

Mastère Spécialisé

Développement en Python



Mastère Spécialisé Développement en Python

- » Modalité: en ligne
- » Durée: 12 mois
- » Qualification: TECH Euromed University
- » Accréditation: 60 ECTS
- » Horaire: à votre rythme
- » Examens: en ligne

Accès au site web: www.techtitute.com/fr/informatique/master/master-developpement-python

Sommaire

01

Présentation

page 4

02

Objectifs

page 8

03

Compétences

page 12

04

Direction de la formation

page 16

05

Structure et contenu

page 20

06

Méthodologie d'étude

page 30

07

Diplôme

page 40

01

Présentation

Python est un langage de programmation polyvalent et efficace. Tout d'abord, sa syntaxe claire et lisible facilite l'écriture et la compréhension du code, ce qui accélère le processus de développement et réduit la probabilité d'erreurs. En outre, la grande variété de bibliothèques et de Frameworks disponibles permet aux développeurs d'accéder à des outils puissants et préexistants, ce qui permet un développement plus rapide et plus efficace. Sa nature open source et sa communauté active de développeurs contribuent également à l'amélioration et à la actualisation constantes du langage, ce qui garantit un environnement de développement dynamique et moderne. C'est pourquoi TECH Euromed University a mis au point un programme complet, destiné à former les professionnels aux dernières avancées en matière de Développement Python, avec une méthodologie 100 % en ligne.



“

Python est réputé pour son adaptabilité, étant utilisé dans des applications, du développement web à l'Intelligence Artificielle. Qu'attendez-vous pour vous inscrire?”

Python est un langage de Programmation de haut niveau, largement utilisé parmi les informaticiens, car il dispose d'un large éventail de bibliothèques et de cadres qui simplifient les tâches courantes, permettant aux développeurs de se concentrer sur la logique de leurs applications au lieu de perdre du temps sur des détails techniques. De plus, sa polyvalence est un autre avantage important, car il peut être utilisé dans une variété d'environnements, du développement web à l'analyse de données et à l'apprentissage automatique.

C'est ainsi qu'est né ce Mastère Spécialisé en Développement en Python, un programme qui couvrira la manipulation avancée des données et des types en Python, en explorant les identificateurs, les mots-clés, les types intégraux et booléens, ainsi que le formatage avancé des strings et des encodages. En outre, les collections telles que les tuples, les listes et les dictionnaires seront examinées, ainsi que les techniques d'itération et les fonctions lambda, ce qui permettra d'acquérir une solide base dans les principes fondamentaux du langage.

En outre, le développement d'applications Python sera approfondi, l'accent étant mis sur les meilleures pratiques et les méthodologies modernes. De l'architecture des applications au déploiement et maintenance, seront abordés des aspects tels que la conception et la modélisation avancées, la gestion des dépendances, les modèles de conception, testings et debugging, l'optimisation des performances, les stratégies de déploiement et de distribution.

De même, l'informaticien se plongera dans le développement web et mobile avec Python, couvrant des frameworks tels que Django et Flask, ainsi que le développement d'API et de services web. En outre, il se concentrera sur la conception d'interfaces et l'expérience utilisateur (UI/UX), de l'utilisation d'outils de conception à l'amélioration de l'accessibilité et de la facilité d'utilisation. Enfin, la gestion et l'analyse des données seront abordées à l'aide de Python et d'outils tels que NumPy, Pandas et Matplotlib.

Ce diplôme est donc présenté comme une opportunité unique, à travers une proposition académique entièrement en ligne et adaptable. Grâce à cette approche, les professionnels bénéficieront d'une plus grande liberté dans la gestion de leur temps d'accès, ce qui leur permettra d'harmoniser leur travail quotidien et leurs engagements personnels.

Ce **Mastère Spécialisé en Développement en Python** contient le programme le plus complet et le plus actualisé du marché. Ses caractéristiques sont les suivantes:

- Le développement d'études de cas présentées par des experts en Développement Python
- Le contenu graphique, schématique et éminemment pratique de l'ouvrage fournit des informations théorique et pratiques sur les disciplines essentielles à la pratique professionnelle
- Les exercices pratiques où effectuer le processus d'auto-évaluation pour améliorer l'apprentissage
- Il met l'accent sur les méthodologies innovantes
- Cours théoriques, questions à l'expert, forums de discussion sur des sujets controversés et travail de réflexion individuel
- La possibilité d'accéder aux contenus depuis n'importe quel appareil fixe ou portable doté d'une connexion internet



Une formation complète et spécialisée en Développement en Python, qui vous préparera à relever les vrais défis du monde du développement de Software”

“

Grâce à ce Mastère Spécialisé, 100% en ligne, vous approfondirez vos connaissances en matière de conception Frontend et Backend, de gestion de bases de données et de stratégies de publication dans les magasins d'applications”

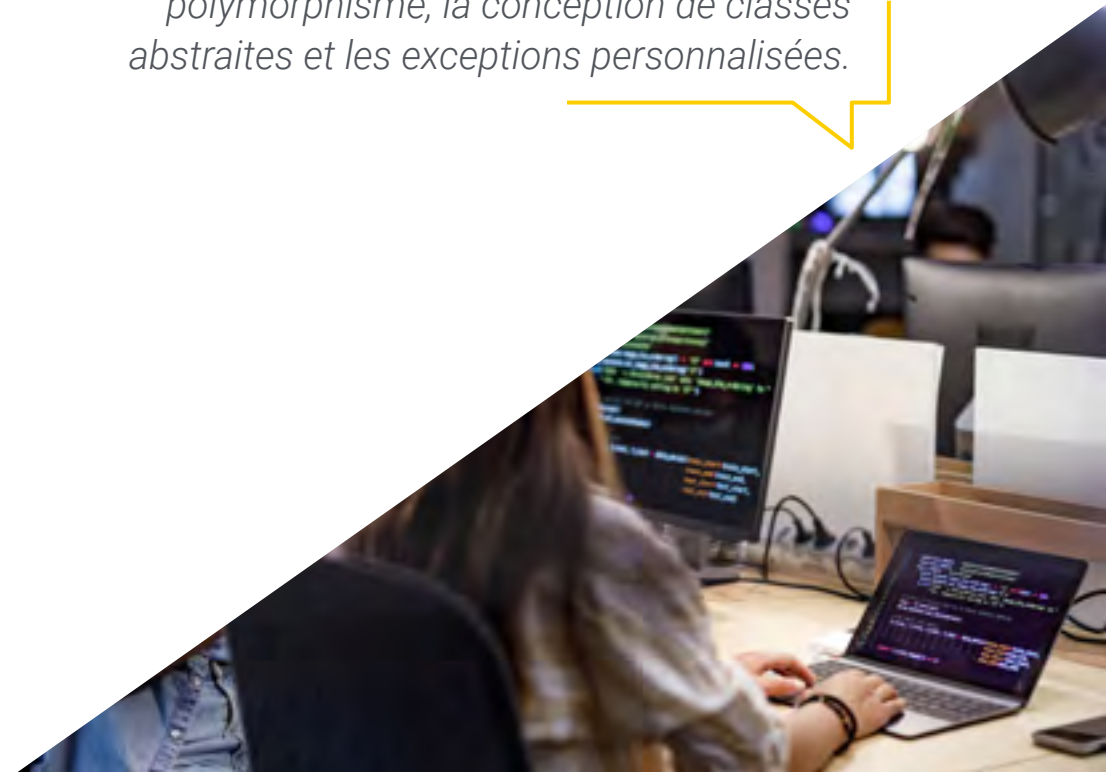
Le corps enseignant du programme englobe des spécialistes réputés dans le domaine et qui apportent à ce programme l'expérience de leur travail, ainsi que des spécialistes reconnus dans de grandes sociétés et des universités prestigieuses.

Grâce à son contenu multimédia développé avec les dernières technologies éducatives, les spécialistes bénéficieront d'un apprentissage situé et contextuel, ainsi, ils se formeront dans un environnement simulé qui leur permettra d'apprendre en immersion et de s'entraîner dans des situations réelles.

La conception de ce programme est axée sur l'Apprentissage par les Problèmes, grâce auquel le professionnel doit essayer de résoudre les différentes situations de la pratique professionnelle qui se présentent tout au long du programme académique. Pour ce faire, l'étudiant sera assisté d'un innovant système de vidéos interactives, créé par des experts reconnus.

Vous apprendrez des notions de base, telles que les variables et les types de données, les techniques plus avancées de visualisation des données et les stratégies d'optimisation des performances et du stockage. Inscrivez-vous dès maintenant!.

