

PROGRAMMATION PYTHON



Objectifs et contexte de la certification

Les objectifs de cette formation Python se présentent comme suit:

- 1- Acquérir une compréhension fondamentale de la syntaxe et de la structure de Python.
- 2- Apprendre à utiliser les variables, les types de données et les opérateurs Python pour manipuler des informations.
- 3- Comprendre les structures de contrôle telles que les boucles et les instructions conditionnelles pour créer des programmes efficaces.
- 4- Explorer les fonctions et les modules Python pour organiser et réutiliser le code
- 5- Apprendre à manipuler des listes, des tuples, des dictionnaires et d'autres structures de données intégrées.
- 5- Comprendre les concepts de programmation fonctionnelle en Python
- 6- Se préparer à poursuivre l'apprentissage autonome pour continuer à améliorer ses compétences en Python

En résumé, ces objectifs fournissent une base solide pour les apprenants afin de maîtriser le langage Python et d'être en mesure d'utiliser ses capacités pour diverses applications.

Temps moyen de formation

25 heures

Pré requis technique

Connaissance de base des outils informatique, environnement Windows. .

Moyens pédagogiques et techniques

- Connexion à la plate-forme de Blended Learning à l'aide d'identifiants personnels qui ont été communiqués par mail avant le début de la formation
- Intégration d'une application de visioconférence autorisant une partie de la formation en visioconférence avec le formateur.
- Cours en ligne comprenant des supports pédagogiques, des exercices de pratique guidée et des vidéos d'aide à la pratique personnelle.
- Exercices et questionnaires en ligne : Questions à trous, Qcm/Qrm, questions correspondance/apparier, questions ouvertes, zones sur image (hotspot), réponse sur image, questions calculées, réordonner des mots, correspondance par glisser/coller, expression orale, compréhension à la lecture....

Délais d'accès

- 48h

Numéro de fiche certification

<https://www.francecompetences.fr/recherche/rs/6202/>

Accessibilité aux handicapés

- Entretien préalable pour définir les modalités d'accès

Contacts

- Référent administratif et handicap : M. IDMOULID Hichem 06 51 54 38 30
- Référent pédagogique : à définir

Compétences attestées

Objectifs et contexte de la certification :

L'objectif de cette certification est double : faciliter pour les employeurs et les chercheurs d'emploi l'identification des compétences nécessaires pour le développement informatique malgré la très grande variété de postes mais également mettre en valeur des compétences réelles de l'individu et faire correspondre postes et candidats efficacement, pour simplifier et accélérer l'embauche.

Compétences attestées :

Niveau opérationnel (score Tosa 551 à 725)

- Utiliser les fonctionnalités de contrôles de flux afin d'automatiser un traitement de données conditionnels et/ou répétitifs
- Créer un programme pour des traitements complets de données, en utilisant les fonctionnalités avancées des contrôles de flux et des opérations arithmétiques
- Reconnaître et créer des objets simples de type intégrés (*built-in*), en vue de créer des programmes manipulant des données
- Reconnaître les objets structurés de type intégré, afin d'effectuer des traitements de multiples variables simples
- Manipuler les objets intégrés simples et structurés et les assembler dans le but de créer des fonctions simples et réutilisables
- Maîtriser les concepts liés aux modules en vue d'inclure des fonctionnalités d'un module à un programme
- Sélectionner et importer des fonctions spécifiques d'un package de l'API afin de les réutiliser dans un traitement de données
- Créer un package simple complet pour créer des fonctionnalités partageables et réutilisables / Stocker et traiter simultanément plusieurs données, en vue de créer un flux de traitement de données simples adapté
- Modéliser un problème et automatiser des manipulations de données, afin de traiter un volume important ou complexe de données.

Niveau avancé (score Tosa 726 à 875)

- Connaître les règles syntaxiques et sémantiques de Python afin de créer des programmes clairs et réutilisables
- Maîtriser les outils d'entrée-sortie, dans le but de lire et modifier des documents externes au programme
- Maîtriser les spécificités des types intégrés afin de traiter efficacement et de manière claire des grands volumes de données
- Créer des fonctions documentées, des classes et leurs méthodes associées, dans le but de créer des fonctionnalités intégrables dans d'autres programmes
- Importer des packages composés pour exploiter l'ensemble des fonctionnalités incluses
- Utiliser les packages fondamentaux de la bibliothèque standard afin de manipuler efficacement des données externes habituelles
- Implémenter des structures de données adaptées et choisir les fonctions adaptées afin de concevoir des programmes à la complexité algorithmique maîtrisée

Niveau expert (score Tosa 876 à 1000)

- Intégrer la gestion d'erreur afin de créer des programmes couvrant tous les cas d'entrée et de gestion de données
- Utiliser les règles et conventions de nommage afin de créer des programmes pouvant s'intégrer dans un environnement complet
- Mettre en œuvre l'intégralité des outils fonctionnels, afin de créer des fonctions adaptées et efficaces
- Utiliser et implémenter les concepts de la programmation orientée objet, afin de créer des structures spécifiques à un projet
- Maîtriser l'ensemble des concepts liés aux modules pour intégrer des packages de la manière la plus performante possible
- Utiliser l'intégralité de la bibliothèque standard pour traiter des données et documents externes, en local ou sur internet
- Mettre en place des outils algorithmiques avancés afin de drastiquement améliorer la performance d'un programme existant

Modalités d'évaluation :

Algorithme adaptatif : le niveau des questions s'adapte au niveau du candidat tout au long du déroulement du test

