SKILLIA

Catalogue des formations

Innovez, Protégez, Transformez : les formations IA, Cyber et Digital signées Skillia

2025-2026

www.skillia.fr

Sommaire

Nos Valeurs

Une démarche axée sur la qualité

Chez Skillia, nous concevons des formations sur mesure, pensées pour répondre aux spécificités de votre organisation, de votre marché et de votre environnement.

La qualité constitue le pilier central de notre approche : pertinence des thématiques, sélection rigoureuse des experts, méthodes pédagogiques éprouvées, organisation soignée et mise à jour continue de notre offre pour anticiper les évolutions technologiques et les besoins du marché.

L'innovation au service de la formation

Créativité, innovation, écoute, réflexion et rigueur sont au cœur de notre ADN.

Ces valeurs guident chacune de nos actions pour vous offrir des formations à la pointe : des approches pédagogiques modernes, des contenus actualisés et une expérience d'apprentissage stimulante, adaptée aux enjeux d'aujourd'hui.

Une écoute active des entreprises

Chaque entreprise bénéficie d'un interlocuteur dédié, véritable consultant en formation.

Sa mission : analyser vos besoins, recommander les solutions les plus pertinentes en matière de contenu, d'organisation, de budget et de financement, puis vous accompagner tout au long du déploiement.

Il veille à la réussite du projet, à l'atteinte de vos objectifs et à la satisfaction des apprenants.

Pourquoi choisir Skillia ?

Des intervenants d’exception

es formateurs Skillia sont des experts reconnus, dotés de plusieurs années d’expérience dans leurs domaines respectifs et ayant occupé des postes à responsabilité en entreprise.

Ils conçoivent et animent eux-mêmes leurs formations, en s’appuyant sur une double expertise : une solide expérience de terrain et une maîtrise pédagogique éprouvée — véritable gage de qualité et de pertinence.

Une offre en constante évolution

Chez Skillia, l’innovation est continue. Nos programmes de formation sont régulièrement actualisés pour intégrer les dernières avancées technologiques et répondre aux nouveaux défis des entreprises.

Une approche résolument opérationnelle

es formations Skillia sont pensées pour une mise en pratique immédiate. Conçues et animées par des professionnels expérimentés, elles comportent au minimum 60 % d’exercices pratiques et d’études de cas inspirés de situations réelles. L’organisation progressive des modules permet d’acquérir les compétences nécessaires à chaque étape, dans les conditions les plus efficaces.

Une réactivité à chaque étape

Nos consultants en formation sont à votre écoute pour définir rapidement les solutions les plus adaptées à vos besoins : sélection de cours ou de parcours, formations intra ou sur mesure, ingénierie pédagogique et dispositifs d’accompagnement spécifiques à votre entreprise.

Former avant de recruter : un levier gagnant !

La reconversion professionnelle est une approche de recrutement innovante et efficace. Elle consiste à identifier des talents motivés, dotés du potentiel nécessaire, puis à leur offrir un parcours de formation sur mesure pour développer les compétences indispensables à leur futur métier.

Cette stratégie représente une opportunité mutuellement bénéfique : elle permet aux entreprises de répondre à leurs besoins en compétences tout en offrant aux candidats une véritable montée en qualification et une nouvelle trajectoire professionnelle.

Avantages pour l'entreprise

Combler le manque de compétences clés sur le marché

Recruter autrement : des profils compétents, motivés et alignés avec vos valeurs

Être en phase avec les dernières avancées technologiques

Avantages pour les candidats

Une reconversion professionnelle vers des métiers d'avenir

Valoriser son employabilité et répondre aux besoins réels du marché du travail.

Informations Pratiques SKILLIA :

Public visé : Professionnels (salariés, indépendants) et particuliers.

Modalités : Formations disponibles en **distanciel** (classe virtuelle synchrone) ou en **présentiel** (dans les locaux de nos partenaires ou en intra-entreprise).

Délais d'accès : Entre **15 et 30 jours** à compter de la signature de la convention et selon le mode de financement sollicité.

Accessibilité Handicap :

- **En distanciel :** Nos parcours sont adaptables (assistance technique, supports numériques accessibles).
- **En présentiel :** Nous nous assurons avec nos partenaires que les lieux d'accueil respectent les normes d'accessibilité (ERP).
- *Pour toute étude personnalisée de vos besoins spécifiques, contactez notre référent handicap :* ***Skander Maalej****.*

Contact : administration@skillia.fr

Notre catalogue

www.skillia.fr

Intelligence Artificielle

Advanced Copilot & AI Agentic Workflows

Prix : 2 950 € HT

Durée : 4 jours (28 heures)

Public cible : Équipe de développement .NET (C#).

Prérequis : GitHub Copilot activé + Visual Studio Code installé et configuré. Expérience de développement .NET (C#). Accès à un environnement de développement fonctionnel avec les projets de l'équipe.

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Comprendre en profondeur les modèles de raisonnement, les LLMs Open Source et le prompting expert
- Concevoir et orchestrer des systèmes agentiques autonomes (Multi-Agents, Tool Use)
- Maîtriser GitHub Copilot dans un environnement .NET sur l'ensemble du SDLC
- Sécuriser, tester et documenter le code généré avec une posture de superviseur

Méthodes et moyens pédagogiques : Formation intensive 100 % hands-on dans l'IDE, appliquée à un environnement et des projets .NET : 10 ateliers pratiques et labs hands-on. Deux formateurs experts, deux approches (fondations GenAI & agentique, puis pratique avancée GitHub Copilot & SDLC .NET). Format présentiel.

Modalités d'évaluation

Évaluation par la réalisation des 10 ateliers pratiques et des labs hands-on tout au long de la formation.

Accessibilité : Formation accessible aux personnes en situation de handicap, contactez notre référent dédié.

Programme

1. **Jour 1 · L'évolution des LLMs**
   - Partie I — Théorie, fondations & l'ère de l'agentique : compréhension approfondie des modèles, LLMs Open Source et conception de systèmes agentiques.
   - Matin — Rappel des bases et prompting expert.
   - Le passage des modèles prédictifs aux modèles de raisonnement natifs
   - Gestion du contexte : fenêtres géantes, attention et prévention de la perte d'information ("lost in the middle")
2. **Jour 1 · Prompt Engineering Expert**
   - Techniques avancées (Chain-of-Thought, Few-Shot structuré, Meta-Prompting) adaptées à l'ingénierie logicielle
3. **Jour 1 · Panorama des LLMs locaux et Open Source**
   - Après-midi — L'écosystème des LLMs Open Source.
   - Exploration de Llama 3, Mistral et de l'écosystème open source
   - Lab pratique : Déploiement local d'un modèle et interaction via API
   - Limites d'exécution : Déterminisme vs Créativité, gestion des hallucinations complexes
4. **Jour 2 · Le Mode Agent — Nouveaux paradigmes**
   - Matin — Le Mode Agent : concepts & fonctionnement.
   - Définition du "Mode Agent" et nouveaux paradigmes (Devin, Claude Engineer)
   - La boucle de rétroaction autonome : Perception, Réflexion, Action, Critique
   - Tool Use (Function Calling) : interaction autonome de l'IA avec un terminal, un système de fichiers ou une API
5. **Jour 2 · Patterns agentiques & systèmes multi-agents**
   - Après-midi — L'ingénierie agentique (Agentic Workflows).
   - Les patterns agentiques clés : Réflexion (Self-Correction), Planification et Systèmes Multi-Agents (collaboration Dev, QA, SecOps)
   - Lab pratique : Création d'un workflow agentique en Python/LangChain pour analyser un log d'erreur, formuler une hypothèse et proposer un correctif
6. **Jour 3 · Contexte & Workspace**

- Partie II — Pratique avancée GitHub Copilot & SDLC augmenté : 100 % hands-on dans l'IDE, appliqué à un environnement et des projets .NET.
- Focus : 100 % hands-on dans l'IDE, appliqué à un environnement .NET. Matin — Spécifications & Architecture .NET.
- Atelier 1 : Maîtriser le contexte de Copilot — utilisation avancée des variables (@workspace, #file, #selection) pour ancrer les réponses dans l'architecture .NET existante

7. Jour 3 · Génération d'Architecture

- Atelier 2 : Traduire des User Stories complexes en diagrammes et squelettes de code structurés pour des microservices en C# — détection des cas limites fonctionnels via Copilot

8. Jour 3 · Refactoring massif

- Après-midi — Revues de code (PR) & dette technique.
- Atelier 3 : Stratégies de refactoring sur du code C# legacy — application de design patterns et migrations de versions de frameworks .NET via des prompts experts

9. Jour 3 · Code Review IA

- Atelier 4 : Automatisation du premier niveau de revue (style, performance, anti-patterns) et nettoyage de fichiers de code obsolètes avec Copilot

10. Jour 4 · Génération de tests robustes

- Focus : tests, sécurisation et intégration des workflows autonomes dans le quotidien du développeur. Matin — QA, Tests & Sécurité.
- Atelier 5 : Création autonome de stratégies de tests (unitaires via xUnit/NUnit, intégration) basées sur l'analyse comportementale du code .NET

11. Jour 4 · Sécurité by Design

- Atelier 6 : Audit de code avec Copilot — identifier, bloquer et corriger l'injection de vulnérabilités (OWASP Top 10) directement dans l'IDE

12. Jour 4 · Création & orchestration multi-agents

- Après-midi — Agents, Personnalisation & Documentation.
- Atelier 7 : Création et utilisation d'agents IA spécialisés — orchestration multi-agents via Microsoft Semantic Kernel ou workflows similaires en .NET

13. Jour 4 · Personnalisation via Skills & Hooks

- Atelier 8 : Développer et brancher des "Skills" personnalisés et utiliser des "Hooks" pour étendre les capacités des agents

14. Jour 4 · Prototypage UI/UX

- Atelier 9 : Utilisation de l'IA pour accélérer la conception et l'intégration de composants UI/UX avec connexion fluide aux services backend .NET

15. Jour 4 · Reporting & Documentation automatisée

- Atelier 10 : Génération de rapports d'analyse de code complets et automatisation de la documentation technique (API, architecture, commentaires de code structurés)

Introduction à l'Intelligence Artificielle

Prix : 700 € H.T.

Durée : 1 jour

Public cible : Professionnels de tous secteurs souhaitant comprendre les enjeux de l'IA (RH, marketing, finance, santé, industrie, etc.). Managers, chefs de projet, décideurs ou consultants impliqués dans des projets de transformation numérique. Profils non techniques ou techniques cherchant une première approche structurée de l'IA. Toute personne curieuse de comprendre les mécanismes et impacts de l'IA

Prérequis : Aucun prérequis technique n'est nécessaire. Une culture générale numérique ou métier est un plus pour l'analyse des cas d'usage

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Comprendre les concepts fondamentaux de l'intelligence artificielle et ses principales approches
- Identifier les grandes familles d'algorithmes : approche symbolique, apprentissage automatique, réseaux neuronaux
- Distinguer les domaines d'application et les limites actuelles de l'IA
- Analyser des cas d'usage concrets de l'IA dans différents secteurs d'activité
- Déconstruire les idées reçues et mieux cerner les impacts réels de l'IA

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. Introduction à l'intelligence artificielle

- Définitions de l'IA : symbolique, connexionniste, hybride
- Courte histoire de l'IA et des grandes ruptures technologiques
- Différences entre IA, machine learning et deep learning

2. Panorama des approches algorithmiques

- Raisonnement symbolique : règles, systèmes experts
- Apprentissage automatique (supervisé, non supervisé, renforcement)
- Réseaux de neurones et deep learning : principes et exemples
- IA générative : émergence et applications récentes

3. Domaines d'application de l'IA

- Vision par ordinateur (reconnaissance d'images, OCR, vidéosurveillance intelligente)
- Traitement automatique du langage naturel (chatbots, analyse de sentiment, traduction)
- Recommandation (e-commerce, contenus, personnalisation)
- Robotique et systèmes autonomes
- Analyse prédictive et aide à la décision

4. Cas d'usage et retours d'expérience sectoriels

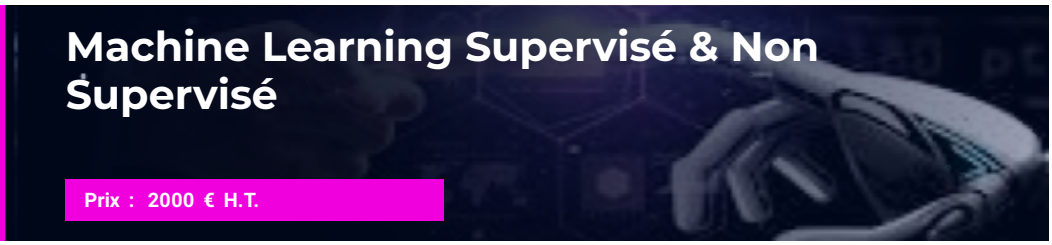
- Santé : diagnostic assisté, médecine personnalisée
- Industrie : maintenance prédictive, automatisation
- Marketing : segmentation, scoring client
- RH : recrutement, détection des soft skills
- Services publics, éducation, finance...

5. Enjeux, limites et perspectives

- IA faible vs IA forte : mythe ou réalité ?
- Biais algorithmiques, explicabilité, gouvernance éthique

- L’IA dans les organisations : risques, opportunités et changements culturels
- Cadres juridiques et réglementaires émergents (RGPD, AI Act...)

Machine Learning Supervisé & Non Supervisé



Prix : 2000 € H.T.

Durée : 3 jours

Public cible : Développeurs, analystes, ingénieurs ou data scientists débutants ayant une première expérience en Python. Professionnels techniques souhaitant approfondir leur compréhension des algorithmes de machine learning. Participants ayant suivi une formation d'introduction à la data science ou équivalent

Prérequis : Maîtrise des bases de Python (structures de données, fonctions, bibliothèques standards). Connaissances élémentaires en statistiques (variance, corrélation, distribution). Une première expérience avec Pandas et NumPy est fortement recommandée

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Comprendre les concepts fondamentaux du machine learning supervisé (régression, classification)
- Appliquer des techniques de machine learning non supervisé (clustering, réduction de dimensionnalité)
- Utiliser la bibliothèque Scikit-learn pour entraîner, évaluer et améliorer des modèles
- Diagnostiquer le sur-apprentissage, utiliser la validation croisée, ajuster les hyperparamètres
- Mettre en œuvre un pipeline complet de modélisation sur des jeux de données réels

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. Introduction au machine learning avec Scikit-learn

- Rappels sur les données tabulaires et la préparation des données
- Principes du machine learning supervisé vs non supervisé
- Scikit-learn : jeux de données intégrés, interface standard des modèles
- Pipelines de transformation, scalers, encodage des variables

2. Modèles supervisés : régression et classification

- Régression linéaire et évaluation des erreurs (RMSE, MAE)
- Régression logistique, arbres de décision, forêts aléatoires
- SVM : classification à grande marge
- Choix des métriques : précision, rappel, F1-score, courbes ROC

3. Apprentissage non supervisé

- Clustering : k-means, DBSCAN (notions)
- Évaluation de la cohérence des clusters (inertie, silhouette)
- Réduction de dimensionnalité : PCA (analyse en composantes principales)
- Visualisation des données en 2D/3D après réduction

4. Validation, réglages et robustesse

- Validation croisée : holdout, k-fold
- Sur-apprentissage et biais/variance
- Recherche d'hyperparamètres : GridSearchCV, RandomizedSearchCV
- Interprétabilité des modèles : importance des variables, SHAP (introduction)

5. Travaux pratiques sur jeux de données réels

- Chargement de jeux de données du monde réel (Kaggle, open data)
- Préparation, exploration, modélisation, évaluation
- Présentation orale ou notebook des résultats

- Retours d’expérience et discussion des choix algorithmiques

Deep Learning avec TensorFlow ou PyTorch

Prix : 2100 € H.T.

Durée : 3 jours

Public cible : Développeurs, data scientists, ingénieurs IA ou R&D souhaitant se spécialiser en deep learning. Participants ayant une première expérience en machine learning supervisé. Toute personne amenée à concevoir ou intégrer des modèles de deep learning dans des projets d'IA

Prérequis : Bonne maîtrise de Python et des bibliothèques NumPy / Pandas. Connaissances de base en machine learning (modèles, sur-apprentissage, métriques). Notions élémentaires en calcul matriciel et en statistiques recommandées.

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Comprendre le fonctionnement des réseaux de neurones artificiels (MLP, CNN, RNN)
- Concevoir et entraîner des modèles simples avec TensorFlow ou PyTorch
- Appliquer le deep learning à des problématiques concrètes (images, texte, séries temporelles)
- Maîtriser les notions clés d'optimisation : fonction de coût, gradient, sur-apprentissage
- Utiliser les techniques modernes comme Dropout, Batch Normalization ou le transfer learning

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. Introduction au deep learning et aux frameworks

- Qu'est-ce que le deep learning ? Rappels sur le machine learning
- Comparaison TensorFlow vs PyTorch : philosophie, syntaxe, écosystèmes
- Présentation des outils : Google Colab, Jupyter, GPU disponibles

2. Réseaux de neurones multicouches (MLP)

- Perceptron multicouche : architecture, fonctions d'activation (ReLU, Sigmoid)
- Calcul du gradient et rétropropagation (backpropagation)
- Entraînement, évaluation, choix de la fonction de perte (MSE, CrossEntropy)
- Visualisation de l'apprentissage, learning rate, epochs, batch size

3. Réseaux convolutifs (CNN)

- Convolutions, pooling, padding : principes et rôle dans la vision par ordinateur
- Architecture type pour classification d'images (MNIST, CIFAR)
- Optimisation : Dropout, Batch Normalization, early stopping
- Visualisation des filtres, erreurs typiques, ajustement des hyperparamètres

4. Réseaux récurrents (RNN de base)

- Introduction aux séquences et séries temporelles
- RNN simples : mémoire courte, structure temporelle
- Notions de LSTM / GRU (présentation sans implémentation détaillée)
- Applications possibles : prévision, texte, détection d'événements

5. Bonnes pratiques et extension des modèles

- Préparation des données : normalisation, encodage, augmentation
- Éviter l'overfitting : régularisation, Dropout, data augmentation
- Introduction au Transfer Learning : fine-tuning de modèles pré-entraînés (ResNet, BERT selon contexte)
- Premiers pas vers la production : export de modèles, compatibilité ONNX, intégration dans une API

Prompt Engineering et IA Générative

Prix : 1800 € H.T.

Durée : 2 jours

Public cible : Professionnels de tous domaines souhaitant tirer parti de l'IA générative dans leur métier. Managers, consultants, créateurs de contenu, responsables innovation, métiers du droit, RH, marketing, etc. Développeurs ou profils techniques découvrant le potentiel des LLMs et outils no-code.

Prérequis : Aucune compétence technique requise (pas de code nécessaire). Une familiarité avec les outils numériques et la logique des workflows est souhaitée. Une première expérimentation personnelle avec des IA génératives (ChatGPT, Copilot, etc.) est un plus

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Comprendre le fonctionnement des modèles de langage (LLM) et leurs limites
- Structurer des prompts efficaces pour générer, reformuler, résumer, analyser ou interagir avec des contenus
- Adapter leurs prompts à différents cas d'usage professionnels : RH, marketing, IT, juridique, etc.
- Utiliser des outils no-code intégrant de l'IA générative pour augmenter leur productivité
- Distinguer les approches prompt, RAG, fine-tuning et connaître les enjeux d'usage éthique

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. Comprendre les IA génératives et les LLMs

- Qu'est-ce qu'un LLM ? ChatGPT, Claude, Mistral, Gemini : aperçu des modèles du marché
- Notions fondamentales : pré-entraînement, génération, embeddings, hallucination
- Introduction à RAG, fine-tuning, agents, context window et limitations
- Différences entre IA conversationnelle, IA générative texte, image, code

2. Fondamentaux du prompt engineering

- Structure d'un bon prompt : rôle, instruction, contexte, format attendu
- Techniques : zero-shot, few-shot, chain-of-thought, prompt imbriqués
- Comparaison de résultats selon la formulation et l'intention
- Astuces pour des prompts robustes, réutilisables, adaptables

3. Panorama des outils IA générative

- Utilisation avancée de ChatGPT (instructions personnalisées, GPTs personnalisés, plugins, navigation web)
- Découverte d'outils no-code : Notion AI, Canva AI, Copilot Microsoft, Jasper, Gamma
- Génération multimodale : images, textes, tableaux, slides, code

4. Cas d'usage métiers et ateliers

- Marketing : génération de contenus, segmentation, stratégie SEO
- RH : aide au recrutement, évaluation de CV, rédaction de fiches de poste
- Juridique : résumé de documents, extraction d'informations, relecture assistée
- IT : rédaction technique, génération de documentation, assistants à la modélisation
- Ateliers pratiques : scénarios sur cas concrets + coaching personnalisé

MLOps : Déploiement de Modèles IA

Prix : 1700 € H.T.

Durée : 2 jours

Public cible : Développeurs, data engineers, data scientists souhaitant industrialiser leurs modèles IA. Équipes techniques impliquées dans des projets de déploiement et de mise en production de solutions d'apprentissage automatique. DevOps souhaitant intégrer des modèles d'IA dans des chaînes CI/CD ou dans des infrastructures conteneurisées

Prérequis : Bonne maîtrise de Python et des bibliothèques de machine learning (scikit-learn, pandas, etc.). Notions de base en Docker, Git, REST API appréciées. Avoir déjà entraîné un modèle ML simple (régression ou classification)

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Préparer un modèle machine learning pour son déploiement en production
- Conteneuriser un modèle avec Docker et l'exposer via une API web
- Mettre en place un pipeline de CI/CD adapté aux workflows IA
- Utiliser des outils de traçabilité et d'orchestration de modèles tels que MLflow et Airflow
- Comprendre les bonnes pratiques de monitoring et de versioning des modèles

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. Introduction au MLOps

- Définition du MLOps : objectifs, enjeux, différences avec DevOps
- Cycle de vie d'un modèle IA : entraînement, test, déploiement, surveillance
- Architecture type d'un projet IA industrialisé

2. Packaging et sauvegarde de modèles

- Sauvegarde de modèles avec Pickle, Joblib, ONNX
- Portabilité, compatibilité entre environnements (CPU/GPU)
- Gestion des dépendances avec virtualenv / requirements.txt

3. Création d'API pour modèles IA

- Mise en place d'une API REST avec FastAPI ou Flask
- Sérialisation des données d'entrée et sortie (JSON, NumPy, Pandas)
- Tests unitaires de l'API et endpoints prédictifs
- Sécurisation et documentation automatique (Swagger/OpenAPI)

4. Conteneurisation avec Docker

- Création d'une image Docker pour le modèle et l'API
- Structure d'un Dockerfile optimisé pour l'IA
- Déploiement local ou distant avec Docker CLI
- Notions de gestion des volumes, logs et configuration

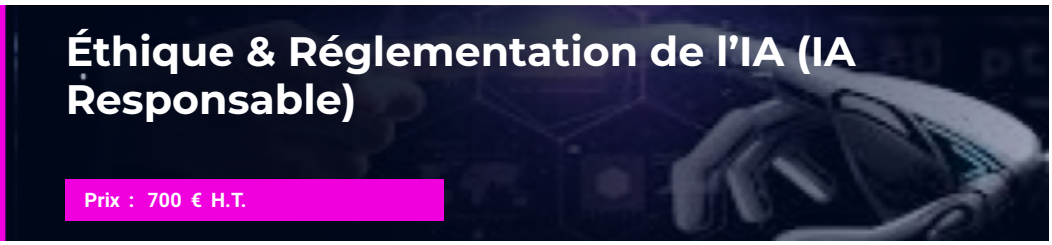
5. Automatisation avec CI/CD

- Introduction à GitHub Actions : pipelines simples pour ML
- Automatisation des tests, du packaging et du déploiement
- Bonnes pratiques de versioning du code et des modèles
- Intégration dans une chaîne DevOps existante

6. Orchestration et suivi des modèles

- Présentation de MLflow : tracking des expériences, versioning des modèles
- Introduction à Apache Airflow : orchestration de tâches ML
- Détection des dérives de performance (concept drift)
- Perspectives : monitoring en production, scalability, cloud-native ML

Éthique & Réglementation de l'IA (IA Responsable)



Prix : 700 € H.T.