Vous vous plongerez dans la Programmation Orientée Objet (POO), ainsi que dans des sujets tels que les classes, l'héritage, le polymorphisme, la conception de classes abstraites et les exceptions personnalisées.



02

Objectifs

Ce Mastère Spécialisé a été conçu dans le but de former des professionnels hautement compétents et spécialisés dans le vaste univers de la programmation dans ce langage très performant. Grâce à des modules complets, les diplômés se plongeront dans la syntaxe et les fonctionnalités de Python, du traitement avancé des données à la maîtrise de la Programmation Orientée Objet (POO) et à la création efficace d'applications web et mobiles. Grâce à une approche pédagogique rigoureuse et actualisée, ce programme garantit une expérience éducative unique qui fera des informaticiens des experts très recherchés dans le secteur du développement de logiciels. Préparez-vous à repousser vos limites et à vous embarquer dans un voyage éducatif qui vous permettra non seulement d'améliorer vos compétences en Python, mais aussi de disposer des outils nécessaires pour exceller et jouer un rôle de premier plan dans le monde dynamique du développement technologique.



“

Devenez un leader dans le domaine du développement de logiciels, en les dotant de connaissances avancées et de compétences spécialisées pour relever les défis technologiques contemporains”



Objectifs généraux

- ♦ Fournir une compréhension globale de Python
- ♦ Former à la manipulation avancée des données et des types en Python
- ♦ Appliquer les principes de la Programmation Orientée Objet (POO) en Python
- ♦ Encourager l'utilisation des meilleures pratiques et des méthodologies modernes dans le développement de logiciels
- ♦ Fournir une formation complète en développement web et mobile avec Python
- ♦ Intégrer les principes UI/UX dans le développement de software
- ♦ Former à la configuration et à l'utilisation d'outils et d'environnements de développement de données
- ♦ Approfondir l'utilisation des structures de données et des fonctions en Python
- ♦ Former sur les techniques avancées de visualisation de données avec Matplotlib
- ♦ Formation aux stratégies d'optimisation des performances et l'entreposage de données



Grâce à la vaste bibliothèque offerte par ce diplôme, remplie des ressources multimédias les plus innovantes, vous obtiendrez les outils nécessaires pour vous démarquer et jouer un rôle de premier plan dans le monde dynamique du développement technologique”





Objectifs spécifiques

Module 1. Programmation en Python

- ♦ Permettre la configuration et l'utilisation efficace de l'environnement de développement Python
- ♦ Comprendre les concepts avancés de la Programmation

Module 2. Données avancées et contrôle de flux avec Python

- ♦ Maîtriser les conventions et les pratiques de manipulation des identificateurs et des mots-clés
- ♦ Appliquer des structures de données complexes et leurs opérations

Module 3. Programmation Orientée Objet (POO) en Python

- ♦ Maîtriser la création et l'utilisation de classes et d'objets en Python
- ♦ Appliquer l'héritage et le polymorphisme en Python

Module 4. Développement d'applications en Python

- ♦ Spécialisation dans la conception et la modélisation avancées des applications
- ♦ Formation à l'optimisation, au déploiement et à la maintenance des applications

Module 5. Développement web et mobile avec Python

- ♦ Utiliser les Frameworks web Python les plus connus
- ♦ Se préparer au développement et à la publication d'applications mobiles

Module 6. Interface et expérience utilisateur avec Python

- ♦ Former aux techniques de conception réactive et adaptative
- ♦ Se préparer aux tests d'utilisabilité et à l'analyse du comportement de l'utilisateur

Module 7. Traitement des données et *Big Data* avec Python

- ♦ Manipuler les techniques de contrôle de flux et les fonctions pour le traitement des données
- ♦ Promouvoir les meilleures pratiques pour le codage et la gestion des erreurs en Python

Module 8. Structures de données et fonctions en Python

- ♦ Créer et utiliser des fonctions d'une manière avancée
- ♦ Lire et écrire des fichiers, et les traiter

Module 9. Traitement des données en Python avec NumPy et Pandas

- ♦ Créer et manipuler Arrays avec NumPy
- ♦ Promouvoir les compétences en visualisation de données avec Matplotlib

Module 10. Techniques avancées et applications pratiques avec NumPy et Pandas

- ♦ Développer des connaissances spécialisées dans le chargement et le stockage de données depuis et vers différentes sources
- ♦ Apprendre les stratégies avancées de nettoyage et de transformation des données

03

Compétences

Le programme ne transmettra pas seulement des connaissances techniques, mais se concentrera également à cultiver des compétences qui transformeront les diplômés en professionnels exceptionnels dans le domaine du développement de Softwares. Ainsi, la capacité à développer des solutions innovantes et efficaces sera couverte, grâce à une maîtrise experte de Python, qui va du traitement avancé des données à la création d'applications web et mobiles de haute performance. En outre, les informaticiens seront dotés de compétences spécialisées en Programmation Orientée Objet (POO), en conception d'interface utilisateur et en expérience utilisateur (UI/UX), ainsi qu'en analyse de données avancée avec des bibliothèques telles que NumPy et Pandas.

“

Le développement de compétences non techniques essentielles telles que la pensée critique, la résolution de problèmes, la collaboration en équipe et l'adaptabilité sera encouragé”



Compétences générales

- ♦ Développer des compétences pratiques en Programmation
- ♦ Se doter de compétences avancées en matière de contrôle des flux de programmes
- ♦ Gérer la conception et la mise en œuvre de Programmation Orienté Objet
- ♦ Acquérir des compétences dans le développement complet d'applications Python
- ♦ Maîtriser la conception et la gestion d'applications web et mobiles
- ♦ Maîtriser la conception de l'interface utilisateur et de l'expérience utilisateur à l'aide de Python
- ♦ Développer des compétences en gestion et analyse de données avec Python
- ♦ Acquérir des compétences en matière de manipulation avancée de fichiers et de modélisation en Python
- ♦ Développer des compétences dans le traitement avancé des données avec NumPy et Pandas
- ♦ Approfondir la gestion avancée des données avec NumPy et Pandas

“

Vous vous confronterez à des défis concrets et vous vous démarquerez sur un marché de l'emploi compétitif et en constante évolution. Optez pour la TECH Euromed University!”





Compétences spécifiques

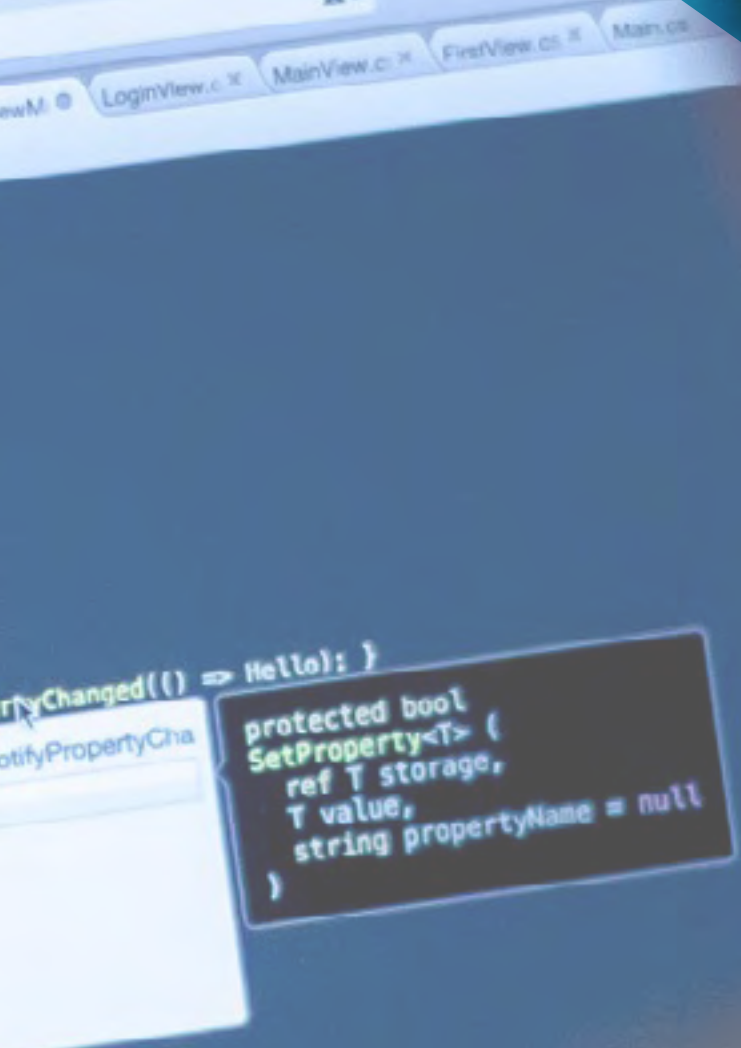
- ♦ Être formé à la gestion avancée des données en Python
- ♦ Maîtriser l'utilisation avancée des fonctions en Python
- ♦ Mettre en œuvre des concepts avancés de Programmation Orientée Objet (POO) tels que les classes abstraites et les exceptions personnalisées
- ♦ Contrôler les tests et le Debugging
- ♦ Développer des API et des services web
- ♦ Maîtriser l'utilisation des outils de conception UI/UX avec Python
- ♦ Utiliser les bibliothèques de données essentielles de Python
- ♦ Appliquer différentes structures de données de manière pratique
- ♦ Utiliser Pandas pour le traitement de données structurées
- ♦ Promouvoir des compétences dans l'analyse et la manipulation de séries temporelles et de données complexes

