MAGELLAN

POLYTECH TOURS · INITIATION · 2026

Comprendre & utiliser l'IA

De la théorie aux usages : modèles de langage, prompts, agents et limites

Andrey Pividori

Intervenant Magellan · Cycle ingénieur 4e & 5e année

Pourquoi ce cours ?

- **En quelques années, l'IA s'est invitée partout** : moteurs de recherche, assistants, code, médecine, industrie.
- **Vous allez travailler avec, toute votre carrière.** Autant la comprendre que la subir.
- **L'enjeu n'est pas de tout savoir,** mais de savoir l'utiliser efficacement… et avec esprit critique.

L'esprit du cours : simple dans l'ensemble, un peu plus pointu sur quelques sur quelques notions clés. Une vraie initiation pour bien démarrer. démarrer.

L'IA EST DÉJÀ LÀ

Santé
diagnostic, imagerie

Mobilité
conduite assistée

Code
assistants de dev

Création
texte, image, son

PROGRAMME

Le cours en huit temps

01		**Qu'est-ce que l'IA ?** Définitions et grandes familles
02		**Principes fondamentaux** Comment ça marche, sans les maths
03		**Rédaction de prompts** Les règles d'un bon prompt + un atelier
04		**La construction d'agents** Pourquoi et comment
05		**Les limites de l'IA** Hallucinations, biais, bon usage
06		**IA & sobriété numérique** Prompts efficaces & impact écologique
07		**Le regard d'ingénieur** Construire avec l'IA, pas seulement l'utiliser
08		**Étude de cas : gstack** Les agents en pratique chez Garry Tan (Y Combinator)

OBJECTIFS

À la fin de cette heure, vous saurez…

- Expliquer ce qu’est, et ce que n’est pas, l’IA
- Comprendre, simplement, comment fonctionne un modèle de langage
- Rédiger des prompts clairs et efficaces
- Comprendre ce qu’est un agent IA et à quoi il sert
- Connaître les limites et utiliser l’IA de façon responsable
- Adopter des réflexes de sobriété : prompts efficaces, moindre empreinte
- Garder le regard d’ingénieur : intégrer, évaluer et sécuriser l’IA

MODULE 1 · 10 MIN

Qu’est-ce que l’IA ?

01

Définitions et grandes familles

≈ 10 min

L'IA, qu'est-ce que c'est ?

L'intelligence artificielle, c'est l'ensemble des techniques qui permettent à une machine de réaliser des tâches que l'on associait à l'intelligence humaine : comprendre un texte, reconnaître une image, décider, créer.

Ce que ça fait

Percevoir, analyser, raisonner, générer du contenu.

