Smartfuture
Certifications | Training | Technology solutions

Formation & Certification • IT Specialist – Python

IT Specialist – Python

Cette formation prépare à la certification IT Specialist – Python et apprend à écrire un code correct pour résoudre des problèmes en utilisant types de données, structures de contrôle, fichiers, fonctions, gestion des erreurs et modules standards.

Distributeur officiel Certiport

Centre d'examen Certiport

Learn • Practice • Certify

Durée 34,5 h (recommandé)	**Examen** IT Specialist – Python
Modalité Distanciel	**Niveau** Fondamental / Junior

INSCRIPTION / RÉSERVATION

Je m'inscris maintenant

- **Learn** : Bases de Python : structures de contrôle, fonctions, fichiers, gestion des erreurs, modules et bonnes pratiques de code.
- **Practice** : Exercices pratiques, mini-projets et préparation ciblée à l'examen.
- **Certify** : passage de la certification IT Specialist – Python dans notre centre Certiport (voucher en option selon l'offre).

OBJECTIFS PÉDAGOGIQUES

- Reconnaître et utiliser les types de données Python et les opérateurs de base.
- Écrire des scripts avec conditions, boucles, fonctions et gestion des exceptions.
- Maîtriser l'entrée/sortie, la documentation et l'utilisation des modules standards.
- Analyser, tester et déboguer le code en corrigeant les erreurs de syntaxe, logique et exécution.

PUBLIC CIBLE

- Pour étudiants, lycéens et débutants souhaitant apprendre Python et obtenir une certification internationale.
- Pour personnes en reconversion vers le développement, la data ou l'automatisation.
- Pour enseignants et formateurs souhaitant créer un parcours d'initiation à Python conforme aux objectifs Certiport.

PRÉREQUIS

- Lecture au niveau collège avec bonnes bases en mathématiques et informatique.
- Capacité à raisonner logiquement et à décomposer les problèmes en étapes.

PROGRAMME DE LA FORMATION – DÉTAILLÉ

Opérations avec les types de données et les opérateurs

Évaluer des expressions pour identifier les types de données

- Reconnaître les types de données que Python assigne aux variables : str,int, float et bool.

Effectuer et analyser des opérations sur les données et les types

- Conversion de types (casting), indexation, slicing de chaînes et de listes, création et manipulation de structures de données simples (listes, opérations sur les listes).

Déterminer l'ordre d'exécution en fonction de la priorité des opérateurs

- Priorité des opérateurs : affectation, comparaison, logiques, arithmétiques, identité (is), appartenance (in).

Choisir les opérateurs pour obtenir le résultat souhaité

- Sélectionner les opérateurs d'affectation, de comparaison, logiques, arithmétiques, d'identité et d'appartenance adaptés à un scénario donné.

Contrôle de flux avec décisions et boucles

Construire et analyser des segments de code utilisant des instructions conditionnelles

- Utilisation de if, elif, else, expressions conditionnelles composées et imbriquées.

Construire et analyser des segments de code utilisant des boucles

- Boucles while et for, instructions break, continue,pass, boucles imbriquées, boucles avec expressions conditionnelles composées.

Opérations d'entrée et de sortie

Construire et analyser des segments de code qui réalisent des E/S fichiers

- Utiliser open, close, read, write, append; vérifier l'existence d'un fichier, le supprimer, utiliser l'instruction with.

Construire et analyser des segments de code pour l'entrée/sortie console

- Lecture de l'entrée via input(), affichage avec print et texte formaté (méthode str.format(), f-strings), utilisation des arguments de ligne de commande.

PROGRAMME DE LA FORMATION – DÉTAILLÉ

Documentation et structuration du code

Documenter des segments de code

- Utiliser l'indentation, les espaces, les commentaires, les chaînes de documentation (docstrings) ; générer une documentation avec pydoc.

Construire et analyser des segments de code incluant des définitions de fonctions

- Signatures d'appel, valeurs par défaut, return, mot-clé def, utilisation de pass pour des squelettes de fonctions.

Dépannage et gestion des erreurs

Analyser, détecter et corriger des segments de code comportant des erreurs

- Identifier et corriger les erreurs de syntaxe, de logique et d'exécution (runtime).

Analyser et construire des segments de code qui gèrent les exceptions

- Utiliser try, except, else, finally, lever des exceptions avec raise.

Réaliser des tests unitaires

- Utiliser le module unittest, écrire des fonctions/méthodes de test, utiliser les assertions assertIsInstance, assertEqual,assertTrue, assertIs, assertIn.

Opérations avec les modules et outils

Réaliser des opérations système et ligne de commande avec les modules intégrés

- Utiliser les modules io, os, os.path, sys(import de modules, ouverture/lecture/écriture de fichiers, arguments de ligne de commande).

Résoudre des problèmes de calcul complexes avec les modules intégrés

- Module math : fabs, ceil, floor,trunc, fmod, frexp, nan, isnan,sqrt, isqrt, pow, pi.
- Module datetime : now, strftime, weekday.
- Module random : randrange, randint, random,shuffle, choice, sample.