04

Direction de la formation

Les professeurs de ce Mastère Spécialisé sont des experts du secteur, dotés d'une vaste expérience pratique dans le développement de Logiciels et d'un palmarès de projets réussis. En mariant la théorie à l'application pratique, ces mentors ne se contenteront pas de transmettre des connaissances avancées sur Python, mais partageront également leur expérience du monde réel, offrant aux étudiants un aperçu précieux des meilleures pratiques et des tendances actuelles dans l'industrie. En outre, son approche pédagogique se distinguera par son engagement à fournir aux diplômés la formation technique la plus actualisée pour exceller dans le domaine dynamique du développement de Logiciels.





“

Vous réussirez avec les meilleurs et acquerez les connaissances et les compétences dont vous avez besoin pour vous lancer dans l'industrie du développement Python”

Direction



M. Matos Rodríguez, Dionis

- ♦ Data Engineer chez Wide Agency Sodexo
- ♦ Data Consultant chez Tokiota
- ♦ Data Engineer chez Devoteam
- ♦ BI Developer chez Ibermática
- ♦ Applications Engineer chez Johnson Controls
- ♦ Database Developer à Suncapital España
- ♦ Senior Web Developer chez Deadlock Solutions
- ♦ QA Analyst chez Metaconcept
- ♦ Master en Big Data & Analytics, EAE Business School
- ♦ Master en Analyse et Conception de Systèmes
- ♦ Licence en Génie Informatique de l'Université APEC

Professeurs

M. Villar Valor, Javier

- ◆ Directeur et Partenaire Fondateur d'Impulsa2
- ◆ Directeur des Opérations (COO) à Summa Insurance Brokers
- ◆ Directeur de la Transformation et de l'Excellence Opérationnelle chez Johnson Controls
- ◆ Master en Coaching Professionnelle
- ◆ Executive MBA de l'Emlyon Business School, France
- ◆ Master en Gestion de Qualité par EOI
- ◆ Ingénieur en Informatique chez l'Université Acción Pro-Education et Culture (UNAPEC)

M. Gil Contreras, Armando

- ◆ Lead Big Data Scientist à Jhonson Controls
- ◆ Data Scientist-Big Data chez Opensistemas S.A
- ◆ Auditeur du Fonds pour la Créativité et la Technologie S.A. (CYTSA)
- ◆ Auditeur du secteur public chez PricewaterhouseCoopers Auditors
- ◆ Master en Data Science au Centro Universitario de Tecnología y Arte
- ◆ Master MBA en Relations et Commerce International au Centro de Estudios Financieros (CEF)
- ◆ Licence en Économie de l'Institut Technologique de Saint-Domingue

Mme Gil Contreras, Milagros

- ◆ Content Creator en MPCTech LLC
- ◆ Gestion de projets
- ◆ Freelance IT Writer
- ◆ MBA de l'Université Complutense de Madrid
- ◆ Licence/Diplôme en Administration des Affaires de l'Institut Technologique de Saint-Domingue

Mme Delgado Feliz, Benedit

- ◆ Assistante Administratif et Opérateur de Surveillance Electronique à la Dirección Nacional de Control de Drogas (DNCD)
- ◆ Service Clientèle en Cáceres y Equipos
- ◆ Réclamations et Service à la Clientèle chez Express Parcel Services (EPS)
- ◆ Spécialiste de Microsoft Office à la École Nationale d'Informatique
- ◆ Communicatrice Social de l'Université Catholique de Santo Domingo



Profitez de l'occasion pour vous informer sur les derniers développements dans ce domaine afin de les appliquer à votre pratique quotidienne"

05

Structure et contenu

Le programme d'études a été méticuleusement conçu pour offrir une expérience complète aux diplômés. D'une plongée profonde dans la syntaxe et les fonctionnalités avancées de Python, à la spécialisation dans la Programmation Orientée Objet (POO), la conception d'applications web et mobiles, et la manipulation experte de données avec des bibliothèques telles que NumPy et Pandas, chaque module sera soigneusement structuré pour fournir des connaissances solides et des compétences pratiques. En outre, des sujets cruciaux tels que la conception de l'interface utilisateur et de l'expérience utilisateur (UI/UX), le traitement avancé des données et l'optimisation des performances et du stockage seront explorés.





“

Non seulement vous acquerez une maîtrise approfondie de Python, mais vous vous préparerez également à relever avec confiance les défis dynamiques de l'industrie technologique”

Module 1. Programmation en Python

- 1.1. Création et exécution de programmes Python
 - 1.1.1. Configurations de l'environnement de développement
 - 1.1.2. Exécution de scripts en Python
 - 1.1.3. Outils de Développement Intégré (IDE)
- 1.2. Les données en Python
 - 1.2.1. Types primitifs (int, float, str)
 - 1.2.2. Conversion et casting des types de données en Python
 - 1.2.3. Immutabilité et stockage des données en Python
- 1.3. Références à des objets en Python
 - 1.3.1. Références en mémoire
 - 1.3.2. Identité vs. Égalité
 - 1.3.3. Gestion des références et collecte des déchets
- 1.4. Données de collecte en Python
 - 1.4.1. Listes et opérations courantes
 - 1.4.2. Les tuples et leur immutabilité
 - 1.4.3. Dictionnaires et accès aux données
- 1.5. Opérations logiques de en Python
 - 1.5.1. Opérateurs booléens
 - 1.5.2. Expressions conditionnelles
 - 1.5.3. Short-Circuit Evaluation
- 1.6. Opérateurs arithmétiques en Python
 - 1.6.1. Opérations arithmétiques en Python
 - 1.6.2. Opérateurs de division
 - 1.6.3. Précédence et associativité
- 1.7. Entrée/sortie en Python
 - 1.7.1. Lire des données à partir de l'entrée standard
 - 1.7.2. Écrire des données sur la sortie standard
 - 1.7.3. Traitement des fichiers
- 1.8. Créer et appeler des fonctions Python
 - 1.8.1. Syntaxe des fonctions
 - 1.8.2. Paramètres et arguments
 - 1.8.3. Valeurs de retour et fonctions anonymes



- 1.9. Utiliser des strings en Python
 - 1.9.1. Manipulation et formatage des strings
 - 1.9.2. Méthodes courantes pour les strings
 - 1.9.3. Interpolation et F-strings
- 1.10. Gestion des erreurs et des exceptions en Python
 - 1.10.1. Types courants d'exceptions
 - 1.10.2. Les blocs try-except
 - 1.10.3. Création des exceptions personnalisés

Module 2. Données avancées et contrôle de flux avec Python

- 2.1. Identificateurs et mots-clés en Python
 - 2.1.1. Règles pour les noms de variables
 - 2.1.2. Mots réservés en Python
 - 2.1.3. Conventions de nommage
- 2.2. Types intégraux et booléens en Python
 - 2.2.1. Les types intégrales
 - 2.2.2. Opérations spécifiques des booléens
 - 2.2.3. Conversions et représentations
- 2.3. Types à virgule flottante et nombres complexes en Python
 - 2.3.1. Précision et représentation
 - 2.3.2. Opérations en virgule flottante
 - 2.3.3. Utilisation de nombres complexes dans les calculs
- 2.4. Formatage des string et encodages en Python
 - 2.4.1. Méthodes avancées de formatage
 - 2.4.2. Encodages Unicode et UTF-8
 - 2.4.3. Travailler avec des caractères spéciaux
- 2.5. Collections: Tuples, Listes et Dictionnaires en Python
 - 2.5.1. Comparaison et contraste entre les types
 - 2.5.2. Méthodes spécifiques aux types
 - 2.5.3. Efficacité et sélection du type approprié
- 2.6. Sets y Frozen Sets en Python
 - 2.6.1. Création et opérations sur les Sets
 - 2.6.2. Frozen Sets
 - 2.6.3. Applications pratiques et performances

