



EN BREF

Référence : PYCPF

Durée : 5 jours

Niveau : débutant

Lieu : Paris, Orléans, sur site ou à distance

Dates : voir site internet ou nous contacter

Tarif (inter-entreprises) : 2860 € HT

Mise à jour : 01/02/2024

NOS FORMATEUR-RICE-S

✓ Expert·e·s dans leur domaine, ils et elles ont l'expérience du développement logiciel et de la réalisation de projets informatiques.

✓ Ils et elles ont dispensé, avec succès, de nombreuses heures de formation vis-à-vis d'un public exigeant d'enseignant·e·s, de technicien·ne·s, d'ingénieur·e·s et de scientifiques.

INSCRIPTION / RENSEIGNEMENTS



02 38 21 30 57

(du lundi au vendredi)



contact@numgrade.com



UN VRAI DÉPART EN PYTHON

Ce stage intensif vous amènera à maîtriser progressivement les fondamentaux et la syntaxe du langage Python ainsi que la programmation orientée objet avec ce langage. Vous serez ainsi capable de développer vos premiers programmes et packages !

L'ENVIRONNEMENT

Python est un langage clair, intuitif, massivement répandu, et devenu incontournable dans le traitement d'une multitude de tâches informatiques.

LA FORMATION

Cette formation permet d'appréhender les bases du langage Python (types simples et avancés, syntaxe, fonctions, modules, paquets, manipulation de fichiers, gestion des exceptions), de disposer d'une vue d'ensemble du langage, y compris de la partie orientée objet, et de faire ses premiers pas en programmation.

Cette formation vous prépare à la **certification** API Society référencée auprès de France Compétences dans le répertoire spécifique sous le numéro [RS6301](#). Toutes les modalités pratiques pour le passage de la certification vous seront communiquées.

FINANCEMENT

La formation est finançable avec votre CPF. Le tarif inclus la formation et le passage de la certification.

OBJECTIFS PÉDAGOGIQUES

À l'issue de la formation, le ou la stagiaire sera capable de :

- Créer et paramétrer son environnement de travail Python
- Posséder les bases du langage Python
- Maîtriser la programmation orientée objet avec Python
- Manipuler des outils de la librairie standard
- Développer ses propres programmes Python

PUBLIC CONCERNÉ

Technicien·ne·s, ingénieur·e·s, développeur·se·s, scientifiques, enseignant·e·s ou tout public souhaitant découvrir le langage Python.

PRÉREQUIS

Connaître les bases de l'algorithmie et un langage de programmation, ou avoir suivi la formation « *Initiation à la programmation avec Python* ». Un questionnaire de pré-formation est proposé pour valider l'adéquation entre le besoin et la formation.





NOS MOYENS PÉDAGOGIQUES

- ✓ Un kit de cours complet en couleur
- ✓ De nombreux exercices pratiques
- ✓ Des programmes régulièrement révisés
- ✓ Un rythme adaptable à chacun·e
- ✓ Un suivi pour toutes vos questions post-formation

INSCRIPTION / RENSEIGNEMENTS



02 38 21 30 57

(du lundi au vendredi)



contact@numgrade.fr

ÉVALUATION DE LA FORMATION

À l'issue de chaque demi-journée de formation, une **feuille d'émargement** est signée, à la fois par le·la stagiaire et le·la formateur·rice.

Des **QCMs** en début et fin de formation permettent de vérifier l'**acquisition des compétences** durant la formation.

Un **certificat de formation**, ainsi qu'une **attestation de compétences** sont remis aux stagiaires en fin de formation, leur permettant de **faire valoir l'acquisition de leurs nouvelles connaissances**.