Durée : 1 jour

Public cible : Managers, chefs de projet, responsables innovation ou transformation numérique. Développeurs IA, data scientists, architectes techniques impliqués dans des projets IA. Juristes, DPO, responsables conformité ou métiers concernés par la régulation de l'IA. Toute personne souhaitant intégrer des principes éthiques dans l'usage ou la conception de systèmes IA

Prérequis : Bonne culture numérique ou première expérience avec des projets impliquant de l'intelligence artificielle. Notions de base en gestion de la donnée (RGPD) recommandées. Aucun prérequis technique n'est nécessaire.

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Identifier les biais algorithmiques et leurs impacts sociétaux ou organisationnels
- Comprendre les enjeux liés à la transparence, l'équité, l'autonomie humaine dans les systèmes IA
- Appliquer les exigences du RGPD et anticiper celles de l'AI Act européen
- Mobiliser des outils et démarches pour améliorer l'explicabilité des modèles (XAI)
- Intégrer des principes d'IA responsable dans les phases de conception et d'évaluation des projets IA

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. Enjeux éthiques de l'IA

- Définition de l'IA responsable et des grands principes éthiques (non-discrimination, sécurité, supervision humaine)
- Biais algorithmiques : sources, types et impacts (biais de données, biais de conception, biais d'usage)
- Exemples concrets : IA et recrutement, justice prédictive, reconnaissance faciale

2. Transparence et explicabilité

- Pourquoi l'explicabilité est-elle cruciale ?
- Introduction aux techniques d'IA explicable (XAI) : SHAP, LIME, arbres de décision, règles locales
- Limites de l'explication dans les modèles complexes (deep learning)
- Rôle des interfaces utilisateurs et de la documentation des modèles

3. Cadre réglementaire européen

- Rappels sur le RGPD : données personnelles, consentement, profilage, droit à l'explication
- Présentation de l'AI Act : structure du texte, niveaux de risque (inacceptable, élevé, modéré, minimal)
- Obligations selon le niveau de risque : transparence, documentation, conformité
- Interactions entre RGPD et AI Act : responsabilités croisées et complémentarités

4. Mise en œuvre d'une IA responsable

- Bonnes pratiques : audit de biais, comité éthique, documentation des datasets, design éthique
- Intégration des principes dès la phase de cadrage d'un projet IA (Privacy by Design, Fairness by Design)
- Étude de cas : diagnostic éthique et conformité d'un projet IA simulé
- Outils et ressources : fiches d'évaluation éthique, checklists, cadres de gouvernance

IA pour non-tech (Acculturation à l'Intelligence Artificielle)

Prix : 700 € H.T.

Durée : 1 jour

Public cible : Professionnels non techniques : fonctions RH, marketing, communication, support client, gestion administrative, commerce. Managers ou collaborateurs impliqués dans des projets d'innovation ou de transformation digitale.

Prérequis : Aucun prérequis technique n'est nécessaire. Une familiarité avec les outils bureautiques et digitaux est un plus. L'envie d'expérimenter en situation réelle est encouragée.

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Comprendre ce qu'est (et ce que n'est pas) l'intelligence artificielle dans un cadre professionnel
- Identifier les principaux outils d'IA no-code et leurs apports métiers
- Utiliser l'IA pour rédiger, résumer, classer, automatiser des tâches simples
- Évaluer les limites, risques et bonnes pratiques associés à l'usage de l'IA au travail
- Gagner en efficacité et en autonomie dans leur quotidien grâce à l'IA générative

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. Introduction à l'IA pour tous

- Qu'est-ce que l'IA ? Définitions accessibles, idées reçues et réalité
- Panorama des grandes familles d'IA : classification, génération, recommandation
- IA générative (textes, images, résumés) : pourquoi cet engouement aujourd'hui ?

2. Outils d'IA no-code pour les métiers

- Présentation d'outils accessibles : ChatGPT, Notion AI, Copilot, Canva IA, Gamma, etc.
- Comparaison des outils selon les usages : rédaction, analyse, brainstorming, création visuelle
- Critères pour choisir un outil adapté à son besoin

3. Usages concrets dans les fonctions support

- RH : rédaction de fiches de poste, synthèse de candidatures
- Marketing / Com : génération de posts, emailing, idées créatives
- Support client : rédaction de réponses types, FAQ assistée
- Commerce / ADV : synthèse d'appels, propositions commerciales automatiques

4. Études de cas et démonstrations pratiques

- Cas concrets analysés et testés en groupe
- Exercices guidés : rédiger, structurer, résumer ou automatiser une tâche avec l'IA
- Débrief collectif sur les résultats obtenus et leurs limites

5. Bonnes pratiques et limites d'usage

- L'IA comme assistant, pas comme décisionnaire
- Identifier les erreurs, biais ou hallucinations des modèles
- Respect des données, confidentialité, RGPD
- Responsabilité de l'humain face aux contenus générés

Outils No-Code & Low-Code : Automatiser et Prototyper sans Développer

Prix : 1910 € H.T.

Durée : 2 jours

Public cible : Responsables métiers, chefs de projet, consultants, fonctions support ou innovation. Product owners, UX designers, responsables marketing ou opérationnels souhaitant créer rapidement des outils internes. Toute personne souhaitant automatiser des tâches ou prototyper des solutions numériques sans programmation complexe.

Prérequis : Bonne culture numérique et maîtrise des outils bureautiques ou collaboratifs (Google Workspace, Office 365…). Aucune compétence en développement n'est requise. Esprit logique et appétence pour les outils digitaux recommandés.

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Comprendre les concepts clés du no-code et du low-code, leurs cas d'usage et leurs limites
- Créer des automatisations simples (workflows, intégrations, reporting) avec des outils visuels
- Concevoir des applications métiers, prototypes ou tableaux de bord sans écrire de code
- Choisir les bons outils selon leurs besoins (formulaires, automatisation, bases de données, UI…)
- Gagner en autonomie pour tester des idées ou automatiser des processus internes

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. Introduction au no-code / low-code

- Définitions et différences : no-code, low-code, full-code
- Avantages, limites, cas d'usage typiques (MVP, automatisation, outils internes)
- Panorama de l'écosystème : Airtable, Zapier, Make, Glide, Notion, Softr, AppSheet, PowerApps, etc.

2. Construire une base de données visuelle

- Introduction à Airtable ou Notion DB
- Structuration de données : tables, vues, filtres, formulaires
- Relations entre tables, vues partagées, collaboration

3. Automatiser ses tâches et workflows

- Découverte de Zapier, Make (ex-Integromat) ou Power Automate
- Cas pratiques :
- Envoyer un email après la soumission d'un formulaire
- Générer un document ou un ticket automatiquement
- Publier sur les réseaux sociaux à partir d'un contenu validé
- Bonnes pratiques d'automatisation (déclencheurs, erreurs, logs)

4. Créer une application métier sans coder

- Utilisation de Glide, Softr, AppSheet ou PowerApps
- Génération d'une interface (mobile/web) à partir de données (Google Sheets, Airtable)
- Personnalisation des écrans, formulaires, filtres, règles métier
- Déploiement et partage de l'application en interne

5. Ateliers pratiques et étude de cas

- Cas métier au choix : gestion de candidatures, CRM simple, reporting automatisé, gestion d'événement, FAQ interactive
- Construction d'un prototype ou automatisation en binôme ou individuel
- Démonstrations croisées entre participants, retours d'expérience

Cybersécurité

Analyste SOC : Security Operations Center

-header-cybersecurite.png

Prix : 5 180 € HT

Durée : 8 jours (56 heures)

Public cible : Techniciens et administrateurs Systèmes & Réseaux, responsables informatiques & responsables techniques, consultants en sécurité & ingénieurs, architectes réseaux & chefs de projets.

Prérequis : Avoir des connaissances en réseau et posséder des connaissances en cybersécurité.

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Le stagiaire sera capable d'assurer les fonctions d'analyste d'un SOC : la détection et l'analyse des intrusions, l'anticipation et la mise en place des protections nécessaires.
- Connaître le rôle et les missions d'un analyste SOC et maîtriser les fondamentaux de la cybersécurité défensive
- Utiliser les outils et technologies du SOC et analyser · corréler les événements de sécurité
- Gérer les incidents de sécurité et rédiger des rapports techniques clairs et exploitables
- Travailler en coordination avec les autres équipes cyber et assurer une veille sur les cybermenaces

Méthodes et moyens pédagogiques : Formation intensive orientée prise de poste : 16 ateliers pratiques, un par demi-journée. Modalité : classe virtuelle.

Modalités d'évaluation

QCM de validation des compétences en fin de session.

Accessibilité : Accessibilité Handicap : en distanciel, nos parcours sont adaptables (assistance technique, supports numériques accessibles). Pour toute étude personnalisée de vos besoins spécifiques, contactez notre référent handicap : Skander Maalej — skander.maalej@skillia.fr

Programme

1. Jour 1 · Panorama de la cybersécurité défensive

- Jour 1 — Introduction à la cybersécurité défensive & fondamentaux de l'analyste SOC. Matin — Introduction à la cybersécurité défensive.
- Fondamentaux de la cybersécurité défensive : principes de protection, détection et réponse
- Actifs, menaces et surfaces d'attaque (CIA, MITRE ATT&CK)
- Organisation des métiers cyber & rôle du SOC
- Atelier : Cartographier les menaces d'un SI avec MITRE ATT&CK

2. Jour 1 · Missions, chaîne de valeur & outillage SOC

- Après-midi — Fondamentaux d'un analyste SOC.
- Missions de l'analyste SOC (L1, L2, L3)
- Chaîne de valeur SOC : surveillance, détection, investigation, reporting
- Coordination avec les autres équipes (CERT, CSIRT, Blue Team, Red Team)
- Rôle de l'analyste dans la chaîne de défense et le cycle de vie d'un incident
- Panorama des outils SOC (SIEM, SOAR, EDR, Threat Intel)
- Atelier : Simulation de triage d'alertes dans un SIEM

3. Jour 2 · Typologie des attaques & vulnérabilités

- Jour 2 — Panorama des menaces cyber & veille. Matin — Panorama des menaces cyber.
- Typologie des attaques (phishing, ransomware, DDoS, APT)
- Exploitation des vulnérabilités (CVE, CVSS)
- Ingénierie sociale et attaques hybrides
- Atelier : Analyse d'un email suspect (phishing)

4. Jour 2 · Veille & Threat Intelligence

- Après-midi — Effectuer sa veille cyber.

- Introduction à la veille cyber (CERT-FR, OSINT, Threat Intel feeds)
- Intégrer la veille dans le quotidien d'un analyste SOC
- Notions de Threat Intelligence (IoC : hash, IP, domaine)
- Atelier : Construire un mini-dashboard de veille cyber

5. Jour 3 · Architecture et principes d'un SIEM

- Jour 3 — SIEM & collecte de logs. Matin — SIEM.
- Architecture et principes d'un SIEM (Splunk, ELK, QRadar)
- Sources de logs : systèmes, réseaux, applicatifs, cloud
- Parsing et normalisation des événements
- Atelier : Collecte et visualisation de logs via ELK

6. Jour 3 · Corrélation & investigation dans un SIEM

- Après-midi — Collecte & corrélation de logs.
- Corrélation de logs pour détecter des patterns simples
- Investigation basique dans un SIEM (authentifications suspectes, brute force)
- Analyser et corréler les événements de sécurité pour identifier des incidents potentiels
- Atelier : Corrélation d'événements réseau et système dans un SIEM

7. Jour 4 · Détection et réponse sur les endpoints

- Jour 4 — EDR & surveillance réseau en temps réel. Matin — Endpoint Detection & Response (EDR).
- Détection et réponse sur les postes de travail (EDR)
- Signatures vs détection comportementale
- Investigation d'un poste compromis
- Atelier : Analyse d'un poste infecté via un EDR

8. Jour 4 · Capture, analyse et détection d'anomalies réseau

- Après-midi — Réseaux et surveillance en temps réel.
- Capture et analyse de trafic réseau (Wireshark, Zeek)
- Signatures IDS/IPS (Snort, Suricata)
- Surveillance des flux Netflow
- Détection d'anomalies réseau et exfiltration de données
- Atelier pratique : Analyse d'un trafic malveillant avec Wireshark

9. Jour 5 · De la détection à la qualification

- Jour 5 — Méthodologie d'investigation & gestion d'incident. Matin — Méthodologie d'investigation SOC.
- Processus de détection →analyse →qualification
- Faux positifs, incidents confirmés
- Introduction au reporting technique (rédiger une qualification d'incident)
- Atelier : Qualification d'une alerte brute en incident confirmé

10. Jour 5 · Cycle de vie de l'incident

- Après-midi — Processus de gestion d'incident.
- Gestion d'incident (NIST SP 800-61, ISO 27035 – simplifié)
- Étapes : préparation, confinement, éradication, retour d'expérience
- Gérer les incidents de sécurité tout au long de leur cycle de vie
- Communication interne et externe en cas d'incident
- Atelier : Table-top exercise sur une attaque ransomware

11. Jour 6 · Acquisition et analyse des preuves

- Jour 6 — Investigation forensic & durcissement. Matin — Investigation et forensic.
- Introduction au forensic : acquisition et conservation des preuves
- Journaux systèmes, disques, mémoire
- Présentation d'outils (Volatility, Autopsy, FTK Imager – démonstration)
- Atelier : Extraction de preuves sur une machine compromise

12. Jour 6 · Durcissement, vulnérabilités & sécurité applicative

- Après-midi — Durcissement & sécurité applicative.
- Bonnes pratiques de durcissement (systèmes & réseaux)
- Gestion des vulnérabilités (scanners, patch management)
- Sécurité applicative vue du SOC (OWASP Top 10 simplifié)
- Atelier : Détection d'une injection SQL dans des logs applicatifs

13. Jour 7 · SOAR, playbooks & scripts

- Jour 7 — Automatisation (SOAR) & reporting. Matin — Automatisation et orchestration SOC (SOAR).
- Automatisation SOC : SOAR (principes, playbooks simples)

- Scripts utiles (Python / PowerShell – niveau accessible)
- Atelier : Script Python simple pour extraire des IoC de logs

14. Jour 7 · Reporting technique & capitalisation

- Après-midi — Reporting & communication.
- Reporting technique SOC (rapports post-incident, indicateurs MTTD/MTTR)
- Communication vers RSSI, DSI, COMEX
- Capitalisation et RETEX (knowledge base SOC)
- Rédiger des rapports techniques clairs et exploitables pour documenter les incidents
- Atelier : Rédaction d'un rapport d'incident + présentation synthétique au COMEX

15. Jour 8 · Compromission d'un SI — de bout en bout

- Jour 8 — Étude de cas fil rouge & préparation à la prise de poste. Matin — Étude de cas fil rouge.
- Étude de cas complète : compromission d'un SI
- Analyse initiale des logs et détection
- Investigation multi-sources (SIEM, EDR, réseau)
- Containment et remédiation
- Atelier : Exercice pratique complet de bout-en-bout

16. Jour 8 · Threat Hunting, éthique & perspectives

- Après-midi — Préparation à la prise de poste analyste SOC.
- Threat Hunting proactif (notions de base)
- Anticipation des menaces émergentes (IoT, supply chain, IA – vulgarisé)
- Éthique et aspects juridiques de la cybersurveillance
- Certifications et perspectives (CEH, CompTIA Security+, SOC Analyst)
- Atelier mise en situation : Simulation d'une première journée en SOC

Sensibilisation à la cybersécurité – Bonnes pratiques pour tous les collaborateurs

-header-cybersecurite.png

Prix : 1010 € H.T.

Durée : 1 jour

Public cible : Tous les collaborateurs de l'entreprise, tous métiers confondus. Services RH, finance, marketing, production, logistique, etc. Managers, assistants, télétravailleurs, utilisateurs de services numériques

Prérequis : Aucun prérequis technique nécessaire. Accès à un ordinateur et à une messagerie professionnelle pour les exercices pratiques.

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Identifier les principaux types de menaces numériques (phishing, ransomware, fuites…)
- Adopter les bons réflexes de cybersécurité au quotidien, au bureau comme en télétravail
- Protéger efficacement ses données personnelles et professionnelles
- Comprendre son rôle dans la cybersécurité de l'organisation
- Réagir correctement en cas d'incident ou de suspicion de fraude

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. Introduction à la cybersécurité en entreprise

- Qu'est-ce que la cybersécurité ?
- Les risques pour l'entreprise et les individus
- Le facteur humain au cœur des incidents
- Le rôle de chacun dans la chaîne de sécurité

2. Menaces courantes et méthodes d'attaque

- Phishing et spear phishing : reconnaître un email frauduleux
- Ransomware : mécanismes, exemples, impacts réels
- Usurpation d'identité, ingénierie sociale, faux ordres de virement
- Deepfakes, QR codes piégés, arnaques téléphoniques

3. Bonnes pratiques numériques au quotidien

- Gestion des mots de passe : robustesse, MFA, gestionnaires
- Sécurité des appareils : mises à jour, antivirus, verrouillage
- Messagerie professionnelle : pièces jointes, liens, contacts suspects
- Naviguer en sécurité : https, cookies, réseaux Wi-Fi publics
- Sécurité des données : sauvegardes, partage, confidentialité

4. Travail à distance et mobilité

- Sécuriser son environnement de télétravail
- VPN, postes sécurisés, outils cloud en entreprise
- Risques liés au BYOD et aux appareils personnels
- Réflexes en déplacement : Wi-Fi, clés USB, impressions

5. Réagir face à un incident ou une suspicion

- Que faire en cas de mail douteux ou comportement anormal ?
- Procédure interne d'alerte / qui contacter
- Importance de ne pas "cacher" un incident
- Comportement à adopter après une fuite ou une compromission

6. Réglementation et cadre légal

- Notions clés du RGPD applicables aux salariés
- Confidentialité, consentement, protection des données personnelles
- Sensibilisation aux obligations légales (charte informatique, politiques SSI)

Sécurité offensive – Introduction à l'Ethical Hacking

-header-cybersecurite.png

Prix : 2100 € H.T.

Durée : 3 jours

Public cible : Développeurs, administrateurs systèmes ou réseaux débutants en cybersécurité. Étudiants, techniciens ou profils non spécialisés souhaitant découvrir les bases du pentest. Responsables IT ou chefs de projets voulant comprendre les méthodes d'attaque et les parades.

Prérequis : Bonnes bases en systèmes (Windows/Linux) et en réseaux (IP, ports, protocoles). Connaissances générales en informatique (structure d'une application, navigateur, serveurs). Aucun prérequis en cybersécurité ni en hacking nécessaire.

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Comprendre le rôle et la posture d'un hacker éthique dans un cadre légal
- Identifier les étapes d'un test d'intrusion standard
- Manipuler les outils essentiels de reconnaissance, scan et analyse de vulnérabilités
- Repérer les failles de sécurité courantes sur le web, les systèmes ou les réseaux
- Savoir interpréter un rapport de vulnérabilités et appliquer les contre-mesures de base

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. Introduction à l'ethical hacking

- Définitions : hacker, white hat, black hat, pentester
- Objectifs d'un test d'intrusion
- Types de tests : black box, white box, grey box
- Cadre légal : RGPD, CNIL, LOPSSI, chartes SSI
- Organisation d'un pentest (phases et livrables)

2. Reconnaissance et scan de vulnérabilités

- Techniques de reconnaissance passive : WHOIS, DNS, Shodan
- Scan de ports avec Nmap : découverte, bannières, services
- Identification des failles : CVE, bases de vulnérabilités publiques
- Introduction aux scanners : Nikto, Nessus, OpenVAS (démonstration)

3. Introduction au pentest Web

- Vulnérabilités les plus fréquentes (OWASP Top 10)
- Injection simple : formulaire, URL, cookies
- Détection XSS, SQLi, mauvaise configuration
- Utilisation de Burp Suite Community (sniffing et test interactif)

4. Pentest sur systèmes et réseaux

- Exploitation d'un partage mal configuré (SMB, FTP)
- Notions de brute force sur services : SSH, RDP
- Introduction à Metasploit Framework (structure, modules)
- Découverte de failles sur systèmes obsolètes (cas pratiques)

5. Sécurité Windows et Active Directory (vue d'ensemble)

- Cartographie d'un domaine AD simplifié
- Détection de comptes faibles et partages non sécurisés
- Notions de Kerberos, hash, enumeration basique (demo guidée)

6. Bonnes pratiques de sécurité défensive

- Renforcement des services exposés
- Principes de défense en profondeur
- Patch management et gestion des droits
- Importance du principe de moindre privilège

7. Atelier pratique simplifié

- Mise en place d'une mini-infrastructure cible dans un environnement lab
- Scénarios simples : scan, identification de faille, exploitation guidée
- Restitution d'un mini-rapport d'audit

Sécurité offensive – Ethical Hacking / Pentest – Niveau Intermédiaire/Avancé

-header-cybersecurite.png

Prix : 3100 € H.T.

