

Mastère Spécialisé

Intelligence Artificielle et Ingénierie des Connaissances



Mastère Spécialisé Intelligence Artificielle et Ingénierie des Connaissances

- » Modalité: en ligne
- » Durée: 12 mois
- » Qualification: TECH Université Technologique
- » Intensité: 16h/semaine
- » Horaire: à votre rythme
- » Examens: en ligne

Accès au site: www.techtute.com/fr/informatique/master/master-intelligence-artificielle-ingenierie-connaissances

Sommaire

01

Présentation

page 4

02

Objectifs

page 8

03

Compétences

page 14

04

Structure et contenu

page 18

05

Méthodologie

page 30

06

Diplôme

page 38

01 Présentation

Le programme est conçu pour immerger les professionnels de l'ingénierie dans le monde passionnant de l'intelligence artificielle et de l'ingénierie de la connaissance. Grâce à une formation hautement compétente, vous serez en mesure de faire un pas solide dans ce domaine, en acquérant les compétences personnelles et professionnelles nécessaires pour exercer en tant qu'expert dans ce domaine. Un programme complet et efficace qui vous propulsera au plus haut niveau de compétence.





“

Devenez l'un des professionnels les plus demandés du moment. Formez-vous avec le Mastère Spécialisé en Intelligence Artificielle et Ingénierie des Connaissances avec ce programme très complet"

Les développements basés sur l'intelligence artificielle ont atteint de nombreuses applications dans le domaine de l'ingénierie. De l'automatisation de nombreuses procédures dans l'industrie et les entreprises, au contrôle des processus lui-même. Cela signifie que les professionnels de l'ingénierie doivent connaître et maîtriser le fonctionnement de ces techniques complexes.

Cette connaissance essentielle devient également la première étape pour accéder à la capacité de développement de ce type de technologie.

Tout au long de cette formation, un panorama réel de travail est proposé afin de pouvoir évaluer la pertinence de son application dans son propre projet, en évaluant ses indications réelles, la manière dont il est développé et les attentes que l'on peut avoir par rapport aux résultats.

Grâce à l'expérience, vous apprendrez à développer les compétences nécessaires pour progresser dans ce domaine de travail. Cet apprentissage, qui nécessite nécessairement de l'expérience, est concilié par l'e-learning et l'enseignement pratique, offrant une option unique pour donner à votre CV le coup de pouce que vous recherchez.



Rejoignez l'élite, avec cette formation pédagogique très efficace et ouvrez de nouvelles voies à votre avancement professionnel"

Ce **Mastère Spécialisé en Intelligence Artificielle et Ingénierie des Connaissances** contient le programme académique le plus complet et le plus actuel du marché. Les principales caractéristiques sont les suivantes:

- ◆ Dernières technologies en matière de softwares d'enseignement en ligne
- ◆ Système d'enseignement intensément visuel, soutenu par des contenus graphiques et schématiques faciles à assimiler et à comprendre
- ◆ Développement d'études de cas présentées par des experts actifs
- ◆ Systèmes vidéo interactifs de pointe
- ◆ Enseignement basé sur la télépratique
- ◆ Systèmes de mise à jour et de recyclage continus
- ◆ Apprentissage auto-adaptatif: compatibilité totale avec d'autres professions
- ◆ Exercices pratiques pour l'auto-évaluation et la vérification de l'apprentissage
- ◆ Des groupes de soutien et synergies éducatives: questions à l'expert, forums de discussion et de connaissances
- ◆ Communication avec l'enseignant et travail de réflexion individuel
- ◆ Les contenus sont disponibles à partir de tout appareil fixe ou portable doté d'une connexion internet
- ◆ Banques de documents justificatifs disponibles en permanence, y compris après le programme



Un Mastère Spécialisé en Intelligence Artificielle et Ingénierie des Connaissances avec la solvabilité d'un professionnel de haut niveau"

Le corps enseignant est composé de professionnels issus de différents domaines liés à cette spécialité. De cette façon, l'objectif visé par la mise à jour pédagogique est atteint. Une équipe multidisciplinaire de professionnels qualifiés et expérimentés dans différents environnements, qui développeront les connaissances théoriques de manière efficace, mais surtout, ils mettront à votre service les connaissances pratiques issues de leur propre expérience: une des qualités différentielles de cette formation.

Cette maîtrise du sujet est complétée par l'efficacité de la conception méthodologique. Développée par une équipe pluridisciplinaire d'experts en e-learning, la méthode intègre les dernières avancées en matière de technologie éducative. Ainsi, vous pourrez étudier avec une gamme d'outils multimédias pratiques et polyvalents qui vous donneront l'opérabilité dont vous avez besoin dans votre formation.

La conception de ce programme est basée sur l'apprentissage par les problèmes: une approche qui conçoit l'apprentissage comme un processus éminemment pratique. Pour y parvenir à distance, nous utilisons le téléenseignement. À l'aide d'un système vidéo interactif innovant et de la méthode Learning From an Expert, vous pouvez acquérir les connaissances comme si vous étiez confronté au scénario que vous êtes en train d'apprendre en ce moment. Un concept qui permet d'intégrer et de fixer votre apprentissage de manière plus réaliste et permanente.

Avec une conception méthodologique basée sur des techniques d'enseignement éprouvées, ce master innovant en intelligence artificielle et ingénierie des connaissances vous fera passer par différentes approches pédagogiques pour vous permettre d'apprendre de manière dynamique et efficace.

Notre concept innovant de télépratique vous donnera l'opportunité d'apprendre à travers une expérience immersive, ce qui vous permettra une intégration plus rapide et une vision beaucoup plus réaliste des contenus: "learning from an expert".



02 Objectifs

Notre objectif est de former des professionnels hautement qualifiés pour une expérience professionnelle. Cet objectif est complété, de manière globale, par la promotion du développement humain qui jette les bases d'une société meilleure. Cet objectif est atteint en aidant les professionnels à accéder à un niveau de compétence et de contrôle beaucoup plus élevé. Un objectif qui, en quelques mois seulement, peut être atteint grâce à un entraînement de haute intensité et de précision.



“

Si votre objectif est de réorienter vos capacités vers de nouvelles voies de réussite et de développement, ce programme est fait pour vous: une formation qui aspire à l'excellence"