- 2.7. Itérer et copier des collections en Python
 - 2.7.1. Boucles for et compréhensions de listes
 - 2.7.2. Copie superficielle vs. Copie profonde
 - 2.7.3. Itérateurs et générateurs
- 2.8. Utiliser les fonctions Lambda en Python
 - 2.8.1. Syntaxe et création des fonctions Lambda
 - 2.8.2. Applications dans les filtres et les cartes
 - 2.8.3. Limites et bonnes pratiques
- 2.9. Structures de contrôle: Conditionnelles et boucles en Python
 - 2.9.1. Structures if-else et elif
 - 2.9.2. Boucles while et for
 - 2.9.3. Contrôle de flux avec break, continue et else
- 2.10. Fonctions et méthodes avancées de Python
 - 2.10.1. Fonctions récursives
 - 2.10.2. Fonctions d'ordre supérieur
 - 2.10.3. Décorateurs de fonctions

Module 3. Programmation Orientée Objet (POO) en Python

- 3.1. Programmation Orientée Objet (POO) en Python
 - 3.1.1. Classes et objets
 - 3.1.2. Encapsulation et abstraction
 - 3.1.3. Programmation Orientée Objet (POO) en Python
- 3.2. Création de classes et d'objets en Python
 - 3.2.1. Les classes dans la POO Python
 - 3.2.2. Méthodes d'instanciation et d'initialisation
 - 3.2.3. Attributs et méthodes
- 3.3. Attributs et méthodes Python
 - 3.3.1. Attributs d'instance vs. Classe
 - 3.3.2. Méthodes d'instance, de classe et statiques
 - 3.3.3. Encapsulation et dissimulation d'informations
- 3.4. Héritage et polymorphisme en Python
 - 3.4.1. Héritage simple et multiple
 - 3.4.2. Écrasement et extension de méthode
 - 3.4.3. Polymorphisme et Duck Typing

- 3.5. Propriétés et accès aux attributs en Python
 - 3.5.1. Getters et Setters
 - 3.5.2. Décorateur @property
 - 3.5.3. Contrôle d'accès et validation
 - 3.6. Classes et collections personnalisées en Python
 - 3.6.1. Créer des types de collections
 - 3.6.2. Méthodes spéciales (__len__, __getitem__,)
 - 3.6.3. Itérateurs personnalisés
 - 3.7. Agrégation et composition dans les classes Python
 - 3.7.1. Relations entre classes
 - 3.7.2. Agrégation vs. Composition
 - 3.7.3. Gestion du cycle de vie des objets
 - 3.8. Utilisation des décorateurs dans les classes Python
 - 3.8.1. Décorateurs pour les méthodes
 - 3.8.2. Décorateurs de classes
 - 3.8.3. Applications et cas d'utilisation
 - 3.9. Classes abstraites et méthodes en Python
 - 3.9.1. Classes abstraites
 - 3.9.2. Méthodes abstraites et implémentation
 - 3.9.3. Utilisation de l'ABC (Abstract Base Class)
 - 3.10. Exceptions et gestion des erreurs de la POO en Python
 - 3.10.1. Exceptions personnalisées dans les classes
 - 3.10.2. Gestion des exceptions dans les méthodes
 - 3.10.3. Bonnes pratiques en matière d'exceptions et de POO
- Module 4. Développement d'applications en Python**
- 4.1. Architecture d'application en Python
 - 4.1.1. Conception du Software
 - 4.1.2. Modèles architecturaux communs
 - 4.1.3. Évaluation des exigences et des besoins
 - 4.2. Conception et modélisation d'applications Python
 - 4.2.1. Utilisation de l'UML et des diagrammes
 - 4.2.2. Modélisation des données et flux d'information
 - 4.2.3. Principes SOLID et conception modulaire
 - 4.3. Gestion des dépendances et des bibliothèques en Python
 - 4.3.1. Gestion des paquets avec Pip
 - 4.3.2. Utilisation d'environnements virtuels
 - 4.3.3. Résoudre les conflits de dépendances
 - 4.4. Modèles de conception dans le développement Python
 - 4.4.1. Modèles créatifs, structurels et de comportement
 - 4.4.2. Application pratique des patrons
 - 4.4.3. Refactorisation et patrons
 - 4.5. Test et Debugging des applications en Python
 - 4.5.1. Stratégies de Test (Unitaire, Intégration)
 - 4.5.2. Utilisation des Frameworks de test
 - 4.5.3. Techniques et outils de Débogage
 - 4.6. Sécurité et authentification en Python
 - 4.6.1. Sécurité des applications
 - 4.6.2. Mise en œuvre de l'authentification et de l'autorisation
 - 4.6.3. Prévention des vulnérabilités
 - 4.7. Optimisation et performance des applications Python
 - 4.7.1. Analyse des performances
 - 4.7.2. Techniques d'optimisation du code
 - 4.7.3. Gestion efficace des ressources et des données
 - 4.8. Déploiement et distribution d'applications Python
 - 4.8.1. Stratégies de déploiement
 - 4.8.2. Utilisation de conteneurs et d'orchestrateurs
 - 4.8.3. Distribution et mises à jour continues
 - 4.9. Maintenance et mises à jour en Python
 - 4.9.1. Gestion du cycle de vie des logiciels
 - 4.9.2. Stratégies de maintenance et de refactorisation
 - 4.9.3. Mises à jour et migration des systèmes
 - 4.10. Documentation et support technique en Python
 - 4.10.1. Créer une documentation efficace
 - 4.10.2. Outils de documentation
 - 4.10.3. Stratégies d'assistance et de communication avec les utilisateurs

Module 5. Développement web et mobile avec Python

- 5.1. Développement web avec Python
 - 5.1.1. Structure et composants d'un site web
 - 5.1.2. Technologies dans le développement web
 - 5.1.3. Tendances en matière de développement web
- 5.2. Frameworks web populaires avec Python
 - 5.2.1. Django, Flask et autres options
 - 5.2.2. Comparaison et sélection des Frameworks
 - 5.2.3. Intégration avec Frontend
- 5.3. Développement de Frontend: HTML, CSS et JavaScript avec Python
 - 5.3.1. HTML et CSS
 - 5.3.2. JavaScript et manipulation du DOM
 - 5.3.3. Frameworks et bibliothèques Frontend
- 5.4. Backend et bases de données avec Python
 - 5.4.1. Développement d'un backend avec Python
 - 5.4.2. Gestion de bases de données relationnelles et non relationnelles
 - 5.4.3. Intégration Backend-Frontend
- 5.5. API et services web avec Python
 - 5.5.1. Conception d'API RESTful
 - 5.5.2. Implémenter et documenter les API
 - 5.5.3. Consommation et sécurité des API
- 5.6. Développement mobile avec Python
 - 5.6.1. Plateformes de développement mobile (Natives, Hybrides)
 - 5.6.2. Outils et environnements de développement
 - 5.6.3. Adaptation d'applications pour les appareils mobiles
- 5.7. Plateformes de développement mobile avec Python
 - 5.7.1. Android et IOS
 - 5.7.2. Frameworks pour un développement croisé
 - 5.7.3. Tests et Deployment sur les appareils mobiles
- 5.8. Design et UX dans les applications mobiles avec Python
 - 5.8.1. Conception d'interfaces mobiles
 - 5.8.2. Facilité d'utilisation et expérience utilisateur avec Python
 - 5.8.3. Outils de prototypage et de conception

- 5.9. Tests mobiles et débogage avec Python
 - 5.9.1. Stratégies de Tsting sur les appareils mobiles
 - 5.9.2. Outils de débogage et de surveillance
 - 5.9.3. Automatisation des tests
- 5.10. Publication dans les boutiques d'applications Python
 - 5.10.1. Processus de publication sur App Store et Google Play
 - 5.10.2. Conformité et politiques des applications
 - 5.10.3. Stratégies de marketing et de promotion

