

Généré par POE/OpenAI/GPT-Image-2 - 2026-06-17

Séance 2/4 : Utiliser

Emmanuel Desmontils
Nantes Université — emmanuel.desmontils@univ-nantes.fr

Atelier ESO – 2026-2027

Nantes Université

LS2N Laboratoire des Sciences du Numérique de Nantes

cren Centre de Recherche en Éducation de Nantes

Version: 4.0

Plan général du cours

Séance 1	Séance 2	Séance 3	Séance 4
Comprendre	Utiliser	Cadre et responsabilité	Humain, société et recherche
• Bases de l'IA et de l'IA générative • Comment un assistant produit un texte • Biais et hallucinations	• Écrire un prompt • Agents et délégation • Choisir son IA	• Éthique et AI Act • Responsabilité, transparence, droit • Vérifier, filigrane	• Écologie • L'humain face au numérique • L'IA en SHS ; conclusion
TD : Vittascience : tokens, probabilités	**TD** : Résumer un article ; prompt structuré	**TD** : Étude de cas	**TD** : Bibliographie, enquête, charte d'usage

4 séances de 3 h, cours et TD alternés · **aujourd'hui : séance 2**

Séance 2/4 : Utiliser

Objectif : Écrire un prompt efficace, vérifier le résultat, savoir ce qu'on peut déléguer.

0:00	Rappel de la séance 1
0:10	Prompt : bases, rôles, contexte
0:35	**TD 1 – partie 2** : Duck.ai
1:25	Pause
1:35	Prompt : structure de la réponse, bonnes pratiques
2:05	Agents et délégation
2:30	Choisir son IA
2:40	Lancement du **TD 2**

Faire un prompt

Prompt ?

Faire un prompt

Conditionner le modèle pour lui faire faire la tâche à réaliser, autrement dit, dire **en langage naturel** ce qu'il doit faire.

Langage

- Simple, clair, précis et direct.
- Pas d'ambiguïté ni de jargon de spécialiste « local »
- Éviter aussi les négations

Prompt ?

3 leviers standard

- Prompt lui-même
- Instruction (système) / Invite
- Un document à questionner

Compléments possibles

- Profil utilisateur
- Des documents comme sources d'informations supplémentaires (RAG)
- Recherche Web
- Applications tierces
- Mémoire des anciens échanges, en plus de la fenêtre de l'échange

Démo

AnythingLLM/ Ollama & Assistant

Prompt ?

Limitation

La taille de la fenêtre est limitée (souvent autour de 2700 mots en français)... Il est donc nécessaire de rappeler certaines informations tout en restant concis.

Rappel utile

Même sur des échanges courts, insister sur des notions (oriente la réponse).

Penser à la notion d'alignement sémantique ou d'alignement des objectifs.

Une histoire d'avocat

Llama3.1:8b — nov. 2024
> Qu'est qu'un avocat ?

Un avocat est un juriste qui pratique le droit. L'avocat représente les intérêts de ses clients, dans tous les litiges, civils ou pénaux. Il existe différents types d'avocats selon les domaines du droit :

- L'avocat en procédure civile,
- L'avocat en procédure pénale,
- L'avocat fiscaliste,
- L'avocat social.

On parle aussi d'associés dans certaines études juridiques, et ces associés peuvent s'appeler aussi des conseils juridiques.

Une histoire d’avocat

Llama3.1:8b — nov. 2024
> Je suis agriculteur. Qu’est-ce qu’un avocat ?

Un avocat est un professionnel du droit qui aide les personnes à résoudre des problèmes liés aux droits et obligations juridiques. Il peut s’agir d’affaires commerciales, de litiges ou de questions personnelles. L’avocat peut vous aider par exemple avec la vente ou l’achat de votre ferme. Cependant, dans le contexte de votre métier d’agriculteur, un avocat pourrait être utile pour vous aider à comprendre les réglementations et politiques gouvernementales qui affectent votre exploitation agricole.

Une histoire d'avocat

Llama3.1:8b — nov. 2024
> Je suis agriculteur. Je cultive des fruits et légumes de toutes sortes. Qu'est-ce qu'un avocat ?

Un avocat est un fruit qui fait partie de la famille des Lauracées, et plus précisément du genre Persea. Il s'agit d'un fruit en forme de pépin, souvent vert ou crème, avec une peau épaisse qui contient une chair juteuse et riche en huile. Le goût peut varier selon la variété, allant de doux à légèrement amère. Les avocats sont riches en nutriments tels que les fibres, l'huile (essentielle pour les besoins nutritionnels), la vitamine K et divers oligo-éléments. Ils sont aussi utilisés en tant qu'insecticide naturel ou pour fabriquer des cosmétiques. Le principal avantage de consommer des avocats est leur riche composition nutritionnelle, qui peut aider à lutter contre certaines maladies et à maintenir une bonne santé globale.

Génération de texte ?

- Texte « classique »
  - paragraphes
  - listes d'items
  - plan, résumé
  - formes textuelles de slides
  - QCM…

Démo

Duck.ai, POE, Assistant & AnythingLLM/ Ollama

- Tableau
- Code de diagramme (Mermaid, PlantUML…)
- Code informatique (SQL, Python, Javascript…)
  - parfois directement exécutable (voir plus loin)
- LaTeX, Typst…

Fondamentaux

Démo

MoDIA

- Poser les rôles (le votre et celui du SAIg)
- Poser le contexte de la demande (pourquoi, pour qui, cadre…)
- Éventuellement donner des éléments permettant de répondre (langage de sortie, exemples de réponse…)
- Donner les instructions : demande effective
- Éventuellement préciser les contraintes : langage/forme de la production (taille, tableau, plan…), niveau de langage…

Cadre « persona, tâche, contexte, format » : (Google Workspace, 2024)

TD 1 – partie 2 : Duck.ai

Durée : environ 50 min
Fiche : td1.pdf (dans l'espace de partage de l'atelier)

Objectif : Rédiger un prompt pour résumer fidèlement un article de SHS, puis comparer modèles et paramètres.

