



UNIVERSITÉ ÉVRY
PARIS-SACLAY

Université d'Évre – Université Paris-Saclay

UFR Sciences et Technologies

Département Informatique

Master MIAGE Ingénierie Logicielle pour le Web

Mémoire de Master

*Comment les modèles de langage (LLM) basés sur
l'intelligence artificielle peuvent-ils améliorer
l'apprentissage de la programmation informatique ?*

par Mathis Beauville

Sous la direction de :



Version publique : les noms des tiers sont caviardés.

Remerciements

Je tiens à exprimer ma profonde gratitude à M. [REDACTED] pour sa supervision, ses précieux conseils et son soutien tout au long de cette année académique.

Je tiens à remercier l'équipe enseignante de l'Université pour le partage de leurs connaissances. Un remerciement particulier à Messieurs [REDACTED], [REDACTED] et [REDACTED]; leur dévouement et leur enthousiasme ont été une véritable source d'inspiration. Leur passion manifeste pour l'enseignement m'a encouragée à approfondir ce sujet d'apprentissage et à m'engager pleinement dans cette thématique.

Je tiens à remercier mes collègues de travail pour leur soutien constant et leurs encouragements, qui ont été essentiels dans le choix et le développement de cette thématique.

Enfin, je remercie ma famille et mes amis pour leur soutien quotidien, leur aide précieuse dans la relecture de ce document et leur accompagnement tout au long de mes recherches cette année.

Table des matières

1	Introduction	4
1.1	Contexte et problématique	5
1.2	Objectifs	6
1.3	Organisation du mémoire	7
2	État de l’art	9
2.1	Fondements théoriques	9
2.2	Programmation web	12
2.3	IA en éducation	13
2.3.1	Apprentissage personnalisé et tutorat intelligent	13
2.3.2	Rôle dans la motivation et la réduction de la frustration	16
2.3.3	Facilitation de l’accès aux ressources et inclusion	17
2.3.4	Enseignants et nouvelles pratiques	19
2.3.5	Comparaison internationale et contextes d’application	21
2.3.6	Limites techniques, éthiques et cognitives	24
2.3.7	Outils existants	26
3	Méthodologie	30
3.1	Question de recherche	30
3.2	Dispositif expérimental	32
3.2.1	Les groupes	33
3.2.2	Protocole expérimental	33
3.3	Recueil de données	35
3.4	Analyse des données	36
4	Mise en oeuvre	38
4.1	Environnement technique	38
4.2	Déroulé expérimental	39
4.3	Ajustements et itérations	42
5	Résultats et analyse	44
5.1	Résultats quantitatifs	44
5.2	Analyse qualitative	47

5.3	Discussion des résultats	48
6	Discussion approfondie et limites	51
6.1	Contributions	51
6.2	Limites	52
6.3	Perspectives	54
7	Conclusion	56
7.1	Synthèse des résultats	56
7.2	Réponse à la problématique	57
7.3	Recommandations	57
7.4	Ouverture	58
	Bibliographie	60
A	Questionnaire de pré-test	62
B	Questionnaire de post-test	65
C	Prompt système du tuteur intelligent	68

Chapitre 1

Introduction

Depuis que j’ai commencé à apprendre la programmation, une question me suit constamment : qu’est-ce que ça veut dire, « apprendre à coder » ? Est-ce comprendre la logique ? Maîtriser la syntaxe ? Être capable de construire un projet seul ? Ou simplement savoir chercher les bonnes réponses au bon moment ? Au fil des années, entre cours magistraux, tutos sur YouTube, et longues soirées à déboguer des fonctions incompréhensibles, j’ai réalisé que la programmation ne s’apprend jamais vraiment en ligne droite. On tâtonne, on copie, on teste, on échoue, et parfois on comprend. Et puis l’intelligence artificielle est arrivée.

Lorsque j’ai découvert ChatGPT pour la première fois, comme beaucoup d’autres, j’ai été impressionné par la capacité du modèle à expliquer, corriger et générer du code. J’ai vu des étudiants l’utiliser comme un assistant de révision, un correcteur, voire un moteur principal de leurs projets. J’ai aussi vu d’autres se perdre dans des réponses vagues, mal contextualisées, ou même erronées. Très vite, une double intuition s’est formée chez moi : ces outils ont un potentiel pédagogique énorme, mais ils posent aussi de vraies questions sur ce que signifie « apprendre » dans un monde où les réponses sont à portée de prompt.

Ce mémoire est né de cette tension. Il ne s’agit pas simplement d’évaluer les performances d’une IA en contexte d’apprentissage, mais d’interroger le rôle qu’elle joue dans le processus même d’appropriation des savoirs. Est-elle une béquille ou un tremplin ? Un raccourci ou un guide ? Et si elle peut aider à apprendre, alors comment l’utiliser au mieux, sans en devenir dépendant ?

Dans les pages qui suivent, je propose d’explorer ces questions à travers un prisme concret : celui de l’apprentissage de la programmation web, avec JavaScript comme terrain d’expérimentation, et deux types d’IA comme compagnons d’étude. Avant d’entrer dans le détail de la problématique, des objectifs et du protocole, je souhaite poser les bases contextuelles et personnelles de cette recherche.

1.1 Contexte et problématique

Depuis plusieurs années, le numérique transforme profondément notre rapport à l'information, au savoir et à la manière dont on apprend. L'éducation, en particulier, n'échappe pas à ces bouleversements. L'arrivée de nouveaux outils numériques, et plus récemment d'intelligences artificielles génératives, change la donne pour les enseignants comme pour les étudiants. Soudain, il devient possible d'obtenir une explication, un exemple de code, une correction ou même une stratégie d'apprentissage en quelques secondes, simplement en posant une question à une IA.

J'ai moi-même été confronté à cette transformation. En tant qu'étudiant, mais aussi comme développeur dans le cadre de mon alternance, j'ai vu ces outils s'intégrer dans mon quotidien. Parfois avec émerveillement, parfois avec méfiance. ChatGPT, par exemple, m'a permis de mieux comprendre certaines notions complexes, de corriger mes erreurs plus rapidement, et même d'explorer des sujets que je n'aurais pas abordés seul. Mais il m'est aussi arrivé de m'appuyer trop vite sur ses réponses, de me reposer sur lui au point d'en oublier le raisonnement derrière une solution.

Ce décalage entre l'aide apportée et l'apprentissage réel m'a particulièrement marqué. À tel point qu'à un moment, j'ai décidé de désactiver GitHub Copilot, pourtant très pratique, pour retrouver une forme d'autonomie dans ma manière de coder. Cette décision m'a permis de mieux comprendre mes erreurs, de progresser plus lentement mais plus solidement.