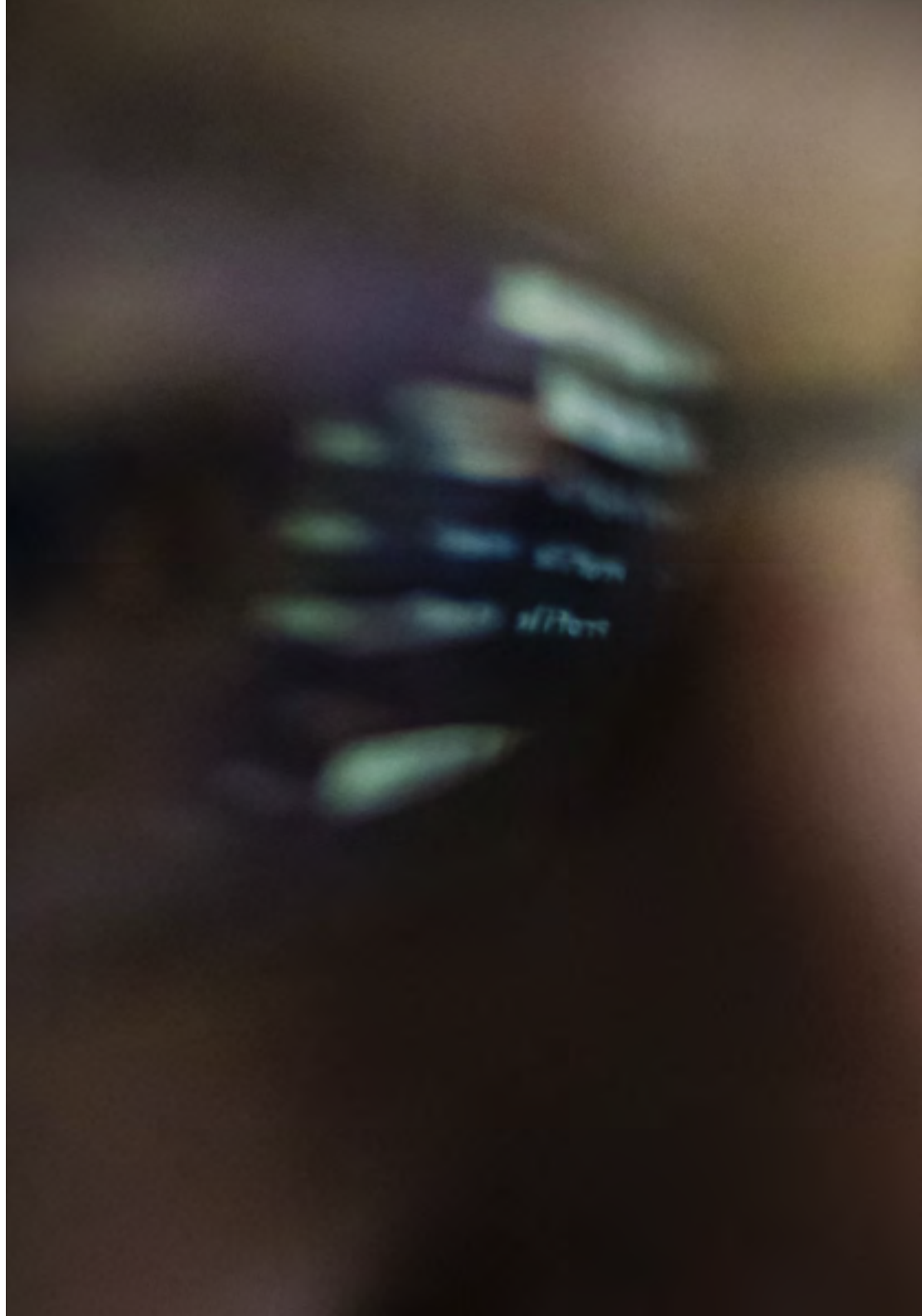
Module 6. Interface et expérience utilisateur avec Python


- 6.1. Conception de l'interface utilisateur avec Python
 - 6.1.1. Conception UI avec Python
 - 6.1.2. Interaction utilisateur-ordinateur avec Python
 - 6.1.3. Conception axée sur l'utilisateur avec Python
- 6.2. Outils de conception UI/UX avec Python
 - 6.2.1. Logiciel de conception et de prototypage
 - 6.2.2. Outils de collaboration et Feedback
 - 6.2.3. Intégration de la conception dans le processus de développement
- 6.3. Conception réactive et adaptative avec Python
 - 6.3.1. Techniques de conception réactive
 - 6.3.2. Adaptation à différents appareils et écrans
 - 6.3.3. Testing et assurance de qualité
- 6.4. Animations et transitions avec Python
 - 6.4.1. Créer des animations efficaces avec Python
 - 6.4.2. Outils et bibliothèques pour les animations
 - 6.4.3. Impact sur UX et la performance
- 6.5. Accessibilité et utilisabilité avec Python
 - 6.5.1. Accessibilité du web
 - 6.5.2. Outils et techniques d'évaluation
 - 6.5.3. Mise en œuvre des meilleures pratiques
- 6.6. Prototypage et Wireframes avec Python
 - 6.6.1. Création de Wireframes et de Mockups
 - 6.6.2. Outils de prototypage rapide
 - 6.6.3. Tests de utilisabilité et Feedback

- 6.7. Tests d'utilisabilité avec Python
 - 6.7.1. Méthodes et techniques de vérification de l'utilisabilité
 - 6.7.2. Analyse et améliorations basées sur les résultats
 - 6.7.3. Outils de test d'utilisabilité
- 6.8. Analyse du comportement des utilisateurs avec Python
 - 6.8.1. Techniques d'analyse et le Tracking
 - 6.8.2. Interprétation des données et métriques
 - 6.8.3. Amélioration continue sur la base des données
- 6.9. Amélioration basée sur le Feedback avec Python
 - 6.9.1. Gestion et analyse du Feedback
 - 6.9.2. Cycles de feedback et amélioration permanente
 - 6.9.3. Stratégies de mise en œuvre d'un changement efficace
- 6.10. Tendances futures de l'UI/UX avec Python
 - 6.10.1. Innovations et tendances émergentes
 - 6.10.2. Impact des nouvelles technologies sur l'UI/UX
 - 6.10.3. Préparer l'avenir du design

Module 7. Traitement des données et Big Data avec Python

- 7.1. Utilisation de Python en matière de données
 - 7.1.1. Python dans la science des données et l'analyse
 - 7.1.2. Bibliothèques essentielles pour les données
 - 7.1.3. Applications et exemples
- 7.2. Mise en place de l'environnement de développement Python
 - 7.2.1. Installation de Python et des outils
 - 7.2.2. Configuration des environnements virtuels
 - 7.2.3. Outils de Développement Intégré (IDE)
- 7.3. Variables, types de données et opérateurs Python
 - 7.3.1. Variables et types de données primitifs
 - 7.3.2. Structures de données
 - 7.3.3. Opérateurs arithmétiques et logiques
- 7.4. Contrôle de flux: Conditionnelles et boucles
 - 7.4.1. Structures de contrôle conditionnel (if, else, elif)
 - 7.4.2. Boucles (for, while) et contrôle de flux
 - 7.4.3. Compréhension de listes et expressions génératrices



- 
- 7.5. Fonctions et modularité avec Python
 - 7.5.1. Utilisation des fonctions
 - 7.5.2. Paramètres, arguments et valeurs de retour
 - 7.5.3. Modularité et réutilisation du code
 - 7.6. Gestion des erreurs et des exceptions avec Python
 - 7.6.1. Erreurs et exceptions
 - 7.6.2. Gestion des exceptions avec try-except
 - 7.6.3. Création des exceptions personnalisés
 - 7.7. Outil IPython
 - 7.7.1. Outil IPython
 - 7.7.2. Utilisation d'IPython pour l'analyse des données
 - 7.7.3. Différences avec l'interpréteur Python standard
 - 7.8. Jupyter Notebooks
 - 7.8.1. Jupyter Notebooks
 - 7.8.2. Utilisation des blocs-notes pour l'analyse des données
 - 7.8.3. Publication des blocs-notes *Jupyter*
 - 7.9. Les meilleures techniques de codage Python
 - 7.9.1. Style et conventions (PEP 8)
 - 7.9.2. Documentation et commentaires
 - 7.9.3. Stratégies de test et de débogage
 - 7.10. Ressources et communautés Python
 - 7.10.1. Ressources en ligne et documentation
 - 7.10.2. Communautés et forums
 - 7.10.3. Apprendre et mettre à jour en Python

Module 8. Structures de données et fonctions en Python

- 8.1. Les ensembles en Python
 - 8.1.1. Opérations et méthodes
 - 8.1.2. Différences et applications pratiques
 - 8.1.3. Itération et compréhension
- 8.2. Les dictionnaires et leur utilisation en Python
 - 8.2.1. Création et manipulation de dictionnaires
 - 8.2.2. Accès et gestion des données
 - 8.2.3. Patrons et techniques avancées