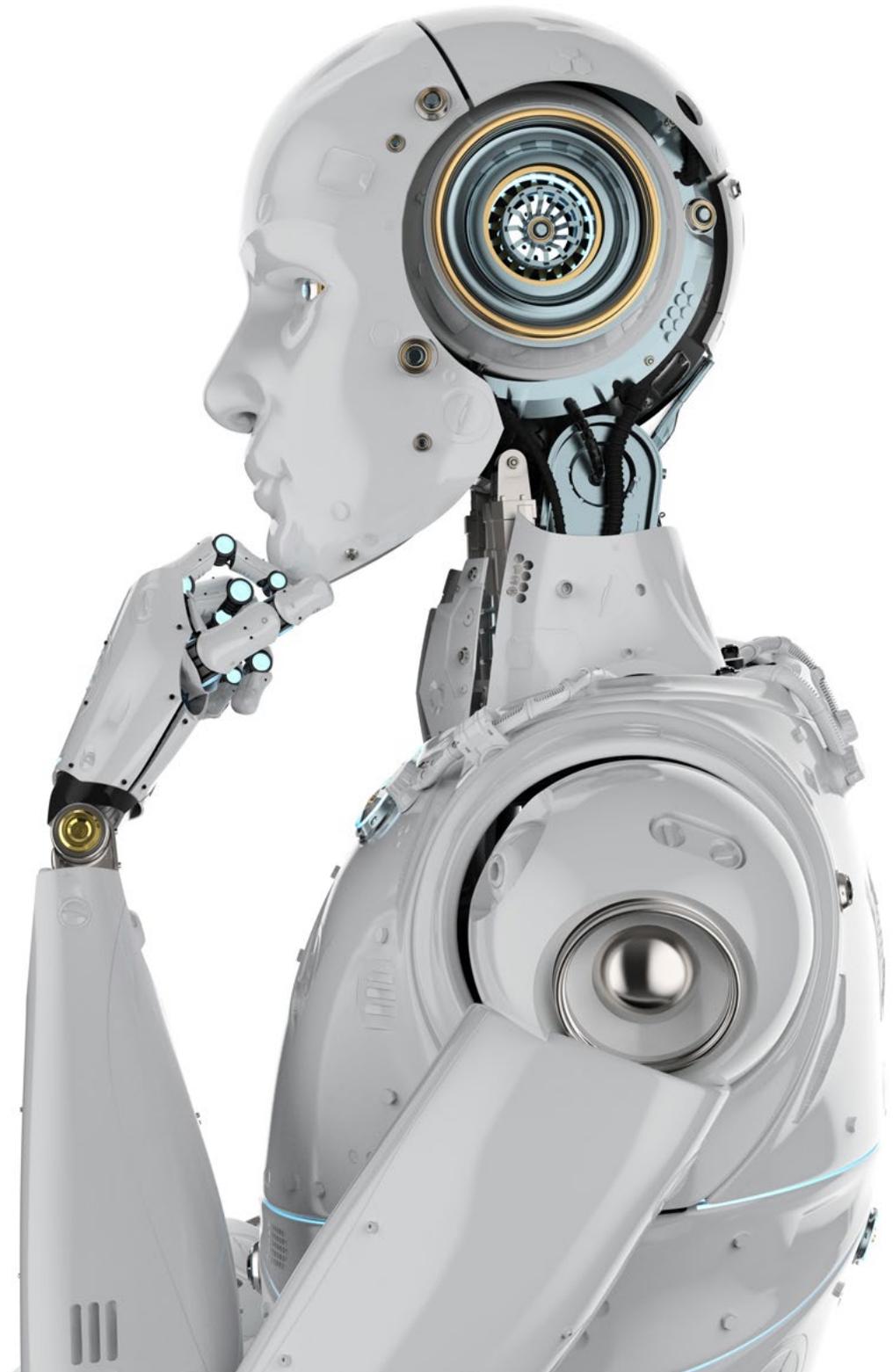


Objectifs généraux

- ◆ Former scientifiquement et technologiquement à la pratique de l'ingénierie informatique
- ◆ Obtenir des connaissances complètes dans le domaine de l'informatique
- ◆ Obtenir des connaissances complètes dans le domaine de la structure informatique
- ◆ Acquérir les connaissances nécessaires en matière de génie logiciel
- ◆ Passer en revue les fondements mathématiques, statistiques et physiques essentiels à cette matière

“

Ne manquez pas l'occasion de vous former concernant les derniers développements dans l'utilisation des antihémorragiques afin de les intégrer dans votre pratique médicale quotidienne”





Objectifs spécifiques

Module 1. Principes fondamentaux de la programmation

- ◆ Comprendre la structure de base d'un ordinateur, les logiciels et les langages de programmation à usage général
- ◆ Apprenez à concevoir et à interpréter des algorithmes, qui constituent la base nécessaire au développement de logiciels
- ◆ Comprendre les éléments essentiels d'un programme informatique, tels que les différents types de données, les opérateurs, les expressions, les instructions, les entrées/sorties et les instructions de contrôle
- ◆ Comprendre les différentes structures de données disponibles dans les langages de programmation à usage général, tant statiques que dynamiques, et acquérir les connaissances essentielles de la manipulation des fichiers
- ◆ Comprendre les différentes techniques de test des logiciels et l'importance de générer une bonne documentation en même temps qu'un bon code source
- ◆ Apprenez les bases du langage de programmation C++, l'un des langages de programmation les plus utilisés dans le monde

Module 2. Structure des données

- ◆ Apprenez les bases de la programmation C++, notamment les classes, les variables, les expressions conditionnelles et les objets
- ◆ Comprendre les types de données abstraits, les types de structures de données linéaires, les structures de données hiérarchiques simples et complexes et leur mise en œuvre en C++
- ◆ Comprendre le fonctionnement des structures de données avancées autres que les structures habituelles
- ◆ Comprendre la théorie et la pratique liées à l'utilisation des monticules de priorité et des files d'attente de priorité
- ◆ Apprenez comment les tables de Hash, en tant que types de données et fonctions abstraites
- ◆ Comprendre la théorie des graphes, ainsi que les algorithmes et concepts avancés des graphes

Module 3. Algorithme et complexité

- ◆ Apprenez les principales stratégies de conception d'algorithmes, ainsi que les différentes méthodes et mesures de calcul d'algorithmes
- ◆ Connaître les principaux algorithmes de tri utilisés dans le développement de logiciels
- ◆ Comprendre comment différents algorithmes fonctionnent avec des arbres, *Heaps* et des graphes
- ◆ Comprendre le fonctionnement des algorithmes *Greedy*, leur stratégie et des exemples de leur utilisation dans les principaux problèmes connus. Nous apprendrons également l'utilisation des algorithmes *Greedy* sur les graphes
- ◆ Nous apprendrons les principales stratégies de recherche du chemin minimal, avec l'approche des problèmes essentiels du domaine et des algorithmes pour leur résolution
- ◆ Comprendre la technique du Backtracking et ses principales utilisations, ainsi que les techniques alternatives

Module 4. Conception d'algorithmes avancés

- ◆ Approfondir la conception d'algorithmes avancés, en analysant les algorithmes récursifs et de type diviser pour régner, ainsi qu'en effectuant des analyses amorties
- ◆ Comprendre les concepts de la programmation dynamique et les algorithmes pour les problèmes NP
- ◆ Comprendre le fonctionnement de l'optimisation combinatoire, ainsi que les différents algorithmes de randomisation et les algorithmes parallèles
- ◆ Connaître et comprendre le fonctionnement des différentes méthodes de recherche locale et de recherche de candidats
- ◆ Apprendre les mécanismes de la vérification formelle des programmes et de la vérification itérative des programmes, y compris la logique du premier ordre et le système formel de *Hoare*
- ◆ Apprenez le fonctionnement de certaines des principales méthodes numériques telles que la méthode de bisection, la méthode de Newton Raphson et la méthode de la sécante

Module 5. Logique computationnelle

- ◆ Apprendre les bases de la logique computationnelle, à quoi elle sert et la justification de son utilisation
- ◆ Connaître les différentes stratégies de formalisation et de déduction en logique propositionnelle, notamment le raisonnement naturel, la déduction axiomatique et naturelle, ainsi que les règles primitives du calcul propositionnel
- ◆ Acquérir des connaissances avancées en logique propositionnelle, en approfondissant sa sémantique et les principales applications de cette logique, comme les circuits logiques
- ◆ Comprendre la logique des prédicats, tant pour le calcul de déduction naturelle des prédicats que pour les stratégies de formalisation et de déduction de la logique des prédicats
- ◆ Comprendre les bases du langage naturel et son mécanisme déductif
- ◆ Introduction à la programmation logique à l'aide du langage PROLOG

Module 6. Intelligence artificielle et Ingénierie des Connaissances

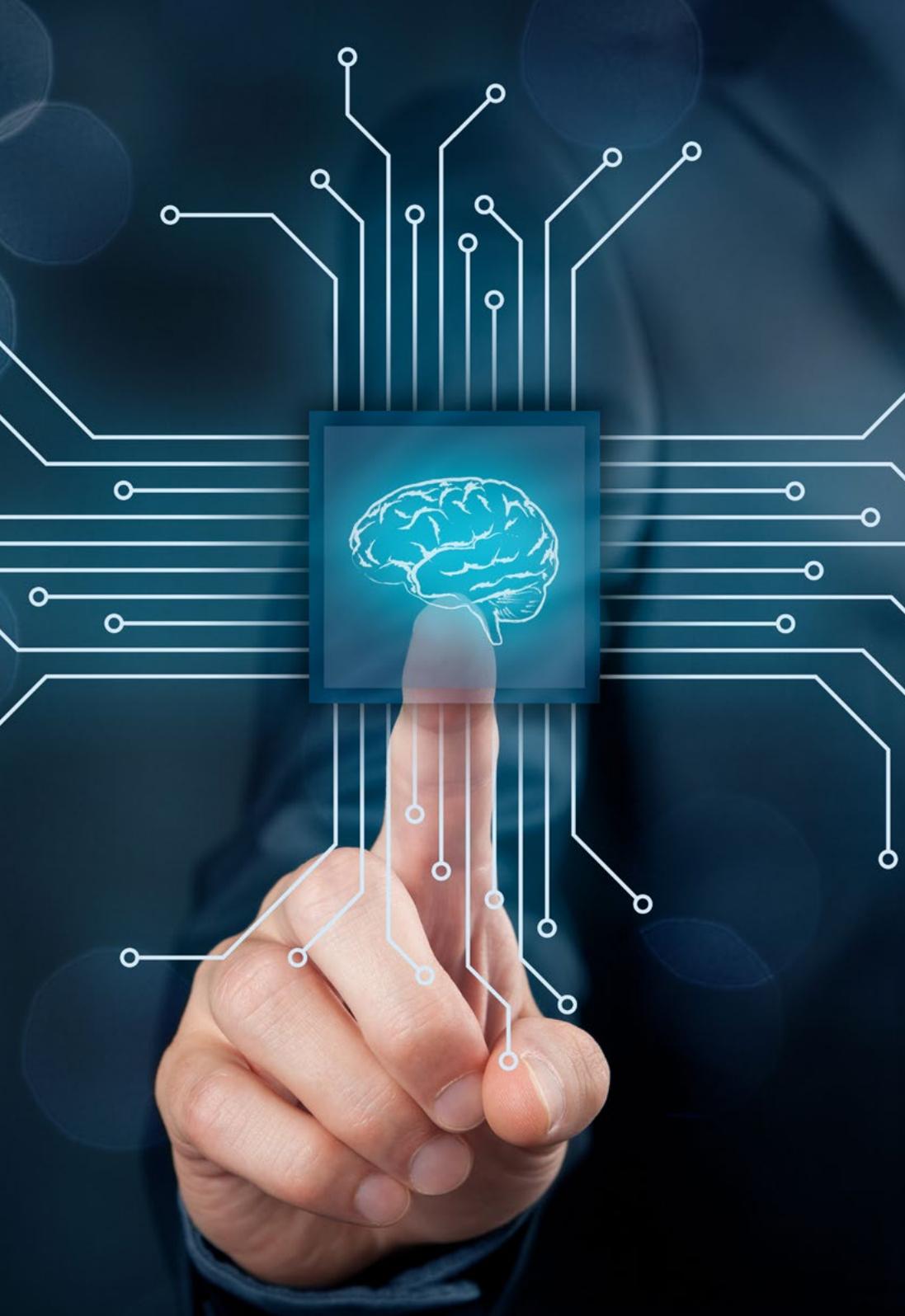
- ◆ Poser les bases de l'intelligence Artificielle et de l'ingénierie de la connaissance, en faisant un bref survol de l'histoire de l'intelligence artificielle jusqu'à aujourd'hui
- ◆ Comprendre les concepts essentiels de la recherche en intelligence artificielle, qu'il s'agisse de recherche informée ou non
- ◆ Comprendre comment l'intelligence artificielle fonctionne dans les jeux
- ◆ Apprenez les concepts fondamentaux des réseaux neuronaux et l'utilisation des algorithmes génétiques
- ◆ Acquérir les mécanismes appropriés pour représenter les connaissances, notamment en tenant compte du web sémantique
- ◆ Comprendre le fonctionnement des systèmes experts et des systèmes d'aide à la décision

