AKADEMIA
FORMATION

PROGRAMME DE FORMATION

Construire avec l'API Claude — De l'Initiation à la Maîtrise

DURÉE	FORMAT	PUBLIC	CERTIFICATION
3 jours	**Inter · Intra**	**Tout collaborateur**	**Incluse**
21 heures	Présentiel ou distanciel		

À PROPOS DE LA FORMATION

Construire avec l'API Claude — De l'Initiation à la Maîtrise.

DURÉE	FORMAT	PÉDAGOGIE	CERTIFICATION
3 jours	**Inter · Intra**	**Active**	**Incluse**
21 heures	Présentiel ou distanciel	Petits groupes	Attestation délivrée

Objectifs pédagogiques

- Maîtriser l'API Messages Claude : authentification, paramétrage, streaming et conversations multi-tours
- Implémenter des techniques de prompt engineering avancées via l'API : system prompts, few-shot, chain-of-thought et structured outputs
- Développer des applications avec Tool Use et Function Calling : définition d'outils, schémas JSON, orchestration multi-tool et boucles agentiques
- Exploiter les fonctionnalités avancées de l'API : vision (images/PDF), citations, prompt caching, Batch API et code execution
- Concevoir des architectures multi-agents avec l'Agents SDK : guardrails, handoffs, tracing et patterns de production
- Déployer et monitorer des applications Claude en production : gestion des rate limits, error handling, sécurité des clés et intégration Bedrock/Vertex
- Optimiser les coûts et performances : prompt caching (-90%), batches (-50%), choix de modèle et évaluation systématique

PROGRAMME

12 modules progressifs pour monter en compétences.

MODULE

01.

1H45H

Fondamentaux de l'API Claude : architecture, authentification et premiers appels

CONTENU PÉDAGOGIQUE

- Architecture de l'API REST Anthropic : endpoints Messages, Batches, Token Counting, Models, Files, Agents et Sessions
- Création et sécurisation des clés API : Console Anthropic, workspaces, variables d'environnement, bonnes pratiques
- Installation des SDK officiels Python et TypeScript : configuration du client, authentification automatique
- Première requête API complète en Python, TypeScript et cURL : requête, réponse JSON, usage tokens, stop_reason
- Panorama des modèles disponibles : Opus 4.6, Sonnet 4.6, Haiku 4.5, Mythos Preview — positionnement, coûts et cas d'usage

MODULE

02.

1H45H

Messages API en profondeur : paramétrage, multi-turn et structured outputs

CONTENU PÉDAGOGIQUE

- Format de requête : paramètres obligatoires (model, max_tokens, messages) et types de content blocks (text, image, document, tool_use, tool_result)
- Conversations multi-tours stateless : gestion de l'historique côté client, messages assistant synthétiques, prefilling
- Paramètres de sampling : temperature, top_p, top_k, stop_sequences, metadata user_id et limites de taille (32 MB)
- Structured Outputs avec output_config : json_schema, constrained decoding, sortie JSON garantie et type-safe
- Atelier pratique : construire un chatbot multi-tour avec mémoire conversationnelle et extraction structurée de données

MODULE

03.

1H35H

Prompt engineering avancé via l'API : system prompts, few-shot et chain-of-thought

CONTENU PÉDAGOGIQUE

- System prompts : paramètre top-level system, définition du rôle et du ton, format texte et tableau de TextBlockParam
- Balises XML pour structurer les prompts complexes : documents multiples, hiérarchie, ancrage dans les citations
- Few-shot prompting via l'API : 3 à 5 exemples dans des balises , diversité, cas limites et évaluation automatique
- Chain-of-thought et thinking adaptatif (effort low/medium/high/max) : calibration automatique sur les modèles 4.6
- Role prompting, long contexte (200K tokens) et contrôle du format de sortie : XML, prose, markdown contrôlé

MODULE

04.

1H55H

Streaming SSE et Extended Thinking

CONTENU PÉDAGOGIQUE

- Streaming Server-Sent Events : paramètre stream:true, implémentation Python (messages.stream) et TypeScript
- Flux d'événements SSE : message_start, content_block_start, content_block_delta (text_delta, input_json_delta, thinking_delta), message_stop
- Obtention du message final sans gestion des événements : get_final_message pour les grandes générations
- Extended Thinking : mode adaptatif (recommandé 4.6+) vs manual (budget_tokens), display summarized/omitted
- Streaming de la réflexion : événements thinking_delta et signature_delta, latence réduite avec display omitted
- Atelier pratique : implémenter un streaming temps réel avec affichage progressif et comparer l'impact du thinking sur la qualité

MODULE

05.

1H35H

Tool Use et Function Calling : fondamentaux et schémas

CONTENU PÉDAGOGIQUE

- Architecture Tool Use : client tools vs server tools (web_search, code_execution, web_fetch), cycle requête-réponse
- Définition d'outils : name (regex ^[a-zA-Z0-9_-]{1,64}$), description détaillée (3-4 phrases min.), input_schema JSON Schema, input_examples
- Contrôle de l'utilisation avec tool_choice : auto, any, tool (forcer un outil), none — restrictions avec Extended Thinking
- Strict Tool Use : strict:true pour garantir la conformité exacte au schéma, combinaison avec tool_choice any
- Pricing des outils : tokens supplémentaires (346 tokens Opus 4.6 en auto), overhead system prompt automatique

MODULE

06.

