

PROGRAMME DE FORMATION

RAG SOUVERAIN — CONSTRUIRE UNE IA SUR SES PROPRES DONNÉES

FICHE SYNTHÉTIQUE DE LA FORMATION

Intitulé	RAG souverain — Construire une IA sur ses propres données
Référence	IA-RAG-SOUV-2J
Durée	2 jours (14 heures)
Modalités	Présentiel ou distanciel synchrone (classe virtuelle)
Tarif	1 400 € HT / participant (inter) — intra sur devis
Délais d'accès	Inscription possible jusqu'à 14 jours avant le début de la session
Lieu	Paris 17e ou distanciel via Zoom
Effectif	4 participants maximum par session
Contact	contact@pythonia.fr – 06 66 06 38 59
Accessibilité PSH	Formation accessible aux PSH. Contactez notre référent handicap pour toute adaptation.
Dernière mise à jour	Octobre 2026

1. Public visé

Cette formation s'adresse aux développeurs, data scientists, data engineers, architectes et équipes DSI qui doivent mettre à disposition de leurs utilisateurs une IA capable de répondre à partir des documents et bases de connaissances internes, sans exposer ces données à un fournisseur externe : banques, assurances, cabinets, industrie, secteur public.

2. Prérequis

- Maîtriser les bases de Python (fonctions, listes, dictionnaires, modules, environnements virtuels)
- Avoir déjà utilisé un assistant IA générative et connaître la notion de prompt
- Avoir suivi « Installer et exploiter un LLM en local » ou disposer de connaissances équivalentes (recommandé)
- Poste avec 16 Go de RAM minimum, 30 Go d'espace disque libre, droits administrateur et Docker installé

Un test de positionnement est proposé avant l'entrée en formation afin de vérifier l'adéquation du parcours avec le profil du participant et la compatibilité de son poste.

3. Objectifs pédagogiques

À l'issue de cette formation, le stagiaire sera capable de :

- Expliquer le fonctionnement d'un système RAG et ses composants : ingestion, découpage, embeddings, index, retrieval, génération
- Construire une chaîne d'ingestion robuste pour des documents internes hétérogènes (PDF, Word, pages web, bases SQL)
- Mettre en œuvre une recherche sémantique et hybride performante avec une base vectorielle hébergée en interne
- Évaluer objectivement la qualité d'un RAG (pertinence, fidélité, citations) et l'améliorer
- Sécuriser le système : cloisonnement des droits, protection contre les injections de prompt, journalisation, conformité RGPD

4. Programme détaillé

Fil rouge : les participants construisent, sur deux jours, un assistant documentaire complet fonctionnant sur une infrastructure maîtrisée, depuis l'ingestion des documents jusqu'au déploiement d'une API sécurisée.

JOUR 1 — Fondations : ingestion, embeddings et recherche (7h)

Matin (3h30) — Architecture d'un RAG souverain

Pourquoi le RAG ?

- Limites des LLM : connaissances figées, hallucinations, absence de sources
- RAG vs fine-tuning vs prompt long : quand choisir quoi
- Souveraineté : modèles open-weight, hébergement interne ou cloud européen, RGPD et AI Act

Architecture de référence

- Chaîne complète : ingestion → découpage → embeddings → index → retrieval → génération
- Briques open source : Ollama, sentence-transformers, Chroma, Qdrant, pgvector, LlamaIndex, LangChain
- Dimensionnement : volumétrie, latence, coûts matériels

Ingestion de documents

- Extraction de texte : PDF (y compris tableaux), Word, HTML, e-mails
- Nettoyage, métadonnées (source, date, auteur, niveau de confidentialité)
- Stratégies de découpage (chunking) : taille fixe, récursive, sémantique, par structure

Travaux pratiques :

→ TP1 : pipeline d'ingestion d'un corpus de documents internes avec extraction des métadonnées

Après-midi (3h30) — Embeddings et recherche sémantique

Embeddings

- Principe des représentations vectorielles et de la similarité cosinus
- Choisir un modèle d'embeddings multilingue local (bge-m3, multilingual-e5, nomic-embed)
- Calcul en lot, normalisation et mise à jour incrémentale

Bases vectorielles

- Chroma, Qdrant, pgvector : comparaison et critères de choix
- Index, filtres sur métadonnées, persistance et sauvegarde

Améliorer la recherche

- Recherche hybride : sémantique + mots-clés (BM25)
- Re-ranking avec un cross-encoder local
- Reformulation de requêtes et requêtes multiples