- 8.3. Compréhension de listes et de dictionnaires en Python
 - 8.3.1. Syntaxe et exemples
 - 8.3.2. Efficacité et lisibilité
 - 8.3.3. Applications pratiques
- 8.4. Fonctions sur les données en Python
 - 8.4.1. Créer des fonctions
 - 8.4.2. Portée et espace de noms
 - 8.4.3. Fonctions anonymes et Lambda
- 8.5. Arguments des fonctions et valeurs de retour en Python
 - 8.5.1. Arguments positionnels et nommés
 - 8.5.2. Valeurs de retour multiples
 - 8.5.3. Arguments variables et mots-clés
- 8.6. Fonctions Lambda et fonctions d'ordre supérieur en Python
 - 8.6.1. Utiliser les fonctions lambda
 - 8.6.2. Fonctions Map, Filter et Reduce
 - 8.6.3. Applications dans le traitement des données
- 8.7. Traitement des fichiers en Python
 - 8.7.1. Lire et écrire des fichiers
 - 8.7.2. Manipuler des fichiers binaires et des textes
 - 8.7.3. Bonnes pratiques et gestion des exceptions
- 8.8. Lire et écrire des fichiers texte et binaires en Python
 - 8.8.1. Formats de fichiers et encodage
 - 8.8.2. Gestion des fichiers volumineux
 - 8.8.3. Sérialisation et désérialisation (JSON, pickle)
- 8.9. Contextes et opérations sur les fichiers
 - 8.9.1. Utiliser le gestionnaire de contexte (avec)
 - 8.9.2. Techniques de traitement des fichiers
 - 8.9.3. Sécurité et gestion des erreurs
- 8.10. Bibliothèques de modélisation Python
 - 8.10.1. Scikit-learn
 - 8.10.2. TensorFlow
 - 8.10.3. Pytorch

Module 9. Traitement des données en Python avec NumPy et Pandas

- 9.1. Créer et manipuler des Arrays avec NumPy
 - 9.1.1. NumPy
 - 9.1.2. Opérations de base avec Arrays
 - 9.1.3. Manipulation et transformation de Arrays
- 9.2. Opérations vectorisées avec Arrays
 - 9.2.1. Vectorisation
 - 9.2.2. Fonctions universelles (ufunc)
 - 9.2.3. Efficacité et performance
- 9.3. Indexation et segmentation dans NumPy
 - 9.3.1. Accès aux éléments et Slicing
 - 9.3.2. Indexation avancée et booléenne
 - 9.3.3. Réorganisation et sélection
- 9.4. Séries Pandas et DataFrames
 - 9.4.1. Pandas
 - 9.4.2. Structures de données dans Pandas
 - 9.4.3. Manipulation des DataFrames
- 9.5. Indexation et sélection dans Pandas
 - 9.5.1. Accès aux données dans les séries et les DataFrames
 - 9.5.2. Méthodes de sélection et de filtrage
 - 9.5.3. Utilisation de loc et iloc
- 9.6. Opérations avec Pandas
 - 9.6.1. Opérations arithmétiques et alignement
 - 9.6.2. Fonctions d'agrégation et de statistiques
 - 9.6.3. Transformations et application de fonctions
- 9.7. Traiter des données incomplètes dans Pandas
 - 9.7.1. Détection et traitement des valeurs nulles
 - 9.7.2. Remplissage et suppression des données incomplètes
 - 9.7.3. Stratégies de traitement des données incomplètes
- 9.8. Fonctions et applications dans Pandas
 - 9.8.1. Concaténation et fusion de données
 - 9.8.2. Regroupement et agrégation (groupby)
 - 9.8.3. Pivot Tables y Crosstabs

- 9.9. Visualisation avec Matplotlib
 - 9.9.1. Matplotlib
 - 9.9.2. Création de graphiques et personnalisation
 - 9.9.3. Intégration avec Pandas
- 9.10. Personnalisation des graphiques dans Matplotlib
 - 9.10.1. Styles et paramètres
 - 9.10.2. Graphiques avancés (scatter, bar, etc.)
 - 9.10.3. Création de visualisations complexes

Module 10. Techniques avancées et applications pratiques avec NumPy et Pandas

- 10.1. Chargement de données à partir de différentes sources
 - 10.1.1. Importation à partir de CSV, Excel et de bases de données
 - 10.1.2. Lecture de données d'API et du web
 - 10.1.3. Stratégies de traitement des données volumineuses
- 10.2. Stockage de données en Python
 - 10.2.1. Exporter vers différents formats
 - 10.2.2. Efficacité du stockage
 - 10.2.3. Sécurité et confidentialité des données
- 10.3. Stratégies de nettoyage des données en Python
 - 10.3.1. Identifier et corriger les incohérences
 - 10.3.2. Normalisation et transformation des données
 - 10.3.3. Automatisation des processus de nettoyage
- 10.4. Transformation avancée des données dans Pandas
 - 10.4.1. Techniques de manipulation et de transformation
 - 10.4.2. Combiner et restructurer des DataFrames
 - 10.4.3. Utilisation des expressions régulières dans Pandas
- 10.5. Combiner des DataFrames dans Pandas
 - 10.5.1. Merge, Join et concaténation
 - 10.5.2. Gestion des conflits et des clés
 - 10.5.3. Stratégies de combinaison efficaces
- 10.6. Transformation avancée et pivotement des données dans Pandas
 - 10.6.1. Pivot et Melt
 - 10.6.2. Techniques de reshape et de transposition
 - 10.6.3. Applications dans l'analyse des données
- 10.7. Séries temporelles dans Pandas
 - 10.7.1. Traitement des dates et des heures
 - 10.7.2. Resampling et Window Functions
 - 10.7.3. Analyse des tendances et de la saisonnalité
- 10.8. Gestion avancée des index dans Pandas
 - 10.8.1. Indices multiniveaux et hiérarchiques
 - 10.8.2. Sélection et manipulation avancées
 - 10.8.3. Optimisation des requêtes
- 10.9. Stratégies d'optimisation des performances
 - 10.9.1. Amélioration de la vitesse et de l'efficacité
 - 10.9.2. Utilisation de Cython et Numba
 - 10.9.3. Parallélisme et traitement distribué
- 10.10. Projets pratiques de manipulation de données
 - 1.10.1. Développement d'exemples concrets
 - 10.10.2. Intégration de techniques Python
 - 1.10.3. Stratégies pour résoudre des problèmes de données complexes



Ce programme innovant est adapté pour vous permettre de rester au courant des dernières tendances et technologies, vous assurant ainsi d'être à la pointe de l'innovation en matière de développement de software"

06

Méthodologie d'étude

TECH Euromed University est la première au monde à combiner la méthodologie des **case studies** avec **Relearning**, un système d'apprentissage 100% en ligne basé sur la répétition guidée.

Cette stratégie d'enseignement innovante est conçue pour offrir aux professionnels la possibilité d'actualiser leurs connaissances et de développer leurs compétences de manière intensive et rigoureuse. Un modèle d'apprentissage qui place l'étudiant au centre du processus académique et lui donne le rôle principal, en s'adaptant à ses besoins et en laissant de côté les méthodologies plus conventionnelles.



“

*TECH Euromed University vous prépare
à relever de nouveaux défis dans des
environnements incertains et à réussir
votre carrière”*

L'étudiant: la priorité de tous les programmes de TECH Euromed University

Dans la méthodologie d'étude de TECH Euromed University, l'étudiant est le protagoniste absolu.

Les outils pédagogiques de chaque programme ont été sélectionnés en tenant compte des exigences de temps, de disponibilité et de rigueur académique que demandent les étudiants d'aujourd'hui et les emplois les plus compétitifs du marché.

Avec le modèle éducatif asynchrone de TECH Euromed University, c'est l'étudiant qui choisit le temps qu'il consacre à l'étude, la manière dont il décide d'établir ses routines et tout cela dans le confort de l'appareil électronique de son choix. L'étudiant n'a pas besoin d'assister à des cours en direct, auxquels il ne peut souvent pas assister. Les activités d'apprentissage se dérouleront à votre convenance. Vous pouvez toujours décider quand et où étudier.