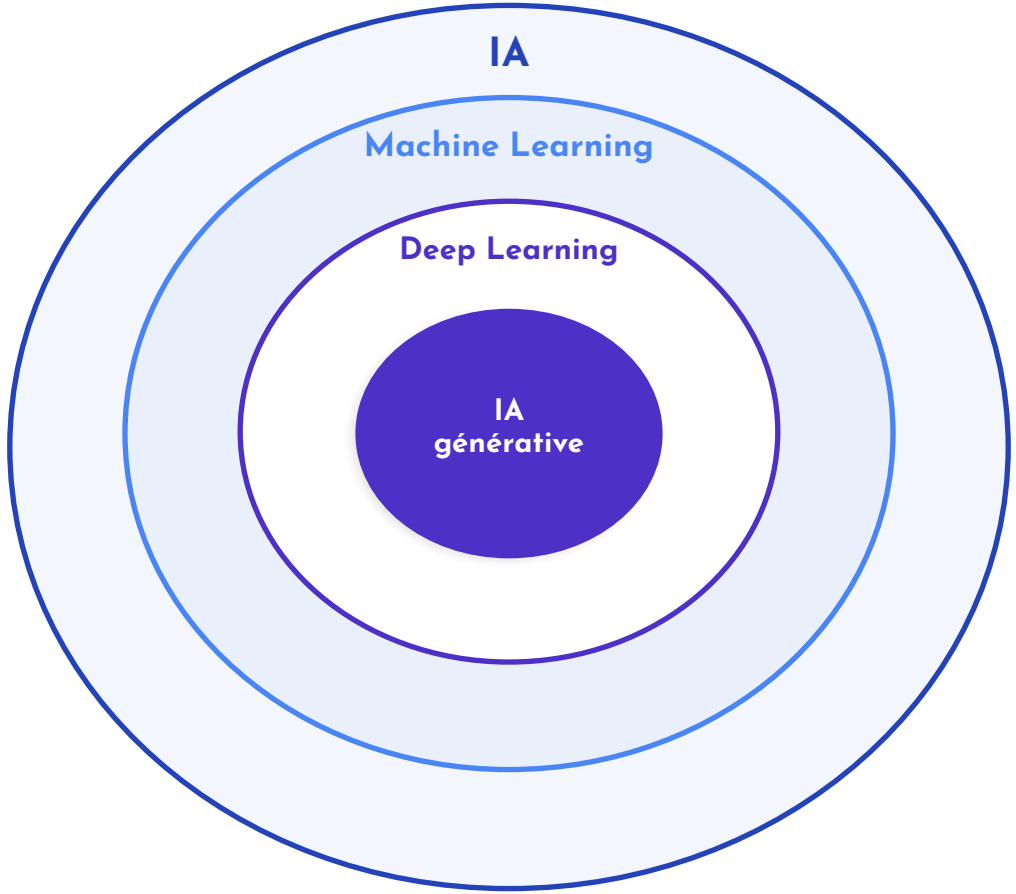
Comment

En apprenant à partir de données et d'exemples, pas avec des règles écrites à la main.

Ce que ce n'est pas

Ni magie, ni conscience. Au fond : des mathématiques et des statistiques.

IA, Machine Learning, Deep Learning



IA

Le domaine global : faire réaliser à une machine des tâches « intelligentes ».

Machine Learning

L'IA qui apprend à partir de données, au lieu de suivre des règles écrites.

Deep Learning

Du ML fondé sur des réseaux de neurones « profonds ». Très puissant sur l'image et l'image et le langage.

IA générative

Le deep learning qui crée du contenu (texte, image, code). C'est ce qu'on utilise au qu'on utilise au quotidien.

IA étroite vs IA générale

IA étroite (faible)

Aujourd'hui

- Excelle sur UNE tâche précise : traduire, jouer aux échecs, générer du générer du texte.
- C'est toute l'IA qui existe réellement, y compris les plus impressionnantes.

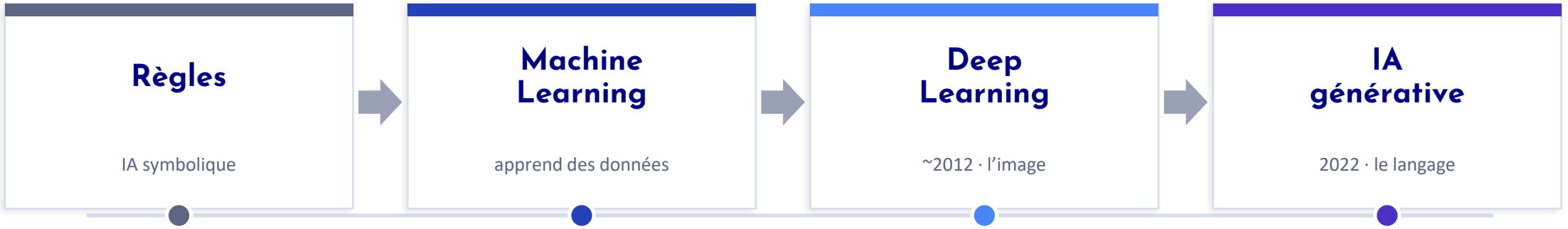
IA générale (forte)

Hypothétique

- Capable de tout faire aussi bien qu'un humain, sur n'importe quel sujet.
- N'existe pas (encore). Sujet de recherche, et de débats.

À retenir : aucun système actuel n'est « conscient » ni vraiment polyvalent. Même les IA les plus bluffantes restent des spécialistes statistiques, très forts dans leur domaine, et seulement là.