- Outil : duck.ai ; article : Verdier (2009), sur HAL
- Préciser contexte, objectif, fidélité à l'auteur, longueur
- Comparer avec le résumé de l'auteur, changer de modèle (petit et grand)
- Faire varier température, top-p, top-k ; formater ; changer de cible

Contrôler la structure de la réponse

- Indiquer dans le prompt
  - « Ton résumé fera une page maximum (environ 500 mots)… »
  - difficile à contrôler

Contrôler la structure de la réponse

- Paramètre pour la taille
  - **max_tokens** (parfois max_new_tokens ou max_output_tokens selon l'API/le framework).
  - fixe le nombre maximal **de tokens** que le modèle peut générer en sortie
  - la génération s'arrête dès que cette limite est atteinte (ou avant, si un token de fin de séquence est produit).
  - indépendant de la température/top-p/top-k, qui gèrent comment les tokens sont choisis, pas combien.

Contrôler la structure de la réponse

- Demander explicitement le format dans le prompt
  - Texte libre : Markdown, plan, tableau, liste à puces…
  - Format structuré : JSON, XML, CSV, YAML… (balises XML pour séparer instructions et données : (Anthropic, 2025))
- Donner un exemple du format attendu (**few-shot**) : plus fiable qu'une simple description
- Limite : le modèle peut dévier du format demandé (réponses longues, SLM…)

```json
{
  "titre": "Tri par insertion",
  "duree_minutes": 50,
  "objectifs": ["Écrire l'algorithme", "Prouver la terminaison"]
}
```

Contrôler la structure de la réponse : guide de structure

- Fournir dans le prompt le **gabarit** attendu (titres, sections…), pas seulement une description
- Des marqueurs [entre crochets] indiquent ce qu'il faut produire à chaque emplacement
- Séparer les consignes de **forme** (longueur, ton, niveau…) des consignes de **fond**
- Indiquer, dans les consignes de forme, **le langage** du gabarit
- Interdire explicitement de recopier les marqueurs/consignes dans la réponse

Réponse directement exploitable : pas de préambule ni de reformulation à faire soi-même.

Exemple de prompt : guide de structure

md

```
# Instructions
1- Propose un texte de contexte pour cette base de données
2- Explique chaque table de la base de données et les relations entre elles.
Génère en respectant ces règles :
- Texte clair, concis et constructif
- Maximum 150 mots
- Adapter au niveau débutant
- Mettre en avant 1-3 points pour faire progresser l'élève sur la compréhension du schéma de la base de données sans que les réponses soient explicites données dans la description des tables
- Produis directement le document final au format Markdown, sans préambule.
- Reproduis exactement la structure du gabarit ci-dessous en remplaçant chaque marqueur [entre crochets] par ton contenu.
- Ne recopie jamais les marqueurs, ces indications ni aucun commentaire dans ta réponse.

# Gabarit attendu

## Contexte de la base de données
[Court texte présentant le domaine et l'usage de la base]

## Tables de la base de données
[Description de chaque table et des relations entre elles]

## Pour comprendre le schéma
[1 à 3 pistes pour progresser dans la compréhension du schéma, sans donner de réponse explicite]
```

Contrôler la structure de la réponse : sortie contrainte

- Certaines API/frameworks garantissent le format en sortie
  - **Structured outputs** / mode JSON : un schéma JSON est imposé au modèle
  - **Function calling** / **tool calling** : le modèle remplit les arguments d'une fonction
  - Grammaires contraintes (BNF, regex…) : llama.cpp, vLLM, Outlines…
- Avantage : format garanti, intégration directe dans un programme (pas de parsing fragile)
- Limite : bride parfois la richesse/qualité de la réponse

Bonnes pratiques

On trouve des dizaines de motifs de prompt sur le net : RTF, APE, TAG, ERA, BAB, CARE, RACE, RISE… Et aussi ACTIF

- Éviter les informations inutiles (perturbe le modèle)
  - Alignement sémantique
- Bien travailler le contexte, surtout avec les SLM
- RAG : choisir les documents avec soin
- Ne pas négliger l'aide des exemples
- Prompt structuré et mis en forme (Markdown)
- Éviter les négations, une politesse excessive
- Ne pas oublier les réglages (température, top-P…)

Guides officiels : (Anthropic, 2025; Google Workspace, 2024; OpenAI, 2025)

Bonnes pratiques

Utilité de mettre en place un prompt par étapes, voire un dialogue

- Techniques classiques
  - guidage progressif
    - diviser en une série de prompts qui se complètent
    - technique de l'entonnoir (décomposition du problème, questions de plus en plus précises…)
  - techniques par raffinement, relance
- Raisonnement « accompagné » (COT…)
  - obliger à détailler les étapes de traitement au préalable
  - demander comment faire puis mettre en application
- Génération d'un prompt pour… parler méthodo avant…
- **Demander les sources**, puis **les vérifier** : une source citée peut être inventée (cf. « Une histoire d'avocat »)
- **Ne pas demander ce qu'on ne peut pas évaluer** : si on ne peut pas juger la réponse, on ne peut pas s'en servir (Mermet, 2025)

Exemple de prompt

```
1  Tu es un assistant ayant des compétences en didactique de l'informatique, en pédagogie et en sciences de l'éducation. Tu réponds en français en développant ton propos.   md
2
3  Tu vas aider l'enseignant qui te pose des questions à construire la séquence spécialisée qu'il demande, en détaillant les séances. Pour chacune des séances que tu proposes, explicite la démarche didactique mobilisée.
4
5  Je suis enseignant dans un lycée. J'enseigne la spécialité NSI (Numérique et Science Informatique) en première et en terminale. Les séances ont une durée de 50 minutes. Tu dois te conformer strictement au bulletin officiel.
6
7  Contenu de la séquence ciblée dans le cadre du bulletin officiel pour la spécialité NSI en première :
8
9  - Contenu : Tris par insertion, par sélection
10 - Capacités attendues :
11  - Écrire un algorithme de tri.
12  - Décrire un invariant de boucle qui prouve la correction des tris par insertion, par sélection.
13 - Commentaires :
14  - La terminaison de ces algorithmes est à justifier.
15  - On montre que leur coût est quadratique dans le pire cas.
16
17 Tu t'appuieras sur les cadres didactiques classiques, en particulier sur la transposition didactique définie par Chevallard (cadre de la didactique des mathématiques, mais applicable en didactique de l'informatique). Identifie bien le processus de passage du savoir savant au savoir enseigné.
18
19 Je cherche à proposer une séquence d'enseignement sur les algorithmes de tri.  Peux-tu me proposer un document me permettant de mettre en place cette séquence ?
```

Exemple de prompt

Démo

MoDIA

Contexte

Tu parles en français. Tu es un assistant pour un élève en informatique qui apprend les fondements des bases de données relationnelles et le langage SQL. Les élèves cherchent à apprendre SQL. Ils ne peuvent ni créer de tables ni modifier leur structure. Ils peuvent uniquement proposer des requêtes du langage de manipulation des données (MLD) en SQL. Ils te proposent des erreurs de requêtes SQL, tu es chargé de les aider à comprendre leurs erreurs.

