
Programme de Formation

Python - Programmation Objet

Description

Apprenez sur des cas concrets à créer, optimiser, et déployer des modèles prédictifs de Machine Learning en utilisant Python et la librairie scikit-learn.

Points Forts

Perfectionnez vos compétences en programmation orientée objet avec Python : apprenez à concevoir des applications modulaires et évolutives, à structurer votre code pour une meilleure maintenabilité, et à utiliser les concepts avancés.



Public visé

Ingénieurs, développeurs débutants ou intermédiaires souhaitant acquérir les fondamentaux Python et POO.



Objectifs pédagogiques

Cette formation prépare aux compétences du niveau Opérationnel (551-725 points) du référentiel Programmer et automatiser des tâches avec Python (Tosa). À l'issue de la formation, vous serez capable de :

- Maîtriser les outils de développement Python (interpréteur, IDE, Notebook)
- Maîtriser la syntaxe Python et les structures de contrôle
- Utiliser les fonctionnalités des modules Python et appliquer les bonnes pratiques
- Maîtriser la programmation orientée objet en Python
- Gérer et optimiser les données avec numpy et pandas
- Tester, déboguer et optimiser le code Python

Organisation

Durée : 35 heures

Mode d'organisation : Mixte

Contenu pédagogique



Description

Programme structuré selon les 4 domaines du référentiel Python (Tosa) niveau Opérationnel (551-725 points).

Module 1 : Environnement de développement Python

- Introduction à Python : historique, comparaison avec autres langages
- Utiliser l'interpréteur Python
- Environnements de développement : IDE, Notebook Jupyter

Module 2 : Maîtriser la syntaxe et les structures de contrôle Python

2.1 Syntaxe de base

- Variables et nombres
- Opérateurs numériques, de comparaison et logiques



- Chaînes de caractères et opérateurs spécifiques

2.2 Structures de contrôle

- Conditions (if, elif, else)
- Boucles (for, while)
- Gestion des exceptions avec try/except

2.3 Fonctions

- Définir et utiliser des fonctions
- Arguments et valeurs de retour
- Fonctions intégrées Python

Module 3 : Structures de données et manipulation

3.1 Conteneurs de base

- Tuples : création, manipulation, immutabilité
- Listes : création, ajout, suppression, slicing
- Ensembles : opérations ensemblistes
- Dictionnaires : clés/valeurs, méthodes
- Itérations sur les conteneurs

3.2 Manipulation avancée des données

- Introduction à NumPy : tableaux multi-dimensionnels
- Introduction à Pandas : création et manipulation de DataFrames
- Import/export de données (CSV, Excel)
- Filtrage et sélection de données

Module 4 : Programmation orientée objet

4.1 Concepts fondamentaux POO

- Objets et classes : définition et instanciation
- Constructeurs et destructeurs
- Attributs d'instance et de classe
- Méthodes d'instance, de classe et statiques

4.2 Concepts avancés POO

- Propriétés (@property)
- Surcharge des méthodes intégrées et opérateurs
- Héritage simple et polymorphisme
- Gestion des exceptions personnalisées

Module 5 : Modules, packages et bonnes pratiques

5.1 Utiliser et créer des modules

- Importer des modules existants (import, from...import)
- Créer ses propres modules
- Structurer le code en packages
- Documentation du code (docstrings)

5.2 Gestion des environnements

- Environnements virtuels (venv)
- Installation de bibliothèques avec pip
- Gestion des dépendances (requirements.txt)

5.3 Manipulation de fichiers et données

- Lecture et écriture de fichiers texte
- Manipulation du système de fichiers
- Variables d'environnement
- Arguments en ligne de commande

Module 6 : Bases de données avec Python

- Introduction à la spécification DB-API
- Connexion et manipulation de bases de données SQLite
- Requêtes SQL avec Python
- Introduction à SQLAlchemy (ORM)

Module 7 : Interfaces graphiques avec TKINTER

- Introduction à TKINTER
- Composants graphiques (widgets) : boutons, labels, entrées
- Gestion des événements

Module 8 : Optimisation et tests

8.1 Tests et débogage

- Linter : vérification syntaxe et style (pylint, flake8)
- Tests unitaires avec unittest
- Techniques de débogage

8.2 Optimisation du code

- Introduction au profiling (mesure performances)
- Bonnes pratiques pour améliorer l'efficacité du code

Module 9 : Entraînement certification TOSA Python Opérationnel

- Exercices pratiques sur les 4 domaines du référentiel
- Manipulation des outils Python niveau Opérationnel
- Activités interactives (QCM, exercices code)
- Mises en situation professionnelles
- Test blanc TOSA complet avec analyse détaillée du score (551-725)

Certification : Formation préparant au passage de la certification Python (Tosa) - RS6962 niveau Opérationnel. Certification obligatoire en financement CPF et optionnelle pour autres financements.

Prérequis

Connaissances de base en programmation (souhaitable mais non obligatoire en langage objet). Ordinateur connecté à internet, micro et caméra (pour formations en classe virtuelle). Test de positionnement obligatoire.

Modalités pédagogiques

Formation en présentiel ou à distance (classe virtuelle) - Inter/Intra - Groupes de 4 à 12 participants. Méthodes : présentation des concepts, discussions techniques, démonstrations, exercices et travaux pratiques.

Matériel fourni :

- En présentiel : Un poste informatique par participant, connecté à Internet
- À distance : PC configuré accessible via outil de prise en main à distance (Meet)



Moyens et supports pédagogiques

- Exercices concrets et cas pratiques
- Plateforme collaborative avec code source des exercices
- Quiz d'évaluation des connaissances



Modalités d'évaluation et de suivi

Positionnement en amont de la formation

- Quiz de consolidation des pré-requis pour adapter la formation

Suivi « pendant »

- Feuilles de présence
- Exercices pratiques
- Évaluation des acquis en fin de formation
- Formulaires d'évaluation de la formation

Évaluation finale

- Test blanc TOSA complet avec analyse détaillée
- Passage de la certification Python (Tosa) en centre agréé

Certification Python (Tosa) - RS6962 - Niveau Opérationnel

Test adaptatif évaluant la maîtrise de Python sur les 4 domaines de compétences.

Durée totale : 90 minutes (35 questions niveau Opérationnel)

Format du test :

- Test numérique adaptatif (difficulté s'adapte au niveau)
- Activités interactives (relier, glisser-déposer, cliquer sur zone, listes déroulantes)
- QCM et exercices de mises en situation professionnelles
- Manipulation de Python dans son environnement

Scoring : Score sur 1000 points (méthode IRT - Item Response Theory)

Niveaux évalués :

- < 551 points : Non certifiable
- Opérationnel (551-725 points - score visé) : Certification obtenue
- Avancé (726-875 points)
- Expert (876-1000 points)

Score minimum requis : 551 points (Niveau Opérationnel) pour obtenir la certification visée par cette formation.

Résultats : Attestation détaillée avec score par domaine de compétence et certificat Python TOSA niveau Opérationnel.

Conditions de passage : En ligne, surveillance en présentiel par un examinateur ou à distance par proctoring sécurisé.

Évaluation à froid

Suivi post-formation : Questionnaire de satisfaction à j+60 mesurant l'impact des compétences acquises en situation professionnelle.