Durée : 5 jours

Public cible : Développeurs, ingénieurs systèmes/réseaux, admins IT souhaitant comprendre et tester les failles de sécurité. Équipes cybersécurité, SOC/Blue Team, responsables techniques ou DevOps. Futurs pentesters ou consultants en sécurité offensive

Prérequis : Bonne maîtrise des systèmes Linux et Windows. Connaissances réseaux : IP, ports, DNS, protocoles courants (HTTP, SMTP, SMB…). Bases en développement ou scripting (Python, Bash ou PowerShell)

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Réaliser un test d'intrusion structuré en respectant les bonnes pratiques éthiques
- Utiliser les outils standards du pentest : Nmap, Burp Suite, Metasploit, etc.
- Identifier et exploiter des vulnérabilités sur des applications web, des systèmes ou des réseaux
- Générer un rapport d'audit clair et exploitable pour la remédiation
- Intégrer la sécurité offensive dans une démarche de cybersécurité globale

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. Introduction à l'ethical hacking

- Cadre légal et éthique : ce qu'on a le droit de faire (LOPSSI, RGPD, CNIL)
- Types de tests d'intrusion : black box, grey box, white box
- Méthodologie standard (PTES, OSSTMM, OWASP)
- Environnement de test : lab, machines virtuelles, VPN, VM vulnérables

2. Phase de reconnaissance et énumération

- Scan réseau avec Nmap : découverte d'hôtes, services et OS
- Énumération DNS, WHOIS, Shodan, reconnaissance passive
- Fingerprinting de services (bannières, version, configuration)
- Mapping d'architecture (topologie, port scanning, hosts discovery)

3. Exploitation Web (OWASP Top 10)

- Injection SQL, XSS, CSRF, file upload, IDOR, path traversal
- Authentification faible, session hijacking, brute force
- Analyse de sécurité avec Burp Suite, Nikto, OWASP ZAP
- Attaques sur API REST, SOAP, JWT et OAuth
- Contournement de WAF ou filtres applicatifs

4. Exploitation système & réseau

- Vulnérabilités connues (CVE) : escalation de privilèges, mauvaises configurations
- Attaques sur services classiques : FTP, SMB, RDP, SSH, LDAP
- Exploitation avec Metasploit Framework
- Pivoting, tunneling, port forwarding et mouvements latéraux
- Dump de mots de passe (LSASS, hashes, mimikatz)

5. Attaques sur Active Directory

- Cartographie d'un domaine Windows
- Exploitation Kerberos (pass-the-ticket, AS-REP roasting)

- Enumeration LDAP, reconnaissance d'ACL
- Outils : BloodHound, CrackMapExec, Impacket

6. Exploitation client et ingénierie sociale (intro)

- Création de payloads, phishing ciblé (spear phishing)
- Contournement antivirus et sandbox (notions de base)
- Attaques sur PDF/Office, macros, droppers
- Post-exploitation : persistance, exfiltration de données

7. Reporting et recommandations

- Rédaction d'un rapport d'audit : vulnérabilités, gravité, recommandations
- Structuration du livrable (executive summary, tech details, scoring CVSS)
- Débriefing avec le client ou l'équipe cible
- Bonnes pratiques de remédiation, patching, renforcement

8. Atelier pratique complet

- Simulation d'un test d'intrusion d'un SI (reconnaissance ⟶exploitation ⟶rapport)
- Cibles réalistes : applications web, serveurs vulnérables, AD, bases de données
- Méthodes d'attaque variées (Web, réseau, système, brute force)
- Rendu de rapport et restitution orale

Cyber Threat Intelligence (CTI) – Niveau Débutant : Premiers pas dans l'analyse des menaces

-header-cybersecurite.png

Prix : 1910 € H.T.

Durée : 2 jours

Public cible : Analystes SOC débutants, techniciens cybersécurité, ingénieurs IT souhaitant s'initier à la CTI. Membres d'une cellule de gestion des incidents, Blue Team, RSSI junior. Étudiants, consultants ou profils reconvertis vers la cybersécurité défensive.

Prérequis : Connaissances générales en cybersécurité (types de menaces, architecture réseau). À l'aise avec l'usage d'un terminal et d'outils en ligne (web, base de données, navigateur). Aucune expérience en renseignement de menace exigée.

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Comprendre les bases du renseignement sur les cybermenaces (CTI)
- Identifier et collecter des indicateurs de compromission (IOC)
- Rechercher des informations sur les attaquants et campagnes (APT, malware)
- Utiliser des sources ouvertes (OSINT) et des outils d'analyse simples
- Contribuer à la détection et anticipation des menaces dans un contexte opérationnel

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. Introduction au Cyber Threat Intelligence

- Qu'est-ce que le CTI ? Définitions, rôles, typologie (stratégique / opérationnelle / tactique / technique)
- Pourquoi intégrer l'Intelligence dans la cybersécurité ?
- Typologie des menaces : APT, hacktivisme, cybercriminalité, insiders
- Modèles de classification : Diamond Model, Cyber Kill Chain, MITRE ATT&CK

2. Les indicateurs de compromission (IOC)

- Types d'IOC : adresses IP, noms de domaine, URLs, hashes, email, user-agent
- IOC vs IOA (indicateurs d'attaque)
- Recherches de base sur IOC avec VirusTotal, AbuseIPDB, urlscan.io
- Fiabilité, temporalité et contextualisation des indicateurs

3. Rechercher l'information (OSINT & sources ouvertes)

- Sources publiques : CERT-FR, CISA, MISP, TheHackerNews, GitHub, forums underground
- Outils d'OSINT : Maltego, Spiderfoot, Google Dorks, Hunter, HaveIBeenPwned
- Exercices guidés : remonter l'origine d'un domaine ou IP suspect
- Limitations, biais et aspects légaux de l'OSINT

4. Analyse d'acteurs et de campagnes

- Groupes APT connus : nomenclature, techniques, motivations
- MITRE ATT&CK : lecture d'un profil d'acteur (ex : APT29, FIN7…)
- Exemple de campagne analysée (phishing, malware, ransomware)
- Introduction à la notion de TTP (Tactics, Techniques, Procedures)

5. Structurer et partager l'information

- Formats standards : STIX/TAXII, MISP, JSON
- Introduction à MISP : plateforme de partage communautaire
- Cycle de vie de l'information CTI (recherche, validation, diffusion)
- Introduction à l'analyse de tendances et reporting simplifié

6. Atelier pratique guidé

- Scénario simple : alerte d'incident ⟶extraction d'IOC ⟶enrichissement avec OSINT
- Lecture d'un rapport CTI réel et extraction des données clés
- Présentation orale : cartographie de l'attaque et mesures de détection proposées

Cyber Threat Intelligence (CTI) – Niveau Intermédiaire : Analyse tactique et anticipation des menaces

Prix : 2300 € H.T.

Durée : 3 jours

Public cible : Analystes SOC, ingénieurs cybersécurité, Blue Team, CERT/CSIRT. Responsables de la détection, de la réponse à incident ou du renseignement cyber. Consultants SSI ou RSSI techniques souhaitant intégrer un volet renseignement à leurs missions

Prérequis : Maîtrise des fondamentaux de la cybersécurité (types d'attaques, vulnérabilités, réseau). Connaissance de base des IOC, OSINT et des familles d'attaquants (APT, malwares, etc.). Avoir suivi une formation CTI débutant ou avoir une première expérience dans un environnement SOC ou CERT.

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Identifier, enrichir et corréler des IOC et TTP issus de sources diverses
- Cartographier une menace selon MITRE ATT&CK et Diamond Model
- Structurer un rapport CTI tactique (pour SOC ou RSSI)
- Intégrer les données CTI dans des outils de détection ou de corrélation (SIEM, MISP, etc.)
- Anticiper l'évolution d'une menace en observant les campagnes ou groupes d'attaquants

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. Rappel et structuration du cycle CTI

- Intelligence vs information : cycle de renseignement, niveau tactique vs stratégique
- Taxonomies et vocabulaire : threat actor, campaign, intrusion set
- Modèles d'analyse : Diamond Model, Cyber Kill Chain, ATT&CK Navigator
- Corrélation entre IOC, TTP et objectifs adverses

2. Enrichissement et corrélation avancée d'IOC

- Extraction d'IOC depuis différents formats (PDF, JSON, logs, rapports)
- Corrélation IOC ⟷campagnes ⟷TTP (utilisation de VirusTotal, PassiveDNS, Greynoise, ThreatFox…)
- Scripts d'automatisation pour l'enrichissement (via APIs publiques)
- Introduction à Sigma, YARA pour détection associée

3. Analyse MITRE ATT&CK et TTP adverses

- Structure du framework ATT&CK (tactics, techniques, sub-techniques)
- Analyse d'un adversaire ou d'un rapport Mandiant / CERT-FR avec MITRE
- Utilisation de ATT&CK Navigator et ThreatMap
- Cartographie d'une attaque à partir de logs ou IOC

4. Plateformes CTI et gestion collaborative

- Utilisation de MISP : création d'événements, ajout d'attributs, taxonomies
- Intégration avec TheHive ou un SIEM (vue d'ensemble)
- Création de feeds internes / publics, gestion de la fiabilité, partage et TLP
- Comparatif avec d'autres solutions CTI : OpenCTI, ThreatConnect (vue d'ensemble)

5. Rapport de CTI tactique & restitution

- Structuration d'un rapport CTI tactique (formel mais exploitable)
- Intégration dans une veille opérationnelle ou une alerte SOC
- Recommandations de détection / mitigation associées aux TTP détectés

- Communication vers les équipes techniques et non techniques

6. Atelier complet : Analyse d'une campagne réelle

- Cas pratique : analyse complète d'une campagne (APT ou malware)
- Extraction des IOC, identification des patterns, attribution possible
- Construction d'un ATTACK Mapping + rapport de synthèse opérationnel
- Présentation et restitution collective

Sécurité dans le Cloud – Bonnes pratiques et outils pour AWS, Azure et GCP

-header-cybersecurite.png

Prix : 2300 € H.T.

Durée : 3 jours

Public cible : Ingénieurs cloud, DevOps, architectes IT. Administrateurs systèmes ou réseaux impliqués dans la migration ou l'exploitation cloud. RSSI, analystes cybersécurité, consultants sécurité techniques

Prérequis : Connaissances de base des services cloud (AWS, Azure ou GCP). Bonnes notions réseaux, systèmes et sécurité informatique générale. Une expérience préalable dans la configuration ou supervision d'un cloud est un plus

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Comprendre les enjeux de sécurité dans les environnements cloud publics
- Identifier les risques liés au modèle de responsabilité partagée
- Mettre en place des politiques de sécurité (IAM, réseau, chiffrement, audit) sur AWS, Azure ou GCP
- Détecter les erreurs de configuration fréquentes (S3 publics, ports ouverts…)
- Exploiter les outils de sécurité natifs pour auditer, protéger et surveiller leur environnement

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. Enjeux et cadre de la sécurité cloud

- Menaces typiques dans le cloud : exposition de ressources, mauvaise configuration, shadow IT
- Modèle de responsabilité partagée (Cloud Provider vs Client)
- Normes et référentiels : CIS Benchmarks, ISO 27017, RGPD, NIS2
- Différences entre IaaS, PaaS, SaaS du point de vue sécurité

2. Gestion des identités et des accès (IAM)

- Principes fondamentaux : utilisateurs, rôles, politiques, principes de moindre privilège
- IAM sur AWS, RBAC sur Azure, IAM sur GCP
- MFA, rotation des clés, délégation d'accès sécurisée
- Outils d'audit IAM : IAM Access Analyzer, Azure Privileged Identity Management

3. Sécurité réseau et isolation des ressources

- Sécurisation des VPC/VNet : firewall, security groups, NACL, bastions
- Exposition publique vs privée des ressources (S3, buckets, machines virtuelles)
- Configuration de points de terminaison privés et DNS interne
- Gestion du trafic sortant / entrant via proxy ou gateway

4. Chiffrement et protection des données

- Chiffrement au repos / en transit : KMS, HSM, TLS
- Sécurisation des bases de données, disques, objets (S3, Blob, Cloud Storage)
- Gestion des secrets et credentials : AWS Secrets Manager, Azure Key Vault, GCP Secret Manager
- Bonnes pratiques de gestion des données sensibles dans le cloud

5. Surveillance, audit et détection d'anomalies

- Services natifs de journalisation et d'audit : CloudTrail, Azure Monitor, Stackdriver
- Analyse d'événements suspects (logins anormaux, modifications de politiques, accès publics)
- Configuration d'alertes avec CloudWatch, Azure Sentinel, Chronicle
- Introduction aux solutions SIEM cloud-native

6. Conformité et remédiation automatique

- Outils d'évaluation de la conformité : AWS Config, Azure Security Center, GCP Security Command Center
- Détection d'erreurs de configuration critiques
- Scripts ou règles automatisées de remédiation (Terraform, CloudFormation, Bicep)
- Scans réguliers et audit de posture de sécurité

7. Atelier pratique multi-cloud (ou centré sur un provider)

- Mise en place d'une infrastructure simple sur AWS, Azure ou GCP
- Analyse de failles de configuration volontairement introduites (S3 public, ports ouverts, rôles excessifs)
- Correction et sécurisation guidée de l'environnement
- Génération d'un rapport de posture de sécurité

DevSecOps – Intégrer la sécurité dans les chaînes CI/CD

-header-cybersecurite.png

Prix : 2100 € H.T.

Durée : 3 jours

Public cible : Développeurs, DevOps, ingénieurs CI/CD. Responsables de la qualité logicielle, architectes cloud. Équipes sécurité (SecOps, RSSI techniques) souhaitant collaborer avec les équipes DevOps

Prérequis : Connaissances pratiques en développement logiciel (Git, scripts, gestion de dépendances). Maîtrise des bases des pipelines CI/CD (GitLab CI, GitHub Actions, Jenkins, etc.). Notions de sécurité applicative et réseau recommandées

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Comprendre les principes et objectifs du DevSecOps
- Intégrer des contrôles de sécurité dans les pipelines CI/CD (SAST, DAST, SCA, IaC scanning)
- Automatiser la détection de vulnérabilités dès la phase de développement
- Sécuriser les environnements de build, les secrets et les conteneurs
- Mettre en place une surveillance continue de la sécurité des applications en production

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. Introduction au DevSecOps

- DevOps vs DevSecOps : enjeux, objectifs, bénéfices
- Shifting left : détecter les failles plus tôt dans le cycle
- Culture, collaboration et outillage
- Points d'intégration typiques dans le cycle CI/CD

2. Sécurité du code source et des dépendances

- Analyse statique de code (SAST) : principes et outils (SonarQube, Semgrep, CodeQL…)
- Analyse de dépendances (SCA) : gestion des vulnérabilités dans les librairies (OWASP Dependency-Check, Snyk, Trivy)
- Intégration dans Git, PRs, et pipelines (GitHub Actions, GitLab CI…)
- Démo : pipeline avec SAST + SCA déclencheur d'alerte ou de blocage

3. Sécurité des conteneurs et images Docker

- Bonnes pratiques de création d'images : minimisation, non-root, versions figées
- Scans d'image Docker avec Trivy, Grype ou Clair
- Signature, SBOM (Software Bill of Materials), attestation de provenance
- Stockage sécurisé dans un registre privé (Harbor, ECR, GitLab registry)

4. Sécurité de l'infrastructure as code (IaC)

- Détection de failles dans les fichiers Terraform, Ansible, Kubernetes YAML
- Outils IaC scanners : Checkov, TFSec, KICS
- Inclusion dans les pipelines avant déploiement
- Gestion des secrets : Vault, Doppler, GitHub Secrets, trousseaux sécurisés

5. Analyse dynamique et tests en environnement de staging

- DAST (Dynamic App Security Testing) avec OWASP ZAP, Nikto
- Scan de vulnérabilités réseau et applicatif sur environnement QA
- Tests de pénétration automatisés (baseline OWASP)
- Intégration conditionnelle (run si SAST ok, ou en nightly)

6. Surveillance continue post-déploiement

- Journalisation et alerting de sécurité dans l'environnement de prod (logs, SIEM)
- Intégration avec Prometheus, Grafana, Loki, ou outils natifs cloud (CloudWatch, Azure Monitor. . .)
- Détection d'anomalies, honeypots, détection de changement de configuration
- Exemple de pipeline de "security monitoring" post-mise en production

7. Atelier complet CI/CD sécurisé

- Mise en place d'un pipeline DevSecOps complet (fourni ou à intégrer au projet de l'équipe)
- Analyse de code, dépendances, scan d'image, IaC, DAST, alertes
- Revue de sécurité automatisée et remontée dans le dashboard
- Génération d'un rapport de sécurité global

DevSecOps – Niveau Avancé : Sécurité continue des pipelines, conteneurs et infrastructure cloud

-header-cybersecurite.png

Prix : 3300 € H.T.

Durée : 5 jours

Public cible : Développeurs senior, ingénieurs DevOps / cloud, architectes CI/CD. Consultants sécurité, SecOps, Blue Team techniques. Équipes en charge de la sécurisation des processus de delivery dans un environnement conteneurisé ou multi-cloud

Prérequis : Maîtrise d'un outil CI/CD (GitLab CI, Jenkins, GitHub Actions, etc.). Expérience avec les conteneurs (Docker), l'IaC (Terraform, Ansible) et un cloud provider (AWS, Azure, GCP). Bonnes bases en sécurité applicative (SAST, secrets, gestion des dépendances, OWASP)

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Intégrer des contrôles de sécurité multi-niveaux dans des pipelines CI/CD complexes
- Auditer et sécuriser des environnements conteneurisés (Docker, Kubernetes)
- Gérer la sécurité de l'IaC et des déploiements cloud automatisés
- Superviser les composants sécurité (logs, compliance, vulnérabilités) en production
- Appliquer des politiques de sécurité automatisées (policy-as-code)

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. DevSecOps en environnement multi-cloud & microservices

- DevSecOps dans le cycle de développement agile et les microservices
- Modèle "Everything as Code" : infra, pipelines, policies
- Intégration sécurité dans des pipelines complexes (multi-env, multi-branche)
- Comparaison des pratiques DevSecOps sur AWS, Azure, GCP

2. Sécurité avancée dans les pipelines CI/CD

- Optimisation des jobs SAST, DAST, SCA dans les chaînes CI/CD
- Déclenchement conditionnel de scans, gestion des seuils de criticité
- Intégration sécurité dans GitOps (ArgoCD, FluxCD)
- Utilisation avancée des artefacts, variables, secrets et journaux sécurisés
- Gestion des dépendances multi-langages et attestation SBOM (CycloneDX, Syft)

3. Sécurisation des conteneurs et de Kubernetes

- Build d'images sécurisées et traçables (Dockerfile avancé, best practices)
- Scans approfondis des images : Trivy, Grype, Anchore
- Sécurisation Kubernetes : RBAC, PodSecurity, NetworkPolicy, SecComp
- Admission controllers (OPA/Gatekeeper, Kyverno) pour validation à l'entrée
- Analyse de clusters avec kube-bench, kube-hunter, Polaris

4. Sécurité de l'Infrastructure as Code (IaC)

- Détection de secrets, erreurs de configuration dans les fichiers Terraform, CloudFormation, K8s YAML
- Checkov, TFSec, KICS : configuration, intégration CI
- Remontée des alertes critiques avec scoring CVSS/CIS
- Contrôle des modifications sensibles via git commit hooks ou MR/Pull Request policies
- Gestion centralisée des secrets : HashiCorp Vault, AWS Secrets Manager, Doppler

5. Surveillance sécurité & compliance as code

- Centralisation des logs et alertes : Prometheus, Loki, ELK, CloudWatch Logs

- Intégration à un SIEM cloud-native (Azure Sentinel, Chronicle, Splunk cloud)
- Suivi de conformité en continu avec AWS Config, Azure Defender, Open Policy Agent
- Dashboards de vulnérabilités, exposition, erreurs de configuration
- Monitoring applicatif et alerting de sécurité (anomalies, règles comportementales)

6. Policy-as-code & Zero Trust dans DevSecOps

- Concepts de Policy-as-Code (OPA, Rego, Sentinel)
- Création de politiques d'admission, accès, conformité (Rego)
- Application dans Kubernetes, Terraform, GitHub Actions
- Introduction à Zero Trust et RBAC fin dans les environnements cloud-native

7. Atelier complet : pipeline DevSecOps sécurisé multi-niveau

- Conception et mise en place d'un pipeline complet :
- →SAST + SCA + secrets + IaC scanning + image scanning + DAST
- Intégration des outils : Git, GitLab/GitHub Actions, Trivy, Checkov, SonarQube, OPA
- Déploiement sécurisé d'un microservice sur Kubernetes (avec admission control)
- Mise en place d'un tableau de bord de sécurité global (alertes, conformité, logs)
- Présentation d'un rapport sécurité automatisé (PDF, JSON, Excel)

Analyse Forensique – Niveau Expert : Investigation avancée, mémoire et reverse engineering

Prix : 3300 € H.T.

Durée : 5 jours

Public cible : Analystes forensiques confirmés, enquêteurs numériques. Experts IR (Incident Response), chercheurs en sécurité. Consultants cybersécurité spécialisés en investigation et malware analysis

Prérequis : Maîtrise des fondamentaux forensiques (analyse disque, logs, mémoire). Expérience pratique avec outils forensiques classiques (Volatility, Autopsy…). Connaissances solides en systèmes Windows et Linux, scripting (Python, PowerShell, Bash)

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Réaliser des analyses approfondies de la mémoire vive et des artefacts complexes
- Détecter, analyser et extraire des malwares, rootkits, techniques d'obfuscation et persistence
- Mettre en œuvre des techniques avancées de reverse engineering forensique
- Automatiser des tâches d'analyse et créer des outils personnalisés
- Produire des rapports d'enquête professionnels et exploitables en contexte légal

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. Mémoire volatile avancée : acquisition et analyse

- Techniques avancées d'acquisition mémoire (WinDbg, FireEye tools, LiME avancé)
- Analyse avancée avec Volatility 3 : extraction de processus cachés, injection de code, hooks, threads suspects
- Analyse des artefacts de la mémoire : credential dumping, dumping de clés de registre, forensics réseau live
- Identification de rootkits et techniques d'évasion mémoire

2. Analyse forensique avancée disque et fichiers

- Analyse de systèmes de fichiers corrompus, cryptés, ou chiffrés
- Techniques d'extraction et reconstruction de fichiers effacés ou partiellement corrompus
- Analyse des métadonnées avancées et horodatages (NTFS USN journal, shadow copies)
- Extraction d'informations cachées dans les fichiers (stéganographie, métadonnées exotiques)

3. Reverse engineering appliqué à la forensic

- Introduction aux outils (IDA Pro, Ghidra, Radare2) et méthodologie d'analyse statique et dynamique
- Analyse comportementale de malware, unpacking, détection d'obfuscation
- Extraction d'indicateurs de compromission (IOC) et techniques anti-forensiques
- Automatisation de l'analyse avec scripts Python / plugins Volatility personnalisés

4. Forensic réseau avancé et corrélation multi-sources

- Analyse approfondie de captures réseau, reconstruction d'attaques complexes
- Corrélation avec logs systèmes et mémoire (timeline multi-sources)
- Détection de techniques avancées d'exfiltration (DNS tunneling, steganographie réseau)
- Utilisation d'outils comme Zeek, Suricata, Moloch (Arkime)

5. Automatisation et scripts forensiques

- Écriture de scripts Python/PowerShell pour automatiser la collecte et l'analyse de preuves
- Création de plugins Volatility spécifiques aux besoins client
- Intégration avec plateformes SIEM et outils de triage (TheHive, Cortex)
- Gestion de gros volumes de données et analyses multi-machines

6. Cadre légal, rédaction de rapports et présentation

- Aspects juridiques avancés et bonnes pratiques de la preuve numérique
- Structuration et rédaction d'un rapport d'expertise forensic détaillé
- Présentation orale des conclusions techniques à un public technique et non-technique
- Gestion de la confidentialité et de la traçabilité

7. Atelier pratique intensif

- Étude de cas complexe : compromission avancée, analyse de mémoire, extraction de malware, corrélation multi-source
- Mise en œuvre complète de la méthodologie d'investigation avancée
- Rédaction d'un rapport final d'enquête et restitution orale

Gouvernance et Conformité en Cybersécurité

-header-cybersecurite.png

Prix : 1450 € H.T.