Module 7. Systèmes intelligents

- ◆ Apprendre tous les concepts liés à la théorie des agents, à l'architecture des agents et à leur processus de raisonnement
- ◆ Assimiler la théorie et la pratique qui sous-tendent les concepts d'information et de connaissance, ainsi que les différentes manières de représenter la connaissance
- ◆ Comprendre la théorie liée aux ontologies, ainsi qu'apprendre les langages ontologiques et les logiciels de création d'ontologies
- ◆ Apprendre différents modèles de représentation des connaissances, tels que les vocabulaires, les taxonomies, les thésaurus et les cartes heuristiques, entre autres
- ◆ Comprendre le fonctionnement des raisonneurs sémantiques, des systèmes à base de connaissances et des systèmes experts
- ◆ Connaître le fonctionnement du web sémantique, son état actuel et futur, ainsi que les applications basées sur le web sémantique

Module 8. Apprentissage automatique et exploration de données

- ◆ Introduire les processus de découverte des connaissances et les concepts de base de l'apprentissage automatique
- ◆ Apprenez les méthodes d'exploration et de prétraitement des données, ainsi que différents algorithmes basés sur des arbres de décision
- ◆ Comprendre le fonctionnement des méthodes bayésiennes, de la régression et des méthodes à réponse continue
- ◆ Comprendre les différentes règles de classification et l'évaluation des classificateurs en apprenant à utiliser les matrices de confusion et l'évaluation numérique, la statistique de Kappa et la courbe ROC
- ◆ Acquérir les connaissances essentielles liées à l'exploration de texte, au traitement du langage naturel (NLP) et au *Clustering*
- ◆ Approfondir la connaissance des réseaux neuronaux, des réseaux neuronaux simples aux réseaux neuronaux récurrents



Module 9. Systèmes multi-agents et perception computationnelle

- ◆ Comprendre les concepts de base et avancés liés aux agents et aux systèmes multi-agents
- ◆ Étudier la norme FIPA relative aux agents, en tenant compte, entre autres, de la communication, de la gestion et de l'architecture des agents
- ◆ approfondir l'apprentissage de la plateforme JADE (*Java Agent DEvelopment Framework*) en apprenant à y programmer des concepts de base et avancés, y compris les thèmes de la communication et de la découverte d'agents
- ◆ Poser les bases du traitement du langage naturel, comme la reconnaissance automatique de la parole et la linguistique computationnelle
- ◆ Comprendre en profondeur le fonctionnement de la vision artificielle, l'analyse d'images numériques, la transformation et la segmentation d'images

Module 10. Informatique bio-inspirée

- ◆ Introduire le concept de calcul bio-inspiré, ainsi que comprendre le fonctionnement de différents types d'algorithmes d'adaptation sociale et d'algorithmes génétiques
- ◆ Approfondir l'étude des différents modèles de calcul évolutif, en connaissant leurs stratégies, leur programmation, leurs algorithmes et les modèles basés sur l'estimation des distributions
- ◆ Comprendre les principales stratégies d'exploration-exploitation de l'espace pour les algorithmes génétiques
- ◆ Comprendre le fonctionnement de la programmation évolutionnaire appliquée aux problèmes d'apprentissage et aux problèmes multi-objectifs
- ◆ Apprenez les concepts essentiels liés aux réseaux neuronaux et comprenez comment ils fonctionnent dans des cas d'utilisation réels appliqués à des domaines aussi divers que la recherche médicale, l'économie et la vision par ordinateur

03

Compétences

Ce Mastère Spécialisé en Intelligence Artificielle et Ingénierie des Connaissances a été créé comme un outil de formation de haut niveau pour les professionnels. Votre formation intensive vous préparera à travailler dans tous les domaines liés à l'intelligence artificielle avec l'assurance d'un expert en la matière.



“

Le Mastère Spécialisé en Intelligence Artificielle et Ingénierie des Connaissances vous fournira les compétences personnelles et professionnelles essentielles pour jouer un rôle approprié dans toute situation professionnelle dans ce domaine d'intervention"

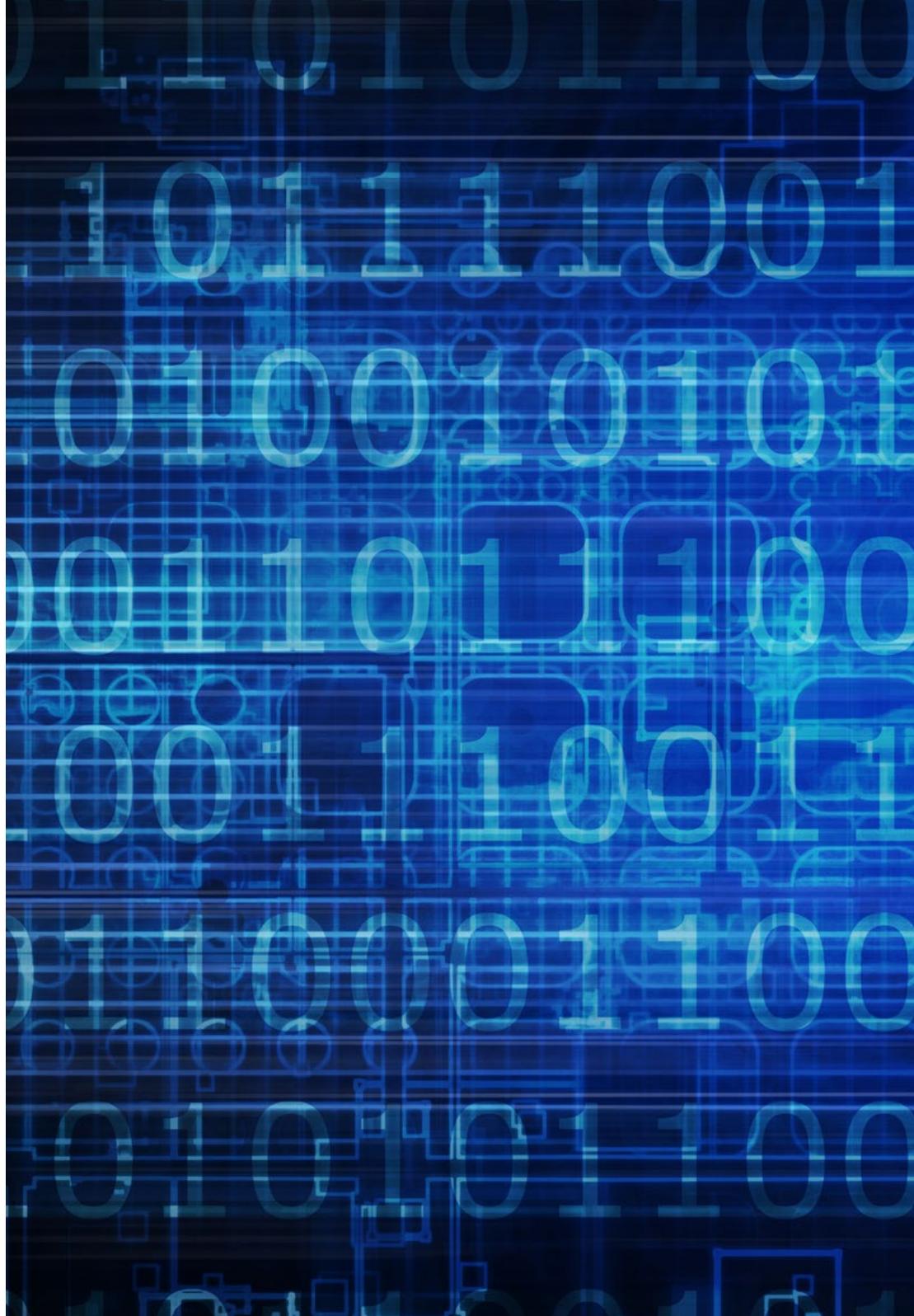


Compétence générale

- ◆ Acquérir les compétences nécessaires à la pratique professionnelle de l'ingénierie informatique avec la connaissance de tous les facteurs nécessaires pour la mener à bien avec qualité et solvabilité

