

Formation Python pour les développeurs : La syntaxe

Formation éligible au CPF, contactez-nous au 09 72 37 73 73

■ Durée :	3 jours (21 heures)
■ Tarifs inter-entreprise :	1 975,00 € HT (standard - hors certification) 1 580,00 € HT (remisé - hors certification)
■ Public :	Développeurs dans des langages concurrents (PHP, C/C++, Java, ...)
■ Pré-requis :	Avoir la maîtrise d'un langage concurrent type PHP, Java, C/C++
■ 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
■ Certification :	La formation prépare à la certification Python détenue par Isograd, enregistrée le 18/12/2024 sous le numéro 6962 au Répertoire Spécifique de France Compétences . Taux de réussite : pas de données disponibles, calculé le 01/07/2025
■ Modalités pédagogiques, techniques et d'encadrement :	<ul style="list-style-type: none">• 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.

<ul style="list-style-type: none"> ■ Modalités d'évaluation : 	<ul style="list-style-type: none"> • 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.
<ul style="list-style-type: none"> ■ Sanction : 	Attestation de fin de formation mentionnant le résultat des acquis
<ul style="list-style-type: none"> ■ Référence : 	PYT102406-F
<ul style="list-style-type: none"> ■ Note de satisfaction des participants: 	4,79 / 5
<ul style="list-style-type: none"> ■ Contacts : 	commercial@dawan.fr - 09 72 37 73 73
<ul style="list-style-type: none"> ■ Modalités d'accès : 	Possibilité de faire un devis en ligne (www.dawan.fr, moncompteformation.gouv.fr, maformation.fr, etc.) ou en appelant au standard.
<ul style="list-style-type: none"> ■ Délais d'accès : 	11 jours ouvrés en cas de financement CPF.
<ul style="list-style-type: none"> ■ 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

Correspondance de motif structurel (match...case)

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

Comprendre les collections immuables (tuples, frozenset, etc.)

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

Comprendre les générateurs et le mot clé yield

Passer des arguments positionnels aux fonctions

Utiliser des arguments avec valeurs par défaut

Comprendre et utiliser l'argument liste (indirection)

Comprendre et utiliser l'argument double étoile (indirection dictionnaire)

Comprendre les arguments étoile et slash

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

Tour des fonctions natives de la bibliothèque standard

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 l'héritage multiple, appréhender le MRO

Comprendre la notion de polymorphisme en Python

Aborder le concept d'encapsulation en Python

Comprendre la différence entre attributs d'objet et de classes

Comprendre, déclarer et utiliser des méthodes statiques

Comprendre, déclarer et utiliser des méthodes de classes

Comprendre, déclarer et utiliser des propriétés

Déclarer et utiliser des métaclasse

Comprendre les interfaces et utiliser les classes abstraites de base

Extra : Utiliser les fonctionnalités simples d'introspection (getattr, dir, isinstance, etc.)

Bonus : Conception de projets en langages objet (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