Durée : 2 jours

Public cible : Managers sécurité, RSSI, chefs de projet sécurité. Auditeurs IT, consultants en cybersécurité. Responsables conformité, DPO, équipes juridiques. Toute personne impliquée dans la gouvernance de la sécurité informatique

Prérequis : Connaissances générales en cybersécurité et gestion des risques. Notions de base sur les normes et réglementations IT

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Comprendre les concepts clés de la gouvernance en cybersécurité
- Identifier les exigences réglementaires et normatives principales (RGPD, ISO 27001, NIS2, LPM, etc.)
- Mettre en place un dispositif de conformité adapté aux besoins de l'organisation
- Piloter une démarche de gestion des risques et audit de sécurité
- Communiquer efficacement avec les parties prenantes internes et externes

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. Introduction à la gouvernance en cybersécurité

- Définition et enjeux de la gouvernance IT et cybersécurité
- Cadres et bonnes pratiques : COBIT, ISO 27001/27002, NIST
- Rôles et responsabilités (RSSI, DSI, comité de pilotage)
- Maturité sécurité et indicateurs clés (KPI)

2. Normes et réglementations clés

- RGPD : principes, droits, obligations et impact sur la sécurité
- Directive NIS2 et autres directives européennes
- Loi de Programmation Militaire (LPM) et cybersécurité nationale
- PCI-DSS, HIPAA, et autres normes sectorielles
- Cadres de contrôle et certification (ISO, SOC, HDS)

3. Gestion des risques et audit

- Identification, évaluation et traitement des risques cybersécurité
- Méthodes et outils d'analyse des risques (EBIOS, MEHARI, ISO 27005)
- Plan de traitement des risques et plan de continuité d'activité (PCA)
- Préparation et conduite d'audits internes et externes

4. Dispositif de conformité et pilotage

- Élaboration de politiques et procédures de sécurité conformes
- Mise en place d'un dispositif de contrôle et suivi (dashboard, reporting)
- Gestion des incidents de sécurité et obligations de notification
- Formation et sensibilisation des collaborateurs à la conformité

5. Communication et gouvernance transverse

- Collaboration avec les équipes juridiques, RH, métiers
- Communication auprès de la direction et des autorités de contrôle
- Gestion des tiers et chaînes d'approvisionnement sécurisées

ISO 27001 – Mise en œuvre et audit du SMSI

-header-cybersecurite.png

Prix : 2100 € H.T.

Durée : 3 jours

Public cible : RSSI, responsables sécurité, consultants et auditeurs. Responsables qualité, compliance, DSI. Toute personne impliquée dans la gestion de la sécurité de l'information

Prérequis : Connaissances générales en cybersécurité et gestion des risques. Notions de base sur les systèmes de management

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Comprendre le référentiel ISO 27001 et son vocabulaire
- Mettre en place un système de management de la sécurité de l'information conforme à la norme
- Piloter une démarche d'audit interne et préparer une certification externe
- Identifier et évaluer les risques selon ISO 27005
- Rédiger et maintenir la documentation et les politiques de sécurité

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. Introduction à ISO 27001 et contexte réglementaire

- Historique et structure de la norme ISO 27001
- Terminologie et concepts clés (SMSI, contexte, parties intéressées)
- Liens avec les autres normes (ISO 27002, ISO 27005, RGPD)

2. Exigences de la norme et mise en œuvre du SMSI

- Contexte organisationnel et définition du périmètre
- Leadership et engagement de la direction
- Planification : analyse des risques et traitement (méthode ISO 27005)
- Support et sensibilisation : compétences, communication, documentation

3. Réalisation et exploitation du SMSI

- Mise en œuvre des mesures de contrôle (Annexe A – bonnes pratiques)
- Gestion opérationnelle et gestion des incidents
- Surveillance, mesure, analyse et évaluation de la performance
- Actions correctives et amélioration continue

4. Audit interne et préparation à la certification

- Planification et organisation d'un audit interne ISO 27001
- Techniques d'audit : entretiens, revue documentaire, échantillonnage
- Rédaction des constats et rapport d'audit
- Préparation à la certification externe : processus et exigences

5. Exercices pratiques et études de cas

- Mise en situation sur l'analyse de risques et choix des mesures
- Simulation d'audit interne avec retours et recommandations
- Construction d'un plan d'amélioration continue

Certification ISO 27001 Lead Implementer / Lead Auditor

-header-cybersecurite.png

Prix : 3200 € H.T.

Durée : 5 jours

Public cible : RSSI, responsables sécurité, consultants, auditeurs internes. Responsables qualité, conformité et gestion des risques. Toute personne en charge de la mise en œuvre, du pilotage ou de l'audit d'un SMSI

Prérequis : Connaissances de base en cybersécurité et normes ISO. Expérience en gestion de projet ou audit recommandée

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Comprendre en profondeur le référentiel ISO 27001 et les exigences du SMSI
- Piloter la mise en œuvre complète d'un SMSI conforme à la norme
- Réaliser un audit interne ou externe selon la méthodologie ISO 19011
- Analyser les écarts et proposer des mesures correctives adaptées
- Préparer une organisation à la certification ISO 27001

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. Introduction et contexte

- Historique et portée d'ISO 27001
- Concepts clés et vocabulaire du SMSI
- Exigences réglementaires et cadre légal

2. Mise en œuvre d'un SMSI (Lead Implementer)

- Définition du périmètre et analyse du contexte organisationnel
- Leadership et gouvernance
- Gestion des parties intéressées et communication
- Analyse des risques selon ISO 27005 et plan de traitement
- Élaboration des politiques, procédures et plans d'action
- Gestion documentaire et sensibilisation

3. Méthodologie d'audit (Lead Auditor)

- Norme ISO 19011 et principes d'audit
- Planification d'un audit ISO 27001
- Conduite d'audit : collecte des preuves, entretiens, revue documentaire
- Identification des non-conformités et analyse des causes
- Rédaction du rapport d'audit et recommandations
- Clôture de l'audit et suivi

4. Cas pratiques et simulations

- Simulation de mise en œuvre d'un SMSI
- Analyse de cas réels d'audit ISO 27001
- Jeux de rôles : conduite d'audit, gestion des résistances
- Travaux en groupe sur rapport d'audit

5. Préparation à la certification

- Exigences des organismes certificateurs
- Préparation de l'organisation à la visite d'audit externe
- Gestion de la surveillance et amélioration continue post-certification

ISO 27005 – Gestion des risques en sécurité de l'information

-header-cybersecurite.png

Prix : 1450 € H.T.

Durée : 2 jours

Public cible : RSSI, responsables sécurité, gestionnaires de risques. Auditeurs, consultants en cybersécurité. Toute personne impliquée dans la gestion et le traitement des risques sécurité

Prérequis : Connaissances de base en cybersécurité et systèmes d'information. Notions sur la gouvernance et normes ISO 27001

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Comprendre la méthodologie ISO 27005 et son intégration dans un SMSI
- Identifier, évaluer, traiter et surveiller les risques liés à la sécurité de l'information
- Utiliser des outils et techniques d'analyse des risques adaptés à leur organisation
- Documenter et piloter une démarche de gestion des risques efficace

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. Introduction à la gestion des risques

- Définitions, concepts clés et vocabulaire ISO 27005
- Relations entre ISO 27001 et ISO 27005
- Importance de la gestion des risques dans la cybersécurité

2. Processus de gestion des risques selon ISO 27005

- Contexte organisationnel et définition du périmètre
- Identification des actifs, menaces et vulnérabilités
- Estimation et évaluation des risques (méthodes qualitatives et quantitatives)
- Traitement des risques : options et choix des mesures de sécurité
- Acceptation, communication et documentation des risques

3. Outils et méthodes d'analyse des risques

- Méthodes d'identification (brainstorming, checklists, interviews)
- Évaluation qualitative vs quantitative (matrices, scoring, ALE)
- Techniques complémentaires (ARIS, BowTie, scénario d'attaque)
- Utilisation d'outils logiciels (exemples et démos)

4. Mise en œuvre pratique

- Planification et pilotage d'une analyse des risques dans un projet réel
- Intégration dans le cycle de vie du SMSI
- Exemples d'étude de cas et ateliers pratiques

5. Suivi et amélioration continue

- Revue périodique des risques et mise à jour du registre
- Indicateurs de performance et reporting
- Interaction avec la gestion des incidents et audit

RGPD (GDPR) – Sensibilisation et mise en œuvre

-header-cybersecurite.png

Prix : 1410 € H.T.

Durée : 2 jours

Public cible : DPO (Délégués à la Protection des Données), responsables conformité. Managers, chefs de projet, équipes IT et métiers. Toute personne concernée par la gestion des données personnelles

Prérequis : Notions de base en sécurité informatique et gestion des données. Compréhension générale des enjeux de la protection des données.

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Comprendre les principes et obligations du RGPD
- Identifier les données personnelles et leurs traitements
- Mettre en place une gouvernance conforme au RGPD
- Gérer les droits des personnes concernées et les incidents de données
- Préparer et accompagner les audits et contrôles CNIL / autorités de protection

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. Introduction au RGPD

- Historique et cadre juridique européen
- Concepts clés : données personnelles, traitement, responsable, sous-traitant
- Principes fondamentaux (licéité, transparence, minimisation, etc.)

2. Gouvernance et organisation RGPD

- Rôle et missions du DPO
- Cartographie des traitements de données
- Analyse d'impact (DPIA)
- Politique de confidentialité et documentation obligatoire

3. Droits des personnes et gestion des demandes

- Accès, rectification, effacement, portabilité
- Consentement et gestion des oppositions
- Gestion des violations de données (data breach)

4. Sécurité et conformité technique

- Mesures techniques et organisationnelles recommandées
- Sensibilisation et formation des collaborateurs
- Intégration dans les projets et cycle de vie des données

5. Audit, contrôle et sanctions

- Préparation aux audits CNIL et autres autorités
- Processus de contrôle et bonnes pratiques
- Sanctions et jurisprudences

Cyber Résilience : Prévention, réponse et continuité face aux cyberattaques

-header-cybersecurite.png

Prix : 2500 € H.T.

Durée : 3 jours

Public cible : RSSI, responsables sécurité, équipes SOC/CSIRT. Managers IT, responsables continuité d'activité (PCA/PCI). Consultants et auditeurs en cybersécurité. Toute personne impliquée dans la gestion des incidents et la résilience

Prérequis : Connaissances de base en cybersécurité et gestion des risques. Compréhension des concepts d'attaque et défense informatique

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Comprendre le concept de cyber résilience et ses composantes clés
- Mettre en place une stratégie de prévention efficace contre les cybermenaces
- Répondre aux incidents cyber de manière coordonnée et rapide
- Assurer la continuité des activités critiques en cas d'attaque
- Développer une culture organisationnelle orientée résilience

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. Introduction à la cyber résilience

- Définitions, enjeux et contexte actuel
- Différence entre cybersécurité traditionnelle et cyber résilience
- Cadres et standards (NIST CSF, ISO 22301, ISO 27031)

2. Prévention et réduction des risques

- Gouvernance et politique de sécurité intégrée
- Gestion des vulnérabilités et patch management
- Sensibilisation et formation des utilisateurs
- Architecture sécurisée et segmentation réseau

3. Détection et réponse aux incidents

- Mise en place d'un SOC/CSIRT efficace
- Détection des attaques : outils, techniques, SIEM
- Processus de gestion des incidents : identification, confinement, éradication
- Communication en situation de crise

4. Continuité d'activité et reprise après sinistre

- Plan de continuité d'activité (PCA) et plan de reprise informatique (PRI)
- Tests, exercices de simulation et retours d'expérience
- Sauvegarde et restauration sécurisée des données
- Redondance et résilience des infrastructures critiques

5. Culture et amélioration continue

- Gouvernance de la résilience et rôle des parties prenantes
- Intégration de la cyber résilience dans la stratégie globale
- Mesure de la maturité et indicateurs de performance
- Capitalisation post-incident et évolution des pratiques

Sécurité et Intelligence Artificielle : enjeux, risques et bonnes pratiques

-header-cybersecurite.png

Prix : 1740 € H.T.

Durée : 2 jours

Public cible : RSSI, responsables sécurité, data scientists. Développeurs IA, ingénieurs ML, équipes DevSecOps. Managers et décideurs impliqués dans des projets IA. Toute personne concernée par la sécurité des systèmes IA.

Prérequis : Connaissances de base en cybersécurité et IA. Notions en développement ou data science recommandées

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Comprendre les spécificités des risques liés à l'IA
- Identifier les vulnérabilités propres aux systèmes d'intelligence artificielle
- Mettre en œuvre des mesures de sécurité adaptées aux projets IA
- Intégrer la sécurité dans le cycle de vie des modèles IA (MLOps sécurisé)
- Gérer les enjeux éthiques et réglementaires liés à la sécurité de l'IA

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. Introduction à la sécurité de l'IA

- Enjeux et contexte actuel
- Spécificités des systèmes IA vs systèmes classiques
- Typologie des menaces (data poisoning, model inversion, adversarial attacks)

2. Vulnérabilités et attaques ciblant l'IA

- Attaques sur les données d'entraînement et de test
- Manipulation et perturbation des modèles (adversarial examples)
- Vol de modèles et extraction d'informations sensibles
- Attaques par rétro-ingénierie et backdoors

3. Mesures de protection et bonnes pratiques

- Sécurisation des données et pipelines ML (data governance, chiffrement)
- Techniques de robustesse des modèles (adversarial training, détection d'anomalies)
- Authentification et contrôle d'accès aux modèles et API
- Surveillance et détection d'attaques en temps réel

4. Intégration de la sécurité dans le cycle MLOps

- Développement sécurisé et revue de code IA
- Automatisation des tests de sécurité IA (fuzzing, tests adversariaux)
- Déploiement sécurisé et gestion des versions
- Gestion des incidents et réponse aux attaques IA

5. Enjeux éthiques, conformité et réglementation

- Gouvernance responsable des projets IA
- RGPD, AI Act et autres cadres réglementaires
- Transparence, auditabilité et explicabilité des modèles
- Gestion des biais et risques éthiques en sécurité IA

Data et Big Data

Snowflake Core Architecture & Ingestion Pipelines

Prix : 2 580 € HT

Durée : 4 jours (28 heures)

Public cible : Ingénieurs Data & Développeurs, administrateurs de plateformes analytiques, Data Engineers & Architectes Data, candidats à la certification Snowflake.

Prérequis : Maîtrise du langage SQL (requêtes, jointures, DDL/DML). Connaissance des concepts de bases de données relationnelles et de l'entreposage de données (data warehousing). Une première exposition au cloud (AWS, Azure ou GCP) est un plus. Accès à un compte Snowflake (trial ou entreprise) pour les hands-on labs.

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Provisionner et verrouiller l'infrastructure Snowflake : architecture 3 couches, Warehouses, sécurité réseau et chiffrement
- Implémenter le modèle RBAC et la gouvernance des données avec Snowflake Horizon (masquage, Row Access, tagging)
- Construire des pipelines d'ingestion et de transformation : Snowpipe, Streams, Tasks (DAG) et Dynamic Tables
- Maîtriser le FinOps, la résilience (Time Travel, Cloning), l'IA Cortex et les pièges de la certification

Méthodes et moyens pédagogiques : Formation intensive et hands-on : 4 hands-on labs SQL de bout en bout. Format : présentiel ou distanciel. Finalité : préparation à la certification Snowflake.

Modalités d'évaluation

Évaluation en continu à travers les 4 hands-on labs SQL réalisés tout au long de la formation ; parcours orienté préparation à la certification Snowflake.

Accessibilité : Formation accessible aux personnes en situation de handicap, contactez notre référent dédié.

Programme

1. Jour 1 · Architecture & Provisioning de Ressources

- Objectif du Jour 1 : provisionner l'infrastructure et verrouiller les accès au niveau réseau et compte.
- Les 3 couches : Storage, Compute (Virtual Warehouses) et Cloud Services
- Cycle de vie et gestion des Warehouses (Sizing, Multi-cluster, auto-suspend/resume, scaling vertical vs horizontal)
- Création et gestion des ressources : Databases, Schemas (Transient vs Permanent vs Temporary)

2. Jour 1 · Sécurité Réseau & Authentification

- Configuration des Network Policies (autoriser/bloquer des IPs ou VPCs)
- Concepts de sécurité : MFA, SSO et chiffrement des données (clés gérées par Snowflake vs Tri-Secret Secure)
- Hands-on Lab : Déploiement par script SQL d'un environnement complet isolé, sécurisé par une politique réseau, avec affectation de profils de Warehouses distincts

3. Jour 2 · Sécurité & Modèle de Responsabilité (RBAC)

- Objectif du Jour 2 : implémenter le contrôle d'accès, la gouvernance de la donnée et les flux d'entrée/sortie.
- Gestion des Users et cycle de vie des comptes
- Hiérarchie des Roles natifs (ACCOUNTADMIN à PUBLIC) et création de rôles sur-mesure (principe du moindre privilège)
- Attribution fine des privilèges (GRANT / REVOKE)

4. Jour 2 · Gouvernance des Données (Snowflake Horizon)

- Mise en place de Dynamic Data Masking (masquage de colonnes) et Row Access Policies (sécurité à la ligne)
- Tagging d'objets pour le suivi de la sensibilité des données

5. Jour 2 · Ingestion de Données & Connectivité

- Configuration des Stages internes et externes — ingestion Batch via COPY INTO
- Ingestion continue via Snowpipe (et introduction à Snowpipe Streaming)
- Hands-on Lab : Implémentation d'une politique de masquage de données RGPD sur un rôle de développeur et configuration d'un pipeline d'ingestion automatisé

6. Jour 3 · Manipulation de Données Semi-Structurées

- Objectif du Jour 3 : manipuler les formats complexes et automatiser les pipelines de données (ELT / CDC).
- Stockage et requêtage natif du JSON, XML et Parquet via le type VARIANT
- Fonctions d'extraction : notation par point, FLATTEN, LATERAL FLATTEN

7. Jour 3 · Pipelines de données (CDC) & Automatisation

- Capture des changements de données via les Streams
- Orchestration native : création de Tasks, gestion des dépendances (DAGs) et stratégies de Scheduling (CRON / Serverless)
- Utilisation des Dynamic Tables pour simplifier les pipelines déclaratifs
- Hands-on Lab : Création d'un pipeline d'ingestion de logs JSON bruts, traitement automatique des deltas via un Stream, et exécution planifiée via un arbre de Tasks (DAG)

8. Jour 4 · FinOps & Gouvernance des Coûts

- Objectif du Jour 4 : maîtriser les coûts, assurer la reprise après sinistre, utiliser l'IA et valider les pièges de la certification.
- Facturation (Compute, Storage, Cloud Services) — suivi via ACCOUNT_USAGE
- Mise en place de garde-fous : Resource Monitors (niveaux Account et Warehouse)

9. Jour 4 · Observabilité & Alerting

- Configuration du framework d'Alerting natif (CREATE ALERT) pour notifier l'équipe par email / webhook (échec d'une Task, pic de coût…)

10. Jour 4 · Résilience & Cycle de vie de la donnée

- Fonctionnement précis du Time Travel (0 à 90 jours selon l'édition) et du Fail-safe (7 jours non configurables)
- Utilisation du Zero-Copy Cloning pour le clonage instantané d'environnements de Dev / Test

11. Jour 4 · Data Marketplace & IA Cortex

- Partage de données et utilisation du Snowflake Marketplace
- Introduction et cas d'usage des fonctions IA intégrées (Snowflake Cortex LLM & Machine Learning)
- Hands-on Lab : Restauration d'une table supprimée via Time Travel, création d'une alerte FinOps sur dépassement de crédit, et appel d'une fonction d'analyse de sentiment via Snowflake Cortex

Apache Hop — Intégration de données visuelle

Prix : 2 100 € HT

Durée : 2 jours (14 heures)

Public cible : Développeurs, ingénieurs de données.

Prérequis : Maîtriser les bases du SQL. Une compréhension de base des concepts d'intégration de données serait un atout. Prérequis d'installation : avoir un compte Google avec un accès au Dataflow de Google Cloud service (crédit de 300 $ offert à la première inscription) ; être admin de son poste, avec une DB active (type PostgreSQL).

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Maîtriser la conception de flux d'intégration visuelle
- Exploiter les fonctionnalités avancées d'Apache Hop
- Créer des workflows d'intégration complexes

Méthodes et moyens pédagogiques : Stage pratique : 60 % pratique, 40 % théorie. Support de formation distribué au format numérique à tous les participants. Le cours alterne les apports théoriques du formateur, soutenus par des exemples, des séances de réflexion et du travail en groupe. La formation se base sur la dernière version en date, Apache Hop 2.11.

Modalités d'évaluation

Positionnement à l'entrée par un questionnaire d'auto-évaluation (conforme aux critères qualité Qualiopi). En fin de session, un questionnaire à choix multiples permet de vérifier l'acquisition des compétences. Une attestation est remise à chaque stagiaire ayant suivi la totalité de la formation.

Accessibilité : Formation accessible aux personnes en situation de handicap, contactez notre référent dédié.