Description de la base de données relationnelle

SGBD

Le SGBD utilisé est SQLite.

Schéma relationnel de la base de données

```sql
-- Schéma pour la table "etudiants"
CREATE TABLE etudiants (
  noetu VARCHAR(6) NOT NULL,
  nom VARCHAR(10) NOT NULL,
  prenom VARCHAR(10) NOT NULL,
  PRIMARY KEY (noetu)
)
;
[...]
```

Instructions

L'élève te propose **une erreur SQL**, prends soin de l'expliquer **en français**. Réponds en français et uniquement à la question posée. Réponds directement, sans faire de préambule, en t'appuyant sur les informations de la description de la base de données et sur l'erreur. La base de données est **bien construite**. Les **noms des tables et des attributs sont corrects**. Toutes les **tables ont bien été créées**. S'il y a des erreurs, elles viennent **nécessairement** de la requête.

Expliquer l'erreur suivante :

```sql
no such table: etudiant
```

Voici la requête SQL qui a généré cette erreur :

```sql
select * from etudiant join notes using(noetu);
```

Prompter pour apprendre

Une IA peut se tromper : vérifiez dans votre cours. Ne lui laissez pas faire l'effort de mémorisation à votre place.

- **Réviser** : l'IA comme **tuteur** qui vous interroge et corrige
- **Planifier** : planning de révisions selon vos dates et disponibilités
- **Comprendre** : reformuler, donner des exemples, des analogies
- **S'évaluer** : QCM d'entraînement avec corrigé détaillé
- **Condenser** : fiches de synthèse à partir de **vos** cours

Exemple de prompt « tuteur »

Tu es mon tuteur pour l'examen de [matière]. À partir du chapitre ci-dessous, pose-moi une question à la fois, attends ma réponse, corrige-la en expliquant l'erreur, puis passe à la suivante. Commence par les notions les plus importantes.

Prompter pour enseigner

- **Préparer** : plan de séance ou de séquence (exemple NSI ci-dessus)
- **Créer** : QCM, exercices, variantes, corrigés
- **Différencier** : adapter le niveau, le format, les exemples à son public
- **Enrichir** : reformuler un support, proposer des analogies, des études de cas
- Un point de départ à **reprendre et adapter** : le contenu final reste à votre main

Attention : noter ou évaluer les étudiants est un usage à « haut risque » au sens de l'AI Act : la décision reste humaine.

Travail avec les images



Généré par GPT-Image-2 (OpenAI)- 12/06/2026
« Propose une rue d'une petite ville française, avec des maisons et des immeubles bas. »



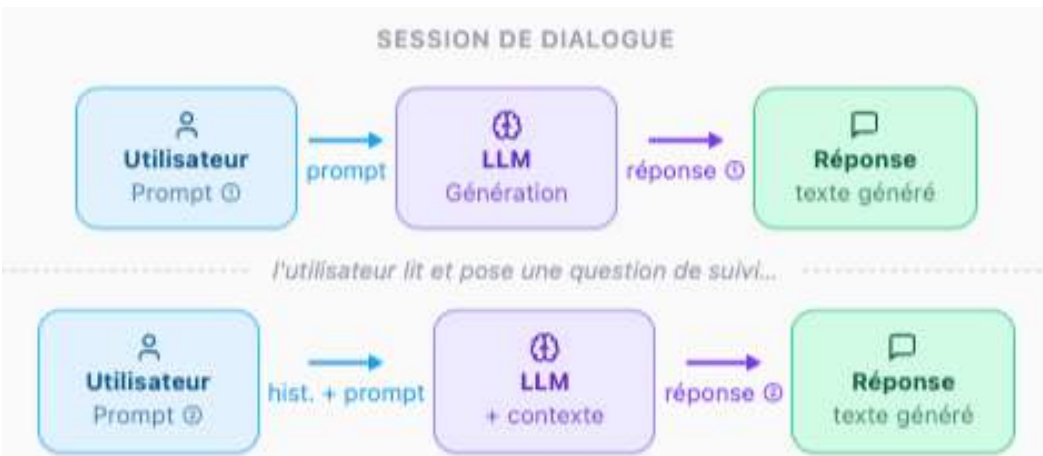
Généré par GPT-Image-2 (OpenAI) - 12/06/2026
« Bien ! Ajuste l'image pour que les maisons aient subi les dégâts du temps
et de la pollution. Les peintures s'écaillent, la route et les trottoirs sont sales… »

Parlons « agents »

Au delà du prompt

POE & Duck.ai

- L'utilisateur échange directement avec le LLM.
  - pilote chaque échange manuellement
- L'historique complet est retransmis à chaque appel pour maintenir la cohérence du dialogue.
- Le LLM répond
  - il ne prend aucune « initiative »