L'IA générative a tout changé



Les modèles de langage (LLM)

Entraînés sur d'immenses quantités de texte, ils savent dialoguer, rédiger, résumer, traduire, coder. Surtout : ils sont accessibles à tous, par une simple conversation. **C'est le sujet de tout le reste du cours.**

MODULE 2 · 12 MIN

Principes fondamentaux

02

Comment ça marche, sans les maths

≈ 12 min

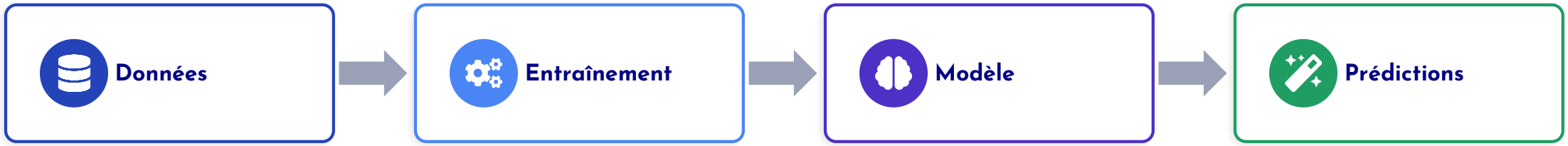
L'IA apprend, elle n'est pas programmée

Programmation classique

L'humain écrit toutes les règles, à la main : « SI… ALORS… ». La machine machine applique, sans rien déduire.

Apprentissage automatique

On montre beaucoup d'exemples ; la machine déduit elle-même les même les règles. C'est ça, le Machine Learning.



Conséquence directe : la qualité d'une IA dépend de la qualité et de la quantité des données vues à l'entraînement. De mauvaises données → un données → un mauvais modèle.

Supervisé, non supervisé, par renforcement

Supervisé

On apprend avec des exemples étiquetés : une : une photo + l'étiquette « chat ». Le plus plus courant.

Reconnaissance d'images, filtres anti-spam

Non supervisé

On cherche des structures cachées, sans étiquettes : regrouper des éléments qui se ressemblent.

Segmentation de clients, détection d'anomalies

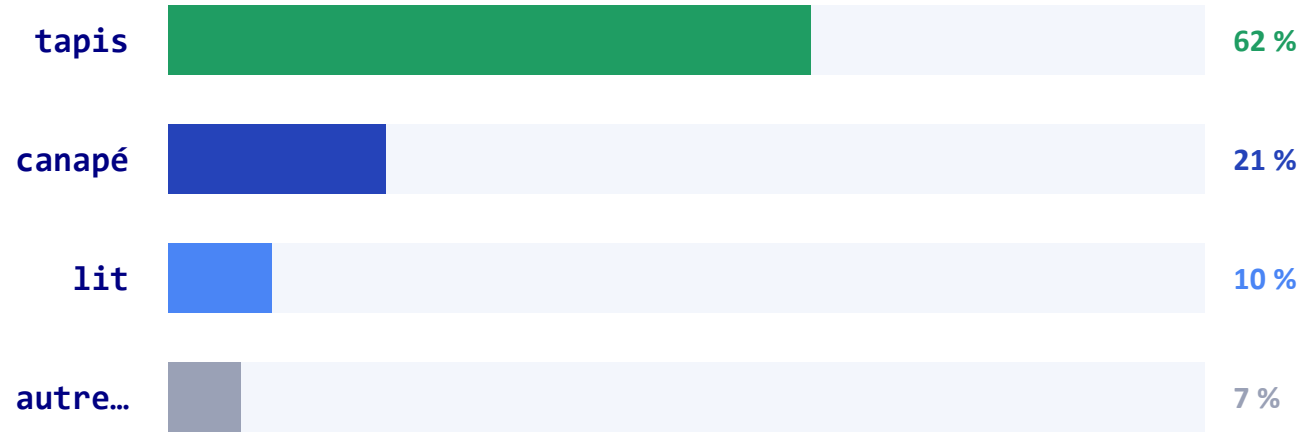
Par renforcement

On apprend par essais et erreurs, guidé par des par des récompenses.