“

À TECH Euromed University, vous n'aurez PAS de cours en direct (auxquelles vous ne pourrez jamais assister)”



Les programmes d'études les plus complets au niveau international

TECH Euromed University se caractérise par l'offre des itinéraires académiques les plus complets dans l'environnement universitaire. Cette exhaustivité est obtenue grâce à la création de programmes d'études qui couvrent non seulement les connaissances essentielles, mais aussi les dernières innovations dans chaque domaine.

Grâce à une mise à jour constante, ces programmes permettent aux étudiants de suivre les évolutions du marché et d'acquérir les compétences les plus appréciées par les employeurs. Ainsi, les diplômés de TECH Euromed University reçoivent une préparation complète qui leur donne un avantage concurrentiel significatif pour progresser dans leur carrière.

De plus, ils peuvent le faire à partir de n'importe quel appareil, PC, tablette ou smartphone.

“

Le modèle de TECH Euromed University est asynchrone, de sorte que vous pouvez étudier sur votre PC, votre tablette ou votre smartphone où vous voulez, quand vous voulez et aussi longtemps que vous le voulez”

Case studies ou Méthode des cas

La méthode des cas est le système d'apprentissage le plus utilisé par les meilleures écoles de commerce du monde. Développée en 1912 pour que les étudiants en Droit n'apprennent pas seulement le droit sur la base d'un contenu théorique, sa fonction était également de leur présenter des situations réelles et complexes. De cette manière, ils pouvaient prendre des décisions en connaissance de cause et porter des jugements de valeur sur la manière de les résoudre. Elle a été établie comme méthode d'enseignement standard à Harvard en 1924.

Avec ce modèle d'enseignement, ce sont les étudiants eux-mêmes qui construisent leurs compétences professionnelles grâce à des stratégies telles que *Learning by doing* ou le *Design Thinking*, utilisées par d'autres institutions renommées telles que Yale ou Stanford.

Cette méthode orientée vers l'action sera appliquée tout au long du parcours académique de l'étudiant avec TECH Euromed University. Vous serez ainsi confronté à de multiples situations de la vie réelle et devrez intégrer des connaissances, faire des recherches, argumenter et défendre vos idées et vos décisions. Il s'agissait de répondre à la question de savoir comment ils agiraient lorsqu'ils seraient confrontés à des événements spécifiques complexes dans le cadre de leur travail quotidien.



Méthode Relearning

À TECH Euromed University, les *case studies* sont complétées par la meilleure méthode d'enseignement 100% en ligne: le *Relearning*.

Cette méthode s'écarte des techniques d'enseignement traditionnelles pour placer l'apprenant au centre de l'équation, en lui fournissant le meilleur contenu sous différents formats. De cette façon, il est en mesure de revoir et de répéter les concepts clés de chaque matière et d'apprendre à les appliquer dans un environnement réel.

Dans le même ordre d'idées, et selon de multiples recherches scientifiques, la répétition est le meilleur moyen d'apprendre. C'est pourquoi TECH Euromed University propose entre 8 et 16 répétitions de chaque concept clé au sein d'une même leçon, présentées d'une manière différente, afin de garantir que les connaissances sont pleinement intégrées au cours du processus d'étude.

Le Relearning vous permettra d'apprendre plus facilement et de manière plus productive tout en développant un esprit critique, en défendant des arguments et en contrastant des opinions: une équation directe vers le succès.



Un Campus Virtuel 100% en ligne avec les meilleures ressources didactiques

Pour appliquer efficacement sa méthodologie, TECH Euromed University se concentre à fournir aux diplômés du matériel pédagogique sous différents formats: textes, vidéos interactives, illustrations et cartes de connaissances, entre autres. Tous ces supports sont conçus par des enseignants qualifiés qui axent leur travail sur la combinaison de cas réels avec la résolution de situations complexes par la simulation, l'étude de contextes appliqués à chaque carrière professionnelle et l'apprentissage basé sur la répétition, par le biais d'audios, de présentations, d'animations, d'images, etc.

Les dernières données scientifiques dans le domaine des Neurosciences soulignent l'importance de prendre en compte le lieu et le contexte d'accès au contenu avant d'entamer un nouveau processus d'apprentissage. La possibilité d'ajuster ces variables de manière personnalisée aide les gens à se souvenir et à stocker les connaissances dans l'hippocampe pour une rétention à long terme. Il s'agit d'un modèle intitulé *Neurocognitive context-dependent e-learning* qui est sciemment appliqué dans le cadre de ce diplôme d'université.

D'autre part, toujours dans le but de favoriser au maximum les contacts entre mentors et mentorés, un large éventail de possibilités de communication est offert, en temps réel et en différé (messagerie interne, forums de discussion, service téléphonique, contact par courrier électronique avec le secrétariat technique, chat et vidéoconférence).

De même, ce Campus Virtuel très complet permettra aux étudiants TECH Euromed University d'organiser leurs horaires d'études en fonction de leurs disponibilités personnelles ou de leurs obligations professionnelles. De cette manière, ils auront un contrôle global des contenus académiques et de leurs outils didactiques, mis en fonction de leur mise à jour professionnelle accélérée.



Le mode d'étude en ligne de ce programme vous permettra d'organiser votre temps et votre rythme d'apprentissage, en l'adaptant à votre emploi du temps"

L'efficacité de la méthode est justifiée par quatre acquis fondamentaux:

1. Les étudiants qui suivent cette méthode parviennent non seulement à assimiler les concepts, mais aussi à développer leur capacité mentale au moyen d'exercices pour évaluer des situations réelles et appliquer leurs connaissances.
2. L'apprentissage est solidement traduit en compétences pratiques ce qui permet à l'étudiant de mieux s'intégrer dans le monde réel.
3. L'assimilation des idées et des concepts est rendue plus facile et plus efficace, grâce à l'utilisation de situations issues de la réalité.
4. Le sentiment d'efficacité de l'effort investi devient un stimulus très important pour les étudiants, qui se traduit par un plus grand intérêt pour l'apprentissage et une augmentation du temps passé à travailler sur le cours.

La méthodologie universitaire la mieux évaluée par ses étudiants

Les résultats de ce modèle académique innovant sont visibles dans les niveaux de satisfaction générale des diplômés de TECH Euromed University.

L'évaluation par les étudiants de la qualité de l'enseignement, de la qualité du matériel, de la structure du cours et des objectifs est excellente. Il n'est pas surprenant que l'institution soit devenue l'université la mieux évaluée par ses étudiants selon l'indice global score, obtenant une note de 4,9 sur 5.

Accédez aux contenus de l'étude depuis n'importe quel appareil disposant d'une connexion Internet (ordinateur, tablette, smartphone) grâce au fait que TECH Euromed University est à la pointe de la technologie et de l'enseignement.

Vous pourrez apprendre grâce aux avantages offerts par les environnements d'apprentissage simulés et à l'approche de l'apprentissage par observation: le Learning from an expert.



Ainsi, le meilleur matériel pédagogique, minutieusement préparé, sera disponible dans le cadre de ce programme:



Matériel didactique

Tous les contenus didactiques sont créés par les spécialistes qui enseignent les cours. Ils ont été conçus en exclusivité pour le programme afin que le développement didactique soit vraiment spécifique et concret.

Ces contenus sont ensuite appliqués au format audiovisuel afin de mettre en place notre mode de travail en ligne, avec les dernières techniques qui nous permettent de vous offrir une grande qualité dans chacune des pièces que nous mettrons à votre service.



Pratique des aptitudes et des compétences

Vous effectuerez des activités visant à développer des compétences et des aptitudes spécifiques dans chaque domaine. Pratiques et dynamiques permettant d'acquérir et de développer les compétences et les capacités qu'un spécialiste doit acquérir dans le cadre de la mondialisation dans laquelle nous vivons.