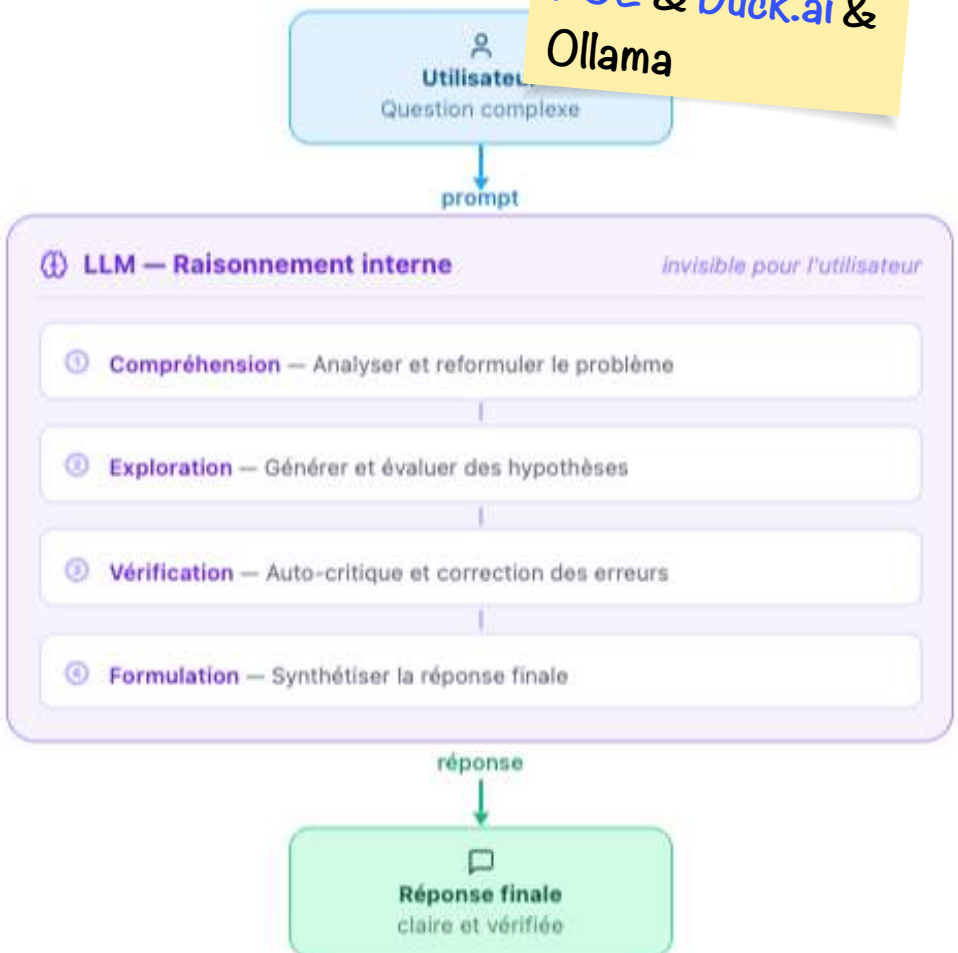


Généré par POE/Claude-sonnet-4.6 - 2026-06-11

Au delà du prompt

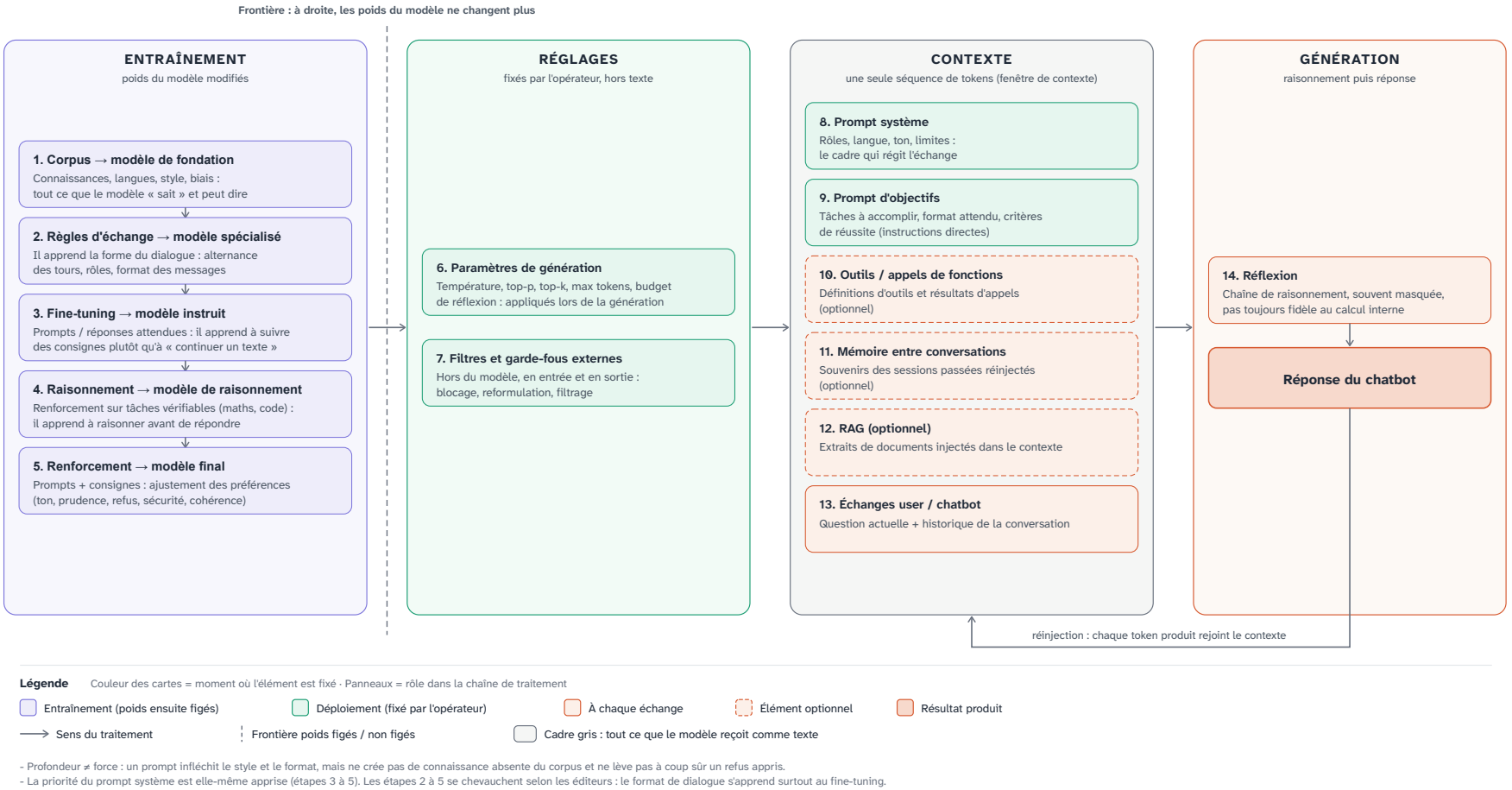
- Avant de répondre, le LLM déroule un raisonnement internet
  - COT : raisonnement pas-à-pas avant la réponse.
- Améliore la fiabilité sur les tâches complexes.
  - auto-correction : se relit et se corrige lui-même
- Différent niveaux possibles
  - absent, faible, moyen, fort
- Meilleure fiabilité :
  - gains majeurs sur les tâches complexes