Programme

1. Introduction à Apache Hop

- Présentation d'Apache Hop et son rôle dans le processus d'intégration
- Comprendre les fondamentaux de l'intégration de données
- Installation et configuration initiale — premiers pas avec Apache Hop
- Exploration de l'interface utilisateur graphique (GUI)

PRATIQUE Création de flux de données simples

2. Les concepts clés d'Apache Hop

- Apprentissage des concepts de base
- Les transformations
- Les étapes
- Les flux de données
- Exploration approfondie des composants de conception de flux
- Utilisation des métadonnées
- Exemples pratiques de manipulation de données avec Apache Hop
- Développement de compétences avancées dans la conception de flux complexes
- Création de flux réutilisables avec Apache Hop

3. Intégration de bases de données avec Apache Hop

- Connexion et extraction de données
- Chargement de données dans des bases de données
- Optimisation des performances
- Ateliers pratiques sur des cas d'utilisation courants
- Gestion des erreurs et récupération dans les flux de base de données
- Utilisation de variables pour une flexibilité accrue

4. Gestion des flux de données

- Introduction aux concepts de gestion des flux de données
- Planification et automatisation des flux avec Apache Hop
- Surveillance et débogage des flux en temps réel
- Gérer efficacement les flux de données
- Création de rapports et de tableaux de bord pour la surveillance
- Gestion des versions des flux

5. Intégration avec d'autres technologies

- Connexion à des services cloud
- Utilisation d'API REST
- Intégration de technologies émergentes dans les flux de données
- Cas d'étude pratique sur l'intégration multi-technologique
- Configuration de connexions sécurisées
- Optimisation des flux

6. Bonnes pratiques et optimisation

- Bonnes pratiques de conception de flux d'intégration
- Optimisation des performances pour des opérations d'intégration rapides
- Sécurité des données dans les flux d'intégration
- Études de cas sur l'application de bonnes pratiques
- Utilisation avancée des options de performance d'Apache Hop

SQL Avancé : Requêtes Complexes et Optimisation

Prix : 1580 € H.T.

Durée : 2 jours

Public cible : Développeurs, data analysts, data engineers ou administrateurs de base de données. Utilisateurs réguliers de SQL souhaitant aller au-delà des requêtes classiques. Chefs de projet ou métiers manipulant des bases de données relationnelles complexes

Prérequis : Maîtrise des bases du langage SQL : SELECT, WHERE, GROUP BY, JOIN. Connaissance d'un SGBD relationnel (PostgreSQL, MySQL, Oracle, SQL Server, etc.). Expérience pratique recommandée sur au moins un outil de requêtage.

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Maîtriser les jointures complexes, sous-requêtes et expressions conditionnelles
- Utiliser les fonctions avancées de traitement de données (fenêtres, CTE, agrégats)
- Optimiser les requêtes SQL en analysant leur performance
- Manipuler efficacement des ensembles de données structurées pour des analyses avancées
- Appliquer les bonnes pratiques d'écriture, de lisibilité et de performance SQL

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. Rappels et bonnes pratiques

- Requêtes imbriquées, jointures internes et externes
- Clauses GROUP BY, HAVING, ORDER BY avancées
- Astuces pour écrire du SQL lisible et maintenable

2. Sous-requêtes et expressions conditionnelles

- Sous-requêtes corrélées vs non corrélées
- Utilisation avancée de CASE, COALESCE, NULLIF
- Requêtes avec EXISTS / NOT EXISTS

3. Fonctions d'agrégation et analytiques

- Fonctions d'agrégation conditionnelle
- Fonctions analytiques (fenêtres) : ROW_NUMBER(), RANK(), LAG(), LEAD(), etc.
- PARTITION BY, ORDER BY dans les fonctions de fenêtre
- Applications : classement, cumul, comparaison temporelle

4. Common Table Expressions (CTE)

- Introduction aux CTE (WITH) : lisibilité et modularité
- CTE récursives pour explorer des hiérarchies ou graphes
- Comparaison avec les sous-requêtes imbriquées classiques

5. Manipulation d'ensembles complexes

- UNION, INTERSECT, EXCEPT
- Traitement de données temporelles (différences de dates, regroupements par périodes)
- Requêtes dynamiques et pivotements de données (PIVOT / UNPIVOT ou alternatives)

6. Optimisation et performance

- Comprendre les plans d'exécution (EXPLAIN / ANALYZE)
- Index, statistiques, tri et filtrage efficace

- Détection des goulots d’étranglement et surcharge mémoire
- Bonnes pratiques : éviter les requêtes inefficaces, limiter la charge serveur

7. Atelier pratique

- Étude de cas à partir d’une base relationnelle métier (ex : ventes, RH, support, finance)
- Résolution de requêtes complexes à plusieurs étapes
- Comparaison de solutions et optimisation collaborative

SQL Analytique : Fonctions de Fenêtre et Traitement Avancé des Données

Prix : 1880 € H.T.

Durée : 2 jours

Public cible : Data analysts, data scientists, contrôleurs de gestion, business analysts. Développeurs ou ingénieurs souhaitant approfondir l'analyse de données avec SQL. Toute personne manipulant régulièrement des bases de données pour produire des indicateurs ou des reporting

Prérequis : Maîtrise des bases du SQL : SELECT, WHERE, GROUP BY, JOINS. Expérience pratique avec un outil de requêtage ou un SGBD relationnel (PostgreSQL, SQL Server, MySQL, Oracle...). Connaissances élémentaires en statistiques ou reporting métier appréciées

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Utiliser les fonctions analytiques (fenêtres) pour produire des analyses complexes
- Maîtriser les opérations de cumul, classement, variation, comparaison temporelle
- Segmenter dynamiquement les données avec PARTITION BY, ORDER BY, LAG, LEAD, etc.
- Appliquer SQL à des cas concrets d'analyse métier (KPI, churn, comportement utilisateur)
- Optimiser les requêtes analytiques pour les rendre lisibles et performantes

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. Rappels SQL pour l'analyse

- Clauses fondamentales : GROUP BY, HAVING, filtres multiples
- Jointures internes / externes pour enrichir les jeux de données
- Sous-requêtes vs Common Table Expressions (CTE)

2. Introduction aux fonctions de fenêtre (window functions)

- Structure : OVER(), PARTITION BY, ORDER BY
- Différences avec les agrégats classiques
- Cas d'usage : moyenne mobile, cumul, classement dans un segment

3. Fonctions analytiques courantes

- ROW_NUMBER(), RANK(), DENSE_RANK() : classement et détection de doublons
- LAG(), LEAD() : comparaison entre lignes consécutives
- FIRST_VALUE(), LAST_VALUE() : repérage de bornes dans une série
- NTILE(), PERCENT_RANK(), CUME_DIST() : segmentation en quantiles

4. Analyse temporelle et séries

- Calculs sur périodes glissantes : moyenne sur 7 jours, évolution mensuelle
- Études d'évolution entre deux dates (delta, variation, croissance)
- Fenêtres mobiles : RANGE vs ROWS
- Cas d'usage : churn, comportements utilisateurs, suivi de cohortes

5. Études de cas métiers

- Analyse du chiffre d'affaires cumulé par client et par mois
- Classement des meilleurs vendeurs par zone / période
- Détection de récurrence ou d'événements inhabituels
- Analyse de la rétention ou de la fréquence d'utilisation

6. Optimisation et lisibilité

- Structuration des requêtes analytiques complexes
- Utilisation de CTE pour segmenter les étapes d'analyse
- Lecture d'un plan d'exécution pour diagnostiquer une requête lente
- Bonnes pratiques de rédaction pour les dashboards automatisés

Apache Spark – Niveau Avancé : Optimisation, Streaming et Traitements Scalables

Prix : 2100 € H.T.

Durée : 3 jours

Public cible : Data engineers, développeurs ou data scientists utilisant Spark en production. Architectes ou tech leads responsables de pipelines de données distribués. Équipes travaillant sur des projets Big Data avec des contraintes de performance, de volume ou de temps réel

Prérequis : Pratique régulière de Spark (PySpark ou Scala). Maîtrise des DataFrames, transformations de base, Spark SQL. Connaissances en cluster, format de données, et environnement distribué

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Optimiser les performances de leurs jobs Spark : partitionnement, cache, joins, formats
- Structurer des pipelines complexes et robustes
- Gérer des traitements en streaming temps réel avec Spark Structured Streaming
- Implémenter des workflows de machine learning distribué avec Spark MLlib
- Déployer Spark efficacement dans un cluster (YARN, Kubernetes, Databricks…)

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. Architecture approfondie de Spark

- Analyse détaillée : DAG, stages, tasks, shuffles
- Fonctionnement du Catalyst Optimizer et du Tungsten Engine
- Paramètres critiques : mémoire, nombre d'exécuteurs, partitions, parallélisme

2. Optimisation des performances

- Partitionnement personnalisé (repartition, coalesce, partitionBy)
- Joins performants : broadcast, sort-merge, bucketed joins
- Cache/persist : usages et limites
- Choix des formats de données : Parquet, ORC, Delta
- Utilisation de explain() et Spark UI pour profiler un job

3. Gestion avancée des données

- Manipulation de fichiers volumineux / structurés
- Lecture/écriture optimisées : compression, schema evolution, merge
- Traitement de données hiérarchiques (nested structures, arrays)

4. Spark Structured Streaming avancé

- Fenêtrage temporel, agrégations par fenêtre
- Gestion de l'état (stateful operations)
- Stratégies de déclenchement (triggering)
- Tolérance aux pannes : checkpointing, exactly-once
- Intégration Kafka ⟷Spark : lecture, écriture, offset management

5. Machine Learning distribué avec MLlib

- Pipelines de traitement : VectorAssembler, StringIndexer, StandardScaler
- Modèles supervisés : régression, classification, arbres de décision
- Évaluation, grid search, cross-validation en mode distribué
- Sauvegarde/rechargement de modèles Spark MLlib

- Différences Spark MLlib vs scikit-learn / TensorFlow

6. Structuration de projets Spark

- Bonnes pratiques de structuration de code (fonctions, scripts modulaires)
- Déploiement en production : packaging avec spark-submit, intégration CI/CD
- Variables d'environnement, configuration dynamique
- Exécution sur Databricks, EMR, Azure Synapse ou Kubernetes

7. Atelier final

- Conception et exécution d'un pipeline complet :
- Ingestion Kafka ou fichiers
- Traitement batch + streaming
- Pipeline ML distribué
- Export dans stockage optimisé
- Profiling et tuning

Apache Kafka – Niveau Avancé : Administration, Optimisation et Streaming Temps Réel

Prix : 2180 € H.T.

Durée : 3 jours

Public cible : Administrateurs, data engineers, développeurs et architectes techniques utilisant Kafka en production. Équipes responsables de la scalabilité, sécurité, et haute disponibilité des clusters Kafka. Professionnels souhaitant concevoir et optimiser des architectures de streaming distribuées complexes

Prérequis : Bonne connaissance des concepts de base Kafka (topics, partitions, producteurs, consommateurs). Expérience pratique avec Kafka en environnement de développement ou production. Notions d'administration système et de réseaux recommandées

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Installer et configurer un cluster Kafka scalable et résilient
- Gérer la sécurité : authentification, autorisation, chiffrement
- Optimiser la production et la consommation pour des performances maximales
- Mettre en œuvre Kafka Connect pour l'intégration avec d'autres systèmes
- Développer des applications streaming avancées avec Kafka Streams
- Surveiller et dépanner un cluster Kafka en production

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. Architecture avancée et administration de cluster

- Installation multi-nœuds, configuration de brokers, Zookeeper / KRaft
- Réplication, ISR (In-Sync Replicas), stratégies de partitionnement
- Gestion des logs, configuration de la rétention, nettoyage des topics
- Gestion des quotas et contrôle du débit

2. Sécurité dans Kafka

- Authentification : SSL, SASL (PLAIN, SCRAM, Kerberos)
- Autorisation : ACLs (Access Control Lists)
- Chiffrement des données en transit et au repos
- Meilleures pratiques et audit de sécurité

3. Performance et optimisation

- Paramètres de production et consommation (batch.size, linger.ms, fetch.min.bytes)
- Compression des messages (gzip, snappy, lz4)
- Gestion des délais, backpressure, et gestion des erreurs
- Tuning du GC, configuration mémoire JVM pour brokers

4. Kafka Connect et intégrations

- Présentation de Kafka Connect et connecteurs disponibles (bases, fichiers, cloud)
- Configuration et gestion de connecteurs source et sink
- Transformation simple des données dans les pipelines
- Déploiement et monitoring des connecteurs

5. Kafka Streams avancé

- Concepts clés : topologies, KStream, KTable, GlobalKTable
- Opérations stateless vs stateful

- Fenêtrage, agrégations, joins, transformations personnalisées
- Tolérance aux pannes, sauvegarde d'état (state stores)
- Monitoring et debugging d'applications Kafka Streams

6. Monitoring et maintenance

- Outils de monitoring : Kafka Manager, Confluent Control Center, Prometheus, Grafana
- Logs, métriques JMX, alerting
- Diagnostic des problèmes : lag, déséquilibre des partitions, ISR, erreurs réseau
- Sauvegarde, migration, mises à jour du cluster

7. Cas pratiques et ateliers

- Mise en place d'un cluster Kafka multi-nœuds sécurisé
- Déploiement de connecteurs source/sink avec transformation des données
- Développement d'une application Kafka Streams avancée
- Analyse et résolution de problèmes de performance et de stabilité

Apache Hadoop – Niveau Avancé : Administration, Optimisation et Écosystème

Prix : 2200 € H.T.

Durée : 3 jours

Public cible : Administrateurs systèmes et data engineers responsables d'un cluster Hadoop. Architectes Big Data et équipes techniques en charge de projets Hadoop complexes. Développeurs avancés souhaitant optimiser et gérer des traitements distribués à grande échelle

Prérequis : Maîtrise des concepts fondamentaux Hadoop et Big Data. Expérience pratique avec HDFS et MapReduce. Connaissances en administration Linux et en réseaux

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Installer, configurer et administrer un cluster Hadoop multi-nœuds
- Optimiser les performances de HDFS et des jobs MapReduce
- Maîtriser YARN et la gestion des ressources dans Hadoop
- Utiliser et administrer les composants de l'écosystème Hadoop (Hive, HBase, Oozie, etc.)
- Assurer la sécurité, la haute disponibilité et la résilience du cluster

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. Architecture avancée de Hadoop

- Détails de HDFS : NameNode HA, Federation
- Gestion des DataNodes, balancement de charge
- Fonctionnement de YARN : ResourceManager, NodeManager, Scheduler
- Cycle de vie des applications et gestion des ressources

2. Installation et administration d'un cluster Hadoop

- Installation multi-nœuds (distribution Apache ou Cloudera/Hortonworks)
- Configuration avancée des fichiers XML (core-site, hdfs-site, yarn-site)
- Démarrage, arrêt, mise à jour du cluster
- Surveillance via JMX, logs, outils (Ambari, Cloudera Manager)

3. Optimisation des performances

- Gestion des fichiers et blocs dans HDFS, taille de bloc optimale
- Compression et formats de fichiers adaptés (Parquet, ORC)
- Optimisation des jobs MapReduce et des ressources YARN
- Tuning du Garbage Collector, JVM, et paramètres réseau

4. Sécurité et gestion des accès

- Authentification Kerberos et intégration LDAP
- Contrôle d'accès basé sur les rôles (RBAC) et permissions HDFS
- Chiffrement des données au repos et en transit
- Audit et traçabilité des accès

5. Écosystème Hadoop avancé

- Hive : optimisation des requêtes, partitionnement, bucketing
- HBase : architecture, administration, scalabilité
- Oozie : orchestration de workflows Hadoop
- Autres outils : Sqoop, Flume, ZooKeeper

6. Haute disponibilité et résilience

- Configuration NameNode HA et failover automatique
- Backup et restauration des métadonnées
- Gestion des pannes DataNodes et récupération
- Stratégies de sauvegarde et reprise d'activité

7. Atelier pratique

- Mise en place d'un cluster Hadoop multi-nœuds en mode pseudo-distribué avancé
- Configuration et tuning d'un job MapReduce complexe
- Mise en œuvre de la sécurité Kerberos et gestion des utilisateurs
- Orchestration d'un workflow Hive/Oozie

Apache Hop – Niveau Débutant : Orchestration Visuelle de Flux de Données

Prix : 1780 € H.T.

Durée : 2 jours

Public cible : Data engineers, développeurs ETL/ELT et intégrateurs de données débutants sur Apache Hop. Chefs de projet ou architectes souhaitant découvrir l'orchestration visuelle de pipelines. Toute équipe technique comparant Hop à d'autres outils d'intégration (Talend, Pentaho, Airflow…)

Prérequis : Connaissances de base en SQL et en architecture ETL/ELT. Notions de manipulation de données (fichiers plats, JSON, bases relationnelles). Java JDK 8+ installé et environnement Hop Studio configuré

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Installer et configurer Apache Hop Studio
- Concevoir, tester et déployer des pipelines (transformations) et des workflows (orchestration)
- Utiliser efficacement les composant clés pour sources, cibles et traitements
- Gérer métadonnées, variables, paramètres et réutilisabilité
- Automatiser, superviser et optimiser des flux de données simples à complexes

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. Introduction à Apache Hop

- Présentation d'Apache Hop et rôle dans l'intégration de données
- Fondamentaux de l'ETL/ELT et cas d'usage
- Installation, configuration initiale et premiers pas dans l'interface graphique (GUI)
- Création d'un premier pipeline simple

2. Concepts clés d'Apache Hop

- Transformations vs Workflows
- Étapes (steps) et flux de données (hops)
- Métadonnées partagées et gestion des repositories
- Exemples pratiques : manipulation de données, filtrage, enrichissement
- Création de pipelines réutilisables

3. Intégration de bases de données

- Connexion JDBC et extraction de données (Table Input, Table Output)
- Chargement dans une base cible et optimisation des performances
- Ateliers pratiques sur cas courants (batch insert/update)
- Gestion des erreurs, transactions et reprise sur incident
- Variables et paramètres pour pipelines dynamiques

4. Gestion et supervision des flux

- Planification et automatisation via Hop Server ou scheduler externe
- Débogage en temps réel, points d'arrêt, aperçu de lignes
- Création de rapports et tableaux de bord de supervision
- Versioning des pipelines et bonnes pratiques de lifecycle

5. Intégration avec d'autres technologies

- Connexion à services cloud (S3, Azure Blob, Google Storage)
- Appels à des API REST et traitement JSON/XML

- Exemples de pipelines multi-technologie (Kafka, Hadoop, Spark)
- Sécurisation des connexions (SSL, authentification)
- Optimisation et tuning des flux hétéroclites

6. Bonnes pratiques et optimisation

- Principes de conception de pipelines robustes et maintenables
- Astuces pour accélérer les traitements et réduire la latence
- Sécurité des données dans Hop : masquage, chiffrement, accès
- Études de cas : application des bonnes pratiques sur un projet réel
- Utilisation avancée des options de performance (parallelisme, clustering)

Elasticsearch – Niveau Avancé : Performance, Sécurité et Scalabilité

Prix : 2180 € H.T.

Durée : 3 jours

Public cible : Administrateurs, data engineers et architectes chargés de clusters Elasticsearch en production. Développeurs et DevOps souhaitant optimiser, sécuriser et faire évoluer un environnement Elastic. Équipes en charge de cas d'usage critiques : logs à fort volume, analytics temps réel, recherche d'entreprise

Prérequis : Maîtrise des fondamentaux d'Elasticsearch (indexation, requêtes DSL, agrégations). Expérience pratique avec un cluster mono- ou multi-nœuds. Connaissances de base en administration Linux et en réseaux

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Concevoir et dimensionner un cluster Elasticsearch scalable et résilient
- Optimiser les performances d'indexation et de recherche pour de gros volumes de données
- Mettre en place des pipelines d'ingestion robustes et sécurisés
- Assurer la haute disponibilité et la reprise après sinistre (HA/DR)
- Sécuriser l'accès et garantir la conformité (authentification, chiffrement, audit)
- Superviser et diagnostiquer finement l'état du cluster

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. Architecture distribuée et dimensionnement

- Topologie de cluster : nœuds master, data, ingest, coordinating
- Stratégies de sharding et de réplication selon le volume et la criticité
- Dimensionnement CPU, mémoire, stockage, I/O
- Migration et extension de cluster en production

2. Performance et tuning avancé

- Optimisation de l'indexation : bulk, refresh interval, merge policy
- Choix des formats de stockage : compression, doc_values, store
- Tuning des requêtes : caches, circuit breaker, préférences de nœuds
- Analyse de performance via _nodes/stats, _cluster/stats et Profiler API

3. Ingest Pipelines & Transformations

- Configuration d'Ingest Pipelines : processors (grok, date, geoip, script)
- Enrichissement des documents à l'ingestion
- Utilisation de Logstash et Beats en complément
- Gestion des erreurs et des documents DLQ (dead letter queue)

4. Sécurité et gouvernance

- Configuration X-Pack Security : utilisateurs, rôles, permissions
- Authentification : LDAP, Active Directory, SAML, OAuth
- Chiffrement TLS inter-nœud et client-nœud
- Audit Logging et conformité RGPD

5. Haute Disponibilité et Résilience

- Backup et snapshots automatisés avec Snapshot API
- Stratégies de restauration et PRA/PCA
- Multi-cluster et cross-cluster replication (CCR)

- Maintenance sans interruption : rolling upgrade, allocation filtering

6. Scripting, Painless et fonctions avancées

- Écriture de scripts Painless pour scoring et transformations
- Scripted fields et runtime fields
- Utilisation de Stored Scripts et Script Templates
- Cas pratiques : calculs dynamiques, filtres géospatiaux, déduplication

7. Monitoring et alerting

- Intégration avec Elastic Stack Monitoring, Prometheus et Grafana
- Configuration de métriques clés et seuils d'alerte
- Visualisation des indicateurs de performance et de santé du cluster
- Analyse des logs de cluster et détection d'anomalies

8. Atelier pratique

- Mise en place d'un cluster multi-nœuds avec HA et sécurité
- Tuning d'un index volumineux et optimisation de requêtes réelles
- Création d'un ingest pipeline pour un cas métier (logs ou analytics)
- Simulation de sinistre et restauration via snapshots

Power BI – Niveau Débutant

Prix : 1580 € H.T.