“

*Une expérience de formation
unique, clé et décisive pour stimuler
votre développement professionnel”*





Compétences spécifiques

- ◆ Développer la programmation dans le domaine de l'intelligence artificielle en tenant compte de tous les facteurs de développement de l'intelligence artificielle
- ◆ Une connaissance compétente de la structure des données en programmation C++
- ◆ Conception d'algorithmes de base et avancés
- ◆ Comprendre la logique computationnelle et l'appliquer à la conception de projets
- ◆ Connaître l'intelligence artificielle, ses utilisations et ses développements et la mettre en œuvre
- ◆ Sachez ce qu'ils sont, comment ils fonctionnent et comment travailler avec des systèmes intelligents
- ◆ Maîtriser les bases de l'apprentissage automatique
- ◆ Connaissance de JADE, FIPA, vision artificielle et autres systèmes multi-agents
- ◆ Comprendre les algorithmes informatiques bio-inspirés et les stratégies pour les utiliser

04

Structure et contenu

Le contenu de ce Mastère Spécialisé a été élaboré par différents experts du domaine, avec un objectif clair: s'assurer que les étudiants acquièrent chacune des compétences nécessaires pour devenir de véritables experts dans tout ce qui concerne l'intelligence artificielle.

Un programme complet et bien structuré qui vous mènera vers les plus hauts standards de qualité et de réussite.



MA
LEA

Search
Engine
Optimiz
A
AUTOMATION

PROBLEM
SOLVING

MACHINE
LEARNING

“

Un programme d'enseignement très complet, structuré en unités didactiques très développées, orienté vers un apprentissage compatible avec votre vie personnelle et professionnelle"

Module 1. Fondamentaux de programmation

- 1.1. Introduction à la programmation
 - 1.1.1. Structure de base d'un ordinateur
 - 1.1.2. Software
 - 1.1.3. Langage de programmation
 - 1.1.4. Cycle de vie d'une application logicielle
- 1.2. Conception d'algorithmes
 - 1.2.1. La résolution de problèmes
 - 1.2.2. Techniques descriptives
 - 1.2.3. Éléments et structure d'un algorithme
- 1.3. Éléments d'un programme
 - 1.3.1. Origine et caractéristiques du langage C++
 - 1.3.2. L'environnement de développement
 - 1.3.3. Concept de programme
 - 1.3.4. Types de données fondamentales
 - 1.3.5. Opérateurs
 - 1.3.6. Expressions
 - 1.3.7. Phrases
 - 1.3.8. Entrée et sortie de données
- 1.4. Phrases de contrôle
 - 1.4.1. Phrases
 - 1.4.2. Bifurcations
 - 1.4.3. Loops
- 1.5. Abstraction et modularité: fonctions
 - 1.5.1. Conception modulaire
 - 1.5.2. Concept de fonction et d'utilité
 - 1.5.3. Définition d'une fonction
 - 1.5.4. Flux d'exécution dans un appel de fonction
 - 1.5.5. Prototype d'une fonction
 - 1.5.6. Retour des résultats
 - 1.5.7. Appel d'une fonction: paramètres
 - 1.5.8. Passage de paramètres par référence et par valeur
 - 1.5.9. Identification de la zone
- 1.6. Structures de données statiques
 - 1.6.1. Arrays
 - 1.6.2. Matrices. Polyèdres
 - 1.6.3. Recherche et tri
 - 1.6.4. Chaînes Fonctions entrées/sorties pour les chaînes
 - 1.6.5. Structures Unions
 - 1.6.6. Nouveaux types de données
- 1.7. Structures de données dynamiques: pointeurs
 - 1.7.1. Concept. Définition de pointeur
 - 1.7.2. Opérateurs et opérations avec des pointeurs
 - 1.7.3. Arrays de pointeurs
 - 1.7.4. Pointeurs et Arrays
 - 1.7.5. Pointeurs vers des chaînes
 - 1.7.6. Pointeurs vers des structures
 - 1.7.7. Indirectivité multiple
 - 1.7.8. Pointeurs vers les fonctions
 - 1.7.9. Passage de fonctions, de structures et de tableaux comme paramètres de fonction
- 1.8. Fichiers
 - 1.8.1. Concepts de base
 - 1.8.2. Opérations sur fichiers
 - 1.8.3. Types de fichiers
 - 1.8.4. Organisation de fichiers
 - 1.8.5. Introduction aux fichiers C++
 - 1.8.6. Traitement des fichiers
- 1.9. Récursivité
 - 1.9.1. Définition de la récursion
 - 1.9.2. Types de récursion
 - 1.9.3. Avantages et inconvénients
 - 1.9.4. Considérations
 - 1.9.5. Conversion récursive-itérative
 - 1.9.6. La pile de récursion

- 1.10. Preuves et documentation
 - 1.10.1. Test de programmes
 - 1.10.2. Test de la boîte blanche
 - 1.10.3. Test de la boîte noire
 - 1.10.4. Outils de tests
 - 1.10.5. Documentation du programme

Module 2. Structure des données

- 2.1. Introduction à la programmation C++
 - 2.1.1. Classes, constructeurs, méthodes et attributs
 - 2.1.2. Variables
 - 2.1.3. Expressions conditionnelles et boucles
 - 2.1.4. Objets
- 2.2. Types de données abstraites (ADT)
 - 2.2.1. Types de données
 - 2.2.2. Structures de base et TAD
 - 2.2.3. Vecteurs et Arrays
- 2.3. Structures de données linéaires
 - 2.3.1. TAD liste. Définition
 - 2.3.2. Listes liées et listes doublement liées
 - 2.3.3. Listes ordonnées
 - 2.3.4. Listes en C++
 - 2.3.5. TAD Pile
 - 2.3.6. TAD Queue
 - 2.3.7. Pile et Queue en C++
- 2.4. Structures de données hiérarchiques
 - 2.4.1. TAD Arbre
 - 2.4.2. Sentiers
 - 2.4.3. Arbres N-aires
 - 2.4.4. Arbres binaires
 - 2.4.5. Arbres de recherche binaires
- 2.5. Structures de données hiérarchiques: arbres complexes
 - 2.5.1. Arbres parfaitement équilibrés ou arbres de hauteur minimale
 - 2.5.2. Arbres à trajets multiples
 - 2.5.3. Références bibliographiques
- 2.6. Monticules et queue de priorité
 - 2.6.1. Monticules TAD
 - 2.6.2. TAD File d'attente prioritaire
- 2.7. Tables Hash
 - 2.7.1. TAD Table *Hash*
 - 2.7.2. Fonctions *Hash*
 - 2.7.3. Fonction *Hash* en tables Hash
 - 2.7.4. Redispersion
 - 2.7.5. Tables Hash ouvertes
- 2.8. Networks
 - 2.8.1. TAD Network
 - 2.8.2. Type de Network
 - 2.8.3. Représentation graphique et opérations de base
 - 2.8.4. Design de réseaux
- 2.9. Algorithmes et concepts graphiques avancés
 - 2.9.1. Problèmes sur les graphiques
 - 2.9.2. Algorithmes sur les routes
 - 2.9.3. Algorithmes sur les routes
 - 2.9.4. Autres algorithmes
- 2.10. Autres structures de données
 - 2.10.1. Sets
 - 2.10.2. Arrays parallèles
 - 2.10.3. Tableaux des symboles
 - 2.10.4. Tries

Module 3. Algorithme et complexité

- 3.1. Introduction aux stratégies de conception d'algorithmes
 - 3.1.1. Récursivité
 - 3.1.2. Diviser pour mieux régner
 - 3.1.3. Autres stratégies
- 3.2. Efficacité et analyse des algorithmes
 - 3.2.1. Mesures d'efficacité
 - 3.2.2. Mesurer la taille de l'entrée
 - 3.2.3. Mesure du temps d'exécution
 - 3.2.4. Pire cas, meilleur cas et cas moyen
 - 3.2.5. Notation asymptotique
 - 3.2.6. Critères d'analyse mathématique pour les algorithmes non récursifs
 - 3.2.7. Analyse mathématique des algorithmes récursifs
 - 3.2.8. Analyse empirique des algorithmes
- 3.3. Algorithmes de tri
 - 3.3.1. Concept de tri
 - 3.3.2. Tri de la bulle
 - 3.3.3. Tri par sélection
 - 3.3.4. Ordre d'insertion
 - 3.3.5. Triage par fusion (*Merge_Sort*)
 - 3.3.6. Tri rapide (*QuickSort*)
- 3.4. Algorithmes avec arbres
 - 3.4.1. Concept d'arbre
 - 3.4.2. Arbres binaires
 - 3.4.3. Parcours d'arbres
 - 3.4.4. Représentation des expressions
 - 3.4.5. Arbres binaires ordonnés
 - 3.4.6. Arbres binaires équilibrés
- 3.5. Algorithmes avec *Heaps*
 - 3.5.1. Les *Heaps*
 - 3.5.2. L'algorithme *HeapSort*
 - 3.5.3. Files d'attente prioritaires