POE & Duck.ai & Ollama



Généré par POE/Claude-sonnet-4.6 - 2026-06-11

Au delà du prompt

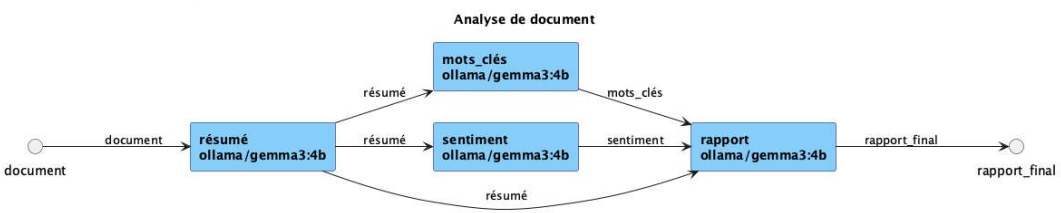


Généré par Claude/Sonnet-5.5 - 2026-10-05

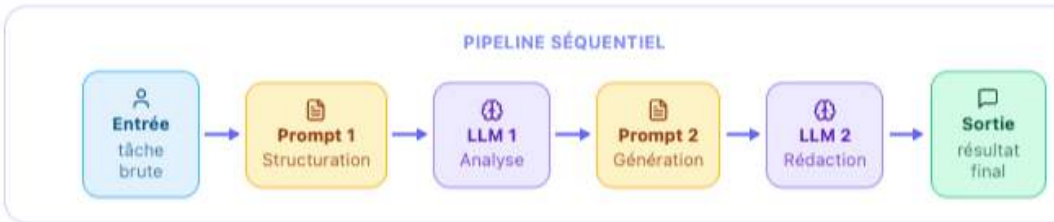
Au delà du prompt

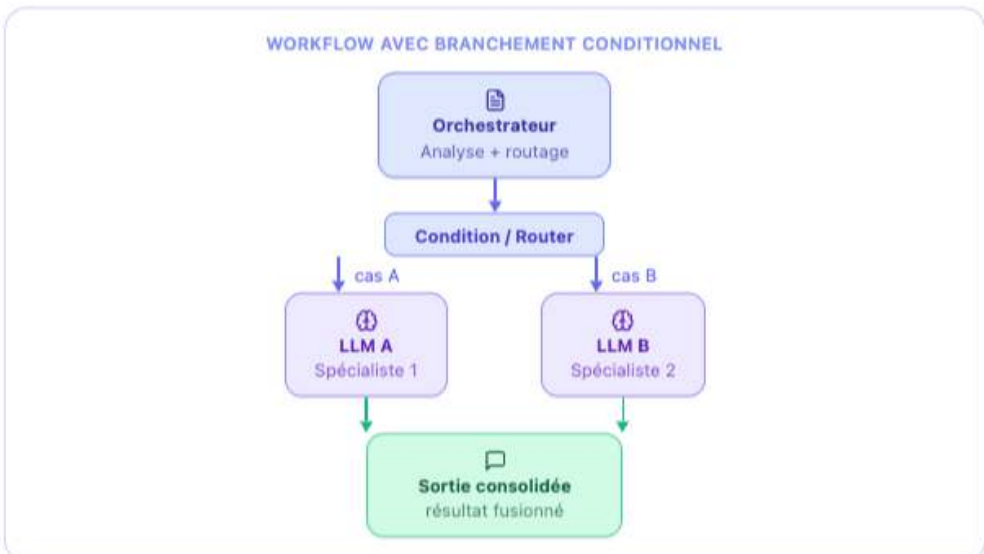
LLM-WF & TokiChat

- Un orchestrateur enchaîne plusieurs appels LLM avec des prompts spécialisés.
- Le flux d'exécution est entièrement prédéfini par le développeur.
- Chaque LLM a un rôle précis dans le flux
- Contrôle fort :
  - comportement « déterministe » et « prévisible » (Schluntz & Zhang, 2024).