Travaux pratiques :

→ TP2 : indexer le corpus dans Qdrant et comparer recherche sémantique, hybride et avec re-ranking

✓ **Évaluation formative : quiz de 10 questions et revue collective des TP en fin de journée**

JOUR 2 — Génération, évaluation, sécurité et déploiement (7h)

Matin (3h30) — Génération et évaluation

Génération augmentée

- Construction du prompt : contexte, consignes, format de réponse
- Citations des sources et refus de répondre hors corpus
- Gestion de la fenêtre de contexte et des documents longs
- Conversation multi-tours et mémoire

Évaluer un RAG

- Constituer un jeu de questions/réponses de référence
- Métriques : précision du retrieval, fidélité, pertinence de la réponse (approche RAGAS)
- LLM-as-a-judge local et limites de l'évaluation automatique
- Boucle d'amélioration : diagnostic des erreurs et réglages

Travaux pratiques :

- TP3 : connecter le retrieval à un LLM local avec citations
- TP4 : évaluer l'assistant sur 30 questions de référence et améliorer ses scores

Après-midi (3h30) — Sécurité et mise en production

Sécuriser le RAG

- Contrôle d'accès documentaire : filtrage par droits de l'utilisateur
- Injections de prompt indirectes via les documents, fuites de données
- Anonymisation et données personnelles (RGPD), durée de conservation
- Journalisation, traçabilité des réponses et supervision

Déploiement

- Exposer le RAG via une API (FastAPI) et une interface (Open WebUI ou Streamlit)
- Conteneurisation avec Docker, mise à jour de l'index
- Feuille de route : du prototype au service interne

Travaux pratiques :

- TP5 : ajouter le filtrage par droits et tester la résistance aux injections
- TP6 : déployer l'assistant sous forme d'API conteneurisée

✓ **Évaluation sommative : QCM final de 20 questions et démonstration de l'assistant — critère de réussite : 60 %**

5. Méthodes et moyens pédagogiques

Méthodes mobilisées

- Apports théoriques : exposés interactifs avec diaporama et démonstrations en direct
- Mise en pratique : TP progressifs construisant un assistant complet sur deux jours
- Pédagogie active : cas d'usage apportés par les participants
- Alternance théorie/pratique : ratio 30 % théorie / 70 % pratique

Moyens techniques

- Salle de formation équipée : vidéoprojecteur, paperboard, connexion Wi-Fi haut débit
- Poste informatique par stagiaire (ou poste personnel répondant aux prérequis)
- En distanciel : Zoom avec partage d'écran, tableau blanc collaboratif
- Supports de cours, notebooks et code source complet remis aux stagiaires
- Corpus de documents d'exemple et jeu de questions de référence fournis

6. Modalités d'évaluation

Positionnement à l'entrée

Avant l'entrée en formation, chaque participant complète un test de positionnement évaluant son niveau en Python, ses connaissances en IA générative et la configuration de son poste.

Évaluations en cours de formation

- Évaluation formative : quiz de 10 questions en fin de jour 1
- Évaluation sommative : QCM final de 20 questions et démonstration de l'assistant en fin de jour 2
- Critère de réussite : obtenir 60 % au QCM final
- Attestation de fin de formation : remise à l'issue, mentionnant objectifs, durée et résultats

Satisfaction

- Questionnaire de satisfaction en fin de formation (formateur, contenu, logistique, atteinte des objectifs)

7. Accessibilité aux personnes en situation de handicap

Pythonia s'engage à rendre ses formations accessibles aux personnes en situation de handicap. Notre référent handicap est disponible pour étudier toute demande d'adaptation :

- Adaptation des supports pédagogiques (taille de police, contraste, format audio)
- Aménagement des conditions de passage (temps majoré, pauses adaptées)
- Orientation vers nos partenaires spécialisés : Agefiph, Cap Emploi, MDPH

Contact référent handicap : contact@pythonia.fr – 06 66 06 38 59

8. Engagement et suivi

- Suivi de l'assiduité : feuille d'émargement signée par demi-journée
- Prévention des abandons : suivi individualisé, adaptation du rythme, relance en cas d'absence
- Réclamations : toute réclamation peut être adressée par email à contact@pythonia.fr. Elle sera traitée sous 48h ouvrées.

PythonIA — Centre de Formation Python & Intelligence Artificielle

255 boulevard Pereire, 75017 Paris | contact@pythonia.fr | 06 66 06 38 59