PROGRAMME DE LA FORMATION

Introduction

- Historique
- Python 2 vs Python 3
- Installer Python
- Installer des bibliothèques
- Caractéristiques du langage (multi-paradigme, typage dynamique fort, syntaxe claire)
- Principe de fonctionnement de l'interpréteur (bytecode)
- Interpréteur officiel CPython et autres interpréteurs
- Interpréteur interactif
- Environnement de développement

Premiers pas

- L'arithmétique de base et opérateurs //, % et **
- Les variables
- Fonctions intégrées élémentaires : print(), type()
- Les types de données simples : int, float, complex, bool, str
- Notations binaire, octale et hexadécimale, fonctions hex(), oct(), bin()
- Les opérateurs >>, <<, |, &
- Ressources documentaires : python.org, help()
- Commentaire
- Mots-clés réservés
- Conventions de nommage

Logique de contrôle de flux

- Les structures conditionnelles : if, else, elif
- Principe de l'indentation pour délimiter les blocs d'instruction
- Les opérateurs de comparaison
- Les mots clés and, or, not et in
- Opérateur ternaire
- La boucle while
- La boucle for
- Instructions break et continue
- Bloc else sur structure répétitive
- Fonctions intégrées élémentaires : input(), len(), range()





NOS PRESTATIONS

- ✓ Formations en INTER ou INTRA
- ✓ Développement logiciel
- ✓ Conseil et assistance technique

INSCRIPTION / RENSEIGNEMENTS



02 38 21 30 57
(du lundi au vendredi)



contact@numgrade.fr

FORMATIONS CONNEXES

- ✓ [Python Interfaces Graphiques](#)
- ✓ [Python Perfectionnement](#)
- ✓ [Python Scientifique](#)

NOS CLIENT·E·S TÉMOIGNENT

« Très bonne formation expliquée de manière pédagogique »

« Très bon formateur, très pédagogue et sympa ainsi que compétent »

« I have got what I expected from the course. It was clear, interesting and with normal speed of explanation »

« La formation était très bien organisée et le fait qu'elle soit à distance n'était d'aucune façon limitante.

Le cours a complètement répondu à mes attentes, le formateur, à l'écoute et bienveillant, maîtrisait parfaitement le sujet »

Les fonctions

- Création d'une fonction
- La docstring
- Retourner des valeurs avec l'instruction return
- Les fonctions génériques (duck typing), les valeurs par défaut et le passage par étiquette
- Les fonctions lambda
- L'objet None
- Passer un nombre d'arguments arbitraire (*args, **kwargs)
- Les fonctions eval(), exec()
- Les modules et la méthode import
- Les fonctions help() et dir()
- Le bloc if `__name__ == "__main__"`
- Créer et importer un paquet (`__init__.py`)
- Les exceptions : principe de fonctionnement et les Instructions `try ... except ... else ... finally`
- Les notions de générateur et l'instruction yield

Types avancés

- Données non-modifiables vs données modifiables
- Utilité des types non-modifiables (optimisation mémoire)
- Reconnaître un objet non-modifiable avec la fonction hash()
- Principe des séquences ordonnées (str, tuple et list) et collections (dict, set)
- La chaîne de caractères unicode (str), définition avec simples et doubles guillemets, chaînes multilignes avec triples, simples ou doubles guillemets
- Indixage positif et négatif, tranche de valeurs (slice), opérateurs + et *
- Méthodes de chaîne de caractères : split(), replace(), lower(), upper(), strip(), join()
- Les chaînes de caractères formatées (%s, %d, %f), la méthode format() et la f-string
- Les fonctions str() et repr()
- Les tableaux d'octets (bytes)





MOYENS PÉDAGOGIQUES

- Un kit de cours en couleur, reprenant l'intégralité du contenu présenté
- Des quiz et de nombreux exercices pratiques en cohérence avec les objectifs de la formation
- Des programmes révisés régulièrement afin de répondre à l'état de l'art
- Un rythme adaptable à tous
- Un suivi pour toutes questions post-formation

- **En visio-conférence :**

Des outils de partage de code et des commentaires en temps réels sur les exercices pratiques pour aider les stagiaires à progresser.

- **En présentiel :**

Des salles agréables et fonctionnelles.

- **En entreprise :**

Nous nous déplaçons partout en France et à l'étranger pour vous former sur vos propres sites.