- 3.6. Algorithmes avec graphes
 - 3.6.1. Représentation
 - 3.6.2. Parcours en largeur
 - 3.6.3. Visite en profondeur
 - 3.6.4. Ordre topologique
 - 3.7. Algorithmes *Greedy*
 - 3.7.1. La stratégie *Greedy*
 - 3.7.2. Éléments de La stratégie *Greedy*
 - 3.7.3. Change de devises
 - 3.7.4. Le problème du voyageur
 - 3.7.5. Problème de sac à dos
 - 3.8. Recherche de chemins minimaux
 - 3.8.1. Le problème du chemin minimal
 - 3.8.2. Arcs et cycles négatifs
 - 3.8.3. Algorithme de Dijkstra
 - 3.9. Algorithmes *Greedy* sobre Grafos
 - 3.9.1. L'arbre à chevauchement minimal
 - 3.9.2. L'algorithme de Prim
 - 3.9.3. L'algorithme de L'algorithme de Kruskal
 - 3.9.4. Analyse de complexité
 - 3.10. *Backtracking*
 - 3.10.1. Le *Backtracking*
 - 3.10.2. Techniques alternatives
-
- Module 4. Conception d'algorithmes avancés**
- 4.1. Analyse des algorithmes récursifs et des algorithmes de type "diviser pour régner"
 - 4.1.1. Poser et résoudre des équations de récurrence homogènes et non-homogènes.
 - 4.1.2. Aperçu de la stratégie "diviser pour régner"
 - 4.2. Analyse amortie
 - 4.2.1. Analyse des agrégats
 - 4.2.2. La méthode de comptabilisation
 - 4.2.3. La méthode du potentiel
 - 4.3. Programmation dynamique et algorithmes pour les problèmes NP
 - 4.3.1. Caractéristiques de la programmation dynamique
 - 4.3.2. Backtracking: retour en arrière
 - 4.3.3. Branchements et élagage
 - 4.4. Optimisation combinatoire
 - 4.4.1. Représentation des problèmes
 - 4.4.2. Optimisation 1D
 - 4.5. Algorithmes de randomisation
 - 4.5.1. Exemples d'algorithmes de randomisation
 - 4.5.2. Le théorème de Buffon
 - 4.5.3. Algorithme de Monte Carlo
 - 4.5.4. Algorithme de Las Vegas
 - 4.6. Recherche locale et recherche de candidats
 - 4.6.1. Garcent *Ascent*
 - 4.6.2. Hill *Climbing*
 - 4.6.3. *Simulated Annealing*
 - 4.6.4. *Tabu Search*
 - 4.6.5. Recherche de candidats
 - 4.7. Vérification formelle des programmes
 - 4.7.1. Spécification d'abstractions fonctionnelles
 - 4.7.2. Le langage de la logique du premier ordre
 - 4.7.3. Le système formel de Hoare
 - 4.8. Vérification des programmes itératifs
 - 4.8.1. Les règles système formel de Hoare
 - 4.8.2. Concept d'itérations invariantes
 - 4.9. Méthodes numériques
 - 4.9.1. La méthode de bisection
 - 4.9.2. Méthode de Newton Raphson
 - 4.9.3. La méthode sécante
 - 4.10. Algorithmes parallèles
 - 4.10.1. Opérations binaires parallèles
 - 4.10.2. Opérations parallèles avec les réseaux
 - 4.10.3. Le parallélisme dans le principe "diviser pour régner"
 - 4.10.4. Le parallélisme dans la programmation dynamique

Module 5. Logique computationnelle

- 5.1. Justification de la logique
 - 5.1.1. Objet de l'étude de la logique
 - 5.1.2. A quoi sert la logique?
 - 5.1.3. Composantes et types de raisonnement
 - 5.1.4. Composants d'un calcul logique
 - 5.1.5. Sémantique
 - 5.1.6. Justification de l'existence d'une logique
 - 5.1.7. Comment vérifier qu'une logique est adéquate?
- 5.2. Calcul de la déduction naturelle des déclarations
 - 5.2.1. Langage formel
 - 5.2.2. Mécanisme déductif
- 5.3. Formalisation et stratégies de déduction pour la logique propositionnelle
 - 5.3.1. Stratégies de formalisation
 - 5.3.2. Raisonnement naturel
 - 5.3.3. Lois et règles
 - 5.3.4. Déduction axiomatique et déduction naturelle
 - 5.3.5. Le calcul de la déduction naturelle
 - 5.3.6. Règles primitives du calcul propositionnel
- 5.4. Sémantique de la logique propositionnelle
 - 5.4.1. Tables de vérité
 - 5.4.2. Équivalence
 - 5.4.3. Tautologies et contradictions
 - 5.4.4. Validation des phrases propositionnelles
 - 5.4.5. Validation au moyen de tables de vérité
 - 5.4.6. Validation à l'aide d'arbres sémantiques
 - 5.4.7. Validation par réfutation
- 5.5. Applications de la logique propositionnelle: circuits logiques
 - 5.5.1. Les portes de base
 - 5.5.2. Circuits
 - 5.5.3. Modèles mathématiques de circuits
 - 5.5.4. Minimisation
 - 5.5.5. La deuxième forme canonique et la forme minimale en produit de sommes
 - 5.5.6. Autres portes

- 5.6. Calcul de la déduction naturelle des prédicats
 - 5.6.1. Langage formel
 - 5.6.2. Mécanisme déductif
- 5.7. Stratégies de formalisation pour la logique des prédicats
 - 5.7.1. Introduction à la formalisation en logique des prédicats
 - 5.7.2. Stratégies de formalisation avec les quantificateurs
- 5.8. Stratégies de Déduction pour la logique des prédicats
 - 5.8.1. Motif de l'omission
 - 5.8.2. Présentation des nouvelles règles
 - 5.8.3. La logique des prédicats comme calcul de la déduction naturelle
- 5.9. Applications de la logique des prédicats: introduction à la programmation logique
 - 5.9.1. Présentation informelle
 - 5.9.2. Éléments de Prolog
 - 5.9.3. Réévaluation et coupure
- 5.10. Théorie des ensembles, logique des prédicats et sa sémantique
 - 5.10.1. Théorie des ensembles intuitionnistes
 - 5.10.2. Introduction à la Sémantique des prédicats

Module 6. Intelligence artificielle et Ingénierie des Connaissances

- 6.1. Introduction à l'intelligence artificielle et à l'ingénierie des connaissances
 - 6.1.1. Brève histoire de l'intelligence artificielle
 - 6.1.2. L'intelligence artificielle aujourd'hui
 - 6.1.3. Ingénierie des Connaissances
- 6.2. Recherche
 - 6.2.1. Concepts de recherche courants
 - 6.2.2. Recherche non informée
 - 6.2.3. Recherche informée
- 6.3. Satisfaisabilité booléenne, satisfaisabilité des contraintes et planification automatique
 - 6.3.1. Satisfaisabilité booléenne
 - 6.3.2. Problèmes de satisfaction des contraintes
 - 6.3.3. Planification automatique et PDDL
 - 6.3.4. La planification en tant que recherche heuristique
 - 6.3.5. Planification avec le SAT

- 6.4. L'intelligence artificielle dans les jeux
 - 6.4.1. La théorie des jeux
 - 6.4.2. Élagage Minimax et Alpha-Beta
 - 6.4.3. Simulation: Monte Carlo
- 6.5. Apprentissage supervisé et non supervisé
 - 6.5.1. Introduction à l'apprentissage automatique
 - 6.5.2. Classification
 - 6.5.3. Régression
 - 6.5.4. Validation des résultats
 - 6.5.5. Regroupement (*Clustering*)
- 6.6. Réseaux neuronaux
 - 6.6.1. Fondements biologiques
 - 6.6.2. Modèle de calcul
 - 6.6.3. Réseaux neuronaux supervisés et non supervisés
 - 6.6.4. Perceptron simple
 - 6.6.5. Perceptron multicouche
- 6.7. Algorithmes génétiques
 - 6.7.1. Histoire
 - 6.7.2. Base biologique
 - 6.7.3. Codage des problèmes
 - 6.7.4. Génération de la population initiale
 - 6.7.5. Algorithme principal et opérateurs génétiques
 - 6.7.6. Évaluation des individus: *Fitness*
- 6.8. Thésaurus, vocabulaires, taxonomies
 - 6.8.1. Vocabulaires
 - 6.8.2. Taxonomies
 - 6.8.3. Thésaurus
 - 6.8.4. Ontologies
- 6.9. Représentation des connaissances: web sémantique
 - 6.9.1. Web sémantique
 - 6.9.2. Spécifications: RDF, RDFS et OWL
 - 6.9.3. Inférence/raisonnement
 - 6.9.4. Linked Data

