

# Python - « Développer en langage Python orienté objet » préparant à la certification API Society

4 jours (28h00) | -/5 | Code RS : RS6301 | Certification API Society (non incluse) | Évaluation qualitative de fin de stage | Formation délivrée en présentiel ou distanciel <sup>(1)</sup>

Formations Informatique > Langages et développement > Développeur Python

Document mis à jour le 19/03/2024

## Objectifs pédagogiques

À l'issue de la formation, le participant sera en mesure de : Connaître la syntaxe du langage Python

- Connaître et manipuler les types de données fondamentaux
- Utiliser et définir des fonctions, des modules, des paquets, des exceptions, manipuler des fichiers
- Connaître la théorie de la Programmation orientée objet et sa mise en pratique en Python
- Connaître et utiliser les modules incontournables de la bibliothèque standard

## Compétences attestées par la certification

- La certification est composée de deux épreuves :
  - une épreuve théorique durant laquelle le candidat doit répondre à un questionnaire de type QCM
  - une épreuve pratique durant laquelle le candidat doit écrire du code Python au travers d'exercices de programmation
- Ce score déterminera un niveau (opérationnel, avancé ou expert) selon lequel vos compétences vous seront attribuées

Retrouvez le détail des compétences attestées sur la fiche RS :

<https://www.francecompetences.fr/recherche/rs/6301/>

## Niveau requis

Connaître un langage de programmation.

**Public concerné** : salariés, consultants, personnes en reconversion professionnelle, demandeur d'emploi. Métiers : Développeurs, Administrateurs système et réseau, Technicien support applicatif.

## Financement : AIF par Pôle emploi, CPF, par votre employeur avec OPCO

Contactez-nous par courriel en indiquant votre identifiant Pôle emploi, nous réaliserons les démarches pour vous auprès de votre conseiller, ainsi vous trouverez toutes les informations sur votre compte Pôle emploi.

(1) Modalités et moyens pédagogiques :

Formation délivrée en présentiel ou distanciel \* (e-learning, classe virtuelle, présentiel à distance). Le formateur alterne entre méthodes \*\* démonstrative, interrogative et active (via des travaux pratiques et/ou des mises en situation). La validation des acquis peut se faire via des études de cas, des quiz et/ou une certification. Les moyens pédagogiques mis en œuvre (variables suivant les formations) sont : ordinateurs Mac ou PC (sauf pour les cours de l'offre Management), connexion internet, 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 au sein de AWC, 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.

## 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 AWC
- Bénéficie d'un suivi de son exécution par une feuille de présence émargée par demi-journée par les stagiaires et le formateur.

# Programme

## Introduction

- Historique (auteur, date de la première version)
- Versions de Python (branches 2 et 3)
- Caractéristiques du langage (multi–paradigme, typage dynamique fort, syntaxe claire)
- Panorama de la bibliothèque standard Modules d'extension et commande pip
- Principe de fonctionnement de l'interpréteur (bytecode PYC)
- Interpréteur officiel CPython et autres interpréteurs (micropython, brython, pypy, numba)
- Ressources (site Internet python.org, accès aux documentations)
- Fonction help() et chaînes documentaires
- Principe de l'indentation pour délimiter les blocs d'instruction
- Commentaire Mots–clés réservés
- Conventions de nommage
- Interpréteur interactif
- Programme autonome
- Fonctions intégrées élémentaires : print(), type(), input(), len()

## Types de données non-modifiables

- Utilité des types non–modifiables (optimisation mémoire), fonctions id() et hash(), opérateur is
- Principe des séquences ordonnées (str, tuple et list) et collections (dict, set)
- Booléen (bool), objets True et False
- Nombre (int, float, complex), constructeurs, opérateurs >>, <<, |, &, ^, //, % et \*\*
- Notations exponentielle, binaire, octale et hexadécimale, fonctions hex(), oct(), bin(), chr(), ord()
- Chaîne de caractères unicode (str), définition avec simple et double guillemets, chaînes multilignes avec triple simple ou double guillemets, mode raw, constructeur
- Indilage positif et négatif, tranche de valeurs (slice), opérateurs +, \* et in, itération
- Méthodes incontournables de str : split(), replace(), lower(), upper(), strip(), join()
- Chaîne de caractères formatée (%s, %d, %f), méthode format() et f–string
- Principe du unpacking de variables
- Tableau d'octets (bytes), constructeur
- Tuple (tuple), constructeur, indilage, itération, opérateurs +, \* et in, méthodes count() et index()
- Objet None et fonction repr()

## Types de données modifiables

- Listes (list), constructeur, indilage, itération, opérateurs +, \* et in, méthodes append(), insert(), fonction del(), méthodes sort(), reverse(), remove(), extend(), pop(), clear()
- Listes : manipulation de références, copie superficielle via la méthode copy() ou l'indilage [:], copie en profondeur avec la fonction deepcopy() du module copy
- Fonction sorted()
- Principe de fonctionnement des objets itérables
- Fonctions reversed() et range()
- Dictionnaires (dict), constructeur, indilage, opérateur in, fonction del(), méthodes keys(), values(), items(), update(), get()
- Dictionnaires : manipulation de références, copie superficielle via la méthode copy(), copie en profondeur avec la fonction deepcopy() du module copy
- Set (set), constructeur, opérateurs – | & et ^