Jeux, robotique, optimisation

Un LLM prédit le mot suivant

« Le chat dort sur le _____ »



Il choisit un mot probable, puis recommence avec le suivant… **et ainsi de suite. Ce n'est pas une encyclopédie qui récite : c'est une machine à prédire. machine à prédire.**

Tokens et fenêtre de contexte

Le texte est découpé en petites unités appelées « tokens » (souvent des morceaux de mots) :

`L’` `intel` `ligence` `arti` `ficielle` *≈ 5 tokens*

Tokens

Le modèle ne raisonne pas en mots, mais en tokens. C’est aussi l’unité de facturation des IA payantes.

Fenêtre de contexte

Tout ce que le modèle « voit » à un instant : votre question + l’historique. l’historique. C’est sa mémoire de travail, et elle est limitée.

En pratique : au-delà de la fenêtre, le modèle « oublie » le début. Donnez le contexte utile dans votre message, et méfiez-vous des méfiez-vous des conversations très longues, qui perdent en précision.

Pré-entraînement, ajustement… et probabilités

1 · Pré-entraînement

- Le modèle « lit » d'énormes quantités de texte et apprend la langue et langue et des connaissances générales.
- Des milliards de paramètres ajustés, un entraînement long et coûteux.

2 · Ajustement (alignement)

- À partir de retours humains, on l'oriente vers des réponses utiles, honnêtes et sûres.
- C'est ce qui transforme un « moteur à texte » brut en assistant fréquentable.

L'idée clé à garder en tête

Un LLM est **probabiliste**, pas déterministe : la même question peut donner des réponses légèrement différentes. Il modélise des régularités du langage, **il ne « comprend » pas comme un humain.**

MODULE 3 · 14 MIN

Rédaction de prompts

Les règles d’un bon prompt + un exemple

≈ 14 min

03

Qu'est-ce qu'un prompt ?

Le prompt, c'est l'instruction que vous donnez au modèle, votre message. C'est votre seul levier de pilotage : toute la qualité de la réponse en dépend.

« Garbage in, garbage out »

Une demande floue donne une réponse floue. Une demande précise et bien cadrée donne une réponse utile.

Bonne nouvelle : bien « prompter » s'apprend, et quelques règles suffisent à faire une grosse différence.

Les ingrédients d'un bon prompt

Donnez un rôle	« Tu es un expert en… » : cela cadre le ton et le niveau d'expertise.
Donnez le contexte	Le modèle ne sait que ce que vous lui dites. Précisez la situation, le public, le but.
Soyez précis sur la tâche	Une demande claire, une seule à la fois. Dites exactement ce que vous attendez.
Précisez le format	Longueur, structure, style : « en 5 puces », « ton formel », « tableau »…
Montrez des exemples / contraintes	Donnez un exemple du résultat voulu, et dites ce que vous ne voulez pas.

Et surtout : *itérez. Un bon prompt se construit en plusieurs essais, ajustez à partir de la réponse obtenue.*

Le même besoin, deux prompts

prompt_faible

```
Écris un email.

# Problème
# Aucun rôle, aucun
# contexte, aucun format.
# Le modèle doit tout
# deviner.

→ réponse générique,
  à côté du besoin.
```

prompt_efficace

```
# Rôle
Tu es mon assistant com.
# Contexte
Réunion projet, 4 collègues,
jeudi 14h, salle B12.
# Tâche
Rédige un email d'invitation.
# Format
Objet court · 5 lignes max ·
finis par une demande de
confirmation.
```

Même IA, même outil, seule l'instruction change. La précision du prompt fait la qualité (et l'utilité) de la réponse.

Trois techniques utiles

Zero-shot

On demande directement, sans exemple. Rapide, suffisant pour les tâches simples.

Few-shot

On donne quelques exemples du résultat voulu. Le modèle imite le motif.

Chaîne de pensée

« Explique étape par étape » : le modèle raisonne avant de répondre. Bien meilleur sur la logique et les calculs.

Astuce qui marche souvent : ajoutez « raisonne étape par étape » à une question difficile, la qualité du raisonnement s’améliore nettement.