- 6.10. Systèmes experts et DSS
 - 6.10.1. Systèmes experts
 - 6.10.2. Systèmes d'aide à la décision

Module 7. Systèmes intelligents

- 7.1. Théorie des agents
 - 7.1.1. Histoire du concept
 - 7.1.2. Définition de l'agent
 - 7.1.3. Les agents en intelligence artificielle
 - 7.1.4. Les agents dans l'ingénierie logicielle
- 7.2. Architectures d'agents
 - 7.2.1. Le processus de raisonnement d'un agent
 - 7.2.2. Agents réactifs
 - 7.2.3. Agents déductifs
 - 7.2.4. Agents hybrides
 - 7.2.5. Comparaison
- 7.3. Informations et connaissances
 - 7.3.1. Distinction entre données, informations et connaissances
 - 7.3.2. Évaluation de la qualité des données
 - 7.3.3. Méthodes de capture des données
 - 7.3.4. Méthodes d'acquisition de l'information
 - 7.3.5. Méthodes d'acquisition des connaissances
- 7.4. Représentation des connaissances
 - 7.4.1. L'importance de la représentation des connaissances
 - 7.4.2. Définir la représentation des connaissances à travers leurs rôles
 - 7.4.3. Caractéristiques d'une représentation des connaissances
- 7.5. Ontologies
 - 7.5.1. Introduction aux métadonnées
 - 7.5.2. Concept philosophique d'ontologie
 - 7.5.3. Concept informatique d'ontologie
 - 7.5.4. Ontologies de domaine et ontologies de niveau supérieur
 - 7.5.5. Comment construire une ontologie?

- 7.6. Langages ontologiques et logiciels pour la création d'ontologies
 - 7.6.1. Triplés RDF, Turtle et N3
 - 7.6.2. RDF Schema
 - 7.6.3. OWL
 - 7.6.4. SPARQL
 - 7.6.5. Introduction aux différents outils de création d'ontologies
 - 7.6.6. Installation et utilisation du Protégé
- 7.7. Le web sémantique
 - 7.7.1. L'état actuel et l'avenir du web sémantique
 - 7.7.2. Applications du web sémantique
- 7.8. Autres modèles de représentation des connaissances
 - 7.8.1. Vocabulaires
 - 7.8.2. Vue d'ensemble
 - 7.8.3. Taxonomies
 - 7.8.4. Thésaurus
 - 7.8.5. Folksonomies
 - 7.8.6. Comparaison
 - 7.8.7. Mapas mentales
- 7.9. Évaluation et intégration des représentations des connaissances
 - 7.9.1. Logique d'ordre zéro
 - 7.9.2. Logique du premier ordre
 - 7.9.3. Logique descriptive
 - 7.9.4. Relation entre les différents types de logique
 - 7.9.5. Prolog: programmation basée sur la logique du premier ordre
- 7.10. Raisonneurs sémantiques, systèmes à base de connaissances et systèmes experts
 - 7.10.1. Concept de raisonneur
 - 7.10.2. Applications d'un raisonneur
 - 7.10.3. Systèmes basés sur la connaissance
 - 7.10.4. MYCIN, l'histoire des systèmes experts
 - 7.10.5. Éléments et architecture des systèmes experts
 - 7.10.6. Création de systèmes experts

Module 8. Apprentissage automatique et exploration de données

- 8.1. Introduction aux processus de découverte de connaissances et aux concepts de base de l'apprentissage machine
 - 8.1.1. Concepts clés du processus de découverte de connaissances
 - 8.1.2. Perspective historique sur les processus de découverte de connaissances
 - 8.1.3. Étapes des processus de découverte de connaissances
 - 8.1.4. Techniques utilisées dans les processus de découverte de connaissances
 - 8.1.5. Caractéristiques des bons modèles d'apprentissage automatique
 - 8.1.6. Types d'informations sur l'apprentissage automatique
 - 8.1.7. Concepts d'apprentissage de base
 - 8.1.8. Concepts d'apprentissage de base non supervisé
- 8.2. Exploration et prétraitement des données
 - 8.2.1. Traitement des données
 - 8.2.2. Traitement des données dans le flux d'analyse des données
 - 8.2.3. Types de données
 - 8.2.4. Transformations de données
 - 8.2.5. Visualisation et exploration des variables continues
 - 8.2.6. Visualisation et exploration de variables catégorielles
 - 8.2.7. Mesures de corrélation
 - 8.2.8. Représentations graphiques les plus courantes
 - 8.2.9. Introduction à l'analyse multivariée et à la réduction de la dimensionnalité
- 8.3. Arbres de décision
 - 8.3.1. Algorithme ID3
 - 8.3.2. Algorithme C4.5
 - 8.3.3. Sobreentrenamiento y poda
 - 8.3.4. Analyse des résultats
- 8.4. Évaluation des classificateurs
 - 8.4.1. Matrices de confusion
 - 8.4.2. Matrices d'évaluation numérique
 - 8.4.3. Statistique de Kappa
 - 8.4.4. La courbe ROC

- 8.5. Règles de classification
 - 8.5.1. Mesures d'évaluation des règles
 - 8.5.2. Introduction à la représentation graphique
 - 8.5.3. Algorithme de superposition séquentielle
- 8.6. Réseaux neuronaux
 - 8.6.1. Concepts de base
 - 8.6.2. Réseaux neuronaux simples
 - 8.6.3. Algorithme de *Backpropagation*
 - 8.6.4. Introduction aux réseaux neuronaux récurrents
- 8.7. Méthodes bayésiennes
 - 8.7.1. Concepts de base de probabilité
 - 8.7.2. Théorème de Bayes
 - 8.7.3. Naive Bayes
 - 8.7.4. Introduction aux réseaux bayésiens
- 8.8. Modèles de régression et de réponse continue
 - 8.8.1. Régression linéaire simple
 - 8.8.2. Régression linéaire multiple
 - 8.8.3. Régression logistique
 - 8.8.4. Arbres de régression
 - 8.8.5. Introduction aux machines à vecteurs de support (SVM)
 - 8.8.6. Mesures de la qualité de l'ajustement
- 8.9. *Clustering*
 - 8.9.1. Concepts de base
 - 8.9.2. Clustering hiérarchique
 - 8.9.3. Méthodes probabilistes
 - 8.9.4. Algorithme EM
 - 8.9.5. Méthode B-Cubed
 - 8.9.6. Méthodes implicites
- 8.10. Exploration de textes et traitement du langage naturel (NLP)
 - 8.10.1. Concepts de base
 - 8.10.2. Création du corpus
 - 8.10.3. Analyse descriptive
 - 8.10.4. Introduction à l'analyse des sentiments

Module 9. Systèmes multi-agents et perception computationnelle

- 9.1. Agents et systèmes multi-agents
 - 9.1.1. Concept d'agent
 - 9.1.2. Architectures
 - 9.1.3. Communication et coordination
 - 9.1.4. Langages et outils de programmation
 - 9.1.5. Applications d'agents
 - 9.1.6. La FIPA
- 9.2. La norme pour les agents: FIPA
 - 9.2.1. Communication entre les acteurs
 - 9.2.2. La gestion des agents
 - 9.2.3. Architecture abstraite
 - 9.2.4. Autres spécifications
- 9.3. La plateforme JADE
 - 9.3.1. Agents logiciels selon JADE
 - 9.3.2. Architecture
 - 9.3.3. Installation et mise en œuvre
 - 9.3.4. Packages JADE
- 9.4. Programmation de base avec JADE
 - 9.4.1. La console de gestion
 - 9.4.2. Création d'un agent de base
- 9.5. Programmation avancée avec JADE
 - 9.5.1. Création d'un agent de avancée
 - 9.5.2. Communication entre les acteurs
 - 9.5.3. Découverte des agents
- 9.6. Vision artificielle
 - 9.6.1. Traitement et analyse des images numériques
 - 9.6.2. Analyse d'images et vision par ordinateur
 - 9.6.3. Traitement d'images et vision humaine
 - 9.6.4. Système de capture d'images
 - 9.6.5. Formation et perception des images

- 9.7. Analyse d'images numériques
 - 9.7.1. Les étapes du processus d'analyse d'image
 - 9.7.2. Pré-traitement
 - 9.7.3. Opérations de base
 - 9.7.4. Filtrage spatial
- 9.8. Transformation d'images numériques et segmentation d'images
 - 9.8.1. Transformations de Fourier
 - 9.8.2. Filtrage de la fréquence
 - 9.8.3. Concepts de base
 - 9.8.4. Segmentation
 - 9.8.5. Détection des contours
- 9.9. Reconnaissance des formes
 - 9.9.1. Extraction de caractéristiques
 - 9.9.2. Algorithmes de classification
- 9.10. Traitement du langage naturel
 - 9.10.1. Reconnaissance automatique de la parole
 - 9.10.2. Linguistique informatique