Dans le cadre de mon alternance, l'usage de l'IA était perçu autrement. Là, pas question de « comprendre avant tout » : il fallait être efficace. Générer des requêtes, automatiser des tâches, trouver la solution la plus rapide. Dans ce contexte, l'IA n'était plus un outil d'apprentissage, mais un levier de productivité. Cette dualité d'usage, entre apprentissage et production, m'a poussé à réfléchir plus en profondeur aux effets réels de ces technologies selon l'objectif poursuivi.

C'est dans ce contexte que j'ai formulé la problématique centrale de ce mémoire : *comment les modèles de langage basés sur l'intelligence artificielle peuvent-ils améliorer l'apprentissage de la programmation informatique ?* Une question large, à laquelle j'ai choisi de répondre en me concentrant sur un domaine que je connais bien : la programmation web, et plus précisément JavaScript.

Ce langage, souvent le premier enseigné dans les cursus orientés web, a été pour moi une porte d'entrée vers des projets concrets. Il est suffisamment accessible pour être utilisé par des débutants, mais assez riche pour permettre une vraie réflexion algorithmique. C'est aussi un langage omniprésent dans l'industrie, ce qui en fait un choix pertinent pour interroger les usages de l'IA dans des contextes réels, proches du terrain.

Ce mémoire s'inscrit donc à la croisée de plusieurs chemins : une technologie en pleine expansion (les LLM), un domaine d'apprentissage vivant (la programmation web), et une interrogation personnelle sur les manières d'apprendre et de progresser à l'ère des IA. Mon objectif est de contribuer à une meilleure compréhension de cette transformation, en prenant le temps d'analyser les effets (positifs comme négatifs) de ces nouveaux outils sur les apprentissages concrets.

1.2 Objectifs

L'objectif de ce mémoire est de mieux comprendre comment les intelligences artificielles, et plus précisément les modèles de langage comme ChatGPT, peuvent influencer l'apprentissage de la programmation web. Derrière cette question se cachent plusieurs enjeux : pédagogiques, techniques, mais aussi humains. Car apprendre à programmer, ce n'est pas seulement apprendre une syntaxe ; c'est apprendre à raisonner, à décomposer un problème, à surmonter l'échec. Et c'est précisément dans ces moments de doute ou de blocage que l'IA peut intervenir... pour le meilleur ou pour le pire.

Plus concrètement, j'ai cherché à comparer deux types d'IA dans un contexte d'apprentissage : une IA généraliste comme ChatGPT, utilisée telle quelle, et une version plus spécialisée, que j'ai moi-même configuré à l'aide d'un prompt conçu pour guider, encourager, et favoriser la réflexion. L'idée n'était pas de tester « qui répond le mieux », mais plutôt de voir si une IA pensée pour accompagner l'apprentissage pouvait avoir un effet tangible sur la progression, la compréhension ou même la motivation des apprenants.

Pour cela, j'ai mis en place un dispositif expérimental accessible en ligne, dans lequel les participants étaient répartis aléatoirement entre ces deux versions de chatbot. Ils devaient résoudre un exercice de programmation en JavaScript, précédé et suivi d'un questionnaire pour évaluer leur niveau, leur ressenti et leur perception de l'outil. Cette approche m'a permis de collecter à la fois des données quantitatives (évolution des scores, fréquence des interactions, etc.) et des données qualitatives (commentaires, ressentis, stratégies d'usage).

Au-delà de cette comparaison, ce travail vise aussi à explorer une dimension plus intime de l'apprentissage assisté par IA : celle du vécu. J'ai voulu comprendre ce que les participants ressentaient face à ces outils : leur confiance, leurs doutes, leurs frustrations ou leur enthousiasme. Car l'IA n'est pas neutre : elle façonne une manière d'apprendre, elle influence l'apprenant, elle redéfinit parfois la frontière entre comprendre et suivre.

Cette réflexion est nourrie par mon propre usage de ces outils, au quotidien, dans mon apprentissage comme dans mon travail. J'ai pu expérimenter leurs apports, mais aussi leurs limites. Et c'est précisément cette ambivalence qui m'a donné envie de creuser la question, non pas comme un développeur, mais comme un apprenant parmi d'autres.

En résumé, ce mémoire poursuit trois objectifs complémentaires :

- **Comprendre** les effets différenciés d'une IA généraliste et d'une IA spécialisée sur l'apprentissage du JavaScript.
- **Expérimenter** un dispositif simple permettant de comparer ces deux approches, en mobilisant à la fois des outils techniques et des retours d'expérience.
- **Proposer** des pistes concrètes pour un usage plus réfléchi et plus pédagogique de l'IA dans les formations à la programmation.

L'ambition n'est pas de trancher définitivement sur l'utilité ou la dangerosité de ces outils, mais d'apporter un éclairage nuancé, à partir d'une expérience de terrain, sur un phénomène encore en construction.

1.3 Organisation du mémoire

Ce mémoire s'organise autour de sept chapitres, qui suivent le fil de ma réflexion et de ma démarche expérimentale. J'ai conçu cette structure pour qu'elle accompagne progressivement le lecteur, depuis les premières interrogations jusqu'aux résultats concrets et aux recommandations finales.

Le **Chapitre 1**, que vous êtes en train de lire, pose le cadre général de la recherche. J'y expose le contexte dans lequel cette réflexion est née, la problématique que je souhaite explorer, les objectifs poursuivis, ainsi que le cheminement personnel qui a nourri ce travail.

Le **Chapitre 2** est consacré à l'*état de l'art*. Il regroupe les principales ressources théoriques et techniques mobilisées : les fondements de l'intelligence artificielle, l'évolution des modèles de langage, les particularités de la programmation web, et les usages actuels des IA génératives dans l'éducation. Cette revue m'a permis de mieux situer ma recherche dans le paysage existant et d'identifier les zones encore peu explorées.

Le **Chapitre 3** décrit la *méthodologie* adoptée. J'y détaille la question de recherche, le dispositif expérimental que j'ai mis en place, les outils de recueil de données utilisés (questionnaires, entretiens), ainsi que les méthodes d'analyse. J'y aborde aussi les choix éthiques, en lien avec le traitement des données personnelles.

Le **Chapitre 4** revient sur la *mise en œuvre concrète* du dispositif. Je présente les choix techniques faits pour développer la plateforme, le déroulé des sessions avec les participants, et les ajustements apportés en cours de route. Ce chapitre est plus technique, mais toujours centré sur l'expérience utilisateur.

Le **Chapitre 5** est dédié à la présentation des *résultats* de l'expérimentation. J'y analyse les données récoltées, en distinguant les aspects quantitatifs (évolution des scores, fréquence des usages) et qualitatifs (perceptions, commentaires). L'objectif est de faire émerger des tendances claires, tout en tenant compte de la diversité des profils.