Les pièges à éviter

À éviter

- Un prompt vague (« parle-moi de… »).
- Empiler plusieurs tâches dans un même message.
- Supposer que le modèle connaît votre contexte non dit.
- Accepter la première réponse sans la relire.
- Demander des sources… puis ne pas les vérifier.

Les bons réflexes

- Une demande = un objectif clair.
- Découper une tâche complexe en étapes.
- Donner tout le contexte utile, explicitement.
- Relire, puis demander une amélioration ciblée.
- Vérifier les faits importants à la source.

ATELIER

À vous de jouer : améliorez ce prompt

Le contexte : vous devez résumer un article de 10 pages pour des collègues pressés, qui veulent l'essentiel et les décisions à décisions à prendre.

prompt_de_depart

```
Résume ce texte.

# Trop vague :
# pour qui ?
# quelle longueur ?
# quel format ?
# quoi mettre en avant ?
```

Votre mission

Réécrivez ce prompt en appliquant les cinq ingrédients :

- Rôle, contexte, tâche précise
- Format et longueur attendus
- Ce qu'il faut mettre en avant

Puis comparez vos versions entre vous.

Objectif : voir, en direct, la différence de qualité entre un prompt vague et un prompt cadré.

MODULE 4 · 12 MIN

La construction d'agents

Pourquoi et comment

≈ 12 min

04

Du chatbot à l'agent

Un chatbot répond

- Vous posez une question, il renvoie du texte.
- Il ne sait rien faire d'autre que produire une réponse.
- Exemple : « Explique-moi la photosynthèse. »

Un agent agit

- Il poursuit un objectif en plusieurs étapes.
- Il utilise des outils : web, fichiers, API, code.
- Exemple : « Trouve les 3 meilleurs vols et réserve. »

En une phrase : un agent, c'est un modèle de langage à qui on a donné des outils et un objectif, il décide quoi faire pour l'atteindre.

Ce que les agents apportent

Tâches multi-étapes

Enchaîner automatiquement des étapes vers un objectif, sans tout tout piloter à la main.

Données fraîches & réelles

Aller chercher l'information à jour (web, fichiers, bases) au lieu de se limiter à sa mémoire.

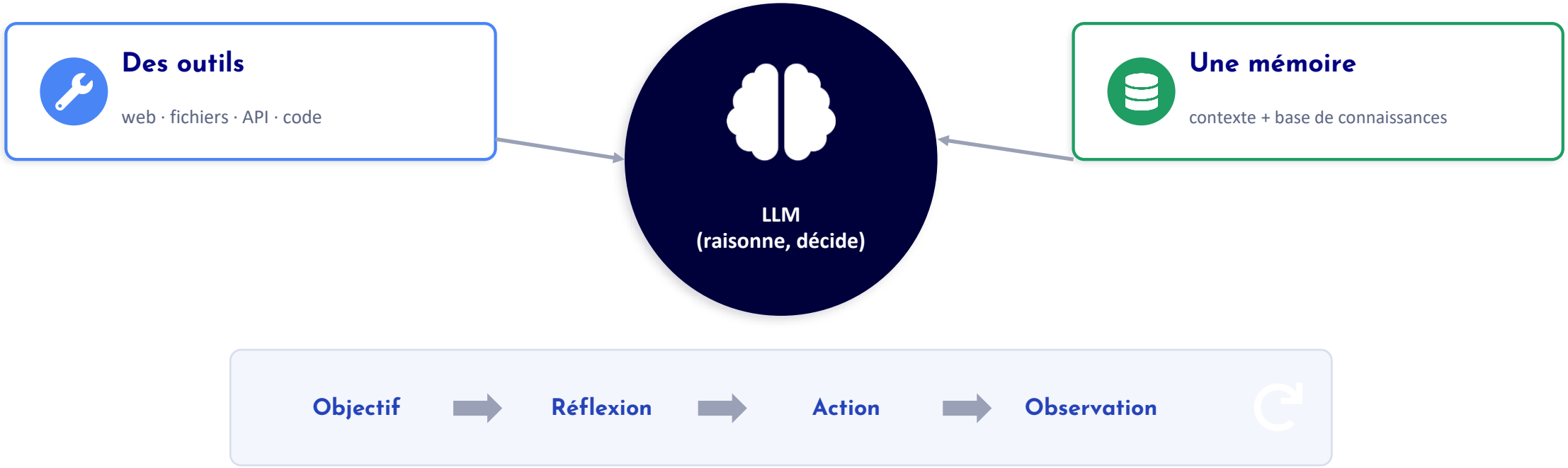
Agir, pas seulement conseiller

Remplir un formulaire, envoyer un message, modifier un fichier, lancer un calcul.