Résumés interactifs

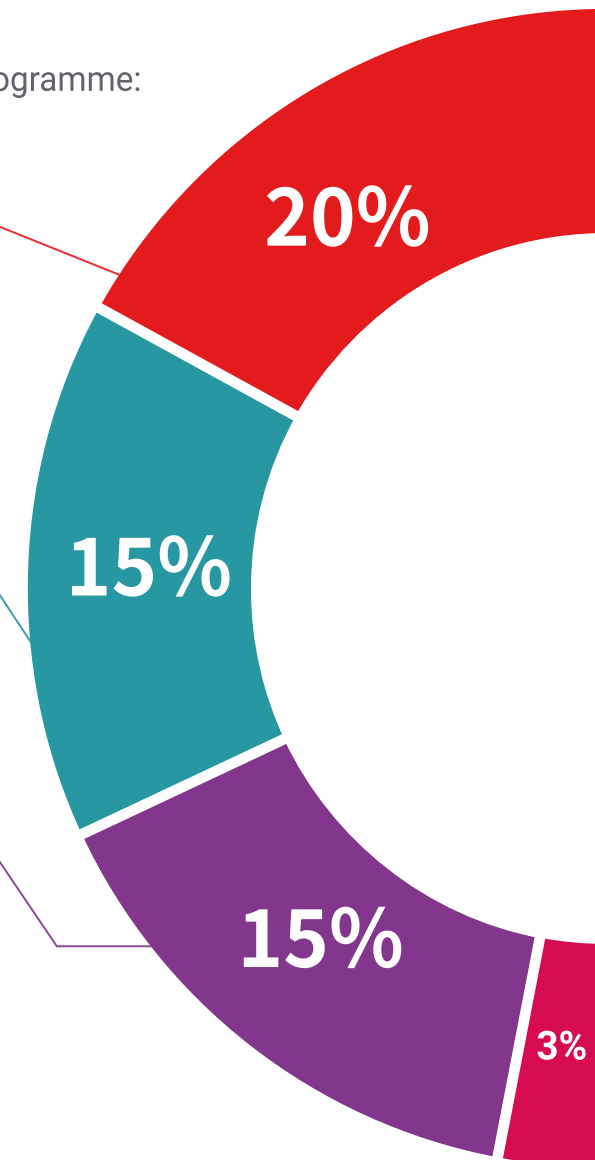
Nous présentons les contenus de manière attrayante et dynamique dans des dossiers multimédias qui incluent de l'audio, des vidéos, des images, des diagrammes et des cartes conceptuelles afin de consolider les connaissances.

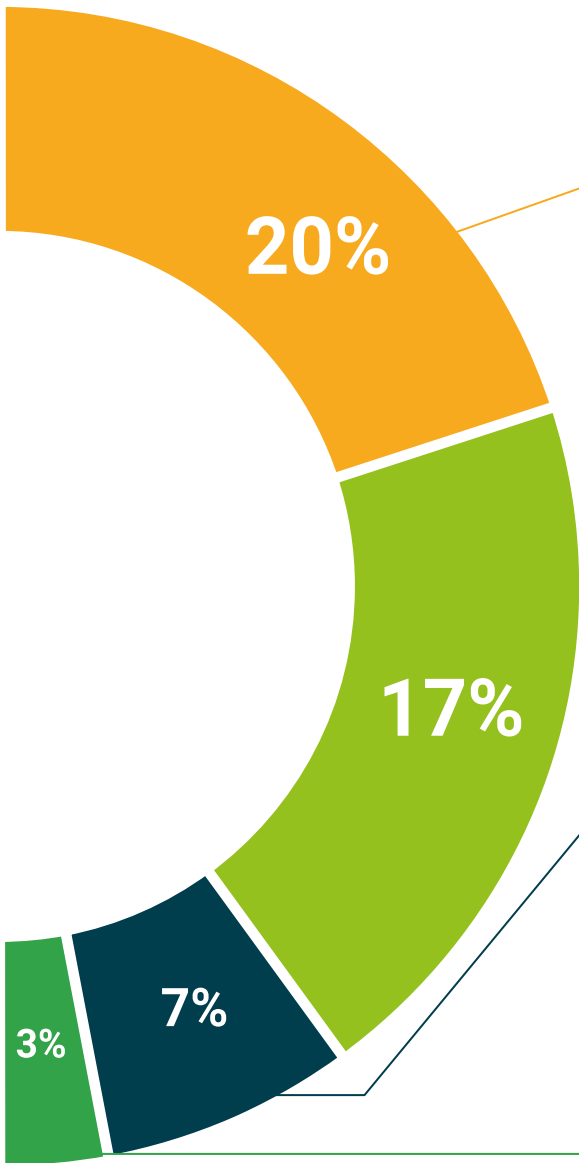
Ce système éducatif unique de présentation de contenu multimédia a été récompensé par Microsoft en tant que «European Success Story».



Lectures complémentaires

Articles récents, documents de consensus, guides internationaux, etc... Dans notre bibliothèque virtuelle, vous aurez accès à tout ce dont vous avez besoin pour compléter votre formation.





Case Studies

Vous réaliserez une sélection des meilleures case studies dans le domaine. Des cas présentés, analysés et encadrés par les meilleurs spécialistes internationaux.



Testing & Retesting

Nous évaluons et réévaluons périodiquement vos connaissances tout au long du programme. Nous le faisons sur 3 des 4 niveaux de la Pyramide de Miller.



Cours magistraux

Il existe des preuves scientifiques de l'utilité de l'observation par un tiers expert. La méthode Learning from an Expert permet au professionnel de renforcer ses connaissances ainsi que sa mémoire, puis lui permet d'avoir davantage confiance en lui concernant la prise de décisions difficiles.



Guides d'action rapide

TECH Euromed University propose les contenus les plus pertinents du programme sous forme de fiches de travail ou de guides d'action rapide. Un moyen synthétique, pratique et efficace pour vous permettre de progresser dans votre apprentissage.



07 Diplôme

Le Mastère Spécialisé en Développement en Python garantit, outre la formation la plus rigoureuse et la plus actualisée, l'accès à un diplôme de Mastère Spécialisé délivré par TECH Global University, et un autre par Euromed University of Fes.



“

*Terminez ce programme avec succès
et recevez votre diplôme sans avoir
à vous soucier des déplacements ou
des formalités administratives”*

Le programme du **Mastère Spécialisé en Développement en Python** est le programme le plus complet sur la scène académique actuelle. Après avoir obtenu leur diplôme, les étudiants recevront un diplôme d'université délivré par TECH Global University et un autre par Université Euromed de Fès.

Ces diplômes de formation continue et d'actualisation professionnelle de TECH Global University et d'Université Euromed de Fès garantissent l'acquisition de compétences dans le domaine de la connaissance, en accordant une grande valeur curriculaire à l'étudiant qui réussit les évaluations et accrédite le programme après l'avoir suivi dans son intégralité.

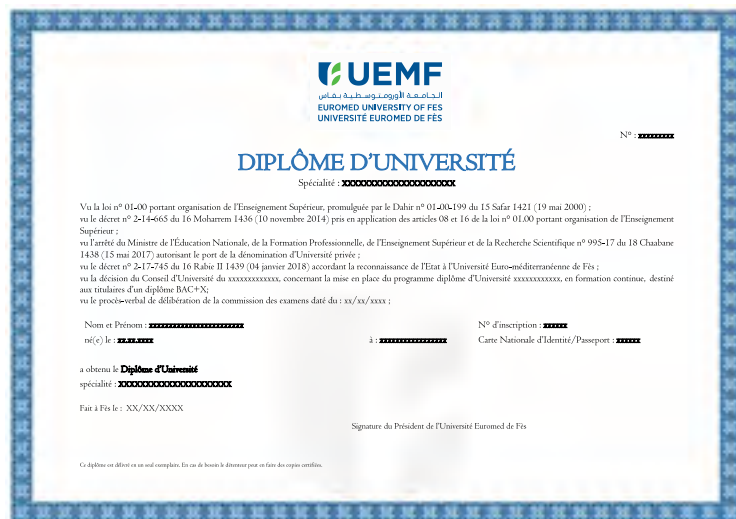
Ce double certificat, de la part de deux institutions universitaires de premier plan, représente une double récompense pour une formation complète et de qualité, assurant à l'étudiant l'obtention d'une certification reconnue au niveau national et international. Ce mérite académique vous positionnera comme un professionnel hautement qualifié, prêt à relever les défis et à répondre aux exigences de votre secteur professionnel.

Diplôme: **Mastère Spécialisé en Développement en Python**

Modalité: **en ligne**

Durée: **12 mois**

Accréditation: **60 ECTS**



*Si l'étudiant souhaite que son diplôme version papier possède l'Apostille de La Haye, TECH Euromed University fera les démarches nécessaires pour son obtention moyennant un coût supplémentaire.

future

santé confiance personnes

éducation information tuteurs

garantie accréditation enseignement

institutions technologie apprentissage

communauté engagement

service personnalisé innovation

connaissance présent qualité

en ligne formation

développement institutions

classe virtuelle langues

tech Euromed
University

Mastère Spécialisé Développement en Python

- » Modalité: en ligne
- » Durée: 12 mois
- » Qualification: TECH Euromed University
- » Accréditation: 60 ECTS
- » Horaire: à votre rythme
- » Examens: en ligne

Mastère Spécialisé

Développement en Python