Le **Chapitre 6** propose une *discussion critique* des résultats. J'y croise les observations avec la littérature, j'en tire des enseignements, mais j'en souligne aussi les limites. Ce chapitre permet d'élargir la réflexion, en explorant les implications possibles de ces résultats pour l'enseignement, la recherche, ou même le développement d'outils pédagogiques.

Enfin, le **Chapitre 7** conclut ce travail. J'y synthétise les principaux apports du mémoire, je réponds à la problématique initiale, et je propose des recommandations concrètes pour les enseignants, les étudiants ou les concepteurs de solutions éducatives.

Le mémoire se termine par une bibliographie complète, suivie d'annexes contenant les éléments utilisés pendant l'expérimentation (captures d'écran, questionnaires, prompt système, etc.). Mon souhait est que ce travail puisse à la fois documenter une réalité en mouvement, et proposer des repères utiles à celles et ceux qui souhaitent utiliser l'IA dans une logique d'apprentissage éclairée et responsable.

Chapitre 2

État de l’art

2.1 Fondements théoriques

L’intelligence artificielle (IA) désigne la capacité d’une machine à reproduire certaines fonctions cognitives humaines, telles que la perception, le raisonnement, l’apprentissage ou encore la résolution de problèmes. Elle repose sur des algorithmes capables de traiter des données complexes pour accomplir des tâches généralement réservées à l’intelligence humaine, notamment la compréhension du langage naturel, la prise de décision autonome ou la génération de contenus.

Au cœur de l’IA contemporaine se trouve l’apprentissage automatique (*machine learning*), qui permet aux systèmes de s’adapter à partir d’exemples sans être explicitement programmés pour chaque tâche. Les modèles construits apprennent à détecter des motifs dans les données d’entraînement, puis généralisent ces connaissances pour effectuer des prédictions ou des classifications sur de nouvelles données. Cette approche repose sur des algorithmes capables de modifier leurs paramètres internes en fonction de l’expérience acquise.

L’une des avancées les plus significatives dans ce domaine est l’apprentissage profond (*deep learning*), une sous-catégorie de l’apprentissage automatique utilisant des réseaux neuronaux profonds. Ces architectures, composées de multiples couches de traitement, permettent l’extraction automatique de caractéristiques à différents niveaux d’abstraction. Elles sont particulièrement efficaces pour analyser des données non structurées comme le texte, les images ou les signaux audio.

L’architecture Transformer, introduite par Vaswani et al. [15], a marqué un tournant dans le traitement automatique du langage naturel (TALN). En s’appuyant sur un mécanisme d’attention, les Transformers permettent au modèle de pondérer l’importance relative des éléments d’une séquence en fonction du contexte, améliorant ainsi la gestion des longues dépendances linguistiques. Cette innovation a permis l’émergence de modèles de langage à grande échelle, ou *Large Language Models* (LLMs), dont les performances dépassent celles des méthodes précédentes.

Les modèles de type GPT (*Generative Pre-trained Transformer*) illustrent pleinement cette évolution. Entraînés sur d’immenses corpus textuels disponibles en ligne, ils acquièrent une connaissance statistique approfondie des structures linguistiques. Leur capacité à prédire les éléments suivants dans une séquence textuelle leur confère des compétences remarquables en génération de texte, en traduction automatique, en réponse à des questions ou encore en assistance à la programmation. L’évolution de GPT-2 à GPT-4 témoigne d’un saut qualitatif notable, tant en termes de cohérence linguistique que de compréhension sémantique [13].

Ces modèles s’appuient sur un paradigme d’apprentissage auto-supervisé (*self-supervised learning*) dans lequel l’étiquetage explicite des données n’est pas requis. Le modèle apprend à combler des lacunes dans des séquences textuelles, ce qui l’amène à capturer les régularités et structures du langage de manière implicite. Cette capacité donne naissance à des propriétés dites émergentes, c’est-à-dire des aptitudes non directement programmées mais acquises par l’échelle de l’entraînement.

Dans ce contexte, le *prompt engineering* émerge comme une technique fondamentale. Il s’agit de l’art de concevoir des instructions textuelles optimisées (ou *prompts*) pour guider les modèles vers des réponses pertinentes. Une formulation précise du prompt peut significativement améliorer la qualité, la pertinence et la fidélité des réponses générées. Cette maîtrise constitue un enjeu crucial dans les usages éducatifs ou professionnels des LLM.

Cependant, malgré leurs capacités impressionnantes, les LLM demeurent perfectibles. Des phénomènes tels que les hallucinations¹ ou les biais algorithmiques² posent des défis majeurs pour une utilisation éthique et fiable. De plus, leur manque d’interprétabilité rend complexe l’analyse de leur fonctionnement interne, ce qui nuit à leur adoption dans des contextes sensibles, comme l’éducation ou la santé.

Sur le plan historique, les LLM constituent l’aboutissement d’une longue évolution. Ils succèdent à des approches statistiques fondées sur les n-grammes, puis à des modèles neuronaux comme Word2Vec, GloVe, RNN ou LSTM. L’introduction du pré-entraînement massif, suivi d’un ajustement spécifique à une tâche cible (*fine-tuning*), a permis une généralisation des capacités linguistiques à de nombreux domaines d’application.

1. production d’informations incorrectes ou incohérentes

2. dérivés des données d’entraînement

Enfin, il convient de souligner que l'entraînement de ces modèles mobilise d'importantes ressources computationnelles, tant en termes de puissance de calcul que de consommation énergétique. Ces contraintes posent la question de la soutenabilité technique et environnementale de ces technologies, notamment dans les contextes éducatifs où les infrastructures peuvent être limitées.

Il convient également à ce stade de distinguer deux approches majeures pour adapter un LLM à un usage spécifique : le *fine-tuning* et le *prompt engineering*.

Le *fine-tuning* consiste à réentraîner le modèle de base sur un corpus spécialisé afin d'ajuster ses représentations internes aux exigences d'une tâche ou d'un domaine donné. Cette méthode offre une précision accrue, notamment dans les contextes où un vocabulaire spécifique ou une structuration particulière des connaissances est nécessaire (par exemple, le droit ou la médecine). Toutefois, cette approche requiert des ressources techniques importantes : un accès à l'architecture du modèle, une puissance de calcul élevée, et des données d'entraînement de qualité. Elle soulève également des enjeux éthiques et juridiques, en particulier en matière de confidentialité des données.

