- Concepts : TDD, AAA (Arrange / Act / Assert), couverture de code
- Module unittest (bibliothèque standard)
- Framework pytest : fixtures, paramétrage
- Mocks et stubs (unittest.mock)
- Mesure de couverture (coverage)

Optimisation et performance

- Complexité algorithmique : $O(1)$, $O(n)$, $O(n^2)$
- Choix de structures et fonctions performantes

Exemple de travaux pratiques (à titre indicatif)

- Création d'une fixture paramétrée
- Mock d'une dépendance externe avec unittest.mock
- Mesure de couverture avec coverage et identification des lignes non testées

Certification (en option, hors inscription via le CPF)

- Prévoir l'achat de la certification en supplément (ne concerne pas les inscriptions via le CPF pour lesquelles la certification est incluse)
- L'examen (en français) s'effectuera ultérieurement en distanciel (e-surveillance) à la date et au créneau de votre choix
- Il est composé de 35 questions, s'effectuera en ligne et durera en moyenne 1h30
- A l'issue de la certification, le candidat se verra attribuer un score (0 à 1000), correspondant à un niveau (Initial / Basique / Opérationnel / Avancé / Expert) :
 - En deçà du score de 551 points, le candidat se verra attribuer une attestation de passage de la certification. Ces niveaux Initial et Basique ne valident cependant pas l'obtention de la certification
 - Les niveaux Opérationnel, Avancé et Expert permettent, quant à eux, au titulaire de la certification de faire valoir ses compétences et d'attester de son opérationnalité selon son profil et l'emploi qu'il exerce

Financement

Plusieurs dispositifs de financement sont accessibles via les OPCO (Opérateurs de Compétences), organismes agréés par le ministère du Travail dont le rôle est d'accompagner, collecter et gérer les contributions des entreprises au titre du financement de la formation professionnelle.

Pour plus d'information, une équipe de gestionnaires SKOLAE Formation spécialisée vous accompagne dans le choix de vos formations et la gestion administrative.

[Nous contacter pour en savoir plus →](#)

Modalités pédagogiques

Formation délivrée en présentiel ou distanciel* (blended-learning, e-learning, classe virtuelle, présentiel à distance).

Le formateur alterne entre méthode** démonstrative, interrogative et active (via des travaux pratiques et/ou des mises en situation).

Variables suivant les formations, les moyens pédagogiques mis en oeuvre sont :

- Ordinateurs Mac ou PC (sauf pour certains cours de l'offre Management), connexion internet fibre, tableau blanc ou paperboard, vidéoprojecteur ou écran tactile interactif (pour le distanciel)
- Environnements de formation installés sur les postes de travail ou en ligne
- Supports de cours et exercices

En cas de formation intra sur site externe à M2I, le client s'assure et s'engage également à avoir toutes les ressources matérielles pédagogiques nécessaires (équipements informatiques...) au bon déroulement de l'action de formation visée conformément aux prérequis indiqués dans le programme de formation communiqué.

* nous consulter pour la faisabilité en distanciel

** ratio variable selon le cours suivi

[Accessibilité aux personnes en situation de handicap](#)

Détails de la session

Du 15 décembre 2026 au 17 décembre 2026 (3 jours / 21 heures)

Strasbourg

Du 15 décembre 2026 au 17 décembre 2026

Strasbourg

- 15 décembre 2026 9h00 – 16h00
- 16 décembre 2026 9h00 – 16h00
- 17 décembre 2026 9h00 – 16h00