Gagner du temps

Automatiser les workflows répétitifs et chronophages.

Les briques d'un agent



La boucle se répète **jusqu'à atteindre l'objectif.** Pour puiser dans des connaissances externes à jour, on utilise le RAG (recherche puis génération).
puis génération).

Exemples & points de vigilance

DES AGENTS, POUR QUOI FAIRE ?

Assistant de code

écrit, teste, corrige

Support client

répond et agit sur le compte

Veille & recherche

collecte et synthétise

Automatisation

tâches bureautiques répétitives

Avantage

Puissants et autonomes : ils prennent en charge des tâches entières, pas juste un conseil.

À surveiller

Une erreur peut s'enchaîner sur plusieurs étapes. Ils exigent supervision, garde-fous, et attention aux coûts et à la sécurité.

MODULE 5 · 10 MIN

Les limites de l'IA

05

Ce qu'il faut savoir pour bien l'utiliser

≈ 10 min

Les hallucinations

Une hallucination, c'est quand l'IA invente une information fausse… mais présentée avec assurance et de façon parfaitement plausible.

Pourquoi ça arrive

Souvenez-vous : le modèle prédit le mot le plus plausible, il ne vérifie pas la vérité. Quand il ne « sait » pas, il comble le vide avec quelque chose de crédible. Citations, dates, références : tout peut être inventé.

Les bons réflexes

- Vérifier toute information importante à la source.
- Se méfier des chiffres, citations et références précises.
- Recouper, surtout pour une décision qui compte.

Biais, date de connaissance, excès de confiance

Des biais

L'IA reflète les biais présents dans ses données d'entraînement (stéréotypes, sous-représentations).

Une date de connaissance

Ses connaissances sont figées à une date. Sans accès au web, elle ignore l'actualité récente.

Un excès de confiance

Elle peut se tromper… en restant tout aussi assurée. Le ton confiant n'est pas une preuve.

À garder en tête : une réponse assurée et bien écrite n'est pas forcément exacte, récente, ou neutre.

Confidentialité, éthique, environnement

Vos données

Ne saisissez pas d'informations sensibles, personnelles ou confidentielles : confidentielles : elles peuvent être conservées ou réutilisées.

Propriété & plagiat

Le contenu généré peut s'inspirer d'œuvres existantes. Citez, vérifiez, n'usurpez pas la paternité.

Environnement

Entraîner et utiliser ces modèles consomme énergie et eau. Un usage utile, pas gadget.

Responsabilité

En cas d'erreur, c'est l'humain qui décide qui reste responsable. L'IA responsable. L'IA n'assume rien.

L'IA est un copilote, pas un pilote

Gardez l'humain dans la boucle. L'IA propose, accélère, déblaie, mais c'est vous qui décidez, validez et assumez. C'est un assumez. C'est un formidable copilote, pas un pilote automatique.

Vérifiez

recoupez les faits importants

Protégez vos données

rien de sensible dans le prompt

Gardez l'esprit critique

l'IA n'a pas toujours raison

Servez-vous-en

pour gagner du temps, pas pour penser à votre place

MODULE 6 · 8 MIN

IA & sobriété numérique

Optimiser ses prompts, réduire son empreinte

≈ 8 min

06

Une requête n'est pas « gratuite »

Derrière chaque prompt, un data center calcule. Cela consomme de l'électricité, de l'eau (refroidissement des serveurs) et du matériel. Invisible, mais bien réel.

Énergie

≈ 0,3 Wh pour une requête texte typique, comme une ampoule LED pendant ~2 min.

Eau

Une fraction de millilitre par requête, pour refroidir les serveurs.

Matériel

Fabrication des puces (GPU) : ressources, extraction, fin de vie.