En revanche, le *prompt engineering* repose sur la conception soignée d'instructions textuelles adressées à un modèle déjà entraîné. Il ne modifie pas les poids du modèle mais exploite sa capacité à adapter dynamiquement ses réponses selon le contexte fourni dans le prompt. Cette approche, plus légère et plus accessible, permet une personnalisation rapide sans infrastructure complexe, ce qui en fait une solution privilégiée dans les environnements éducatifs. Elle permet aux enseignants de guider les modèles vers des comportements pédagogiquement pertinents, par exemple en formulant des prompts qui encouragent la réflexion ou la reformulation plutôt que la simple réponse directe.

D'un point de vue pédagogique, le *prompt engineering* favorise l'expérimentation et l'autonomisation des enseignants comme des apprenants, tandis que le *fine-tuning*, plus technique et coûteux, est généralement réservé aux concepteurs de systèmes éducatifs à grande échelle.

Ainsi, les fondements théoriques des LLM reposent sur une convergence de progrès algorithmiques, de puissances de calcul accrues et de volumes de données massifs. Leur appropriation dans le champ éducatif nécessite néanmoins une compréhension fine de leurs mécanismes, de leurs potentiels, mais aussi de leurs limites.

2.2 Programmation web

Avant d’aborder la programmation web en tant que telle, il convient de rappeler que j’ai restreint mon champ d’étude à la programmation web. En effet, la programmation dans son ensemble constitue un champ extrêmement vaste, englobant des paradigmes variés (impératif, fonctionnel, orienté objet), des domaines d’application multiples (IA, développement logiciel, systèmes embarqués, etc.), et un nombre important de langages et d’outils. Face à cette richesse, il était nécessaire de circonscrire l’analyse pour garantir la cohérence et la profondeur du mémoire.

J’ai donc naturellement choisi de m’intéresser à la programmation web, domaine qui correspond à mon champ de prédilection. Ce choix, déjà évoqué dans l’introduction, est ici réaffirmé comme fil directeur de cette partie. La programmation web offre un terrain particulièrement pertinent pour interroger les interactions entre intelligence artificielle, pédagogie et développement logiciel, notamment en raison de sa composante double, de son accessibilité pour les débutants, ainsi que de la richesse des outils mobilisés dans les environnements éducatifs actuels.

La programmation web constitue aujourd’hui un point d’entrée privilégié dans l’apprentissage de la programmation informatique, notamment en raison de son accessibilité et de son potentiel immédiat de visualisation des résultats. Elle permet de concevoir des applications interactives déployées directement via un navigateur, combinant des langages de structuration (*HTML*), de style (*CSS*) et de logique (*JavaScript*). Ce domaine, qui attire aussi bien les novices que les développeurs confirmés, présente une double nature pédagogique. Il est à la fois intuitif dans ses premiers usages, et rapidement complexe lorsqu’il s’agit d’intégrer des *frameworks*³, des architectures client-serveur ou encore des interactions avec des bases de données.

L’évolution de la programmation web reflète la dynamique plus large de la transformation numérique. D’un web essentiellement statique au départ, le domaine s’est enrichi de technologies dynamiques, puis de *frameworks* modernes tels que React, Angular ou Vue.js. Ces outils ont facilité la construction d’interfaces complexes, mais ont aussi introduit une forme de sophistication technique pouvant dérouter les apprenants débutants [16]. En réponse à cette complexité croissante, l’écosystème éducatif s’est adapté : des environnements de développement accessibles en ligne, tels que Replit ou CodePen, permettent une pratique immédiate, favorisant l’expérimentation et le retour rapide sur les erreurs [3].

3. Un *framework* est un ensemble d’outils et de bibliothèques facilitant le développement d’applications en fournissant une structure de base.

Cette accessibilité accrue s’accompagne également de l’émergence d’outils pédagogiques dits *low-code* ou *no-code*, permettant de créer des applications web fonctionnelles sans recourir à la programmation traditionnelle. Bien que ces outils facilitent l’entrée dans le domaine, ils soulèvent des interrogations sur la compréhension des mécanismes sous-jacents. Jošt et al. [7] alertent notamment sur le risque d’une compréhension superficielle lorsque l’usage de ces plateformes précède la maîtrise des concepts fondamentaux.

En parallèle, la diversité des technologies utilisées (JavaScript, TypeScript, bibliothèques front-end, bases de données, gestion des API) peut générer une surcharge cognitive, particulièrement pour les débutants. Dans ce contexte, Boguslawski et al. [2] rappellent l’importance d’un encadrement pédagogique structuré, qui permet de développer progressivement les compétences techniques tout en consolidant la motivation et le sentiment d’efficacité des apprenants.

Ainsi, la programmation web se distingue par sa capacité à offrir un apprentissage progressif, concret et motivant. Elle constitue un espace d’initiation technique autant qu’un laboratoire d’expérimentation pédagogique, dans lequel la variété des outils et des approches doit être articulée autour d’une exigence centrale : comprendre ce que l’on fait, et pourquoi on le fait.

2.3 IA en éducation

2.3.1 Apprentissage personnalisé et tutorat intelligent

L’un des apports les plus prometteurs des modèles de langage de grande taille (LLMs) à l’éducation réside dans leur capacité à simuler un tutorat intelligent, c’est-à-dire à adapter en temps réel leurs réponses et leur accompagnement aux besoins spécifiques de chaque apprenant. Le tutorat intelligent est défini comme un système d’assistance pédagogique capable de diagnostiquer les compétences, les lacunes et les intentions de l’apprenant, afin d’ajuster dynamiquement les contenus et les rétroactions proposés. Historiquement, ce type d’assistance reposait sur des architectures complexes mêlant systèmes experts, arbres de décision et règles pédagogiques codées à la main. Désormais, les LLMs ouvrent une voie nouvelle en permettant une personnalisation des apprentissages à grande échelle, sans nécessiter de modélisation explicite de chaque scénario d’apprentissage.

Les LLMs, lorsqu'ils sont intégrés dans un environnement éducatif, peuvent être considérés comme des tuteurs intelligents adaptatifs. En effet, ils sont capables d'interpréter les réponses des étudiants, de repérer des erreurs conceptuelles, de reformuler des explications, et de proposer des exercices de difficulté graduée. Comme l'expliquent Gan et al. : « Les EduLLMs peuvent être appliqués au tutorat intelligent, en offrant une résolution de problèmes en temps réel, des conseils d'apprentissage et un accompagnement académique à travers le dialogue et l'interaction avec les étudiants » [16]. Ces interactions, dialogiques et contextuelles, permettent une personnalisation de l'apprentissage difficilement atteignable dans un cadre traditionnel, notamment dans des contextes où le rapport enseignant-étudiants est déséquilibré (ex. 1 enseignant pour 50 étudiants).