Module 10. Informatique bio-inspirée

- 10.1. Introduction à l'informatique bio-inspirée
 - 10.1.1. Introduction à l'informatique bio-inspirée
- 10.2. Algorithmes d'adaptation sociale
 - 10.2.1. Calcul bio-inspiré basé sur les colonies de fourmis
 - 10.2.2. Variantes des algorithmes de colonies de fourmis
 - 10.2.3. Informatique en nuage de particules
- 10.3. Algorithmes génétiques
 - 10.3.1. Structure générale
 - 10.3.2. Mise en œuvre des principaux opérateurs
- 10.4. Stratégies d'exploration-exploitation de l'espace pour les algorithmes génétiques
 - 10.4.1. Algorithme CHC
 - 10.4.2. Problèmes multimodaux

- 10.5. Modèles de calcul évolutif (I)
 - 10.5.1. Stratégies évolutives
 - 10.5.2. Programmation évolutive
 - 10.5.3. Algorithmes basés sur l'évolution différentielle
- 10.6. Modèles de calcul évolutif (II)
 - 10.6.1. Modèles d'évolution basés sur l'estimation des distributions (EDA)
 - 10.6.2. Programmation génétique
- 10.7. La programmation du développement appliquée aux troubles de l'apprentissage
 - 10.7.1. Apprentissage basé sur des règles
 - 10.7.2. Méthodes évolutionnaires dans les problèmes de sélection d'instances
- 10.8. Problèmes multi-objectifs
 - 10.8.1. Concept de dominance
 - 10.8.2. Application des algorithmes évolutionnaires aux problèmes multi-objectifs
- 10.9. Réseaux neuronaux (I)
 - 10.9.1. Introduction aux réseaux neuronaux
 - 10.9.2. Exemple pratique avec les réseaux neuronaux
- 10.10. Réseaux neuronaux (II)
 - 10.10.1. Cas d'utilisation des réseaux neuronaux dans la recherche médicale
 - 10.10.2. Cas d'utilisation des réseaux neuronaux en économie
 - 10.10.3. Cas d'utilisation des réseaux neuronaux dans la vision artificielle

- AboutWindow.cpp
- auth.c
- auth.h
 - _auth.h
 - get_access_token.c
- client.c
- client.h
 - _client.h
 - client_block(int, int)
 - client_chunk(int, int)
 - client_connect(char)
 - client_disable()
 - client_enable()
 - client_light(int, int, i)
 - client_login(const cl)
 - client_position(float)
 - client_recv()
 - client_send(char *)
 - client_sign(int, int, i)
 - client_start()
 - client_stop()
 - client_talk(const cha)
 - client_version(int)
 - DEFAULT_PORT
 - get_client_enabled()
- config.h
- ConnectionDialog.cpp
- _make_sphere(float

```

        continue;
    }
    float du = (tiles[i] % 16) * s;
    float dv = (tiles[i] / 16) * s;
    int flip = ao[i][0] + ao[i][3] > ao[i][1] + ao[i][2];
    for (int v = 0; v < 6; v++) {
        int j = flip ? flipped[i][v] : indices[i][v];
        *(d++) = x + n * positions[i][j][0];
        *(d++) = y + n * positions[i][j][1];
        *(d++) = z + n * positions[i][j][2];
        *(d++) = normals[i][0];
        *(d++) = normals[i][1];
        *(d++) = normals[i][2];
        *(d++) = du + (uvs[i][j][0] ? b : a);
        *(d++) = dv + (uvs[i][j][1] ? b : a);
        *(d++) = ao[i][j];
        *(d++) = light[i][j];
    }
}

void make_cube(
    float *data, float ao[6][4], float light[6][4],
    int left, int right, int top, int bottom, int front, int back,
    float x, float y, float z, float n, int w)
{
    int wleft = blocks[w][0];
    int wright = blocks[w][1];
    int wtop = blocks[w][2];
    int wbottom = blocks[w][3];
    int wfront = blocks[w][4];
    int wback = blocks[w][5];
    make_cube_faces(
        data, ao, light,
        left, right, top, bottom, front, back,
        wleft, wright, wtop, wbottom, wfront, wback,
        x, y, z, n);
}
    
```

05 Méthodologie

Ce programme de formation offre une manière différente d'apprendre. Notre méthodologie est développée à travers un mode d'apprentissage cyclique: ***le Relearning***.

Ce système d'enseignement est utilisé, par exemple, dans les écoles de médecine les plus prestigieuses du monde et a été considéré comme l'un des plus efficaces par des publications de premier plan telles que le ***New England Journal of Medicine***.



“

Découvrez Relearning, un système qui renonce à l'apprentissage linéaire conventionnel pour vous emmener à travers des systèmes d'enseignement cycliques: une façon d'apprendre qui s'est avérée extrêmement efficace, en particulier dans les matières qui exigent la mémorisation”

Étude de Cas pour mettre en contexte tout le contenu

Notre programme offre une méthode révolutionnaire de développement des compétences et des connaissances. Notre objectif est de renforcer les compétences dans un contexte changeant, compétitif et hautement exigeant.

“

Avec TECH, vous pouvez expérimenter une manière d'apprendre qui ébranle les fondations des universités traditionnelles du monde entier”



Vous bénéficierez d'un système d'apprentissage basé sur la répétition, avec un enseignement naturel et progressif sur l'ensemble du cursus.



L'étudiant apprendra, par des activités collaboratives et des cas réels, à résoudre des situations complexes dans des environnements commerciaux réels.

Une méthode d'apprentissage innovante et différente

Cette formation TECH est un programme d'enseignement intensif, créé de toutes pièces, qui propose les défis et les décisions les plus exigeants dans ce domaine, tant au niveau national qu'international. Grâce à cette méthodologie, l'épanouissement personnel et professionnel est stimulé, faisant ainsi un pas décisif vers la réussite. La méthode des cas, technique qui constitue la base de ce contenu, permet de suivre la réalité économique, sociale et professionnelle la plus actuelle.

“ Notre programme vous prépare à relever de nouveaux défis dans des environnements incertains et à réussir votre carrière ”

La méthode des cas est le système d'apprentissage le plus largement utilisé dans les meilleures écoles d'informatique du monde depuis qu'elles existent. Développée en 1912 pour que les étudiants en Droit n'apprennent pas seulement le droit sur la base d'un contenu théorique, la méthode des cas consiste à leur présenter des situations réelles complexes afin qu'ils prennent des décisions éclairées et des jugements de valeur sur la manière de les résoudre. En 1924, elle a été établie comme méthode d'enseignement standard à Harvard.

Dans une situation donnée, que doit faire un professionnel? C'est la question à laquelle nous sommes confrontés dans la méthode des cas, une méthode d'apprentissage orientée vers l'action. Tout au long du programme, les étudiants seront confrontés à de multiples cas réels. Ils devront intégrer toutes leurs connaissances, faire des recherches, argumenter et défendre leurs idées et leurs décisions.

Relearning Methodology

TECH combine efficacement la méthodologie des Études de Cas avec un système d'apprentissage 100% en ligne basé sur la répétition, qui associe différents éléments didactiques dans chaque leçon.

Nous enrichissons l'Étude de Cas avec la meilleure méthode d'enseignement 100% en ligne: le Relearning.

En 2019, nous avons obtenu les meilleurs résultats d'apprentissage de toutes les universités en ligne du monde.

À TECH, vous apprendrez avec une méthodologie de pointe conçue pour former les managers du futur. Cette méthode, à la pointe de la pédagogie mondiale, est appelée Relearning.

Notre université est la seule université autorisée à utiliser cette méthode qui a fait ses preuves. En 2019, nous avons réussi à améliorer les niveaux de satisfaction globale de nos étudiants (qualité de l'enseignement, qualité des supports, structure des cours, objectifs...) par rapport aux indicateurs de la meilleure université en ligne.





Dans notre programme, l'apprentissage n'est pas un processus linéaire, mais se déroule en spirale (apprendre, désapprendre, oublier et réapprendre). Par conséquent, chacun de ces éléments est combiné de manière concentrique. Cette méthodologie a permis de former plus de 650.000 diplômés universitaires avec un succès sans précédent dans des domaines aussi divers que la biochimie, la génétique, la chirurgie, le droit international, les compétences en gestion, les sciences du sport, la philosophie, le droit, l'ingénierie, le journalisme, l'histoire, les marchés financiers et les instruments. Tout cela dans un environnement très exigeant, avec un corps étudiant universitaire au profil socio-économique élevé et dont l'âge moyen est de 43,5 ans.

Le Relearning vous permettra d'apprendre avec moins d'efforts et plus de performance, en vous impliquant davantage dans votre formation, en développant un esprit critique, en défendant des arguments et en contrastant les opinions: une équation directe vers le succès.

