



Développeur Python

Python - Par la pratique

4 jours (28h00) | ★★★★★ 4,8/5 | PYT | Code RS ou RNCP : RS6962 | Certification TOSA
Programmer et automatiser des tâches avec Python (non incluse) | Évaluation
qualitative de fin de stage | Formation délivrée en présentiel ou distanciel

Tarif : 2 498 € ht

Formations Informatique › Langages et développement › Développeur Python

Document mis à jour le 06/07/2026

Objectifs de formation

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

- Identifier les usages courants du langage
- Mettre en pratique le scripting en Python
- Structurer votre code en fonction, classes et modules
- Utiliser des modules existants
- Décrire la programmation réseau avec Python
- Expérimenter la programmation objet en Python.

Compétences attestées par la certification

Domaine 1 : Langage et syntaxe

- Maîtriser la syntaxe de base et les structures de contrôle : maîtriser la syntaxe de base de Python, incluant les boucles, les conditionnels et la gestion des erreurs, pour développer des programmes robustes et efficaces
- Définir et utiliser des fonctions : créer des fonctions en Python, comprendre la portée des variables, gérer les arguments et retours, et utiliser les fonctions lambda pour simplifier le code et améliorer sa lisibilité
- Appliquer des concepts avancés de programmation : utiliser les décorateurs, générateurs, compréhensions de listes et techniques asynchrones pour écrire un code Python plus performant et concis

Domaine 2 : Structures de données et objets

- Manipuler des types de données primitifs et composés : utiliser les listes, tuples, dictionnaires et ensembles, ainsi que les opérations courantes sur ces structures pour organiser et manipuler efficacement les données en Python
- Utiliser la programmation orientée objet : créer des classes, utiliser l'héritage et le polymorphisme, et appliquer des principes de conception avancée tels que les classes abstraites et les interfaces pour structurer des programmes orientés objet, facilitant la réutilisation et la maintenance du code
- Gérer et optimiser les données : manipuler les données de manière avancée avec des bibliothèques comme pandas et optimiser les structures de données pour améliorer les performances des programmes Python

Domaine 3 : Modules et packages

- Utilisation et création de modules : importer des modules existants, créer de nouveaux modules et structurer le code en modules réutilisables pour favoriser la modularité et la maintenabilité en Python

- Développer et distribuer des packages : créer et configurer des packages avec setuptools pour assurer leur distribution efficace
- Gérer les environnements et les dépendances : utiliser des environnements virtuels, gérer les dépendances avec pip et automatiser les configurations via Docker pour garantir des environnements de développement cohérents et reproductibles

Domaine 4 : Optimisation de code

- Analyser et profiler le code : utiliser des outils de mesure pour analyser la performance du code, identifier les goulots d'étranglement, et appliquer des méthodologies afin d'optimiser l'efficacité du programme
- Améliorer l'efficacité du code : appliquer des techniques pour réduire la complexité du code, optimiser les boucles et utiliser efficacement les ressources afin d'améliorer la performance globale des programmes
- Tester et déboguer : écrire des tests unitaires et d'intégration, utiliser des frameworks comme 'pytest' et appliquer des stratégies avancées de débogage pour garantir la qualité et la fiabilité du code.

Lien pour visualiser le détail de la certification enregistrée au RS :

<https://www.francecompetences.fr/recherche/rs/6962/>

Modalités, méthodes et moyens 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

Prérequis

Avoir suivi le cours PRG-IN "Algorithmique et programmation structurée" ou avoir les connaissances équivalentes.

Public concerné

Développeurs.

Cette formation :

- Est animée par un consultant-formateur dont les compétences techniques, professionnelles et pédagogiques ont été validées par des diplômes et/ou testées et approuvées par l'éditeur et/ou par M2I Formation
- Bénéficie d'un suivi de son exécution par une feuille de présence élargée par demi-journée par les stagiaires et le formateur.