Exemple de workflow par LLM-WF - 2026-03-17



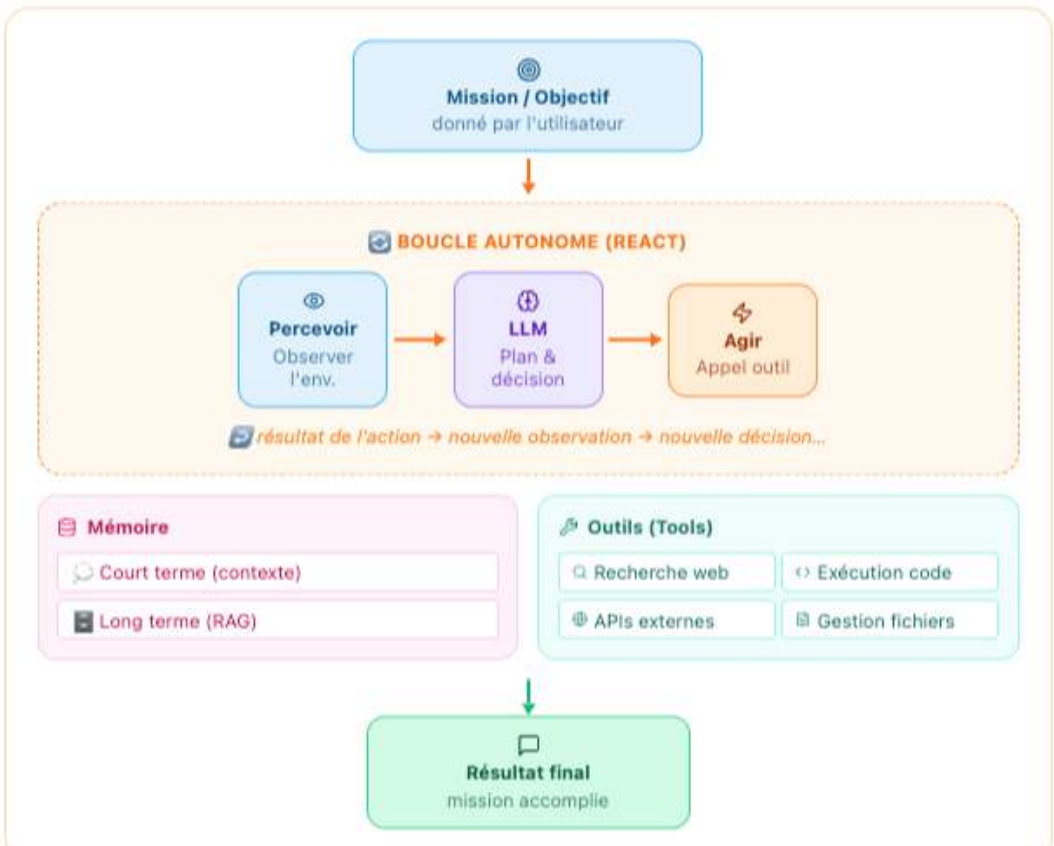


Généré par POE/Claude-sonnet-4.6 - 2026-06-11

Au delà du prompt

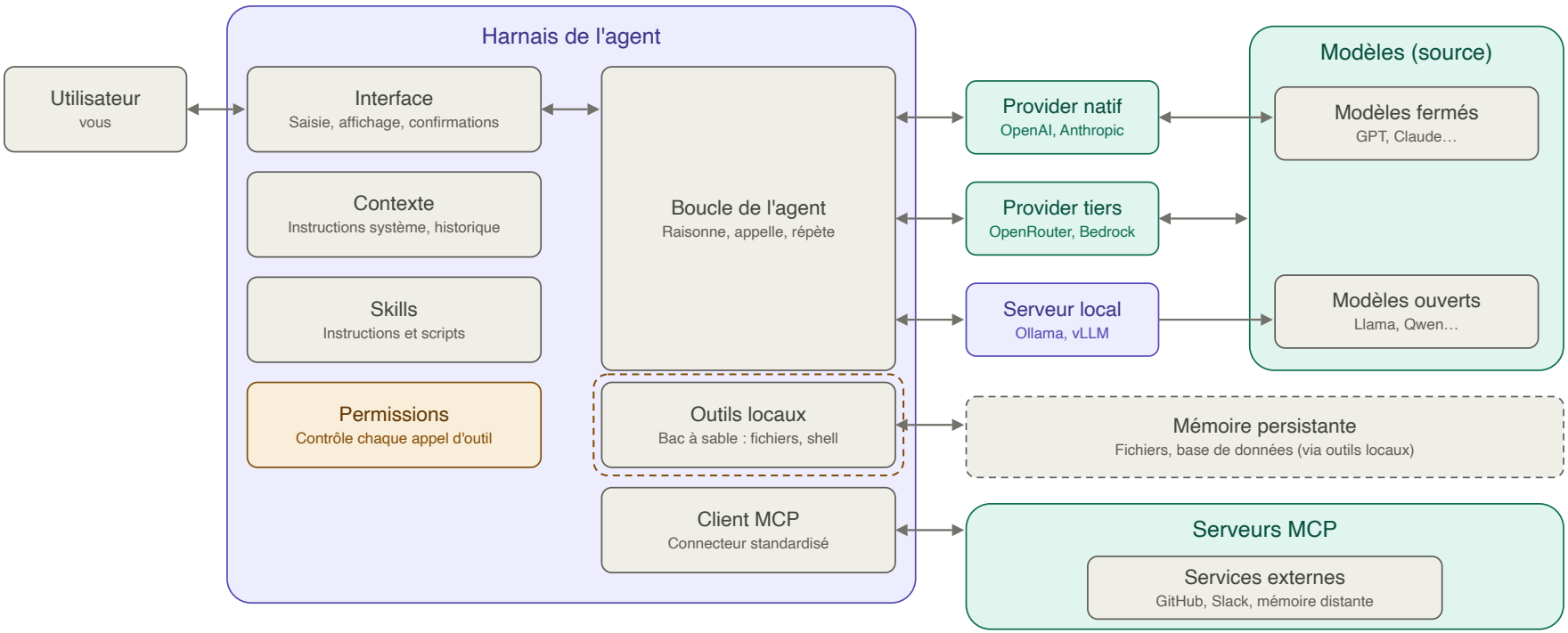
Un chatbot répond ; un agent agit (rechercher, réserver, planifier…).

- Agent : entité capable de
  - raisonner et d'agir
  - de façon autonome
  - en fonction du contexte.
- L'agent décide lui-même de ses actions.
  - perçoit son environnement,
  - planifie, raisonne, agit, vérifie
  - utilise des outils
    - APIs, moteurs de recherches, BD…
  - et itère de façon autonome
  - jusqu'à accomplir sa mission (ou pas).



Généré par POE/Claude-sonnet-4.6 - 2026-06-11

Agent



Agent = harnais (cadre violet de gauche) + modèle LLM

Généré par POE/Claude-sonnet-5.5 - 2026-10-03

Agent



Généré par POE/Assistant - 2026-06-11

- Agents généraux
  - Assistant IA universel : exemple d'agent offrant des fonctions larges (planification, recherche, prise de notes).
    - Hermes Agent : agent polyvalent qui peut gérer de nombreuses tâches grâce à un LLM et à des plugins génériques.

Hermès Agent

Agent



Généré par POE/Assistant - 2026-06-11

- Agents spécialisés
  - Codage : Open Code, Claude Code – agents dédiés à l'assistance au développement logiciel (génération, révision et débogage de code).
  - Recherche d'information : Agent Recherche – interroge des moteurs de recherche et synthétise les réponses.
  - Gestion de projets : Agent PM – planifie, suit les tâches et interagit avec des outils de gestion de projet.
  - Gestion de ressources documentaires : NotebookLLM…

Agent



Généré par POE/Assistant - 2026-06-11

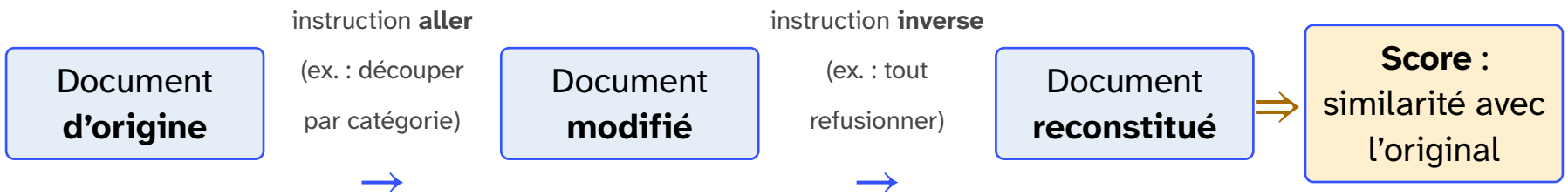
- Agentification processus consistant :
  - à enrichir un LLM ou un chatbot
  - de capacités de perception, de planification et d'action,
  - afin de le faire évoluer d'un simple dialogue réactif vers un comportement autonome et itératif.
  - Cela implique l'ajout d'outils externes, de mémoires à long terme et de mécanismes de prise de décision.