Dans ce contexte, il est pertinent de distinguer les usages éducatifs d'un LLM générique, tel que ChatGPT, d'un tuteur intelligent spécifiquement entraîné sur des données pédagogiques ciblées. Un LLM générique, bien qu'efficace dans la génération de réponses ou d'explications générales, peut manquer de précision disciplinaire ou pédagogique. En revanche, les tuteurs intelligents spécialisés sont conçus pour épouser une logique curriculaire précise, avec des réponses pédagogiquement calibrées. Ce dernier illustre une application concrète d'un tuteur intelligent dans le domaine de l'apprentissage de la programmation, en fournissant aux étudiants des conseils structurés, du feedback en langage naturel, et des aides à la détection d'erreurs sans livrer directement la solution, ce qui préserve le processus d'apprentissage actif.

Ces outils permettent la mise en œuvre d'un apprentissage véritablement différencié, en cohérence avec les principes de l'éducation personnalisée. À titre d'exemple, des plateformes comme Coursera ou Khan Academy ont déjà commencé à intégrer des assistants basés sur l'IA pour guider les étudiants dans leur parcours. Ces assistants adaptent les contenus à la vitesse de progression, proposent des ressources complémentaires, et suggèrent des points de révision en fonction des erreurs identifiées. Comme l'affirme Zvieli-Girshin, ces technologies offrent des expériences d'apprentissage « student-centered », en répondant à des besoins variés tels que « debugging », « problem-solving » ou « code explanation » [17].

De plus, l'interactivité des LLMs contribue à renforcer l'engagement cognitif des apprenants. En fournissant des retours instantanés, contextualisés et reformulables, ces outils réduisent le sentiment de solitude souvent ressenti dans l'apprentissage en ligne. Cela est particulièrement vrai pour les étudiants débutants en programmation, qui peuvent être découragés par des erreurs répétées ou des messages de compilation obscurs. Les LLMs peuvent non seulement corriger le code, mais aussi expliquer la logique des erreurs, proposer des pistes de compréhension et accompagner la correction pas à pas, ce qui favorise une appropriation active des savoirs.

Par ailleurs, l'aptitude des LLMs à ajuster leur discours au niveau de l'utilisateur constitue un avantage majeur. Dans l'étude menée par Ma, Li et Shin'ichi [3] auprès d'étudiants en première année suivant un cours de Python, les étudiants ont reconnu que ChatGPT avait facilité leur compréhension, notamment en adaptant ses réponses à leur niveau de maîtrise. Ces témoignages mettent en lumière la capacité des LLMs à remplir une fonction de médiation pédagogique, à la fois bienveillante et rigoureuse.

Cependant, ces apports doivent être analysés avec nuance. Comme le rappellent Jošt et al. [7], une utilisation excessive ou non guidée des LLMs peut conduire à un contournement du raisonnement et à une perte de l'effort cognitif, notamment dans des tâches de génération de code. Le rôle du tuteur intelligent n'est donc pas de se substituer à l'apprentissage, mais d'en optimiser les conditions, en soutenant la démarche réflexive et en favorisant l'autonomie progressive.

Enfin, le potentiel des tuteurs intelligents s'inscrit dans une vision plus large de l'éducation du futur : adaptative, inclusive, et centrée sur l'apprenant. Dans ce modèle, les LLMs ne remplacent pas l'enseignant mais en deviennent le prolongement, en offrant une assistance granulaire et constante, qui libère du temps pour des activités à plus forte valeur pédagogique. Loin de déshumaniser l'enseignement, l'IA pourrait, si elle est bien intégrée, en renforcer l'humanité.

2.3.2 Rôle dans la motivation et la réduction de la frustration

L'un des enjeux majeurs de l'intégration des modèles de langage en éducation réside dans leur capacité à influencer positivement la motivation des apprenants, notamment dans les disciplines réputées complexes comme la programmation. Les recherches récentes, notamment celles mobilisant la théorie de l'autodétermination (Self-Determination Theory, SDT), soulignent que la motivation d'un étudiant repose sur la satisfaction de trois besoins fondamentaux : le sentiment de compétence, l'autonomie et le lien social [2]. Dans ce cadre, les LLMs peuvent jouer un rôle d'accompagnement significatif, en renforçant les deux premiers leviers.

Le sentiment de compétence, défini comme la perception qu'a un individu dans sa capacité à accomplir une tâche, est souvent mis à mal dans l'apprentissage de la programmation, notamment chez les débutants confrontés à des erreurs fréquentes ou à des concepts abstraits. L'usage des LLMs permet de restaurer ce sentiment en offrant un soutien immédiat, explicatif et non jugeant. Grâce à leur disponibilité permanente, les LLMs réduisent la frustration liée à l'attente d'un retour humain et permettent une exploration plus sereine des concepts techniques.

L'immédiateté des réponses constitue en effet un levier important dans le maintien de la motivation. Dans l'étude conduite par Ma, Li et Shin'ichi [3], plusieurs étudiants rapportent que ChatGPT les a aidés à « rester engagés » face aux difficultés, car les explications obtenues en temps réel les ont empêchés d'abandonner une tâche frustrante. Cette disponibilité permanente de l'assistance renforce l'endurance cognitive et émotionnelle dans des contextes d'apprentissage autonomes ou à distance, où le découragement peut survenir rapidement.

De plus, la reformulation simplifiée des concepts par les LLMs contribue à démystifier certains blocages, réduisant ainsi l'anxiété associée à l'échec. Ce rôle apaisant est particulièrement important dans l'apprentissage de la programmation, où la confrontation à l'erreur est quotidienne. Comme le note Zviel-Girshin, ces outils peuvent atténuer la « charge émotionnelle » liée aux erreurs, en proposant une lecture neutre et constructive des problèmes rencontrés [17]. Dans certains cas, les étudiants comparent même l'IA à un coach bienveillant, voire à un « compagnon de route », soulignant l'aspect relationnel de l'interaction avec l'outil.

Cependant, cette valorisation du sentiment de compétence ne va pas sans limites. Un danger identifié par plusieurs études est celui de l'illusion de maîtrise. Les réponses fournies par les LLMs, lorsqu'elles sont formulées avec aisance et autorité, peuvent induire une confiance excessive chez les utilisateurs, même en présence d'erreurs ou de raisonnements approximatifs. Jošt et al. ont observé que « Nos résultats suggèrent qu'une dépendance excessive aux LLMs pour ces tâches peut freiner le développement de compétences essentielles en résolution de problèmes [...]. De plus, la tendance à la baisse observée dans les notes finales en fonction de l'augmentation de l'usage moyen des LLM, quel que soit le type d'utilisation [...] » [7], suggérant que certains étudiants surestiment leur compréhension réelle. Cette surconfiance peut masquer des lacunes fondamentales et limiter le développement d'un raisonnement autonome.