Durée : 2 jours

Public cible : Analystes métier, responsables reporting, consultants ou toute personne souhaitant créer rapidement des tableaux de bord interactifs. Développeurs et data engineers souhaitant intégrer Power BI dans leur flux de données. Décideurs et managers souhaitant comprendre les possibilités de la plateforme

Prérequis : Connaissances de base en Excel (tableaux, formules simples, graphiques). Notions élémentaires de bases de données relationnelles et SQL. Ordinateur avec Power BI Desktop installé

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Se connecter à différentes sources de données (fichiers, bases, services cloud)
- Préparer et transformer les données avec Power Query
- Modéliser les données (relations, hiérarchies, mesures de base)
- Créer des visualisations interactives et construire un rapport unifié
- Publier, partager et rafraîchir un tableau de bord sur Power BI Service

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. Introduction à Power BI et son écosystème

- Présentation de la suite Power Platform (Desktop, Service, Mobile, Report Server)
- Cas d'usage BI self-service vs BI traditionnelle
- Architecture et flux de travail : data-connect →model →report →publish

2. Connexion et préparation des données (Power Query)

- Connexion à des sources variées : Excel, CSV, SQL Server, SharePoint, APIs
- Transformation : filtres, colonnes calculées, fusion, appariement, pivot/unpivot
- M-language : aperçu des formules et étapes de requête
- Bonnes pratiques de nettoyage et documentation des requêtes

3. Modélisation de données

- Concepts de modèle en étoile : tables factuelles, tables de dimensions
- Définition des relations et cardinalités
- Création de hiérarchies et de tables de dates
- Introduction au DAX : mesures vs colonnes calculées

4. Visualisations et rapports

- Galerie de visuels : tables, matrices, graphiques barres, lignes, cartes, KPI
- Personnalisation : filtres, slicers, bookmarks, boutons d'action
- Conception de pages interactives et navigation entre vues
- Accessibilité et storytelling : mise en page, couleurs, bonnes pratiques UX

5. Power BI Service et collaboration

- Publication depuis Desktop vers Power BI Service
- Création d'espaces de travail, applications et apps Power BI
- Partage, permissions, et gestion des accès
- Configuration du rafraîchissement des jeux de données (gateways)

6. Introduction aux fonctionnalités avancées

- Q&A (requêtes en langage naturel)
- Alertes et abonnements aux rapports
- Introduction aux visuels personnalisés du marketplace
- Aperçu de Power BI Mobile

7. Atelier pratique

- Construction d'un tableau de bord de reporting métier (ventes, finance, RH…)
- Import, transformation, modélisation et mise en forme finale
- Publication et configuration du rafraîchissement automatique

Développement logiciel

Algorithmique – Maîtriser la logique de programmation

Prix : 1200 € H.T.

Durée : 2 jours

Public cible : Toute personne souhaitant acquérir les bases de la logique algorithmique avant d'aborder un langage de programmation ou de comprendre les bases des systèmes informatiques.

Prérequis : Aucun prérequis technique n'est nécessaire, mais une aisance avec l'ordinateur est recommandée.

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Comprendre les fondements de l'algorithmique et des structures de contrôle
- Traduire un raisonnement logique en étapes claires (pseudo-code, organigrammes)
- Concevoir des algorithmes simples pour résoudre des problèmes
- Décomposer un problème en sous-tâches cohérentes
- Se préparer à l'apprentissage d'un langage de programmation (Python, JavaScript, etc.)

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. Introduction à l'algorithmique

- Qu'est-ce qu'un algorithme ? Définition et exemples concrets
- Rôle de l'algorithmique dans l'informatique et la data
- Outils de modélisation : pseudo-code, logigramme, langage naturel

2. Structures de base

- Les types de données (entiers, chaînes, booléens)
- Variables et affectations
- Opérations arithmétiques et logiques

3. Contrôle du flux

- Structures conditionnelles : si, sinon, sinon si
- Boucles : pour, tant que
- Exercices de mise en pratique (calculs, analyse de chaînes, conversions)

4. Fonctions et modularité

- Définir une fonction, passer des arguments, retourner un résultat
- Réutilisation et lisibilité du code
- Décomposer un problème en sous-algorithmes

5. Initiation à la complexité

- Notion intuitive de performance algorithmique
- Meilleure solution vs solution fonctionnelle

6. Cas pratiques

- Algorithmes classiques : recherche, tri, maximum/minimum
- Résolution guidée de petits défis (fizzbuzz, palindrome, factorielle)
- Préparation à l'implémentation en Python (sans codage requis)

Structures de données et algorithmes avancés

Prix : 1900 € H.T.

Durée : 3 jours

Public cible : Développeurs débutants ou intermédiaires, analystes, techniciens ou toute personne souhaitant améliorer la performance et la structuration de son code.

Prérequis : Connaissances de base en algorithmique et en programmation (idéalement Python ou équivalent).

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Identifier et utiliser les structures de données adaptées à un problème donné
- Implémenter des algorithmes classiques et comprendre leur complexité
- Optimiser le traitement de données grâce à une meilleure structuration
- Approfondir leurs compétences pour aborder les domaines de la data science ou de l'IA

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. Rappels fondamentaux

- Variables, types simples, fonctions
- Notion de complexité (temps/espaces, notation Big O)

2. Structures linéaires

- Listes, piles (stack) et files (queue)
- Implémentation, utilisation, complexité
- Cas d'usage : gestion d'historique, files d'attente

3. Structures hiérarchiques

- Arbres binaires : structure, parcours (préfixe, infixe, suffixe)
- Arbres de recherche (BST)
- Cas pratique : tri, organisation hiérarchique

4. Graphes

- Représentation (matrice d'adjacence, listes)
- Parcours en largeur (BFS) et en profondeur (DFS)
- Cas pratique : réseaux, chemins optimaux

5. Algorithmes de tri et de recherche

- Tri par sélection, insertion, bulle, rapide (quicksort), fusion (mergesort)
- Recherche linéaire, dichotomique
- Analyse de performance et choix adapté

6. Structures avancées

- Tables de hachage (dictionnaires)
- Heaps et files de priorité
- Cas d'usage : cache mémoire, planification

7. Mise en pratique

- Choisir la bonne structure pour un cas donné
- Optimisation d'un algorithme existant
- Résolution guidée de problèmes (format coding challenge)

Initiation à la programmation avec Python

Prix : 1910 € H.T.

Durée : 3 jours

Public cible : Toute personne souhaitant s'initier à la programmation à l'aide d'un langage simple, lisible et utilisé dans de nombreux domaines (automatisation, data, IA, web, etc.).

Prérequis : Aucun prérequis en programmation. Il est recommandé d'avoir suivi la formation « Algorithmique » ou d'en maîtriser les bases.

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Comprendre le fonctionnement d'un programme écrit en Python
- Manipuler les types de données, structures de contrôle et fonctions
- Écrire des scripts simples pour automatiser des tâches ou résoudre des problèmes
- Lire et comprendre du code Python standard
- Se préparer à des applications avancées (data, IA, automation)

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. Introduction à Python

- Historique, usages et philosophie du langage
- Installation de l'environnement (Python, IDE type VS Code ou Jupyter Notebook)
- Premiers scripts : structure d'un fichier Python, exécution

2. Types de base et opérations

- Nombres, chaînes de caractères, booléens
- Opérations arithmétiques, concaténation, comparaisons
- Conversion entre types

3. Structures de données

- Listes, tuples, dictionnaires, ensembles
- Parcours et manipulation d'éléments
- Fonctions intégrées utiles (len, range, enumerate, zip...)

4. Contrôle du flux

- Instructions conditionnelles (if, elif, else)
- Boucles for et while
- Gestion des exceptions (try, except)

5. Fonctions et modularité

- Définir une fonction avec ou sans paramètres
- Retourner une valeur, portée des variables
- Modules et importations

6. Introduction aux fichiers et à la console

- Lire et écrire des fichiers texte (.txt, .csv)
- Entrée utilisateur avec input(), impression avec print()

7. Petits projets guidés

- Calculatrice simple, conversion d’unités
- Analyse d’un fichier texte (ex : comptage de mots)
- Mini-jeu texte (ex : devine le nombre)

Manipuler et analyser des données avec Python (Pandas & NumPy)

Prix : 2200 € H.T.

Durée : 3 jours

Public cible : Développeurs, analystes, ingénieurs, data scientists débutants ou toute personne souhaitant apprendre à traiter efficacement des données tabulaires avec Python.

Prérequis : Bonne maîtrise des bases de Python (structures de données, fonctions, boucles). Une familiarité avec les fichiers CSV ou les bases de données est un plus.

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Lire, nettoyer et transformer des données avec Pandas
- Réaliser des analyses statistiques simples
- Comprendre les bases du calcul vectoriel avec NumPy
- Préparer des données pour des visualisations ou des modèles IA

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. Introduction à l'analyse de données

- La data dans les projets numériques : enjeux et étapes
- Présentation de l'écosystème Python pour la data (Pandas, NumPy, Matplotlib, etc.)

2. Manipulation de données avec Pandas

- Séries et DataFrames : création, lecture depuis CSV/Excel
- Accès, filtrage et sélection des données
- Opérations de nettoyage : traitement des valeurs manquantes, renommage, typage

3. Transformations et jointures

- Filtres conditionnels, transformations par colonnes
- Fusion de DataFrames (merge, concat)
- Agrégations et regroupements (groupby, pivot tables)

4. Introduction à NumPy

- Tableaux (ndarray), vecteurs et matrices
- Indexation, slicing, opérations vectorisées
- Performance vs boucles classiques

5. Statistiques descriptives

- Moyenne, médiane, écart-type, corrélation
- Identification des outliers et valeurs atypiques
- Tri croisé, tableaux récapitulatifs

6. Mise en pratique

- Étude de cas sur données réelles (CSV, Excel)
- Mini-projets guidés : analyse de ventes, scoring de clients, tendances temporelles

Visualiser des données avec Python (Matplotlib, Seaborn, Plotly)

Prix : 1460 € H.T.

Durée : 2 jours

Public cible : Professionnels manipulant des données (analystes, chargés d'études, chefs de projet, développeurs) souhaitant produire des visualisations claires et percutantes.

Prérequis : Maîtrise de base de Python et du package Pandas. Avoir suivi la formation "Manipuler et analyser des données avec Python" est recommandé.

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Créer des graphiques avec les bibliothèques Python les plus courantes
- Adapter les types de graphiques aux types de données et aux messages à faire passer
- Personnaliser les visuels pour améliorer leur lisibilité et leur impact
- Réaliser des visualisations interactives avec Plotly

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. Introduction à la visualisation de données

- Rôle de la data visualisation : comprendre, explorer, convaincre
- Choisir le bon graphique selon le message à transmettre
- Présentation des bibliothèques : Matplotlib, Seaborn, Plotly

2. Visualisation avec Matplotlib

- Création de courbes simples, barres, nuages de points
- Personnalisation : couleurs, légendes, titres, axes
- Gestion des sous-graphes (subplot)

3. Visualisation avancée avec Seaborn

- Graphiques statistiques (boxplot, violin, heatmap, pairplot…)
- Utilisation avec DataFrames Pandas
- Représentation multi-variables et catégorielles

4. Visualisations interactives avec Plotly

- Graphiques dynamiques : zoom, filtre, survol
- Intégration dans des notebooks ou dashboards simples
- Comparatif avec les outils statiques

5. Mise en pratique

- Analyse exploratoire guidée avec visualisations
- Construction d'un tableau de bord graphique
- Études de cas : indicateurs commerciaux, analyse de trafic, sondages

Programmation Orientée Objet (POO)

Prix : 1910 € H.T.

Durée : 3 jours

Public cible : Développeurs débutants ou ayant des connaissances en programmation (connaissance de base en Java ou d'un autre langage de programmation impératif)

Prérequis : Connaissances de base en programmation (structures de contrôle, variables, boucles, fonctions)

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Comprendre les principes fondamentaux de la programmation orientée objet
- Concevoir et développer des applications en utilisant la POO
- Implémenter des classes, objets, héritage, polymorphisme et encapsulation
- Appliquer les bonnes pratiques de la POO dans un environnement Java

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. Introduction à la POO

- Définitions de la POO : Object, Class, Encapsulation, Héritage, Polymorphisme
- Historique et avantages de la POO
- Comparaison avec les langages procéduraux

2. Création d'une classe en Java

- Déclaration d'une classe et d'un objet
- Attributs, méthodes, et constructeurs
- Notion de portée (private, public, protected)

3. Encapsulation

- Accès aux attributs via les méthodes getters et setters
- Contrôle de l'accès aux données
- Avantages de l'encapsulation

4. Héritage

- Définition et mécanisme d'héritage (extends)
- Utilisation de la méthode super
- Surcharge et redéfinition des méthodes
- Avantages et limitations de l'héritage

5. Polymorphisme

- Surcharge de méthodes (overloading)
- Redéfinition de méthodes (overriding)
- Utilisation des interfaces pour le polymorphisme
- Exemple d'application du polymorphisme dans des projets réels

6. Abstraction

- Création de classes abstraites
- Définition d'interfaces et leur mise en œuvre
- Différence entre classe abstraite et interface

7. Exemples pratiques et études de cas

- Application des concepts dans des projets réels
- Études de cas sur la conception orientée objet
- Développement d'une mini-application en utilisant les principes de la POO

8. Bonnes pratiques en POO

- Respect des principes SOLID
- Utilisation d'annotations et gestion des exceptions
- Structuration du code et design patterns de base

Java pour Débutants

Prix : 3030 € H.T.

Durée : 5 jours

Public cible : Développeurs débutants ou toute personne souhaitant apprendre les bases du développement avec Java.

Prérequis : Aucune expérience préalable en programmation n'est nécessaire, mais une familiarité avec l'informatique de base (utilisation de systèmes d'exploitation, de logiciels) est recommandée.

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Comprendre la syntaxe fondamentale du langage Java
- Créer des applications simples en Java en utilisant des concepts de programmation de base
- Manipuler les structures de données essentielles (tableaux, listes, etc.)
- Appliquer les bonnes pratiques de codage en Java
- Utiliser un environnement de développement Java moderne (JDK, IDE, Maven)

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. Introduction à Java et à l'environnement de développement

- Historique et caractéristiques du langage Java
- Installation et configuration du JDK et d'un IDE (IntelliJ IDEA, Eclipse, etc.)
- Structure d'un programme Java : classes, méthodes, et la méthode main()
- Compilation et exécution d'un programme Java dans un IDE

2. Les bases du langage Java

- Types primitifs et variables (int, double, char, boolean)
- Utilisation des opérateurs arithmétiques et logiques
- Structures de contrôle : conditions (if, switch) et boucles (for, while, do-while)
- Tableaux : déclaration, initialisation et manipulation de tableaux simples
- Manipulation des chaînes de caractères (String), méthodes utiles

3. Introduction à la Programmation Orientée Objet (POO)

- Concepts clés : classes, objets, attributs et méthodes
- Création et instanciation de classes en Java
- Notion de constructor, gestion de la portée des variables (private, public, protected)
- Différence entre méthode statique et méthode d'instance

4. Gestion des erreurs et exceptions

- Concepts de base des exceptions en Java
- Bloc try/catch pour la gestion des erreurs
- Utilisation des exceptions personnalisées
- Bonnes pratiques de gestion des erreurs et exceptions en Java

5. Les collections et l'API standard de Java

- Introduction aux collections : List, Set, Map
- Création et manipulation de listes avec ArrayList, utilisation des itérateurs
- Manipulation des ensembles (HashSet, TreeSet) et des maps (HashMap, TreeMap)
- Tri des collections, utilisation des Comparator et Comparable

6. Travailler avec les fichiers et les entrées/sorties (I/O)

- Lecture et écriture dans des fichiers texte avec BufferedReader et BufferedWriter
- Sérialisation d'objets en Java
- Introduction aux flux de données en Java (fichiers texte, binaires)

7. Les dernières fonctionnalités de Java (version actuelle)

- Introduction à Java 17 et les fonctionnalités récentes :
- Pattern Matching pour instanceof (Java 16)
- Records pour simplifier la déclaration de classes de données (Java 14)
- Text Blocks pour faciliter la manipulation de chaînes multi-lignes (Java 13)
- Switch Expressions améliorées (Java 12)
- Sealed Classes pour restreindre les héritages (Java 17)
- Utilisation des nouvelles API : java.nio.file, java.util.stream, java.time (dates et heures)

8. Bonnes pratiques et structuration du code

- Respect des conventions de nommage
- Structuration d'un projet Java simple avec Maven
- Introduction à la gestion de versions avec Git
- Introduction aux tests unitaires avec JUnit

9. Exemples pratiques et exercices

- Développement d'un programme simple : gestion d'une bibliothèque, gestion des utilisateurs, ou calculatrice
- Résolution d'exercices pratiques pour appliquer les concepts appris

Java – Niveau Expert

Prix : 3030 € H.T.

Durée : 5 jours

Public cible : Développeurs expérimentés ayant une bonne maîtrise de Java et des concepts de programmation orientée objet, souhaitant approfondir leur expertise dans les domaines avancés du langage et des architectures Java complexes.

Prérequis : Maîtrise des bases de Java, des concepts fondamentaux de la POO, et des outils de développement Java (IDE, Maven). Expérience en développement avec Java ou tout autre langage orienté objet.

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Concevoir et implémenter des architectures Java robustes, évolutives et performantes
- Utiliser les fonctionnalités avancées du langage Java pour résoudre des problèmes complexes
- Appliquer des principes de conception avancés, des design patterns, et des bonnes pratiques d'architecture logicielle
- Optimiser les performances d'applications Java en utilisant les dernières fonctionnalités et outils

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. Introduction aux concepts avancés de Java

- Vue d'ensemble de la plateforme Java et des évolutions récentes (Java 17 et au-delà)
- Comprendre l'architecture de la JVM, le garbage collector, et la gestion de la mémoire
- Analyse des dernières fonctionnalités du langage : Records, Pattern Matching, Sealed Classes, Text Blocks, etc.

2. Optimisation de la performance en Java

- Comprendre et analyser la gestion de la mémoire en Java : heap, stack, et garbage collection
- Optimisation des performances avec la JVM : tuning, monitoring (JVisualVM, JConsole, JProfiler)
- Stratégies de gestion de la mémoire et gestion des fuites mémoire
- Amélioration des performances avec des collections adaptées (ConcurrentHashMap, CopyOnWriteArrayList, etc.)
- Optimisation des entrées/sorties (I/O) : meilleures pratiques avec les flux, buffers, et NIO

3. Conception avancée avec la programmation fonctionnelle

- Lambda expressions et Stream API pour une programmation fonctionnelle en Java
- Utilisation des Optional pour éviter les nulls et améliorer la sécurité du code
- Programmation réactive avec Reactive Streams (frameworks comme Project Reactor)
- Combinaison de la POO et de la programmation fonctionnelle pour des architectures plus flexibles et lisibles

4. Architecture Java avancée

- Design Patterns : Application des patterns de conception pour résoudre des problèmes complexes (Singleton, Factory, Strategy, Observer, etc.)
- Création et gestion des microservices avec Spring Boot, communication entre services via RESTful APIs, gRPC, et WebSockets
- Gestion des transactions et des connexions avec JPA et Hibernate
- Introduction aux architectures hexagonales et clean architecture en Java

5. Programmation concurrente et multi-threading

- Compréhension des threads, ExecutorService, Callable et Future
- Synchronisation des threads : synchronized, Locks, Atomic variables
- Introduction aux CompletableFuture et aux concepts de la programmation asynchrone
- Résolution des problèmes de concurrence et des deadlocks dans un environnement Java multi-threadé

6. Tests unitaires et TDD (Test-Driven Development)

- Mise en œuvre des tests unitaires avancés avec JUnit 5 et Mockito
- Test-Driven Development (TDD) : Stratégies et bonnes pratiques pour la rédaction de tests en amont
- Tests d'intégration et tests fonctionnels
- Outils de couverture de code : JaCoCo, SonarQube
- Introduction à la gestion des tests dans des projets multi-modules

7. Utilisation des outils modernes et des bonnes pratiques de développement

- Mise en œuvre de CI/CD avec Jenkins, GitLab CI, ou GitHub Actions pour Java
- Gestion des dépendances avec Maven et Gradle
- Surveillance et gestion des logs avec SLF4J, Logback, et ELK Stack
- Sécurisation des applications Java : protection contre les attaques courantes (Injection SQL, XSS, CSRF)

8. Gestion de projets complexes et évolutifs

- Stratégies de développement pour des applications modulaires et évolutives
- Modularité avec le système de modules Java (JPMS - Java Platform Module System)
- Gestion des versions et des dépendances dans des environnements complexes
- Utilisation de Docker pour le développement et le déploiement d'applications Java en conteneurs

9. Cas pratiques et projets avancés

- Développement d'une application Java complexe intégrant des microservices, une base de données, et une architecture de communication (REST/gRPC)
- Application des concepts de programmation concurrente et de performances dans un projet réel
- Résolution de cas d'architecture Java complexes en groupe

JavaScript – Niveau Expert

Prix : 3030 € H.T.

Durée : 5 jours

Public cible : Développeurs ayant une maîtrise des bases de JavaScript et une expérience de développement web, souhaitant approfondir leur expertise dans les concepts avancés et les bonnes pratiques du langage JavaScript.

Prérequis : Maîtrise des concepts de base de JavaScript, du DOM, des fonctions et des événements, ainsi que des connaissances de base en HTML et CSS. Une expérience préalable en développement avec JavaScript est fortement recommandée.