À l'échelle d'une requête, c'est peu, et l'efficacité progresse vite. **Le vrai enjeu, c'est le volume : des milliards de requêtes par jour, et des data centers en et des data centers en pleine expansion.**

Ordres de grandeur 2025 (Google, OpenAI, Epoch AI, IEA), très variables selon le modèle et la requête.

Tokens = énergie… et argent

L'énergie, et le prix, d'une requête dépend surtout du nombre de tokens traités, en entrée comme en sortie.

Une réponse longue

Plus de tokens générés = plus de calcul. « 2 pages » coûte plus que « 5 lignes ».

Gros modèle, mode raisonnement

Les modèles qui « réfléchissent » étape par étape peuvent consommer 3 à 4× plus.

De multiples allers-retours

À chaque échange, tout l'historique est relu et recalculé.

Le format

Une image ≈ 10× une requête texte ; une vidéo, bien davantage.

Bonne nouvelle : sobriété = écologie ET économie. Moins de tokens, c'est moins d'énergie… et moins d'euros sur la facture.
sur la facture.

Des prompts plus sobres

Visez juste du premier coup	Un prompt précis évite les itérations, autant de calculs en moins.
La bonne longueur, le bon format	Demandez ce dont vous avez besoin : 5 lignes, pas 2 pages de remplissage.
Repartez d'une conversation neuve	Plutôt que de traîner un très long historique, relu à chaque message.
Le bon modèle pour la tâche	Pas le plus gros modèle par défaut pour une question triviale.
Image et vidéo : si nécessaire	Bien plus coûteuses que le texte, à réserver aux vrais besoins.

En résumé : *bien poser sa demande = bonne réponse du premier coup = moins de calcul gaspillé.*

Concevoir sobre : le réflexe à garder

« Ai-je besoin d'une IA ? »

Une règle, une requête SQL ou une recherche classique suffit souvent, pour souvent, pour bien moins cher.

Dimensionner juste

Un petit modèle, voire un modèle local, plutôt que le plus gros par défaut.

Réutiliser & mettre en cache

Ne pas refaire calculer deux fois la même chose.

Mesurer

Suivre tokens, coûts et énergie ; optimiser là où ça compte vraiment. vraiment.

Numérique responsable : la question n'est pas seulement « comment utiliser l'IA », mais « quand, et à quel coût ». Parfois, la meilleure Parfois, la meilleure requête est celle qu'on ne fait pas.

MODULE 7 · 10 MIN

Le regard d'ingénieur

07

Construire avec l'IA, pas seulement l'utiliser

≈ 10 min

De l'IA-chat à l'IA dans le code

Jusqu'ici, le chat. En pratique professionnelle, on appelle les modèles par API, depuis son propre code : l'IA devient une brique de vos applications.

API / SDK

Appeler un modèle tient en quelques lignes : on envoie un prompt, on reçoit une réponse.

Clé & tokens

Une clé d'API authentifie vos appels. La facturation se fait au token (entrée et sortie).

Un service externe

À gérer comme tout service tiers : version du modèle, coût, latence, disponibilité.

Concrètement : vous intégrerez sans doute l'IA dans un produit avant d'en entraîner une vous-même.

Quand (ne pas) utiliser l'IA

Plutôt l'IA

flou, langage, créativité

- Langage naturel et contenu non structuré.
- Résumer, rédiger, reformuler, classer.
- Explorer, quand une réponse approchée suffit.

Plutôt du code classique

exact, déterministe

- Résultat exact, règles claires, données structurées.
- Calculs, recherches, opérations critiques.
- Plus rapide, déterministe et économique.

Le piège : un LLM pour ce qu'une règle ferait mieux. Le bon réflexe d'ingénieur : choisir l'outil adapté au problème, souvent plus fiable et moins cher.

Comment savoir si la réponse est bonne ?

Une réponse fluide et sûre d'elle n'est pas forcément correcte. On l'évalue, comme on teste du code.

Vérifier sur des cas connus

Comparer la sortie à une « vérité terrain » que vous maîtrisez.

Comparer plusieurs réponses

Confronter différents prompts ou modèles sur la même tâche.