À partir des dernières preuves scientifiques dans le domaine des neurosciences, non seulement nous savons comment organiser les informations, les idées, les images et les souvenirs, mais nous savons aussi que le lieu et le contexte dans lesquels nous avons appris quelque chose sont fondamentaux pour notre capacité à nous en souvenir et à le stocker dans l'hippocampe, pour le conserver dans notre mémoire à long terme.

De cette manière, et dans ce que l'on appelle Neurocognitive context-dependent e-learning, les différents éléments de notre programme sont reliés au contexte dans lequel le participant développe sa pratique professionnelle.

Ce programme offre le support matériel pédagogique, soigneusement préparé pour les professionnels:



Support d'étude

Tous les contenus didactiques sont créés par les spécialistes qui enseigneront le cours, spécifiquement pour le cours, afin que le développement didactique soit vraiment spécifique et concret.

Ces contenus sont ensuite appliqués au format audiovisuel, pour créer la méthode de travail TECH en ligne. Tout cela, avec les dernières techniques qui offrent des pièces de haute qualité dans chacun des matériaux qui sont mis à la disposition de l'étudiant.



Cours magistraux

Il existe des preuves scientifiques de l'utilité de l'observation par un tiers expert.

La méthode "Learning from an Expert" renforce les connaissances et la mémoire, et donne confiance dans les futures décisions difficiles.



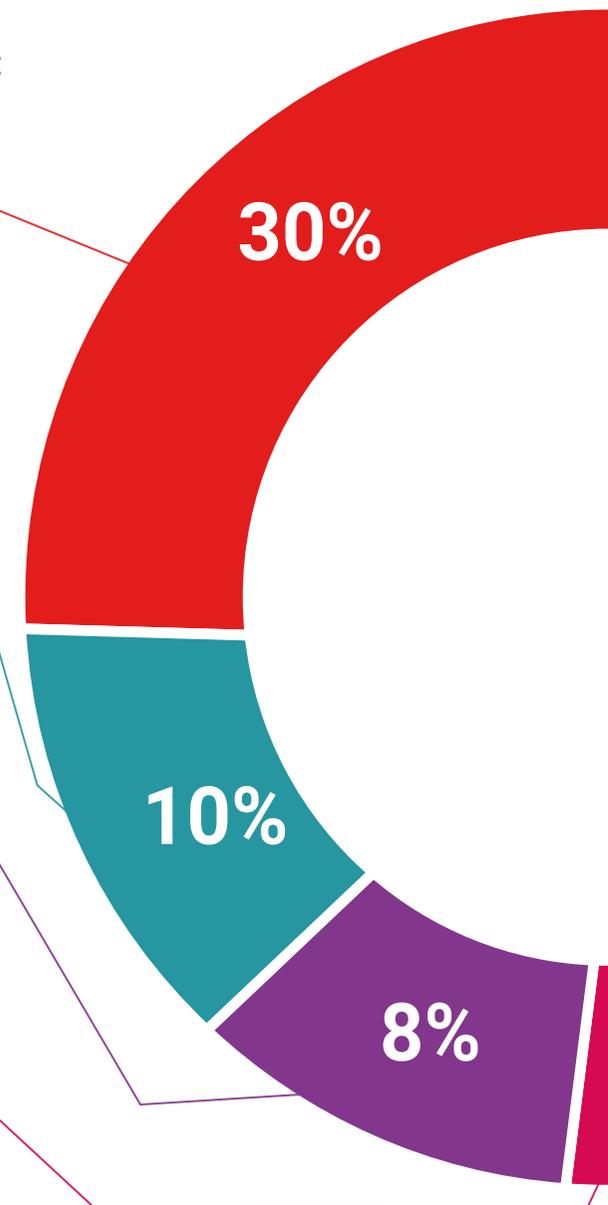
Pratiques en compétences et aptitudes

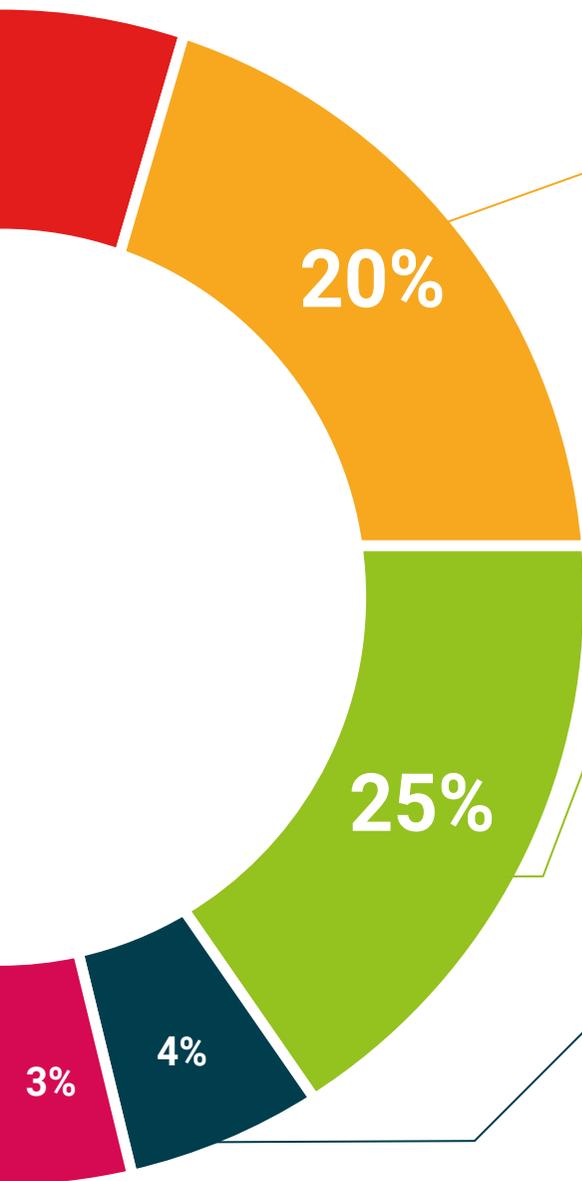
Les étudiants réaliseront des activités visant à développer des compétences et des aptitudes spécifiques dans chaque domaine. Des activités pratiques et dynamiques pour acquérir et développer les compétences et aptitudes qu'un spécialiste doit développer dans le cadre de la mondialisation dans laquelle nous vivons.



Lectures complémentaires

Articles récents, documents de consensus et directives internationales, entre autres. Dans la bibliothèque virtuelle de TECH, l'étudiant aura accès à tout ce dont il a besoin pour compléter sa formation.





Case studies

Ils réaliseront une sélection des meilleures études de cas choisies spécifiquement pour ce diplôme. Des cas présentés, analysés et tutorés par les meilleurs spécialistes de la scène internationale.



Résumés interactifs

L'équipe TECH présente les contenus de manière attrayante et dynamique dans des pilules multimédia comprenant des audios, des vidéos, des images, des diagrammes et des cartes conceptuelles afin de renforcer les connaissances. Ce système éducatif unique pour la présentation de contenu multimédia a été récompensé par Microsoft en tant que "European Success Story".



Testing & Retesting

Les connaissances de l'étudiant sont périodiquement évaluées et réévaluées tout au long du programme, par le biais d'activités et d'exercices d'évaluation et d'auto-évaluation, afin que l'étudiant puisse vérifier comment il atteint ses objectifs.



06 Diplôme

Le Mastère Spécialisé en Intelligence Artificielle et Ingénierie des Connaissances vous garantit, en plus de la formation la plus rigoureuse et la plus actuelle, l'accès à un diplôme universitaire de Mastère Spécialisé délivré par TECH Université Technologique.



“

Finalisez cette formation avec succès et recevez votre diplôme sans avoir à vous soucier des déplacements ou des démarches administratives”

Ce **Mastère Spécialisé en Intelligence Artificielle et Ingénierie des Connaissances** contient le programme le plus complet et le plus à jour du marché.

Après avoir réussi l'évaluation, l'étudiant recevra par courrier postal* avec accusé de réception son correspondant diplôme de **Mastère Spécialisé** délivré par **TECH Université Technologique**.

Le diplôme délivré par **TECH Université Technologique** indiquera la note obtenue lors du Mastère Spécialisé, et répond aux exigences communément demandées par les bourses d'emploi, les concours et les commissions d'évaluation des carrières professionnelles.

Diplôme: **Mastère Spécialisé en Intelligence Artificielle et Ingénierie des Connaissances**

N.º Horas Oficiales: **1.500 h.**



*Si l'étudiant souhaite que son diplôme version papier possède l'Apostille de La Haye, TECH EDUCATION 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 langage

tech université
technologique

Mastère Spécialisé Intelligence Artificielle et Ingénierie des Connaissances

- » Modalité: en ligne
- » Durée: 12 mois
- » Qualification: TECH Université Technologique
- » Intensité: 16h/semaine
- » Horaire: à votre rythme
- » Examens: en ligne

Mastère Spécialisé

Intelligence Artificielle et Ingénierie des Connaissances