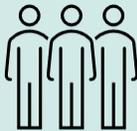


# Formation Programmation PYTHON



## Public concerné

Les développeurs qui souhaitent apprendre à coder dans le langage PYTHON.



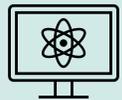
## Nombre d'inscrits en 23

50



## Objectifs pédagogiques

Avec cette formation Programmation PYTHON vous apprendrez le langage de programmation Python au travers de cas pratiques et saurez utiliser les différentes fonctionnalités pour vos projets web.



## Méthodologie

L'ensemble de la formation se déroule par le suivi de 54 leçons de vidéos en ligne et l'accompagnement d'un tuteur..



## Accessibilité à tous :

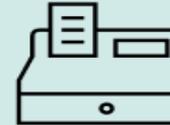
Pas de déplacements nécessaires. Adressez-vous à nous en cas de handicap.

Formation accessible dès 14 jours après l'inscription respectant votre droit de rétractation - art. L221-18 du code de la Conso



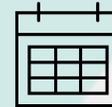
## Certification

TOSA



## Tarif

Selon votre situation, une proposition commerciale vous sera éditée



## Lieu et dates

- En E-learning
- 3 mois d'accès depuis l'inscription
- 25 heures
- tutorat



## Prérequis

Ce programme est basé sur l'entraînement des participants. Il est nécessaire de maîtriser l'outil informatique et les différents navigateurs.



## Contact

- Information, inscription ou devis sur mesure
- Assistance tout au long de la formation
- Tél : 09 77 29 70 51
- Mobile : 07 55 53 62 75
- mail : [maformation@modulage.fr](mailto:maformation@modulage.fr)



# 1. Cas pratique 1

1. Calculer une moyenne, un total et un pourcentage
2. Déterminer le nombre de jours dans un mois
3. Déterminer si un nombre est fort ou non
4. Supprimer les doublons d'une liste
5. Trouver la somme des éléments de chaque ligne et colonne d'une matrice
6. Manipuler une chaîne de caractère
7. Utiliser les fonctions récursives
8. Utiliser les tuples

# 2. Introduction au langage Python

1. Introduction à Python
2. L'environnement de travail Python
3. Créer un fichier et l'exécuter avec Python

# 3. Notions fondamentales en Python

1. Mots-clés et identificateurs en Python
2. Instructions, indentation et commentaires en Python
3. Les principaux types de données
4. Présentation des variables
5. Entrée / sortie en Python : fonctions `input()` et `print()`

# 4. Les structures de contrôle

1. Les opérateurs en Python
2. Les instructions if-else en Python
3. Les boucles en Python
4. Créer des conditions Python complexes
5. Convertir une boucle for imbriquée en une boucle for simple

# 5. Les fonctions

1. Introduction aux fonctions Python
2. Définition et syntaxe des fonctions
3. Notions avancées sur les paramètres et arguments
4. Arguments `*args` et `**kwargs`
5. Contrôle des valeurs de retour d'une fonction
6. La portée des variables en Python
7. La fonction `yield`
8. Quelques fonctions utiles

# 6. Cas pratique 3

1. Créer une classe CompteBancaire
2. Utiliser l'héritage des classes
3. Utiliser le polymorphisme
4. Simuler le tirage d'une loterie
5. Utiliser les décorateurs

# 7. POO : Notions fondamentales

1. Introduction à la Programmation Orientée Objet en Python
2. Constructeurs et destructeurs
3. Classes, objets et attributs
4. Gérer la visibilité des membres de classes

# 8. POO : Notions avancées

1. Héritage
2. Polymorphisme
3. Surcharge des opérateurs
4. Itérateurs et générateurs
5. Listes des méthodes utiles

# 9. Sujets avancés

1. Les métaclasses
2. Les décorateurs
3. Les itérateurs avancés
4. Instruction assert
5. Conclusion

# 10. Les différents types de données

1. Les chaînes de caractères
2. Les listes
3. Les tableaux – module array
4. Les tuples
5. Les ensembles
6. Les dictionnaires
7. Les matrices
8. Récapitulatif

# 11. Cas pratique 2êtage

1. Effectuer des opérations de lecture sur des fichiers
2. Effectuer des opérations d'écriture sur des fichiers
3. Gérer les erreurs
4. Manipuler les dates
5. Générer un mot de passe aléatoire

# 12. Manipulation de fichiers avec Python

1. Introduction à la manipulation de fichiers
2. Opérations sur les fichiers
3. Échange de données avec le module JSON
4. Autres méthodes pour la gestion des fichiers

# 13. La gestion des erreurs en Python

1. Introduction à la gestion d'erreurs ou d'exceptions
2. Gérer les exceptions avec try, except, else et finally
3. Liste des exceptions

# 14. Modules standards et paquets Python

1. Les modules et paquets
2. Modules math, random et statistics
3. Les modules datetime, time et calendar
4. Module re : expressions régulières ou rationnelles