

Formation Python Initiation

Formation éligible au CPF, contactez-nous au 22 519 09 66

■ Durée :	3 jours (21 heures)
■ Tarifs inter-entreprise :	1 995,00 CHF HT (standard) 1 596,00 CHF HT (remisé)
■ Public :	Développeurs
■ Pré-requis :	Aucun
■ Objectifs :	- Devenir syntaxiquement compétent en Python afin de répliquer une expérience préalable dans un autre langage - Être autonome pour réaliser des projets de taille petite à moyenne (scripts et utilitaires) en Python - Connaître des bonnes pratiques minimales dans l'écriture de code en Python - Maîtriser des outils de base utilisés de façon récurrente chez les développeurs Python
■ Modalités pédagogiques, techniques et d'encadrement :	• Formation synchrone en présentiel et distanciel. • Méthodologie basée sur l'Active Learning : 75 % de pratique minimum. • Un PC par participant en présentiel, possibilité de mettre à disposition en bureau à distance un PC et l'environnement adéquat. • Un formateur expert.
■ Modalités d'évaluation :	• Définition des besoins et attentes des apprenants en amont de la formation. • Auto-positionnement à l'entrée et la sortie de la formation. • Suivi continu par les formateurs durant les ateliers pratiques. • Évaluation à chaud de l'adéquation au besoin professionnel des apprenants le dernier jour de formation.
■ Sanction :	Attestation de fin de formation mentionnant le résultat des acquis
■ Référence :	PYT252-F

■ Note de satisfaction des participants:	4,79 / 5
■ Contacts :	commercial@dawan.fr - 09 72 37 73 73
■ Modalités d'accès :	Possibilité de faire un devis en ligne (www.dawan.fr , moncompteformation.gouv.fr , maformation.fr , etc.) ou en appelant au standard.
■ Délais d'accès :	Variable selon le type de financement.
■ Accessibilité :	Si vous êtes en situation de handicap, nous sommes en mesure de vous accueillir, n'hésitez pas à nous contacter à referenthandicap@dawan.fr , nous étudierons ensemble vos besoins

Découverte du langage

Découvrir les origines de Python

Installer et utiliser Python sous Windows ou Linux

Environnements de développement pour Python

Atelier : Exécuter Python et créer un projet

Syntaxe de base de Python

Instructions : Déclarations et expressions

Commencer avec la fonction print

Commentaires

Variables et expressions

Types de données de base : nombres, texte, booléens, octets, valeur None

Structures de contrôle : conditions et boucles (for, while)

Outils syntaxiques supplémentaires : unpacking, opérateur walrus

Objectifs : Connaître les outils de base fournis par Python et résoudre des problèmes en utilisant les fonctionnalités appropriées

Types de données avancés

Collections : manipuler des listes, tuples, ensembles et dictionnaires

Travailler sur des séquences : opération de slicing

Exploiter des chaînes de caractères (méthodes disponibles)

Comprendre les types de chaînes de caractères : formatables, brutes, octets

Objectifs : Être capable d'introduire l'usage de collections lorsqu'un problème le nécessite

Découvrir les fonctions

Découvrir l'utilité des fonctions
Découvrir la syntaxe
Référence vs appel de fonction
Valeurs de retour d'une fonction
Passer des arguments positionnels aux fonctions
Utiliser des arguments avec valeurs par défaut

Atelier : Créer et réutiliser des fonctions de calcul

Objectifs : Savoir factoriser du code en fonctions pour ne pas se répéter

Le droit aux erreurs

Découvrir le concept d'exceptions
Intercepter et gérer des exceptions
Bonnes et mauvaises pratiques pour la gestion des exceptions

Réutiliser du code

Notion de Modules et packages
Découvrir la bibliothèque standard Python et effectuer des imports
Réutiliser votre propre code et écrire des bibliothèques de fonctions
Extra : manipuler des environnements virtuels et utiliser l'outil pip

Atelier : Faire du calcul avec la bibliothèque standard

Atelier : Utiliser une bibliothèque tierce open-source (ex. requests)

Objectifs : Être à l'aise avec le système d'import en Python

Découvrir la programmation orientée objet

Intérêts de la programmation objet
Découvrir la syntaxe de l'objet en Python
Découvrir le modèle de données de Python (article de documentation)
Concepts d'attributs et de méthodes
Différence entre une classe et ses objets
Instancier des objets en passant des arguments
Découvrir l'héritage
Comprendre la notion de polymorphisme en Python
Extra : Utiliser les fonctionnalités simples d'introspection (getattr, dir, isinstance, etc.)

Bonus : Conception de projets en langages objets (Diagramme de classes UML)

Atelier : Représenter une structure (bibliothèque, librairie) avec des classes

Objectifs : Connaître en détail le système d'objets et le modèle de données de Python, et savoir quand décrire une classe est intéressant