Scoring mathématique IRT (Item Response Theory) : score sur 1000

Typologies de questions : activités interactives

Format du test : 35 questions – 90 minutes

Dispositif de suivi de l'exécution et d'évaluation des résultats de la formation

- Suivi en ligne du stagiaire par le e-formateur en temps réel durant toute la durée de la formation en e-learning avec possibilité d'intervention par message interne, chat ou visioconférence en cas de problématique.
 - Intervention du e-formateur par visioconférence en fin de parcours pour récapitulatif de la formation et personnalisation à chaque cas personnel.
 - Mise en place de travaux personnalisés à réaliser sur l'espace dédié à cet effet avec corrections du e-formateur.
- Formulaire d'évaluation en ligne à la fin de la formation débouchant sur la remise d'une Attestation de suivi de formation
- Mise en ligne de feuille d'émargement pour la partie de formation en Visioconférence dite « synchrone » et d'une feuille d'assiduité avec récapitulatif des connexions pour la partie de formation en e-learning dite « asynchrone ».
 - Mise en ligne d'une enquête de satisfaction concernant la formation afin d'évaluer la prestation de formation sous les aspects techniques et pédagogiques.

Points forts

- Certification 6202
- De nombreux ateliers et applications pour appréhender les bases
- Mentorat avec un professionnel-expert
- Contenus supplémentaires d'aide à la valorisation des compétences
- Assistance pédagogique prioritaire par e-mail et tchat illimité
- 10 rendez-vous privés par téléphone ou vidéo-conférence
- Vidéos accessibles en illimité pendant 1 an

Programme

INTRODUCTION

Test de positionnement

BASES DE PYTHON

- 1- Définition des variables
- 2- Applications
- 3- Applications 2
- 3- Listes et dictionnaires
- 4- Applications
- 5- Applications 4
- 6- Fonctions et méthodes
- 7 - Applications
- 8 - Applications 6
- 9- Développer une fonction
- 10- Applications
- 11- Applications 8
- 12- If_Else_while
- 13- Applications
- 14- Applications 10
- 15- Range et boucle for
- 16- Applications
- 17- datetime

NOTIONS PLUS AVANCEES

- 1 - Installation et introduction
- 2 - Commentaire et type de variable
- 3 - Opérateurs arithmétiques et exercice TVA
- 4 - Opérateurs simplifiés et association de variables
- 5 - Découper un texte
- 6 - Importer des librairies
- 7 - Input
- 8 - opérateurs rationnels et conditions if
- 9 - Exercice calculatrice
- 10 - Opérateurs logiques
- 11 - Boucle while
- 12 - Boucle for
- 13 - Break_continue
- 14 - Exercice deviner un numéro
- 15 - Listes
- 16 - Fonctions pour les listes
- 17 - Tuple
- 18 - Dictionnaires
- 19 - Sets
- 20 - Opérations avec les sets
- 21 - Nested types
- 22 - Utiliser la boucle for avec des nested types
- 23 - Dictionnaires dans dictionnaires ou listes dans dictionnaires
- 24 - Document_Dictionnaires dans dictionnaires ou listes dans dictionnaires
- 25 - Extraire les données en utilisant la boucle for
- 26 - Extraire des données d'un dictionnaire inclut dans un autre dictionnaire
- 27 - Items
- 28 - Document_Exercice avec l'utilisation de items
- 29 - Exercice dictionnaire
- 30 - Document_Exercice dictionnaire

- 31 - Comprehension list
- 32 - Document_comprehension list
- 33 - Comprehension dictionary
- 34 - Document_comprehension dictionary
- 35 - Comprehension set
- 36 - generator expression
- 37 - Exercice
- 38 - Fonction
- 39 - Document_fonction
- 40 - Return
- 41 - Module
- 42 - Document_figures
- 43 - Enumeration
- 44 - Document_enumeration
- 45 - Exercice
- 46 - Document_Exercice
- 47 - Mesurer le temps requis d'une fonction
- 48 - Document_Temps requis pour remettre un résultat d'une fonction
- 49 - Appel à une fonction à l'intérieur d'une autre
- 50 - Document_appel a une fonction à l'intérieur d'une autre
- 51 - Argument par défaut
- 52 - Document_argument par default
- 53 - argument nommé et non nommé
- 54 - Exercice
- 55 - Document_Exercice 50
- 56 - Utiliser les astéris avec les arguments
- 57 - Document_utiliser les asterix avec les arguments
- 58 - Variable locale et globale
- 59 - Mutable vs immutable objects
- 60 - Document_Mutable vs immutable objects
- 61 - Mutable vs immutable objects_partie 2
- 62 - Document_Mutable vs immutable objets_partie 2
- 63 - Mutable vs immutable objects_partie 3
- 64 - Document_mutable vs immutable objects_partie 3
- 65 - Copy
- 66 - Deepcopy
- 67 - Document_Deepcpy
- 68 - Fonction anonyme
- 69 - Document_fonction anonyme
- 70 - Nombres aléatoires
- 71 - Document_nombres aléatoires
- 72 - Choice_Choices_Suffle_Sample
- 73 - Création de fichiers
- 74 - Document_création de fichiers
- 75 - Lire un fichier
- 76 - Seek_tell_append
- 77 - Fonctions supplémentaires pour fichier
- 78 - Exercice
- 79 - Document_Exercice 70
- 80 - Fonction generator
- 81 - Exercice
- 82 - Document_Exercice 72

Quiz final