- Les tuples, les opérateurs + et *, les méthodes count() et index()
- Les listes, les opérateurs + et *, et les méthodes append(), insert(), sort(), extend() et pop()
- La fonction enumerate()
- Les listes en intension (comprehension list)
- Les fonctions map, filter et reduce
- Les fonctions sorted(), reversed(), range() et le mot clé del
- Les dictionnaires et les méthodes keys(), values(), items(), update(), get()
- Les sets (set) et les opérateurs - | & et ^

Portée des variables, objets mutables et immutables, références, copie superficielle et profonde

- Espaces de noms local, global, prédéfini (__builtins__)
- Références et fonction id()
- Copie superficielle via la méthode copy() ou l'indilage [:]
- Copie en profondeur avec la fonction deepcopy() du module copy
- Différences entre les opérateurs == et is

Manipuler des fichiers

- Interagir avec le système de fichiers (os.path, glob, pathlib et shutil)
- La fonction open(), la méthode close() et l'attribut closed
- Les méthodes read(), readline() et readlines()
- Itérer sur un fichier
- Utiliser l'instruction with avec les fichiers
- Écrire avec les méthodes write() et writelines()
- Utiliser les méthodes tell() et seek()
- Sérialisation avec les modules pickle et shelve
- Travailler sur d'autres formats de fichiers avec les modules struct, csv, json et xml

INSCRIPTION / RENSEIGNEMENTS



02 38 21 30 57
(du lundi au vendredi)



contact@numgrade.fr





QUELQUES CHIFFRES

- ✓ **4.7/5** de moyenne sur l'ensemble des questionnaires de **satisfaction**
- ✓ **+ 850** personnes formées
- ✓ **+ 19000** heures de formations stagiaires
- ✓ **98.85 %** des stagiaires recommandent nos formations !

INSCRIPTION / RENSEIGNEMENTS



02 38 21 30 57
(du lundi au vendredi)



contact@numgrade.fr

EXIGEZ LA DIFFÉRENCE NUMGRADE

- ✓ Des consultant·e·s expert·e·s et passionné·e·s
- ✓ Concentré·e·s sur la réussite de leurs clients et clientes
- ✓ En quête permanente d'excellence technique

SITUATION DE HANDICAP

NUMGRADE s'engage à accueillir les personnes en situation de handicap. N'hésitez pas à nous contacter pour que le suivi de la formation se déroule dans les meilleures conditions possibles.

La programmation orientée objet en Python

- Concepts fondamentaux de la POO : séparation du code, encapsulation, héritage
- Notions de classe d'objet, d'objet (instance), d'attribut et de méthode
- Définition d'une classe d'objet
- La méthode d'initialisation (`__init__`) et le mot-clé `self`
- Les attributs d'instance et les attributs de classe
- Les méthodes d'instance
- Les méthodes de classe
- Les méthodes statiques
- Instancier un objet et la fonction `isinstance()`
- Explorer une classe
- Espaces de noms global, de l'objet, de la classe
- Définir ses propres exceptions
- Propriété (fonction `property`), accesseur et mutateur
- Les méthodes spéciales : surcharge d'affichage (`__str__` et `__repr__`), surcharge d'opérateurs (`__add__`, `__radd__`, `__sub__`, `__rsub__`, `__mul__`, `__rmul__`, `__truediv__`, `__rtruediv__`, `__eq__`, `__ne__`, `__lt__`, `__le__`, `__gt__`, `__ge__`), surcharge de conteneurs (`__len__`, `__getitem__`, `__setitem__`, `__contains__`)
- L'héritage simple et les fonctions `issubclass()` et `super()`
- L'héritage multiple et la méthode `mro()`
- La composition

Modules de la bibliothèque standard

- Interagir avec l'interpréteur : module `sys`
- Interagir avec le système d'exploitation : module `os`
- Expressions rationnelles : module `re`
- Créer des tests unitaires : module `unittest`
- Tour d'horizon d'autres modules intéressants de la bibliothèque standard : `datetime`, `math`, `timeit`, `urllib`, `collections`, `sqlite3`

Mini-projet

- Réaliser une application Python complète de A à Z