De plus, la facilité d'accès aux réponses peut réduire l'engagement dans l'effort cognitif nécessaire à l'apprentissage profond. Ainsi, la motivation soutenue par l'IA ne doit pas être confondue avec une motivation authentique et durable. Le rôle des enseignants reste donc crucial pour encourager une posture réflexive et une appropriation critique des outils. Comme le rappellent Boguslawski et al., les LLMs ne remplacent pas le besoin de soutien social : « Les LLM actuels ne peuvent pas fournir ni remplacer de manière adéquate le soutien social, qui demeure un facteur essentiel de la motivation des apprenants. » [2].

En somme, les LLMs peuvent constituer un puissant levier de motivation en réduisant la frustration liée à la complexité de l'apprentissage et en renforçant le sentiment de compétence. Toutefois, leur efficacité dépend de l'encadrement pédagogique dans lequel ils s'inscrivent, afin d'éviter les effets pervers d'une utilisation passive ou irréfléchie. Un équilibre reste à trouver entre soutien adaptatif et développement de l'autonomie intellectuelle.

2.3.3 Facilitation de l'accès aux ressources et inclusion

Les modèles de langage de grande taille (LLMs) jouent un rôle croissant dans la démocratisation de l'accès aux ressources éducatives. Leur capacité à générer, reformuler, traduire et adapter les contenus constitue un élément important pour favoriser l'inclusion, notamment dans les contextes d'apprentissage autonome, à distance ou dans des environnements caractérisés par une hétérogénéité linguistique, cognitive ou socio-économique. Ces outils participent ainsi à la réduction de certaines inégalités structurelles dans l'accès au savoir.

Premièrement, les fonctions de traduction automatique et de simplification des contenus offertes par les LLMs permettent de surmonter des barrières linguistiques importantes. Dans un cadre multilingue ou pour des étudiants non natifs dans la langue d’enseignement, l’IA peut agir comme un médiateur de compréhension. À travers des reformulations adaptées au niveau de l’utilisateur, l’apprenant peut accéder aux concepts sans être freiné par un jargon technique ou un niveau linguistique insuffisant. Cette approche est particulièrement pertinente dans l’apprentissage de la programmation, où la maîtrise des termes techniques conditionne la compréhension des erreurs, des documentations ou des forums spécialisés.

Par ailleurs, l’intelligence artificielle offre des opportunités nouvelles en matière d’accessibilité pour les personnes en situation de handicap. Les personnes dyslexiques, par exemple, peuvent bénéficier de textes reformulés en langage simplifié ou d’une segmentation plus claire des informations. Certains systèmes peuvent également générer des descriptions alternatives, proposer une lecture vocale du contenu, ou s’adapter aux préférences sensorielles de l’utilisateur. Comme le souligne Gan et al., les EduLLMs peuvent « a partir de vastes données et corpus éducatifs, les EduLLMs peuvent offrir un accompagnement d’apprentissage personnalisé. » [16], ce qui ouvre la voie à des ajustements fins pour des profils cognitifs spécifiques.

Ces potentialités prennent tout leur sens dans les contextes d’apprentissage en autonomie, comme les MOOC ou les plateformes d’apprentissage en ligne. Dans ces environnements, les apprenants sont souvent livrés à eux-mêmes face à des ressources abondantes mais parfois peu lisibles ou peu contextualisées. Les LLMs permettent de naviguer plus aisément dans cette masse d’informations en facilitant l’accès rapide à des explications ciblées, en synthétisant des concepts complexes, ou en générant des exemples à la demande. L’étude de Silva et al. [4] montre que la majorité des étudiants interrogés reconnaissent dans ChatGPT un outil facilitant l’accès aux connaissances, notamment grâce à sa capacité à répondre à des questions précises en langage naturel.

En comparaison avec des outils non dotés d’IA (tels que les forums traditionnels, les FAQ statiques ou les moteurs de recherche classiques), les LLMs offrent une expérience plus fluide et interactive. Cette dimension dialogique permet à l’utilisateur de clarifier progressivement ses zones d’ombre, en formulant plusieurs requêtes successives, comme dans une véritable discussion. Cette dynamique favorise l’apprentissage par questionnement et par reformulation, deux mécanismes essentiels dans la construction du savoir.

Toutefois, ces avancées technologiques ne doivent pas occulter un risque latent : celui de creuser la fracture numérique. En effet, l'accès aux LLMs suppose non seulement une connexion internet stable, mais également une littérature numérique suffisante, ainsi qu'un équipement adapté. Les étudiants issus de milieux défavorisés, ou vivant dans des zones géographiques peu desservies, peuvent se retrouver exclus de ces dispositifs s'ils ne bénéficient pas d'un accompagnement spécifique. Par ailleurs, certaines plateformes d'IA restent payantes ou nécessitent un abonnement premium pour débloquer toutes les fonctionnalités, ce qui constitue un frein économique à leur adoption massive.

En définitive, si les LLMs offrent des perspectives concrètes pour améliorer l'accessibilité et favoriser l'inclusion dans l'éducation, leur déploiement doit s'accompagner de politiques éducatives volontaristes, visant à garantir l'équité d'accès à ces technologies. Il s'agit notamment de former les utilisateurs à un usage critique et éclairé de ces outils, de favoriser leur intégration dans des environnements d'apprentissage inclusifs, et d'anticiper les obstacles liés à l'infrastructure ou à la fracture socio-numérique.

2.3.4 Enseignants et nouvelles pratiques

L'introduction des LLMs dans le champ éducatif ne transforme pas uniquement les modalités d'apprentissage des étudiants ; elle engage également une reconfiguration profonde des pratiques enseignantes. Cette mutation, bien qu'annonciatrice de nombreuses opportunités, suscite aussi des réticences de la part du corps enseignant, confronté à des enjeux à la fois pédagogiques, techniques, éthiques et institutionnels.

L'un des premiers freins identifiés réside dans la crainte d'un remplacement partiel de la fonction enseignante par l'intelligence artificielle. Ce sentiment, souvent amplifié par la méconnaissance des mécanismes internes des LLMs, se double d'un malaise face à une charge mentale accrue : nouveaux outils à appréhender, responsabilité accrue dans la vérification de l'authenticité des productions étudiantes, sentiment d'obsolescence partielle de certaines méthodes pédagogiques. Comme l'illustrent Jošt et al. [7], les enseignants perçoivent parfois l'IA comme une entité difficile à contrôler, capable de court-circuiter les étapes formatives du raisonnement.