Faire relire (humain)

Pour tout ce qui compte vraiment, la revue humaine reste indispensable.

Industrialiser : les « evals »

Des jeux de tests automatisés mesurent la qualité dans la durée.

Règle d'or : ne jugez pas une IA sur une seule réponse réussie. Mesurez sa fiabilité sur de nombreux cas, surtout avant de l'intégrer à un produit.
à un produit.

L'injection de prompt

L'injection de prompt : quand une application laisse entrer du texte extérieur (page web, document, message), ce texte peut contenir des instructions cachées qui détournent le modèle.

Le risque

- Faire ignorer vos consignes initiales.
- Faire fuiter des données sensibles.
- Déclencher des actions non voulues.

Les bons réflexes

- Ne jamais faire confiance aveuglément à une sortie.
- Valider et filtrer entrées comme sorties.
- Cloisonner et limiter les droits d'un agent.

Une IA dans une application, c'est une surface d'attaque de plus : à traiter avec la même rigueur que le reste.

MODULE 8 · 5 MIN

Étude de cas : les agents en pratique

Garry Tan, Y Combinator et le projet gstack

≈ 5 min

Le CEO de Y Combinator partage sa méthode

Mars 2026 : Garry Tan publie gstack. Le président de Y Combinator ouvre au public sa configuration personnelle de Claude Code : 66 000 étoiles GitHub en quelques semaines.

Qui ?

Investisseur et dirigeant du plus grand accélérateur de startups au monde. Et il code encore, avec des agents.

Quoi ?

23 outils écrits en pur Markdown : des commandes qui assignent un rôle précis à l'agent de code.

Pourquoi ?

Transformer un assistant générique en équipe virtuelle : CEO, designer, eng manager, QA, release manager.

Open source (licence MIT) : github.com/garrytan/gstack. Le repo lui-même a été co-écrit avec Claude.

Une équipe virtuelle, pas un assistant unique

Assistant générique

un seul mode pour tout

- Prend la demande au pied de la lettre, sans la questionner.
- Revue de code superficielle et de qualité variable.
- Beaucoup d'allers-retours pour livrer.

Avec gstack

des rôles explicites

- /plan-ceo-review interroge le produit avant le code.
- Des revues séparées : eng manager, staff engineer, QA.
- /ship enchaîne tests, branche et pull request.

L'idée clé : donner à l'agent un rôle, des priorités et des contraintes. Exactement les principes du module 4, appliqués au quotidien par un dirigeant de la tech.

Ce qu'il faut en retenir (et relativiser)

Les chiffres annoncés : 10 000 lignes de code et 100 pull requests par semaine, sur 50 jours. Impressionnant, mais à lire Impressionnant, mais à lire avec un regard d'ingénieur.

À relativiser

- Les lignes de code ne mesurent pas la valeur.
- Accepter une sortie sans la lire fait perdre l'essentiel.
- C'est un workflow personnel, pas une norme.

À retenir

- De simples prompts structurent tout un workflow.
- La valeur vient de la méthode encodée, pas de l'outil.
- Revue humaine, tests et supervision restent essentiels.

gstack illustre tout le cours : prompts soignés (module 3), agents avec rôles (module 4), regard d'ingénieur (module 7).

SYNTHÈSE

Sept idées à retenir

1. **L'IA d'aujourd'hui est statistique et spécialisée.** ni magique, ni consciente.
2. **Un LLM prédit le mot suivant.** puissant, mais faillible, et probabiliste.
3. **Un bon prompt = rôle + contexte + tâche + format.** et on itère.
4. **Un agent ne fait pas que répondre : il agit.** via des outils, et sous supervision.
5. **Vérifiez toujours.** hallucinations, biais, confidentialité.
6. **Chaque requête a un coût.** énergie, eau, argent : visez des prompts sobres et le bon outil.
7. **Gardez le regard d'ingénieur.** choisir le bon outil, évaluer les sorties, sécuriser l'intégration.

MAGELLAN

MERCI

Des questions ? Et surtout : expérimentez par vous-mêmes.

Andrey Pividori

Intervenant Magellan · Polytech Tours · 2026

andrey@pividori.fr

www.pividori.fr