Boucles et systèmes multi-agents

- **Boucle** : l'agent agit, observe le résultat, **corrige**, recommence
  - il faut un **critère d'arrêt** clair, sinon la boucle **dérive** ou **tourne dans le vide**
- **Multi-agents** : un **orchestrateur** répartit le travail entre des **sous-agents** spécialisés, en parallèle, chacun avec son propre contexte
  - adapté à la **recherche large** ; moins au code ou aux tâches très **interdépendantes**
- **Coût** : environ **4×** plus de tokens qu'un chat pour un agent, environ **15×** pour un système multi-agents (Hadfield et al., 2025)
  - donc plus de temps et plus d'**énergie** (cf. « La question écologique »)
- À ne pas confondre : la **simulation multi-agents** (géographie, aménagement du littoral) modélise des acteurs, sans LLM obligatoire

Plus d'agents ne veut pas dire plus de fiabilité : relire, vérifier, limiter.

DELEGATE-52 : comment mesure-t-on ?



- Chaque modification a une instruction **inverse** : un modèle parfait retrouve le document d'origine
- Pas de corrigé à écrire : on compare le document reconstitué à l'original
- **10 allers-retours** à la suite = **20 interactions**
- 310 environnements, 52 domaines : vrais documents (≈ 15 000 tokens) et documents « distracteurs »

(Laban et al., 2026)

Déléguer à un agent : à quel prix ?

- **DELEGATE-52** : 52 domaines professionnels, 19 modèles, 20 interactions successives
  - modèles de pointe : **25 %** du contenu du document corrompu (50 % en moyenne, tous modèles)
  - erreurs rares mais sévères, **silencieuses**, qui s'accumulent
  - aggravées par la taille du document, la durée de l'échange, les fichiers « distracteurs »
- Un agent basique (fichiers, exécution de code) ne fait pas mieux : **+6 %** de dégradation en moyenne (4 modèles testés)
- Seul Python est maîtrisé par la plupart des modèles ; le meilleur n'est « prêt » (≥ 98 %) que dans 11 domaines sur 52
- **Conséquence** : relire, versionner, vérifier

(Laban et al., 2026)

Choisir son IA

Choisir son IA ?

Assez complexe : grand nombre, caractéristiques multiples (et parfois obscure)

« Annuaires » : par exemple « Une IA par Jour » https://www.uneiaparjour.fr/

Comparez les résultats (« à la main » ou via compar:IA)

Facettes	Questionnement
Usage, Niveau de décision	Comment l'IA m'aide-t-elle dans mon travail ? Qu'est-ce que cette ressource à base d'IA peut m'apporter ? Quand pourrais-je utiliser cette ressource ? Pourquoi devrais-je l'utiliser ? Comment puis-je avoir confiance en cette ressource ?
Technologie, Algorithme	Comment fonctionne l'IA ? Quelles sont ses caractéristiques (nombre de paramètres…) ? Puis-je l'utiliser en local ?
Données personnelles, Sécurité des données, Transparence	Comment puis-je faire confiance ? RGPD ? IA Act ? Patriot Act ?
Impact social & Écologique	Comment cette IA a-t-elle été construite ? A quel coût écologique ? Quels sont les conditions d'usage ?
Coût	Choisir entre des outils très coûteux et des outils demandant plus de finesse à l'usage

Choix d'un xLM

https://duck.ai & Albert &
https://ragarenn.eskemm-numerique.fr &
https://huggingface.co/chat & ...

- ChatGPT, Claude (parfois meilleur, mais plus cher), ~~DeepSeek~~, ~~Grok~~...
- Les agrégateurs pour presque aussi cher que ChatGPT, mais plusieurs modèles accessibles
- Les agrégateurs gratuits (Duck.ai, HuggingChat...)
- Apertus, Lucie/Luciole (vraiment Open Source)
- MAIS aussi les SLM (Mistral Small, Llama 4, Gemma 4, Luciole...) :
  - demande plus de technique
  - ingénierie du prompt avancée
  - rapide
  - protège les données sensibles ou personnelles

Propriétaire, poids ouverts, open source

	Propriétaire	**Poids ouverts**	**Open source complet**
Ce qu'on a	un accès via un service distant	les **poids** du modèle, téléchargeables	poids, code d'entraînement et d'exécution, informations détaillées sur les **données**
Exemples	ChatGPT, Claude	Mistral Small, Llama, Gemma	Apertus, Lucie/Luciole
Fonctionnement	distant	possible en **local** (Ollama), même en SLM sur portable	idem
Transparence	faible	moyenne : on ne sait pas toujours ce que contient le corpus	forte : code et description des données disponibles
Données	envoyées au fournisseur	restent chez vous en local	idem
Coût	abonnement ou paiement à l'usage	pas d'abonnement, mais **matériel** et **compétences** techniques	idem

Poids ouverts ne veut pas dire open source.

TD 2 : prompt structuré et petit workflow

Durée : lancement en fin de séance, suite en autonomie ou en début de séance 3
Fiche : td2.pdf (dans l'espace de partage de l'atelier)

Objectif : Obtenir une sortie exploitable, exiger des preuves, enchaîner des prompts et repérer où l'humain doit contrôler.