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Concevoir des architectures JavaScript complexes et maintenables
- Maîtriser les fonctionnalités avancées du langage (ES6+, async/await, modules, etc.)
- Optimiser les performances des applications JavaScript
- Utiliser des outils modernes de développement (Webpack, Babel, TypeScript)
- Appliquer des design patterns et des techniques avancées pour le développement d'applications web performantes et robustes

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. Rappels et révision des bases avancées

- Fonctionnement de JavaScript : moteur V8, compilation et interprétation
- Les nouveautés ECMAScript 6 (ES6+) : let, const, classes, modules, template literals, destructuration
- Gestion des modules JavaScript avec import et export
- Promises, async/await pour la gestion des tâches asynchrones

2. Les fonctions avancées en JavaScript

- Fonctions de première classe et fonctions d'ordre supérieur
- Fonction bind, call et apply et leur utilisation dans des contextes spécifiques
- Manipulation avancée des closures et des portées (scope)
- Gestion des callbacks et inversion de contrôle (callback hell)
- Utilisation des générateurs (generators) et des itérateurs

3. Programmation fonctionnelle en JavaScript

- Concepts fondamentaux : immutabilité, pureté des fonctions, réutilisabilité
- Utilisation des méthodes de tableau avancées : map(), filter(), reduce(), some(), every()
- Fonctions d'ordre supérieur pour la composition de fonctions
- La notion de currying et des fonctions partielles

4. Gestion des erreurs avancée et debug

- Traitement des erreurs avec try/catch, gestion des erreurs asynchrones (promises)
- Meilleures pratiques pour la gestion des exceptions en JavaScript
- Techniques de débogage avancées (Utilisation de console, breakpoints dans les outils dev)
- Suivi des erreurs en production avec des outils comme Sentry

5. Architecture et design patterns en JavaScript

- Introduction aux design patterns : Singleton, Factory, Observer, Module, et MVC
- Utilisation des patterns pour l'architecture d'applications web complexes
- Gestion des états avec des architectures comme Flux et Redux

- Architecture modulaire avec JavaScript et gestion des dépendances avec ES Modules et Webpack

6. Optimisation des performances JavaScript

- Comprendre la boucle d'événements et le modèle de concurrence
- Gestion des memory leaks et optimisation de la consommation mémoire
- Profilage de performance : analyse du temps d'exécution avec des outils comme Chrome DevTools
- Optimisation des performances du DOM : techniques de batch updates, minimisation des reflows/repaints
- Meilleures pratiques pour le rendu côté client et la gestion des tâches asynchrones

7. Les API modernes en JavaScript

- Manipulation des Fetch API et des WebSockets pour les communications réseau
- Utilisation de l'API Fetch pour remplacer XMLHttpRequest
- Gestion des Service Workers et Web Workers pour la gestion de tâches en arrière-plan
- Introduction aux Progressive Web Apps (PWA)

8. TypeScript et JavaScript : Passage de JS à TS

- Introduction à TypeScript : pourquoi et comment migrer d'un projet JavaScript vers TypeScript
- Utilisation des types statiques et avantages de TypeScript dans la gestion des grandes bases de code
- Compilation et intégration avec des outils comme Webpack et Babel

9. Tests avancés en JavaScript

- Introduction aux tests unitaires avec Jest et Mocha
- Tests d'intégration, tests fonctionnels avec Cypress
- Mise en œuvre de la méthodologie TDD (Test-Driven Development) en JavaScript
- Mocking, spying et tests des fonctions asynchrones

10. Exercices pratiques et mini-projets avancés

- Développement d'une application complexe avec architecture modulaire, gestion d'état, et appels API
- Mise en œuvre de design patterns pour une gestion efficace des différentes parties de l'application
- Optimisation de la performance et des interactions utilisateur avec des techniques avancées de manipulation du DOM
- Intégration de tests unitaires, d'intégration et fonctionnels dans une application JavaScript complète

React – Débutant

Prix : 1900 € H.T.

Durée : 3 jours

Public cible : Développeurs débutants ou toute personne ayant une expérience en JavaScript et souhaitant apprendre à créer des applications web dynamiques avec React.

Prérequis : Maîtrise des bases de JavaScript (notamment les fonctions, objets, tableaux et événements). Une connaissance de base du HTML et du CSS est recommandée.

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Comprendre les principes fondamentaux de React et son écosystème
- Créer des applications web réactives avec React
- Utiliser les composants, les états et les props pour construire des interfaces utilisateur dynamiques
- Gérer les événements utilisateur et manipuler le DOM via React
- Intégrer des API REST pour récupérer et afficher des données

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. Introduction à React

- Qu'est-ce que React et son rôle dans le développement d'applications web modernes ?
- Les avantages de React : Réutilisation de composants, performance avec le Virtual DOM, unidirectional data flow
- Comprendre les composants : Class components vs. Functional components
- Installation de React via Create React App

2. Création d'un premier composant React

- Structure d'une application React : index.js, App.js
- Définition d'un composant avec des fonctions (stateless functional components)
- Rendu dynamique avec JSX (syntax sugar pour écrire du HTML dans JavaScript)
- Introduction aux props : Qu'est-ce que sont les props et comment les utiliser pour passer des données dans les composants

3. Les états dans React (useState)

- Introduction à l'état local dans un composant
- Gestion de l'état avec le hook useState
- Manipulation de l'état : changer l'état en fonction des actions de l'utilisateur (ex. : clic, saisie dans un champ)
- Interaction entre l'état et l'interface utilisateur : rafraîchissement du rendu lorsque l'état change

4. Gestion des événements dans React

- Introduction aux événements dans React (clics, changements, soumissions de formulaires)
- Utilisation des gestionnaires d'événements : onClick, onChange, onSubmit
- Passage des paramètres aux fonctions de gestion d'événements
- Liaisons entre la logique métier et l'interface utilisateur via les événements

5. Composants enfants et communication entre composants

- Passer des données entre composants via les props
- Création de composants réutilisables et dynamiques
- Prop drilling : Comment passer des données profondément dans la hiérarchie des composants
- Solutions pour la gestion de l'état global (introduction à Context API)

6. Introduction au Routing avec React Router

- Installation et configuration de React Router pour la gestion des pages
- Créer des liens de navigation avec Link et NavLink
- Gérer les routes et les paramètres de l'URL pour rendre des composants spécifiques
- Mise en place d'un rendu conditionnel pour afficher différentes vues selon l'URL

7. Travailler avec des API et gestion des données

- Introduction aux API REST : récupérer des données avec fetch()
- Utiliser useEffect pour effectuer des appels API lors du montage du composant
- Manipulation des données reçues : affichage dynamique des informations dans l'interface
- Gestion des erreurs et des chargements (ex : état de chargement, gestion des erreurs)

8. Gestion du formulaire dans React

- Création d'un formulaire simple avec gestion de l'état des champs
- Validation des entrées de formulaire : contrôle des erreurs avant soumission
- Utilisation de useState et onSubmit pour gérer la soumission des formulaires
- Implémentation de champs contrôlés et non contrôlés

9. Bonnes pratiques et organisation du code

- Structurer une application React : Organisation des dossiers et fichiers
- Modularité et réutilisabilité des composants
- Gestion des clés uniques avec key dans les listes de composants
- Introduction à la gestion des erreurs dans les applications React

10. Exercices pratiques et mini-projets

- Création d'une application de liste de tâches avec React (ajouter, supprimer, marquer comme complété)
- Développement d'une application météo avec appel à une API pour récupérer les données météorologiques et affichage des résultats
- Mise en place d'un formulaire interactif de contact avec validation et soumission des données

Angular – Niveau Expert

Prix : 2500 € H.T.

Durée : 4 jours

Public cible : Développeurs Angular confirmés souhaitant maîtriser les aspects avancés du framework pour concevoir, optimiser et maintenir des applications complexes en environnement professionnel.

Prérequis : Solide expérience d'Angular (niveaux débutant et intermédiaire maîtrisés), connaissance approfondie de TypeScript, des composants, services, formulaires réactifs, RxJS et du routing.

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Concevoir des architectures évolutives et maintenables avec Angular
- Implémenter des performances optimisées et du lazy loading avancé
- Maîtriser RxJS pour des flux complexes
- Développer et tester des directives et composants réutilisables
- Gérer l'état de l'application avec NgRx
- Assurer la qualité, sécurité et testabilité du code Angular

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. Architecture avancée d'une application Angular

- Organisation modulaire d'un projet à grande échelle
- Utilisation des FeatureModules, CoreModule, SharedModule
- Hiérarchisation et encapsulation des dépendances
- Définition d'une architecture scalable et testable

2. Optimisation des performances

- Change Detection : stratégie Default vs OnPush
- Utilisation avancée de trackBy, memoization avec pure pipes
- Chargement différé des modules (PreloadingStrategy, Quicklink)
- Optimisation des cycles de vie de composants (ngOnChanges, ngDoCheck)
- Détection et élimination des fuites de mémoire

3. RxJS avancé pour Angular

- Composition de flux avec des opérateurs complexes : concatMap, exhaustMap, combineLatest, withLatestFrom
- Gestion d'erreurs, annulation de requêtes, retry strategies
- Création de services orientés flux de données
- Application réactive : état, UI et appels API liés par RxJS

4. NgRx – Gestion d'état avec Redux

- Présentation du pattern Redux et du store NgRx
- Actions, reducers, selectors, effects : architecture complète
- Gestion des effets secondaires avec @ngrx/effects
- Optimisation des performances et lazy loading du store
- Alternative légère : ComponentStore pour des scopes plus localisés

5. Développement de composants avancés et librairies Angular

- Création de librairies Angular (ng generate library)
- Revue des mécanismes d'encapsulation CSS (ViewEncapsulation)

- Utilisation de ContentProjection et ngTemplateOutlet
- Création de composants dynamiques avec ComponentFactoryResolver
- Packaging et distribution d'une librairie interne

6. Sécurité et bonnes pratiques

- Prévention des vulnérabilités XSS, CSRF dans Angular
- Stratégies de sécurisation du routing et du backend
- Gestion des tokens et headers (JWT, OAuth) via HttpInterceptor
- Audit de dépendances et mises à jour (ng update, npm audit fix)

7. Tests avancés (unitaires et E2E)

- Bonnes pratiques de test avec Jest ou Karma/Jasmine
- Tests de composants avec dépendances : TestBed, mock services
- Tests d'effets avec @ngrx/effects/testing
- Tests E2E robustes avec Cypress : scénarios complexes et mocks

8. Déploiement et intégration continue

- Build production optimisé : ng build –configuration production
- Configuration d'environnement multi-cibles (environment.*.ts)
- Intégration continue : GitHub Actions, GitLab CI/CD
- Hébergement sur Firebase, Vercel ou Docker

9. Atelier final – Projet complet d'entreprise

- Création d'un tableau de bord sécurisé et modulaire
- Lazy loading, NgRx, routing imbriqué, composants partagés
- Déploiement simulé et revue de qualité du code

Vue.js – Débutant

Prix : 1900 € H.T.

Durée : 3 jours

Public cible : Développeurs front-end débutants ou confirmés souhaitant découvrir Vue.js pour créer des interfaces web modernes et interactives.

Prérequis : Bonnes bases en JavaScript, HTML et CSS. Aucune connaissance préalable de Vue.js requise.

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Comprendre l'architecture et les principes de Vue.js 3
- Créer des composants interactifs et modulaires
- Gérer les données, événements, formulaires et directives
- Construire une application monopage simple avec Vue Router
- Organiser un projet Vue moderne avec Vite et Composition API

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. Introduction à Vue.js

- Vue.js : historique, philosophie, place dans l'écosystème JS
- Avantages et cas d'usage
- Présentation des versions (Vue 2 vs Vue 3)
- Mise en place du projet avec Vite ou Vue CLI

2. Fondamentaux de Vue.js

- Vue Instance et cycle de vie
- Le DOM réactif
- Data binding : interpolation, v-bind, v-model
- Gestion des événements : v-on, raccourcis clavier, événements personnalisés

3. Composants Vue

- Définition et structure d'un composant
- Props et communication parent-enfant
- Slots (simples et nommés)
- Gestion de l'état local et réutilisabilité

4. Directives et templates

- Directives intégrées : v-if, v-for, v-show, v-bind, v-on
- Classes et styles dynamiques
- Utilisation des filtres (deprecated), remplacement par computed/methods
- Création de directives personnalisées simples

5. Composition API – Vue 3

- Introduction à la Composition API
- setup(), ref(), reactive(), computed(), watch()
- Comparaison avec Options API
- Organisation du code : composabilité, lisibilité, réutilisabilité

6. Gestion des formulaires

- Binding avec v-model
- Validation de base
- Introduction à des bibliothèques de validation comme VeeValidate

7. Navigation avec Vue Router

- Configuration du router
- Création de routes, redirections, routes dynamiques
- Navigation programmatique
- Passage de paramètres et vues imbriquées

8. Introduction à la gestion d'état (Pinia)

- Présentation de Pinia (successeur de Vuex)
- Création et utilisation d'un store simple
- Communication entre composants via store

9. Exercices pratiques et mini-projets

- Création d'une application TODO ou gestionnaire de notes
- Interface formulaire avec validation simple
- Mini dashboard avec plusieurs vues et composants

Node.js – Niveau intermédiaire

Prix : 1900 € H.T.

Durée : 3 jours

Public cible : Développeurs ayant déjà utilisé Node.js pour créer des scripts ou une API simple, souhaitant structurer des projets backend plus complexes.

Prérequis : Bonne maîtrise de JavaScript moderne et des bases de Node.js (Express, routage, gestion des requêtes et réponses, Promises).

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Structurer une application Node.js avec une architecture modulaire
- Implémenter des middlewares personnalisés et sécuriser les routes
- Intégrer et manipuler une base de données (MongoDB ou PostgreSQL)
- Gérer les erreurs de façon centralisée
- Documenter et tester leur API REST

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. Architecture d'une application Node.js

- Structure MVC simplifiée (routes / contrôleurs / services)
- Modularisation du code, usage des modules ES
- Organisation des middlewares, gestion des dépendances

2. Express avancé

- Création de middlewares personnalisés (authentification, logs, erreurs)
- Route grouping, express.Router(), hiérarchie des routes
- Filtres et limitations d'accès (middleware conditionnel)
- Upload de fichiers avec multer

3. Gestion des erreurs et logs

- Centralisation des erreurs (middleware d'erreur)
- Formatage des erreurs personnalisées (AppError, errorHandler)
- Logging avec morgan, winston, logs structurés

4. Sécurité des API

- Authentification JWT (JSON Web Token)
- Hashage de mots de passe avec bcrypt
- Protection contre les injections, XSS, CSRF
- Limitation de débit (express-rate-limit), nettoyage des entrées (express-mongo-sanitize)

5. Base de données (MongoDB ou PostgreSQL)

- Connexion à une base MongoDB (avec mongoose) ou PostgreSQL (avec pg ou Prisma)
- Création de modèles (schemas, relations)
- Requêtes complexes, pagination, tri, filtres
- Gestion des migrations (pour SQL)

6. Appels asynchrones robustes

- Utilisation avancée de async/await

- Traitement de logique métier asynchrone
- Gérer les cas d'échec réseau ou base de données

7. Documentation de l'API

- Introduction à Swagger (via swagger-ui-express)
- Création d'un fichier de documentation OpenAPI
- Test interactif des endpoints via Swagger UI ou Postman

8. Tests et qualité

- Tests unitaires avec Jest (routes, services)
- Tests d'intégration API (supertest)
- Mock de base de données
- Intégration dans une CI (GitHub Actions en option)

9. Projet pratique

- Conception et développement d'une API complète :
- Authentification, CRUD complet, validation
- Base de données connectée
- Documentation Swagger
- Gestion des erreurs et sécurité

Next.js – Niveau Expert

Prix : 2300 € H.T.

Durée : 4 jours

Public cible : Développeurs expérimentés en React/Next.js souhaitant concevoir des applications web performantes, scalables et prêtes pour la production à grande échelle.

Prérequis : Bonne maîtrise de React, de JavaScript moderne, et de Next.js (routage, Server/Client Components, récupération de données).

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Structurer une application Next.js selon les meilleures pratiques (App Router, modularisation, performance)
- Maîtriser les Server Components, Server Actions, streaming et ISR avancé
- Mettre en œuvre des techniques d'optimisation du rendu, du chargement et du SEO
- Intégrer l'internationalisation, la gestion d'état et des middlewares avancés
- Déployer une application Next.js optimisée, sécurisée et maintenable

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. Architecture avancée avec App Router

- App Router vs Pages Router : état de l'art
- Organisation d'un projet en modules, layouts imbriqués, providers globaux
- Optimisation du lazy loading et des bundles
- Gestion multi-domaines, micro-frontends (aperçu)

2. Server Components & Server Actions

- Approfondissement : composant purement serveur et client hybride
- Server Actions (Next.js 14) : appels directs à des fonctions côté serveur depuis des formulaires
- Sécurité des Server Actions, validation, revalidation automatique

3. Rendu dynamique et streaming

- Streaming partiel des pages (React Server Components)
- Utilisation de Suspense, loading.tsx et error.tsx
- Incremental Static Regeneration avancé (ISR dynamique, revalidateTag, revalidatePath)
- Détection automatique des chemins statiques et dynamiques

4. Internationalisation et routing avancé

- next-intl ou i18n natif de Next.js
- Routage conditionnel selon la langue
- Domaine dédié par locale
- Redirections et réécritures dynamiques (next.config.js)

5. API, middleware et Edge functions

- Création d'API REST et RPC via les routes route.ts
- Middleware Edge : gestion d'authentification, géolocalisation, A/B testing
- Middleware de rewriting conditionnel (edge config, headers dynamiques)
- Introduction à Vercel Edge Runtime

6. Sécurité et bonnes pratiques

- Authentification sécurisée (JWT, OAuth2, NextAuth.js)
- Server Actions et validation des inputs avec zod
- CSP, headers HTTP sécurisés (helmet, next-secure-headers)
- Sécurisation des endpoints API et rate limiting

7. Gestion d'état & intégration des outils

- Appel d'API avec SWR ou React Query
- Gestion d'état globale (Zustand, Redux Toolkit, ou Context API optimisé)
- Intégration de CMS headless : Sanity, Strapi, Contentful
- Intégration d'une base de données avec Prisma + PlanetScale ou PostgreSQL

8. Performances & accessibilité

- Audit Lighthouse, Web Vitals et optimisation des scores
- Optimisation des images, lazy loading et CDN
- Optimisation du temps de TTFB avec le rendu serveur
- Accessibilité : bonnes pratiques ARIA, navigation clavier, focus management

9. Déploiement et CI/CD

- Environnements de staging / production sur Vercel
- Configuration des variables sécurisées, preview URLs
- Pipeline GitHub Actions, checks automatiques (lint, tests, build)
- Monitoring avec Vercel Analytics, LogRocket ou Sentry

10. Projet final – Application professionnelle complète

- Développement d'une app complète avec :
- Authentification NextAuth.js
- Pages dynamiques avec Server Components + Server Actions
- CMS intégré (ou base de données Prisma)
- Middleware Edge pour sécurisation
- Déploiement et documentation

NestJS – Niveau Expert

Prix : 2300 € H.T.

Durée : 4 jours

Public cible : Développeurs back-end confirmés, lead dev ou architectes souhaitant concevoir des API robustes, modulaires et scalables avec NestJS.

Prérequis : Très bonne maîtrise de TypeScript, NestJS (modules, contrôleurs, services, DTO, validation), expérience avec base de données relationnelle ou NoSQL.

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Concevoir une architecture modulaire évolutive (DDD, Clean Architecture, CQRS)
- Gérer des systèmes complexes avec les patterns de NestJS avancés
- Implémenter des microservices, de la communication inter-service (MQ, EventBus)
- Sécuriser les API au niveau middleware et infrastructure
- Assurer la qualité et la maintenabilité via tests, logging, CI/CD

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. Architecture avancée

- Structure DDD et Clean Architecture avec NestJS
- Layers : controllers, use-cases, repositories, entities
- Injection dynamique, configuration multi-modules, dépendances croisées
- Feature-based vs domain-based folder structure

2. CQRS avec NestJS

- Introduction au pattern Command Query Responsibility Segregation
- Mise en place avec le module @nestjs/cqrs
- Gestion des handlers, buses, events
- Séparation stricte des lectures/écritures
- Exemple d'implémentation sur une API CRUD complexe

3. Microservices et communication interservices

- NestJS Microservices : architecture, transporteurs supportés (TCP, Redis, MQTT…)
- Communication via Events (event-driven) et Messages (RPC)
- Mise en place avec Redis / NATS
- Scalabilité, résilience, retry et timeout

4. Sécurité avancée

- Authentification OAuth2, JWT, gestion des rôles (RBAC)
- Guards customisés pour accès granulaire
- Protection CSRF, CORS avancé, Helmet
- Logs d'audit, surveillance de requêtes sensibles

5. Gestion de la configuration et des environnements

- Utilisation avancée de @nestjs/config
- Injection de configuration dynamique (multi-environnement, secrets)
- Séparation dev/staging/prod avec Docker ou CI/CD

6. Tests avancés

- Tests unitaires avec Jest : services, controllers, pipes, guards
- Tests d'intégration : setup in-memory DB (SQLite, Mongo Memory Server)
- Mocking avancé avec jest-mock, ts-mockito
- Tests E2E avec Supertest

7. Observabilité et logs

- Interceptors pour tracking et transformation des réponses
- Logging avancé avec @nestjs/logger, Winston ou Pino
- Intégration avec Elastic, Grafana, Prometheus (via exporters)

8. CI/CD et déploiement

- Pipeline CI avec GitHub Actions / GitLab CI : lint, test, build
- Dockerisation d'une app NestJS avec multi-stage build
- Déploiement sur Kubernetes, Fly.io, Render ou AWS (aperçu)
- Health check, readiness, monitoring d'app

9. Projet final – API hexagonale et scalable

- Création d'une API complexe (ex : gestion RH ou e-commerce) :
- Architecture modulaire + CQRS
- Authentification JWT
- Communication asynchrone via Redis ou NATS
- Tests et Dockerisation

C# – Débutant

Prix : 1950 € H.T.

Durée : 3 jours

Public cible : Développeurs débutants ou développeurs venant d'autres langages (Java, JS, Python) souhaitant apprendre C# pour développer des applications .NET.

Prérequis : Connaissances de base en algorithmique ou en programmation dans un autre langage.

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Comprendre les fondements du langage C# et du framework .NET
- Développer des applications console et poser les bases pour le développement web ou desktop
- Maîtriser les types de données, structures de contrôle, collections et fonctions
- Utiliser la programmation orientée objet avec C#
- Lire, écrire et manipuler des fichiers et des données simples

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. Introduction à C# et .NET

- Présentation de la plateforme .NET (Core, Standard, .NET 8)
- Environnement de développement : Visual Studio, Visual Studio Code
- Structure d'un projet C#
- Compilation, exécution, gestion des packages NuGet

2. Bases du langage

- Types de données : types primitifs, chaînes, dates, booléens
- Opérateurs, conversions, nullabilité
- Structures de contrôle : if, switch, while, for, foreach
- Fonctions, arguments nommés et optionnels
- Notions sur les tuples et pattern matching (C# 8+)

3. Programmation orientée objet

- Classes, objets, constructeurs, propriétés
- Méthodes et surcharge
- Encapsulation, héritage, polymorphisme
- Interfaces et classes abstraites
- Utilisation des records (C# 9+) et init-only properties

4. Collections et structures de données

- Tableaux, listes, dictionnaires
- Itération, recherche, tri
- Introduction aux LINQ (Language Integrated Query)
- Méthodes d'extension (extension methods)

5. Gestion des exceptions

- Try, catch, finally
- Création d'exceptions personnalisées
- Bonnes pratiques de gestion d'erreurs

6. Fichiers et manipulation des données

- Lecture/écriture de fichiers texte (File, StreamReader, StreamWriter)
- Sérialisation JSON (System.Text.Json)
- Manipulation de dates, formats et cultures

7. Introduction aux applications console et à l'interactivité

- Interactions avec l'utilisateur (Console.ReadLine / WriteLine)
- Gestion des arguments en ligne de commande
- Création d'outils CLI simples

8. Projet fil rouge

- Développement d'une mini-application de gestion (contacts, tâches, etc.)
- Interface console simple
- CRUD avec enregistrement en fichier JSON
- Application des principes objets et de la structuration du code

ASP.NET Core – Niveau Expert

Prix : 2900 € H.T.