En parallèle, le manque de formation spécifique à ces outils constitue un obstacle majeur à leur appropriation. Dans de nombreux cas, les enseignants se trouvent confrontés à des dispositifs techniques dont les logiques de fonctionnement, le vocabulaire, ou les potentialités leur échappent partiellement. Ainsi, la maîtrise du *prompt engineering* devient une compétence pédagogique stratégique, permettant d’exploiter pleinement les LLMs comme vecteurs d’apprentissage adaptatif.

Mais au-delà de ces craintes, de nombreuses opportunités émergent pour les enseignants qui intègrent ces outils dans leurs pratiques. Les LLMs peuvent être mobilisés pour générer des contenus personnalisés (exemples de code, quiz, explications différenciées), analyser automatiquement les copies d’étudiants ou détecter des difficultés récurrentes dans les productions. Ils permettent aussi de mettre en œuvre une pédagogie différenciée à grande échelle, en adaptant les supports aux profils cognitifs des apprenants.

Cette transformation redéfinit également le rôle de l’enseignant, non plus comme unique source de savoir, mais comme médiateur humain dans un écosystème informationnel élargi. Le lien pédagogique devient un espace de supervision, de validation, mais aussi de co-interprétation des productions issues de l’IA. Dans ce modèle, l’enseignant accompagne l’apprenant dans l’élaboration d’un regard critique sur les réponses générées, en interrogeant la fiabilité, la cohérence, et la logique sous-jacente des propos de l’IA. Cette posture suppose une vigilance accrue et une redéfinition des objectifs d’apprentissage, en centrant davantage l’évaluation sur les compétences d’analyse, de justification, et de synthèse.

Certaines initiatives pilotes montrent l’efficacité de cette approche intégrative. L’étude de Snipes [14] décrit des expérimentations menées dans plusieurs établissements nord-américains, où des enseignants ont été formés à l’usage pédagogique des LLMs dans des contextes variés (soutien à la rédaction, accompagnement de projets, remédiation ciblée). Ces expériences mettent en évidence une transformation progressive des représentations : l’IA n’est plus perçue comme un substitut menaçant, mais comme un catalyseur pédagogique, à condition d’être encadrée et contextualisée.

Enfin, cette acceptation suppose une adaptation des dispositifs institutionnels. L’inclusion des LLMs dans les pratiques enseignantes nécessite un cadre éthique clair, une politique d’établissement sur leur usage, et une réflexion sur l’évaluation. Il s’agit notamment de repenser la place de la créativité, de la collaboration, et de la vérification dans les modalités d’apprentissage. Dans ce contexte, l’enseignant devient un acteur stratégique de la régulation technopédagogique, garant de la pertinence et de l’intégrité des usages éducatifs de l’IA.

En somme, l’acceptation des LLMs par les enseignants constitue un processus complexe, à la croisée de la formation, de l’innovation, de la confiance technologique et du renouvellement didactique. Ce processus, loin d’être linéaire, révèle les tensions mais aussi les potentialités d’un changement de paradigme éducatif en cours.

2.3.5 Comparaison internationale et contextes d’application

L’intégration des LLMs dans l’éducation ne suit pas une trajectoire homogène à l’échelle mondiale. Les contextes d’appropriation varient fortement selon les régions, les cultures éducatives, les politiques publiques et les infrastructures techniques. Cette diversité d’approches permet d’enrichir la réflexion sur les conditions d’une intégration efficace et éthique de l’intelligence artificielle dans les systèmes éducatifs.

En Asie, plusieurs initiatives publiques et privées ont placé les technologies éducatives au cœur des stratégies nationales de transformation pédagogique. La Chine, par exemple, a fortement investi dans l’« éducation intelligente » en s’appuyant sur des modèles d’IA entraînés localement. Gan et al. notent que « la recherche sur les grands modèles éducatifs (EduLLMs) évolue constamment, proposant de nouvelles méthodes et approches pour atteindre les objectifs d’apprentissage personnalisé, de tutorat intelligent et d’évaluation éducative » [16]. Ce développement s’appuie sur une forte centralisation des données éducatives, un soutien gouvernemental massif, et une culture favorable à l’expérimentation technologique à grande échelle.

En Amérique du Nord, l’intégration des LLMs dans l’enseignement supérieur est portée par des initiatives universitaires pionnières, telles que CS50 à Harvard, où des outils IA ont été spécifiquement développés pour accompagner les étudiants dans leur apprentissage de la programmation [9]. Ces projets témoignent d’une approche localisée, où l’outil est conçu pour répondre aux spécificités d’un cours, tout en respectant les principes d’intégrité académique. Aux États-Unis et au Canada, les discussions sur l’IA en éducation sont également nourries par des préoccupations éthiques : biais des modèles, protection des données étudiantes, impact sur l’évaluation, et formation des enseignants au *prompt engineering*.

L'Europe, quant à elle, adopte une position plus prudente et réglementée. Les projets pilotes sont souvent encadrés par des institutions publiques ou des programmes européens tels qu'Erasmus+ ou Horizon Europe. L'acceptabilité sociale des IA éducatives y est conditionnée par un haut niveau d'exigence en matière de transparence, d'explicabilité des algorithmes et de conformité au RGPD. Dans ce contexte, l'IA est perçue comme un outil complémentaire, à intégrer de manière mesurée dans des logiques pédagogiques existantes. L'étude de Boguslawski et al., menée en Allemagne, illustre cette tension entre potentiel perçu et régulation attendue : « les éducateurs doivent déterminer la meilleure manière de soutenir les étudiants dans ce contexte ».» [2].

Les types d'établissements impliqués dans l'expérimentation de ces outils varient également. Les universités technologiques et les écoles d'ingénieurs sont souvent en première ligne, du fait de leur familiarité avec les environnements numériques et de l'urgence à former des profils adaptés aux besoins de l'industrie numérique. Cependant, de plus en plus d'établissements secondaires ou de formation continue testent l'usage de ChatGPT ou d'outils dérivés, notamment pour soutenir l'autoformation, la remédiation ou la production de supports pédagogiques.

Cette diversité de contextes révèle également des freins spécifiques à l'intégration des LLMs : obstacles linguistiques, accès limité aux infrastructures, résistance culturelle à l'innovation pédagogique, ou encore absence de cadre réglementaire clair. Par exemple, dans certains pays émergents, l'accès irrégulier à Internet ou aux outils numériques rend difficile la généralisation de ces dispositifs, exacerbant ainsi les inégalités éducatives. À l'inverse, dans des pays comme la Finlande ou Singapour, où les politiques éducatives sont historiquement tournées vers l'innovation, les LLMs sont perçus comme une opportunité stratégique pour renforcer l'individualisation de l'enseignement.