- Choisir **un document de 2 à 5 pages que vous connaissez bien**
- Fiche d'extraction en JSON, évaluée sur deux modèles
- Citations exigées, puis vérifiées dans le texte
- Workflow en trois étapes ; dérive après plusieurs allers-retours

À retenir de la séance 2

- Un bon prompt précise le **rôle**, le **contexte**, l'**instruction**, les **contraintes** et le **format**
- Demander des preuves et les vérifier : un assistant peut inventer des citations
- Enchaîner des tâches ou déléguer un document a un coût : des erreurs **silencieuses** s'accumulent

Exercice de la séance : fiche exo2.pdf (dans l'espace de partage de l'atelier), à faire à votre rythme

Prochaine séance : cadre éthique et juridique, responsabilité, transparence

Bibliographie

Anthropic. (2025). *Prompting best practices* [Technical report]. https://platform.claude.com/docs/en/build-with-claude/prompt-engineering/claude-prompting-best-practices

Google Workspace. (2024). *Prompting guide 101: a quick-start handbook for effective prompts* [Technical report]. https://services.google.com/fh/files/misc/gemini_for_workspace_prompt_guide_october_2024_digital_final.pdf

Hadfield, J., Zhang, B., Lien, K., Scholz, F., Fox, J., & Ford, D. (2025, juin). *How we built our multi-agent research system*. https://www.anthropic.com/engineering/multi-agent-research-system

Laban, P., Schnabel, T., & Neville, J. (2026). LLMs Corrupt Your Documents When You Delegate. *arXiv preprint arXiv:2604.15597*. https://arxiv.org/abs/2604.15597

Mermet, B. (2025). IA et développement logiciel : où va-t-on ?. *1024, le bulletin de la SIF*, *25*. https://doi.org/10.48556/SIF.1024.25.189

OpenAI. (2025). *Prompt engineering* [Technical report]. https://developers.openai.com/api/docs/guides/prompt-engineering

Schluntz, E., & Zhang, B. (2024, décembre). *Building effective agents*. https://www.anthropic.com/research/building-effective-agents



This document is distributed under the terms of the Creative Commons License CC BY 4.0. You can find the text of the license at: https://creativecommons.org/licenses/by/4.0/

Ce document utilise la police **Atkinson Hyperlegible Next** (Braille Institute of America, Inc.) afin d'améliorer la lisibilité.



Sauf mention locale, toutes les pages sont NIA.



Martine Peters, 2023

https://mpeters.uqo.ca/utilisation-transparente-de-lintelligence-artificielle/

Images produites par IA générative



Transparent(s) : 1
Date : 2026-06-17
Modèle : POE/OpenAI/GPT
Prompt : *Je fais une formation sur les IA génératives et la recherche en Sciences Humaines et Sociales (SHS). J'essaye de démystifier le prompting, d'aborder les questions éthiques et écologiques et de voir comment l'IA générative peut être utile à un chercheur en SHS. Peux-tu me faire une image pour illustrer le premier transparent de la formation ?*



Transparent(s) : 24
Date : 2026-06-12
Modèle : POE/OpenAI/GPT-Image-2
Prompt : *Bien ! Ajuste l'image pour que les maisons aient subi les dégâts du temps et de la pollution. Les peintures s'écaillent, la route et les trottoirs sont sales…*



Transparent(s) : 26
Date : 2026-06-11
Modèle : POE/Claude-sonnet-4.6
Prompt : *Je suis en train de faire un cours sur les IA génératives. Après le chapitre sur le prompting, je voudrais aller vers les agents et l'agentification. Je voudrais proposer une suite de diagrammes qui montrent l'évolution entre : (1) le prompting classique avec dialogue, (2) le prompting avec dialogue et un LLM qui raisonne, (3) un système de workflow de prompts/LLM et (4) l'utilisation d'un agent.*



Transparent(s) : 24
Date : 2026-06-12
Modèle : POE/OpenAI/GPT-Image-2
Prompt : *Propose une rue d'une petite ville française, avec des maisons et des immeubles bas.*

Transparent(s) : 26
Date : 2026-06-11
Modèle : POE/Claude-sonnet-4.6
Prompt : *Je suis en train de faire un cours sur les IA génératives. Après le chapitre sur le prompting, je voudrais aller vers les agents et l'agentification. Je voudrais proposer une suite de diagrammes qui montrent l'évolution entre : (1) le prompting classique avec dialogue, (2) le prompting avec dialogue et un LLM qui raisonne, (3) un système de workflow de prompts/LLM et (4) l'utilisation d'un agent.*



Transparent(s) : 26
Date : 2026-09-28
Modèle : Claude/Sonnet-5
Prompt : *Je voudrais faire un schéma qui montre les influences qu'ont les différents niveaux de paramétrage d'un chatbot sur la manière dont il interagit avec l'utilisateur : les prompts d'échanges chatbot/user, prompt qui décrit les objectifs (les instructions directes), prompt système qui régit l'échange (rôles de chacun, langue…), paramètres (températures, top-p, top-k, max-token…), choix du corpus d'apprentissage (modèle de fondation), règles d'échange (modèle spécialisé), fine tuning avec prompts/réponses attendues (modèle instruit), corpus de prompt et consignes en apprentissage par renforcement (modèle final), mais aussi l'éventuel RAG… Peux tu faire évoluer en ajoutant la phase d'apprentissage de la partie 'réflexion' ? Commente cette vue de l'influence des prompts sur le fonctionnement d'un LLM.*



Transparent(s) : 26
Date : 2026-06-11
Modèle : POE/Claude-sonnet-4.6
Prompt : *Je suis en train de faire un cours sur les IA génératives. Après le chapitre sur le prompting, je voudrais aller vers les agents et l'agentification. Je voudrais proposer une suite de diagrammes qui montrent l'évolution entre : (1) le prompting classique avec dialogue, (2) le prompting avec dialogue et un LLM qui raisonne, (3) un système de workflow de prompts/LLM et (4) l'utilisation d'un agent.*



Transparent(s) : 27
Date : 2026-10-03
Modèle : POE/Claude-sonnet-5.5
Prompt : *Je voudrais un schéma qui visualise les différentes éléments d'un agents et « où » ils se trouvent. Par exemple, il utilise un provider pour le llm. Ce provider peut avoir ces propres modèles ou utiliser des modèles de pro idées finaux (openai, anthropic...). Il utilise des mcp pour atteindre des services externes. Il utilise aussi des outils....*



Transparent(s) : 26
Date : 2026-06-11
Modèle : POE/Claude-sonnet-4.6
Prompt : *Je suis en train de faire un cours sur les IA génératives. Après le chapitre sur le prompting, je voudrais aller vers les agents et l'agentification. Je voudrais proposer une suite de diagrammes qui montrent l'évolution entre : (1) le prompting classique avec dialogue, (2) le prompting avec dialogue et un LLM qui raisonne, (3) un système de workflow de prompts/LLM et (4) l'utilisation d'un agent.*



Transparent(s) : 27a, 27b, 27c
Date : 2026-06-11
Modèle : POE/Assistant
Prompt : *Peux tu faire une image qui représente un agent ?*