## Structures conditionnelles et répétitives

- Structure conditionnelle if ... elif ... else
- Opérateurs ternaire et morse
- Structure répétitive while
- Structure répétitive for
- Instructions break et continue
- Fonction enumerate()
- Bloc else sur structure répétitive
- Liste en intension (comprehension list), dictionnaire en intension (comprehension dict)

## Fonctions, modules et paquets

- Définition et appel d'une fonction
- Espace de noms local, global, pré-défini (`__builtins__`) et fonction `dir()`
- Retourner des valeurs, instruction `return`
- Fonctions génériques (duck typing)
- Valeurs par défaut
- Passage par étiquette
- Nombre d'arguments arbitraire (`*args`, `**kwargs`)
- Fonctions anonymes (`lambda`)
- Fonctions `eval()`, `exec()`, `map()` et `filter()`
- Importation de modules
- Création d'un module
- Bloc `if __name__ == "__main__"`
- Importation de paquet
- Création d'un paquet (`__init__.py`)
- Instruction `yield`

## Manipulation des fichiers

- Fonction `open()` et méthode `close()`
- Méthodes `readline()` et `readlines()`
- Objet itérable
- Instruction `with` avec les fichiers
- Méthodes `read()` et `write()`
- Méthodes `tell()` et `seek()`
- Méthode `writelines()`
- Modules complémentaires : `struct`, `csv`, `json`, `xml`
- Sérialisation avec le module `pickle`
- Sérialisation avec le module `shelve`

## Programmation Orientée Objet

- 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
- Instanciation d'objets, fonction `isinstance()`
- Constructeur (`__init__`)
- Attributs et méthodes
- Le paramètre `self`
- Surcharge d'affichage (`__str__`)
- Surcharge d'opérateurs (`__eq__`, `__add__`)
- Propriété (fonction spéciale `property`), accesseur et mutateur
- Espaces de noms global, de l'objet, de la classe
- Variable de classe
- Constructeur à nombre d'arguments arbitraire (`*args`, `**kwargs`)
- Agrégation / Composition
- Héritage de classe (généralisation), fonctions `issubclass()`, `super()` et méthode `mro()`

### **Exceptions**

- Principe de fonctionnement
- Exceptions pré-définies et arbre d'héritage
- Instructions try ... except ... else ... finally
- Propagation des exceptions
- Déclenchement d'exceptions
- Définition d'une exception

### **Modules de la bibliothèque standard**

- Interaction avec l'interpréteur : module sys
- Interaction avec le système d'exploitation : modules os et pathlib
- Interaction avec le système de fichiers : module os.path
- Expressions rationnelles : module re
- Tests unitaires : instruction assert, module unittest
- Tour d'horizon d'autres modules intéressants de la bibliothèque standard : datetime, math, timeit, urllib, collections, csv, json, unittest, sqlite3

### **Certification (en option) Prévoir l'achat de la certification en supplément**

- L'examen (en français) sera passé soit à la fin de la formation, soit ultérieurement dans notre centre de formation ou à distance.
- Épreuve théorique s'effectuera en ligne de 20 min – format : 40 Questions – validation : 24 réponses justes.
- Épreuve pratique s'effectuera en ligne de 120 min – format 5 exercices – 11 critères validés sur 16
- Obtention de la certification (le score obtenu attestera d'un niveau de compétence).
- 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 (en option TP avec machine virtuelle en e-formation)
- Et, en fin de formation, par un questionnaire d'auto-évaluation et/ou le passage d'une certification par AWC à distance via un logiciel de certification à télécharger sur son poste ou en présentiel.

### **Moyens de suivi d'exécution et appréciation des résultats**

- Feuille de présence, émarginée par demi-journée par chaque stagiaire et le formateur ;
- Évaluation qualitative de fin de formation, qui est ensuite analysée par l'équipe pédagogique AWC ;
- Attestation de fin de formation, remise au stagiaire en main propre ou par courrier électronique.

### **Qualification du formateur**

La formation est animée par un professionnel de l'informatique et de la pédagogie, dont les compétences techniques, professionnelles et pédagogiques ont été validées par des certifications et/ou testées et approuvées par les éditeurs et/ou notre équipe pédagogique. Il est en veille technologique permanente et possède plusieurs années d'expérience sur les produits, technologies et méthodes enseignés.

Il est présent auprès des stagiaires pendant toute la durée de la formation.

### **Pour les personnes en situation de handicap**

Les formations AWC sont ouvertes à toutes et tous, inclusives et accueillantes pour les personnes en situation de handicap. Concernant les modalités physiques de la formation, c'est environ 7h par jour en place assise, dans un bureau fermé, avec la nécessité de se déplacer dans la salle et d'éventuellement transporter son ordinateur. Concernant les modalités organisationnelles, il y a possibilité de télétravail ou, en présentiel, d'isolement dans une salle pour se reposer, même s'il y a la nécessité de travailler régulièrement en groupe.

La mise en place d'aménagements est possible, n'hésitez pas à prendre contact avec votre référent et notre référent handicap [contact@awcformation.fr](mailto:contact@awcformation.fr) ou par téléphone au 01 85 10 00 33.