1H55H

Orchestration multi-tool et boucles agentiques

CONTENU PÉDAGOGIQUE

- Boucle agentique complète : message → tool_use (stop_reason) → exécution → tool_result → analyse → itération
- Multi-tools et appels parallèles : prompt engineering avec , optimisation des latences
- Server tools intégrés : web_search_20260209, code_execution_20250825/20260120, web_fetch_20260209 — configuration et exemples
- Extended Thinking + Tool Use : restrictions (auto/none uniquement), passage obligatoire des blocs thinking dans la boucle
- Bonnes pratiques : descriptions détaillées, namespacing (github_list_prs), consolidation d'outils, réponses à haute valeur
- Atelier pratique : construire un agent de recherche multi-sources avec 3+ outils, boucle d'orchestration et gestion d'erreurs

MODULE

07.

1H37H

Vision, PDF et Citations documentaires

CONTENU PÉDAGOGIQUE

- Vision API : envoi d'images en base64, URL ou Files API — formats JPEG, PNG, GIF, WebP, limite 600 images/requête
- Calcul du coût images : formule (largeur x hauteur) / 750 tokens, exemples de coûts par taille
- PDF Support : traitement dual image+texte par page, 3 méthodes d'envoi, 1500-3000 tokens/page, limite 32 MB et 600 pages
- Citations : activation avec citations:{enabled:true}, 3 types (char_location, page_location, content_block_location)
- Avantages des citations : cited_text gratuit (pas de tokens output), qualité supérieure au prompting, streaming citations_delta
- Search Result Blocks pour RAG natif : format search_result_block, citations automatiques comme web search

MODULE

08.

1H53H

Prompt Caching, Batch API, Files API et Code Execution

CONTENU PÉDAGOGIQUE

- Prompt Caching : cache_control ephemeral, TTL 5min (1.25x écriture, 0.1x lecture) et 1h (2x écriture, 0.1x lecture)
- Stratégies de caching : automatique multi-turn, breakpoints explicites (max 4), cache sur outils, fenêtre lookback 20 blocs
- Monitoring du cache : cache_read_input_tokens et cache_creation_input_tokens, règles d'invalidation par changement
- Batch API : traitement asynchrone -50% sur tous les prix, 100K requêtes max ou 256 MB, résultats disponibles 29 jours
- Files API : upload unique et réutilisation via file_id, 500 MB/fichier, 500 GB/organisation, opérations gratuites
- Code Execution : sandbox Python/Bash, versions 20250825 (base) et 20260120 (REPL persistant), gratuit avec web search/fetch
- Atelier pratique : optimiser les coûts d'un pipeline — caching multi-turn + batch processing de 1000 évaluations

MODULE

09.

1H45H

Agents SDK et patterns multi-agents

CONTENU PÉDAGOGIQUE

- Architecture de l'Agents SDK : Agents API (configurations réutilisables), Sessions API (stateful en containers managés), Environments API
- Patterns multi-agents : orchestrateur-workers, chaîne séquentielle, parallélisation, spécialisation par domaine
- Handoffs entre agents : transfert de contexte, routage conditionnel, escalade et fallback automatique
- Prompt chaining et auto-correction : générer → évaluer contre des critères → affiner, avec inspection des sorties intermédiaires
- Atelier pratique : concevoir un système multi-agents (trieur → analyseur → rédacteur) avec handoffs et tracing

MODULE

10.

1H45H

Guardrails, tracing et sécurité des agents

CONTENU PÉDAGOGIQUE

- Guardrails de sécurité : validation des inputs/outputs, filtrage de contenu, limites de boucle, budgets de tokens
- Tracing et observabilité : logs structurés des appels d'outils, métriques de performance, debugging des boucles agentiques
- Prompt injection et défenses : séparation données/instructions, input sanitization, metadata user_id pour détection d'abus
- Gestion des erreurs robuste : retries exponentiels (SDK intégrés), circuit breakers, timeouts, fallback entre modèles
- Atelier pratique : sécuriser un agent existant avec guardrails, tracing complet et stratégie de gestion d'erreurs

MODULE

11.

1H45H

Production et Scale : rate limits, monitoring et cloud providers

CONTENU PÉDAGOGIQUE

- Rate limits et tiers d'utilisation : RPM, TPM, spend limits, montée en tier automatique, Priority Tier pour engagement
- Monitoring en production : suivi du cache, usage par workspace, métriques de coût par requête (input + output tokens)
- Déploiement multi-cloud : Amazon Bedrock, Google Vertex AI, Microsoft Azure AI Foundry — différences et délais de fonctionnalités
- Bonnes pratiques de production : rotation des clés, séparation dev/staging/prod par workspace, gestion budgétaire
- Optimisation globale des coûts : choix du modèle par tâche (Haiku pour le simple, Opus pour le complexe), caching systématique, batches pour le non-temps-réel

MODULE

12.

1H45H

Projet fil rouge : construire et déployer une application complète

CONTENU PÉDAGOGIQUE

- Conception de l'architecture : choix du modèle, design des outils avec schémas stricts, stratégie de caching et budget thinking
- Développement guidé : implémentation d'une application multi-outils avec vision, citations et structured outputs
- Évaluation systématique : critères SMAR, exact match, cosine similarity, notation par LLM, A/B testing de prompts
- Sécurisation et mise en production : guardrails, error handling, monitoring des coûts, checklist de déploiement
- Démonstration et revue de code : présentation des projets, feedback personnalisé, plan d'action post-formation

PASSONS À L'ACTION

Construisons ensemble votre session sur-mesure.

Dites-nous vos contraintes (format, lieu, dates, nombre de participants) et recevez une proposition personnalisée sous 24 heures ouvrées.

Akademia Formation

SERVICE ADMINISTRATION DES VENTES

adv@akademiaformation.com

www.akademiaformation.com

Devis personnalisé

RÉPONSE SOUS 24 H OUVRÉES

Format inter · intra

Présentiel ou distanciel

FIN DU PROGRAMME