Programme

Jour 1

Introduction

- Historique
- Champs d'application
- Catégoriser Python: langage interprété à typage fort dynamique
- Environnement de travail
- Versions architectures et plateformes pour Python
- Editeurs et environnements de développement (IDE)
- Installation et configuration de Python
- Conformité du codage à PEP8

Exemples de travaux pratiques (à titre indicatif)

- *Acquisition de l'environnement de développement*
- *Configuration de l'installation*

Programme type

- Programme minimal et point d'entrée
- Lancement de programmes et passage d'arguments
- Versions compilées de Python
 - .py
 - .pyc
- Console Python

Exemples de travaux pratiques (à titre indicatif)

- *Ecriture d'un premier programme de base Python*
- *Echange avec la console*

Types de base

- Les primitifs
- Conversions de types, casting
- Types agrégés : Tuples, List, Set et Dictionnaires

Exemples de travaux pratiques (à titre indicatif)

- *Sur le programme Python démarré, travail sur les types de variable en Python*
- *Ajout de tableaux et travail sur ceux-ci*

Jour 2

Les instructions de base

- Commentaires sur une ligne #
- Notion de blocs et indentations

- Les opérateurs
 - Affectation
 - Arithmétique
 - Logiques
 - Relationnels
- Saisie et affichage : input, print et formatage
- Structure conditionnelle : la directive "Match"
- Les structures de boucle : While, For, Range()
- Break et continue
- La gestion des exceptions
 - Groupes d'exceptions
 - Notes d'exceptions
- L'import de modules

Exemples de travaux pratiques (à titre indicatif)

- Suite du travail sur le programme Python
- Utilisation des structures de contrôle
- Ajout de la gestion des exceptions et compréhension des modules

Procédures et fonctions

- Def, arguments et valeur de retour
- Variables globales et l'instruction globale
- Arguments par défaut, *args et **kwargs -
- Fonctions lambda

Exemples de travaux pratiques (à titre indicatif)

- Définition et exploitation de nouvelles fonctions avec passage de paramètres et types de retour
- Ecriture de fonctions lambda, compréhension de l'avantage d'utiliser celles-ci

Gestion des fichiers

- Gestion de fichiers et répertoires
- Accès séquentiel, aléatoire
- Le module os, os.path, shutil, zlib

Exemples de travaux pratiques (à titre indicatif)

- Création d'un nouveau programme qui accède aux fichiers et répertoires, différents types d'accès étudiés
- Travail sur les modules standards

Jour 3

Les modules

- Définition de modules et des packages
- Instruction import
- Résolution des modules
- Module et programme : `_main_`
- Installation de modules : pip, `easy_install`

Exemples de travaux pratiques (à titre indicatif)

- Utilisation de modules systèmes
- Création de modules regroupant plusieurs fonctions écrites en cours

Les classes

- Approche objets
- Classes et instances / objets : self

- Constructeur : `__init__`
- Données et méthodes membres
- Héritage multiple

Exemples de travaux pratiques (à titre indicatif)

- Dans un nouveau programme, création de classes simples, instanciation de celles-ci pour en récupérer des objets
- Ajout de méthodes et de propriétés
- Ajout de l'héritage et travail sur ce modèle

Jour 4

Les bibliothèques Python

- Accès aux bases de données relationnelles
- Programmation graphique UI avec Tkinter

Exemples de travaux pratiques (à titre indicatif)

- Création d'un programme d'interface utilisateur utilisant la bibliothèque Tkinter
- Ajout dans ce programme d'un accès à une base de données relationnelle puis exploitation de cet accès

Mise au point de programme

- Débogage : exécution pas à pas

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) sera passé soit à la fin de la formation, soit ultérieurement dans nos centres de formation
- 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

Le contenu de ce programme peut faire l'objet d'adaptation selon les niveaux, prérequis et besoins des apprenants.

Modalités d'évaluation des acquis

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation et/ou une certification (proposée en option)

Accessibilité de la formation

pagebreakavoidchecked="true";

Le groupe M2I s'engage pour faciliter l'accessibilité de ses formations. Toutes nos formations sont accessibles aux personnes en situation de handicap : les détails de l'accueil des personnes sont consultables sur la page Politique Handicap.

Modalités et délais d'accès à la formation

pagebreakavoidchecked="true";

Les formations M2I sont disponibles selon les modalités proposées sur la page programme. Les inscriptions sont possibles jusqu'à 48 heures ouvrées avant le début de la formation. Dans le cas d'une formation financée par le CPF, ce délai est porté à 12 jours ouvrés.