Durée : 4 jours

Public cible : Développeurs back-end expérimentés, lead developers, architectes logiciels souhaitant concevoir des API ou des applications web complexes, sécurisées et performantes avec ASP.NET Core.

Prérequis : Très bonne maîtrise du C# (niveau avancé) et d'ASP.NET Core (contrôleurs, middleware, EF Core, routage, configuration).

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Concevoir des API web et des architectures modulaires robustes (Clean Architecture, DDD)
- Implémenter des stratégies avancées de sécurité, de performance et de mise en production
- Utiliser les dernières pratiques en matière de tests, de logs, de CI/CD et d'observabilité
- Gérer des scénarios complexes de persistance, mapping, validation, et monitoring

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. Architecture logicielle avancée

- Clean Architecture, Domain-Driven Design (DDD) appliqué à ASP.NET Core
- Organisation modulaire : séparation entre Web, Application, Domain, Infrastructure
- Principes SOLID, découplage avec interfaces, injection de dépendances avancée
- Utilisation des MediatR, AutoMapper, FluentValidation

2. Création d'API REST avancées

- Conventions REST : statuts, routes imbriquées, HATEOAS (vue d'ensemble)
- Filtrage, pagination, tri dynamiques
- Versioning d'API (URL, headers, négociation de contenu)
- Contrôleurs génériques et conventions

3. Middleware et pipeline HTTP personnalisé

- Middleware custom : gestion centralisée des erreurs, journalisation, authentification
- Filtres (filters) et attributs personnalisés
- Utilisation fine des IApplicationBuilder, IEndpointRouteBuilder

4. Sécurité et authentification

- Authentification JWT et OAuth2 (Azure AD, IdentityServer, Auth0)
- Autorisation par rôles, policies, claims
- Protection CSRF, CORS avancé, rate limiting (AspNetCoreRateLimit)
- Sécurisation des données sensibles (KeyVault, chiffrement, Identity)

5. Entity Framework Core avancé

- Mapping complexe (relations, types possédés, value objects)
- Optimisation des requêtes LINQ et projections avec Select
- Transactions, verrouillages, stratégie de concurrence
- Multi-tenancy, audit, stratégie soft delete

6. Tests, logs et observabilité

- Tests unitaires / d'intégration avec xUnit, WebApplicationFactory, EF InMemory

- Mocks complexes avec Moq, FakeItEasy, Bogus
- Logging structuré avec Serilog et enrichissement (correlation ID, user context…)
- Traces et métriques avec OpenTelemetry, Grafana, Application Insights

7. Performances, résilience et mise en production

- Caching : mémoire, distribué (Redis), HTTP headers (ETag, Last-Modified)
- Retry, circuit breaker avec Polly
- Compression, minification, gestion fine du pipeline
- Publication sur Azure, Docker, Kubernetes ou Render

8. CI/CD et DevOps

- Pipelines GitHub Actions / GitLab CI : lint, tests, build, déploiement
- Dockerisation d'une API ASP.NET Core avec configuration multi-env
- Intégration avec Helm, Kustomize, Terraform (notions)
- Stratégies de migration de base et déploiement blue/green

9. Projet final

- Conception complète d'une API métier modulaire (ex : système RH, e-commerce, CRM)
- Architecture hexagonale avec MediatR
- JWT, logins sécurisés, validation centralisée
- Logs structurés, tests automatisés, déploiement Docker

Codage sécurisé en C/C++

Prix : 2100 € H.T.

Durée : 3 jours

Public cible : Développeurs C et/ou C++ intervenant sur des applications critiques (embarqué, système, temps réel, industriel, finance, cybersécurité), responsables qualité, architectes logiciels.

Prérequis : Maîtrise des fondamentaux du langage C ou C++ (pointeurs, structures, fonctions, gestion mémoire, compilation). Une expérience de développement concret est fortement recommandée.

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Identifier les principales vulnérabilités dans le code C/C++
- Appliquer les bonnes pratiques de programmation défensive
- Rédiger un code robuste, fiable et sécurisé dès la conception
- Utiliser des outils d'analyse statique et dynamique pour prévenir les failles
- Se conformer aux référentiels de sécurité reconnus (CERT, MISRA, ISO/IEC TS 17961)

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. Introduction à la sécurité logicielle

- Enjeux de sécurité dans le développement natif
- Historique et vulnérabilités emblématiques (Heartbleed, Stack Clash…)
- Panorama des normes : CERT C, CERT C++, MISRA C, ISO/IEC TS
- Attaques classiques : buffer overflow, use-after-free, integer overflow, format string

2. Failles courantes en C/C++

- Problèmes liés à la gestion mémoire manuelle :
- Déréférencement de pointeurs invalides
- Double free, fuites mémoire, free non initialisé
- Failles d'initialisation et d'accès concurrent
- Dangereuses fonctions standard (strcpy, sprintf, gets…)

3. Bonnes pratiques de codage sécurisé

- Validation stricte des entrées/sorties (whitelisting, bornes)
- Gestion défensive des pointeurs et allocations dynamiques
- Utilisation correcte des types (size_t, uint32_t, conversions sûres)
- Programmation robuste : assertions, invariants, defensive programming

4. Codage sécurisé en C++

- Exceptions vs codes d'erreur, RAII et gestion des ressources
- Bonnes pratiques avec les smart pointers (unique_ptr, shared_ptr)
- Conteneurs STL sécurisés vs accès mémoire direct
- Gestion de l'héritage, override explicite, règles d'encapsulation

5. Recommandations CERT C/C++

- Principes de la norme CERT (rules vs recommendations)
- Exemples de règles : INT32-C, EXP33-C, ARR30-C, MEM50-CPP
- Cas pratiques : analyse de code et refactoring à partir d'extraits non conformes

6. Outils d’analyse statique et dynamique

- Analyse statique : Clang-Tidy, Cppcheck, SonarQube, Coverity
- Analyse dynamique : Valgrind, AddressSanitizer, UndefinedBehaviorSanitizer
- Intégration dans les chaînes CI/CD (GitHub Actions, GitLab CI)
- Métriques de sécurité et rapports de conformité

7. Sécurisation du cycle de développement

- Intégration des contrôles sécurité dans le SDLC (Secure Development Life Cycle)
- Revue de code orientée sécurité
- Tests fuzzing (libFuzzer, AFL), injection de fautes
- Techniques de durcissement du binaire : ASLR, DEP, stack canaries, PIE

8. Études de cas et atelier final

- Analyse de failles réelles dans des bibliothèques open source
- Réécriture sécurisée d’un module critique (lecture/écriture fichier, socket, parsing)
- Application d’outils d’analyse et rédaction de rapports de non-conformité

Symfony – Niveau Débutant à Intermédiaire

Prix : 2600 € H.T.

Durée : 4 jours

Public cible : Développeurs PHP souhaitant apprendre à concevoir des applications web robustes et maintenables avec le framework Symfony.

Prérequis : Bonne maîtrise du langage PHP orienté objet. Expérience de développement d'applications web (formulaires, bases de données) et notions de Composer, HTTP et MVC

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Comprendre l'architecture d'un projet Symfony
- Créer une application web en s'appuyant sur les composants Symfony
- Gérer les routes, les contrôleurs, les entités, les formulaires et la base de données avec Doctrine
- Appliquer les bonnes pratiques de sécurité et de structuration de projet
- Utiliser les outils intégrés de débogage, tests et configuration

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. Introduction à Symfony

- Présentation du framework Symfony et de l'écosystème PHP-FIG (PSR)
- Architecture MVC, composants réutilisables
- Installation avec Composer et Symfony CLI
- Structure d'un projet Symfony (src, config, templates, public, etc.)

2. Routing et contrôleurs

- Définir une route avec annotations, YAML ou PHP
- Création de contrôleurs (Controller, Response, Request)
- Génération d'URLs, redirections
- Passage de paramètres dans les routes

3. Twig – Le moteur de templates

- Syntaxe de base de Twig
- Héritage de templates et blocs
- Filtres, fonctions et extensions
- Affichage conditionnel et boucles

4. Entités et base de données avec Doctrine ORM

- Configuration de la base de données (.env, Doctrine config)
- Création d'entités et mapping via annotations ou attributs PHP 8
- Migrations, relations (OneToMany, ManyToMany)
- Requêtes via repository et QueryBuilder

5. Formulaires Symfony

- Création et affichage de formulaires avec FormBuilder
- Champs de formulaires (text, textarea, choice, checkbox, file, etc.)
- Validation (constraints) et messages d'erreurs
- Traitement des données et persistance

6. Services, injections et container

- Création de services personnalisés
- Injection automatique (autowiring) et configuration via YAML
- Notion de container de services
- Tags, événements, listeners

7. Sécurité

- Gestion de l'authentification : login, logout, firewall, encodage des mots de passe
- Sécurité des routes (access control, rôle utilisateur)
- Création d'un système d'inscription et de connexion avec SecurityBundle
- CSRF, validation et protection contre les attaques courantes

8. Console, debug et outils

- Commandes utiles (bin/console, debug:*)
- Utilisation du Profiler Symfony (Barre de debug)
- Dumping et logging avec Monolog

9. Tests fonctionnels et unitaires (initiation)

- Mise en place de tests avec PHPUnit
- Tests des contrôleurs et du comportement des routes
- Simulation de requêtes HTTP

10. Mini-projet final

- Mise en œuvre d'une application complète (ex : blog, gestion de tâches, catalogue produits)
- Mise en pratique du routage, des entités, des formulaires et de la sécurité

TypeScript – Prise en main et développement robuste en JavaScript typé

Prix : 1900 € H.T.

Durée : 3 jours

Public cible : Développeurs JavaScript souhaitant structurer et sécuriser leur code. Développeurs front-end ou back-end utilisant des frameworks modernes (React, Angular, Node.js…) Équipes techniques adoptant TypeScript pour l'industrialisation de leurs projets.

Prérequis : Maîtrise des fondamentaux JavaScript (ES6+ : classes, modules, fonctions fléchées, promesses). Connaissances de base en développement web (HTML/CSS facultatif)

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Comprendre les principes du typage statique appliqué à JavaScript
- Configurer et compiler un projet TypeScript efficacement
- Créer des types personnalisés et utiliser les types génériques
- Migrer progressivement un projet JavaScript vers TypeScript
- Intégrer TypeScript avec des bibliothèques/frameworks courants

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. Introduction à TypeScript

- Pourquoi TypeScript ? Typage statique, IntelliSense, refactoring sécurisé
- Fonctionnement du compilateur tsc
- Configuration de projet (tsconfig.json)
- Compilation, surveillance (–watch), intégration avec NPM

2. Types de base

- Types primitifs : string, number, boolean, null, undefined, any, unknown
- Tableaux, tuples, objets typés
- Types union (|), intersection (&)
- Enumérations (enum)
- Typage de fonctions (arguments, retour, callbacks)

3. Typage avancé et structures complexes

- Interfaces et types (interface vs type)
- Types optionnels, readonly, et indexés
- Fonctions génériques (<T>)
- Types utilitaires (Partial, Pick, Omit, Record, etc.)
- Narrowing, assertion de types, contrôle de flux
- Type guards personnalisés

4. Classes et programmation orientée objet

- Classes, constructeurs, modificateurs d'accès (public, private, protected)
- Propriétés et méthodes statiques
- Héritage, interfaces d'implémentation
- Décorateurs (bases, pour Angular ou NestJS)

5. Modules et gestion de projet

- Modules ES vs CommonJS

- Import/export de types et interfaces
- Organisation modulaire d’un projet TypeScript
- Compilation vers différentes cibles (ES2022, ES5, etc.)

6. Interopérabilité avec JavaScript

- Typage de bibliothèques JS via DefinitelyTyped (@types)
- Utilisation de bibliothèques non typées
- Migration progressive d’un projet JavaScript
- Ajout de déclarations globales (fichiers .d.ts)

7. Outils et bonnes pratiques

- ESLint + TypeScript : règles de qualité
- Intégration avec VSCode : IntelliSense, debug, lint
- Build & CI : intégration avec Webpack/Vite, tests (Jest + TS), Git hooks
- Bonnes pratiques de typage, patterns recommandés et erreurs à éviter

COBOL Initiation

Prix : 2100 € H.T.

Durée : 3 jours

Public cible : Cette formation s'adresse aux informaticiens qui souhaitent acquérir une formation opérationnelle pour développer ou maintenir des applications en Cobol.

Prérequis : Connaissances de base en programmation.

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Présenter les intérêts de Mainframe par rapports aux autres infrastructures
- Maîtriser la syntaxe globale du langage COBOL
- Clarifier la structure d'un programme Cobol
- Définir les ressources d'un programme Cobol en données dans un environnement Batch

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. Présentation de l'interface ISPF

- Accès à la machine IBM
- Historique des mainframes
- Architecture générale des mainframes
- Terminologie IBM
- Présentation des choix du menu principal (Edit, View et SDSF)
- View : affichage du contenu de la librairie
- Edit : Création et mise à jour du contenu de la librairie
- SDSF: Spool Search and display Facility
- Commande de base de l'éditeur ISPF.
- Les commandes lignes de l'éditeur

2. Les commandes JCL

- La carte de commande JOB
- La structure des champs de JOB Statements
- La carte de commande EXEC
- La structure des champs de EXEC Statements

3. Présentation du langage Cobol

- Présentation de la structure du langage COBOL
- Description technique de COBOL
- Composition d'un programme COBOL
- Règle de codage sur une ligne COBOL
- Règle d'utilisation des mots clés et des chaînes de caractères
- Déclarations des variables

4. Opérations sur les données (MOVE, COMPUTE, ADD, etc...)

- Traitements des variables MOVE
- Opération de calcul ADD
- Opération de calcul SUBSTRUCT
- Opération de calcul MULTIPLY
- Opération de calcul DIVIDE

- Opération de calcul COMPUTE

5. Procédure de Compilation d'un programme

- Examen des phases de compilation
- Les Librairies de stockage

6. Traitement conditionnel

- Instruction IF .... THEN .... ELSE
- Instruction EVALUATE
- Instruction PERFORM
- Instruction GO ... TO

7. Gestion des Tables (OCCURS)

- Définition d'une Table
- Déclaration d'une Table
- Indice d'adressage d'une Table
- Indexation pour adressage des Tables
- Mise à jour des Index par l'instruction SET

CICS – Développement d'Applications Transactionnelles sur Mainframe

Prix : 2300 € H.T.

Durée : 3 jours

Public cible : Développeurs COBOL, PL/I ou Assembler amenés à développer ou maintenir des programmes CICS, Analystes techniques travaillant sur des applications transactionnelles, administrateurs systèmes ou responsables de production souhaitant mieux comprendre l'architecture CICS. Toute personne intervenant dans un environnement IBM z/OS avec des applications en ligne

Prérequis : Connaissance de la programmation COBOL ou PL/I. Notions de base sur le fonctionnement d'un mainframe et des fichiers VSAM. Expérience préalable avec JCL (souhaitable)

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Comprendre l'architecture CICS et le rôle du système transactionnel
- Développer des programmes CICS en COBOL ou PL/I
- Utiliser les commandes CICS pour la gestion des écrans, des fichiers, et des bases de données
- Gérer les communications entre programmes et utilisateurs via les terminaux
- Intégrer CICS avec des bases de données DB2 ou des programmes batch

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. Introduction à CICS

- Qu'est-ce que CICS ? Rôle et position dans une architecture mainframe
- Différences entre traitement batch et transactionnel
- Structure d'un environnement CICS : régions, tables, programmes, terminaux
- Concepts de transaction, tâche (task), unité de travail

2. Architecture et composants CICS

- Définition des ressources : PPT (Program Processing Table), FCT (File Control Table), TCT (Terminal Control Table)
- Gestion des terminaux et sessions utilisateurs
- Cycle de vie d'une transaction CICS
- Notion de pseudo-conversation

3. Programmation CICS avec COBOL

- Structure d'un programme COBOL CICS
- Utilisation des commandes EXEC CICS (RECEIVE, SEND, LINK, XCTL, RETURN, etc.)
- Échange de données avec les utilisateurs via des écrans BMS (Basic Mapping Support)
- Gestion des commutateurs de programmes et des transitions

4. Gestion des fichiers et des bases de données

- Accès aux fichiers VSAM sous CICS avec les commandes FILE CONTROL
- Gestion de la concurrence, verrouillage, et intégrité des données
- Intégration avec DB2 sous CICS (commandes SQL dans un contexte transactionnel)
- Transactions multi-ressources (Commit, Rollback, Syncpoint)

5. Écrans et interface utilisateur

- Introduction à BMS (Basic Mapping Support)
- Définition et création de maps (MAPSET, MAP)
- Envoi et réception de données via écran 3270

- Gestion des erreurs d'écran, champs protégés, curseur, attributs dynamiques

6. Contrôle de flux et appels de programmes

- Appels de programmes (CALL, XCTL, LINK) et passage de paramètres
- Chaînage de transactions et pseudo-conversations
- Gestion des commutateurs d'écran et navigation utilisateur
- Programmation structurée avec des modules appelés et retour contrôlé

7. Gestion des erreurs et sécurité

- Gestion des codes de retour, gestion des exceptions CICS
- Utilisation de HANDLE CONDITION et IGNORE CONDITION
- Sécurité CICS : contrôle d'accès, autorisations, interactions avec RACF
- Journaux et diagnostics en cas d'erreur

8. Outils de développement et supervision

- Utilisation de CEDA pour la définition des ressources
- Utilisation de CEMT pour la gestion dynamique des ressources (activation, fermeture)
- Supervision des transactions en cours, consommation CPU, et files d'attente
- Bonnes pratiques de développement et de test en environnement CICS

Jakarta EE – Niveau Avancé : Développement d'Applications et Microservices d'Entreprise

Prix : 2500 € H.T.

Durée : 4 jours

Public cible : Développeurs Java expérimentés souhaitant concevoir des applications Jakarta EE modernes et modulaires. Ingénieurs logiciels impliqués dans des projets de migration vers des architectures microservices. Architectes logiciels, DevOps, chefs de projets techniques sur des plateformes Java EE. Équipes techniques souhaitant conteneuriser ou orchestrer leurs applications EE.

Prérequis : Solide expérience avec Java SE et les composants principaux de Jakarta EE (Servlets, EJB, JPA, REST). Bonne connaissance de l'architecture n-tiers et des modèles de conception Java. Expérience souhaitée avec Maven ou Gradle, Git, et un serveur d'application Jakarta EE.

Objectifs pédagogiques

À l'issue de la formation, les participants seront capables de :

- Concevoir des architectures modulaires à base de microservices avec Jakarta EE
- Utiliser Jakarta REST, CDI, JPA et JTA dans un contexte décentralisé
- Sécuriser et orchestrer les services avec JWT, OAuth2, OpenAPI, etc.
- Conteneuriser et déployer leurs applications Jakarta EE dans Docker et Kubernetes
- Intégrer Jakarta EE avec des systèmes de messaging, d'authentification et des outils de supervision

Méthodes et moyens pédagogiques : Travaux pratiques : formation alternant théorie et pratique.

Modalités d'évaluation

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Programme

1. Rappels essentiels Jakarta EE

- Architecture Jakarta EE et standardisation du cloud-native
- Application Monolithique vs Microservices
- Révision rapide des composants clés : EJB, JPA, RESTful, CDI, JTA

2. Conception de microservices avec Jakarta EE

- Découpage fonctionnel et granularité des services
- Patterns : API Gateway, Service Registry, Circuit Breaker (Hystrix-like)
- Communication REST/JSON entre services avec JAX-RS
- Appels interservices via HTTP et gestion de latence

3. CDI et architecture modulaire

- Injection de dépendances avancée avec Jakarta CDI
- Events, Interceptors, Producers
- Portabilité du code métier dans un environnement de services découplés
- Modularisation avec Jigsaw (Java 9+) ou via Maven multi-modules

4. Persistance et transactions distribuées

- Gestion de la persistance dans un environnement microservices
- Réplication vs base de données centralisée
- JPA dans des contextes décentralisés
- Transactions distribuées avec JTA ou alternatives (Sagas)

5. Sécurité et authentification

- Implémentation d'OAuth2 / JWT avec Jakarta Security
- Sécurisation des API REST avec JAX-RS et filtres personnalisés
- Mise en œuvre de rôles, autorisations, et contexte utilisateur

- OpenID Connect (vue d’ensemble)

6. Documentation, supervision, résilience

- Génération de documentation avec OpenAPI / Swagger
- Health checks avec MicroProfile Health
- Configuration centralisée avec MicroProfile Config
- Metrics, tracing, et log centralisé avec Prometheus, Grafana, ELK

7. Conteneurisation et orchestration

- Création d’images Docker pour Jakarta EE (Payara, WildFly, TomEE)
- Déploiement d’un cluster de services sur Kubernetes ou OpenShift
- Gestion de la configuration via secrets, volumes, config maps
- Bonnes pratiques CI/CD avec Jenkins, GitHub Actions ou GitLab

8. Interopérabilité et migration

- Intégration avec des services Spring Boot, Quarkus, Node.js via REST ou Kafka
- Migration progressive d’un monolithe Java EE vers une architecture microservices
- Séparation des responsabilités et évolution continue
- Utilisation de Jakarta EE Core Profile et Lite Profile (Jakarta EE 10+)

Le Sur-Mesure

Avec notre équipe pédagogique, nos formateurs experts et nos consultants dédiés, nous co-construisons des formations personnalisées pour développer l'engagement et la performance de vos collaborateurs.

La conception de formations sur mesure

Votre consultant Skillia organise un entretien de cadrage avec l'expert-formateur pressenti afin de définir précisément les objectifs de la formation et d'évaluer les besoins de préparation. Spécialiste du domaine concerné, l'expert analyse les spécificités du projet, le profil des participants et élabore, avec vous, un cahier des charges détaillé garantissant la pertinence du dispositif.

Des solutions e-learning personnalisées

Vous souhaitez une formation 100 % sur mesure, composée de modules e-learning et de vidéos adaptées à votre métier ?

Skillia met à votre disposition son savoir-faire et ses outils de production pour concevoir des formations digitales performantes, alliant interactivité, efficacité et accompagnement personnalisé par des tuteurs experts.

La gestion de projets de grande envergure

Qu'il s'agisse de former plusieurs équipes, sur différents sites ou dans plusieurs langues (français, anglais…), Skillia dispose de l'expertise nécessaire pour piloter l'ensemble du projet : audit, conception de programmes, construction de parcours, ainsi que coordination technique et logistique.

SKILLIA

Le spécialiste de la formation professionnelle