Enfin, des organismes internationaux tels que l'UNESCO ou l'OCDE appellent à une gouvernance mondiale de l'intelligence artificielle éducative. L'UNESCO, dans sa recommandation sur l'éthique de l'IA, insiste sur le rôle des États dans la régulation des usages éducatifs de l'IA, en garantissant un accès équitable, une transparence des modèles, et une formation adéquate des enseignants et des apprenants. Ces recommandations visent à éviter une adoption non critique ou biaisée des technologies, tout en favorisant une appropriation contextuelle et inclusive.

Ainsi, la comparaison internationale des usages des LLMs en éducation met en évidence des écarts mais aussi des convergences : partout, la question de l’accompagnement humain, de l’équité d’accès et de la qualité pédagogique demeure centrale. L’avenir de ces outils dépendra autant de leurs performances techniques que de leur inscription dans des politiques éducatives responsables, cohérentes et culturellement adaptées.

Afin de clarifier les tendances d’adoption des LLMs à l’échelle mondiale, le tableau suivant synthétise les approches régionales identifiées dans la littérature, en croisant stratégies éducatives, exemples d’outils et défis spécifiques.

TABLE 2.1 – Comparaison internationale des usages éducatifs des LLMs

Région / Pays	Approche dominante	Exemples / Outils	Enjeux / Particularités
Chine	Centralisation, IA comme levier stratégique national	EduLLMs, plateformes locales	Accès massif aux données éducatives, soutien gouvernemental fort, peu de débats publics
États-Unis	Innovation locale, approche expérimentale	CS50 Duck (Harvard), Copilot, ChatGPT	Forte autonomie des institutions, enjeux d’intégrité académique, formation au prompt engineering
Canada	Équilibre entre ouverture et régulation	Intégration dans les LMS, initiatives universitaires ciblées	Préoccupation pour l’équité et l’éthique, encadrement pédagogique
Europe	Régulation éthique et prudente	Erasmus+, projets Open Source (Lingua, Edunao)	RGPD, exigence de transparence, mise en œuvre lente mais réfléchie
Pays émergents	Usage émergent, contraintes infrastructurelles	ChatGPT (version gratuite), outils mobiles	Fracture numérique, accès inégal, dépendance aux outils gratuits
Finlande, Singapour	Intégration stratégique à l’échelle nationale	Assistants IA dans l’enseignement secondaire et supérieur	Systèmes éducatifs déjà tournés vers l’innovation, accompagnement fort des enseignants

2.3.6 Limites techniques, éthiques et cognitives

Malgré leur potentiel transformateur, les modèles de langage de grande taille soulèvent un ensemble de limites qui doivent être prises en compte pour garantir un usage responsable et éducativement pertinent. Ces limites, qu’elles soient techniques, éthiques ou cognitives, affectent directement la fiabilité des outils, la qualité des apprentissages, ainsi que la sécurité et l’intégrité des environnements pédagogiques dans lesquels ces modèles sont déployés.

L’un des problèmes techniques majeurs est celui des hallucinations, c’est-à-dire la génération par l’IA de contenus factuellement incorrects, mais formulés avec assurance. Cette capacité des LLMs à produire des réponses plausibles mais erronées pose un défi de taille pour l’apprentissage, en particulier chez les débutants qui ne disposent pas encore des compétences nécessaires pour vérifier l’exactitude des informations. Dans l’étude de Ma, Li et Shin’ichi [3], certains étudiants ont indiqué qu’ils avaient accepté des explications incorrectes sans les remettre en question, ce qui a parfois mené à des incompréhensions persistantes.

En lien avec ces erreurs de fond, la question de l’opacité des modèles est également préoccupante. Les LLMs, notamment ceux de type Transformer, reposent sur des milliards de paramètres dont les décisions sont difficilement explicables. Cette absence de transparence compromet l’interprétabilité pédagogique, c’est-à-dire la capacité à expliquer pourquoi une réponse est correcte ou non. Or, la compréhension métacognitive, essentielle à un apprentissage en profondeur, repose précisément sur la capacité à justifier les démarches entreprises. Gan et al. soulignent également que « l’interprétabilité et l’équité des LLM4Edu constituent des points centraux, exigeant que les processus décisionnels des modèles soient compréhensibles et qu’ils évitent les biais injustes liés aux données » [16].

Les risques éthiques liés à l’usage des LLMs en contexte éducatif sont tout aussi importants. Le plagiat facilité par ces outils constitue une préoccupation centrale : en produisant du code ou des textes cohérents à la demande, les étudiants peuvent contourner les processus d’apprentissage actif. Cette situation engendre une confusion entre assistance légitime et tricherie académique. Savelka et al. [13] soulignent que les versions récentes de GPT, notamment GPT-4, sont désormais capables de « passer des évaluations universitaires sans intervention humaine », ce qui pose des problèmes d’intégrité académique.

Un autre point critique est celui du transfert de compétence. Une utilisation passive des LLMs peut aboutir à une forme d'apprentissage superficiel, où l'étudiant maîtrise l'usage de l'outil sans comprendre les principes sous-jacents. Jošt et al. ont mis en évidence une corrélation négative entre l'usage intensif des LLMs pour la génération de code et les résultats finaux des étudiants : « suggérant clairement qu'une dépendance accrue aux LLMs est corrélée à une baisse des performances académiques dans les travaux de programmation » [7]. Cette tendance montre que l'IA ne peut se substituer à l'effort cognitif, à la construction de représentations mentales durables et à la résolution de problèmes complexes.

La problématique de l'évaluation constitue un autre défi majeur. Lorsqu'un étudiant obtient une réponse correcte grâce à un LLM, comment savoir s'il a réellement compris la démarche ? L'automatisation du feedback ne garantit pas une vérification de la compréhension. Zvieli-Girshin [17], insiste sur ce point en affirmant que les outils d'IA peuvent masquer une compréhension incomplète en produisant du code fonctionnel, mais non maîtrisé. Cela remet en question la validité des évaluations traditionnelles à l'ère de l'IA.

Enfin, l'usage pédagogique des LLMs soulève des enjeux en matière de vie privée et de protection des données. De nombreuses plateformes collectent les interactions des utilisateurs pour améliorer leurs performances, souvent sans que l'étudiant ait pleinement conscience de la nature des données partagées. Le respect du RGPD en Europe, et plus largement des principes de protection des données à des fins éducatives, devient alors crucial. Des garde-fous techniques et juridiques doivent être mis en place pour éviter toute exploitation abusive ou commercialisation des données d'apprentissage.

En conclusion, les LLMs représentent à la fois une opportunité et un risque pour l'éducation. Leur puissance doit être contrebalancée par une vigilance éthique, une pédagogie encadrée et une réflexion critique sur leur rôle dans l'appropriation des savoirs. Ces outils, s'ils sont utilisés sans accompagnement humain ni stratégie éducative claire, peuvent renforcer des mécanismes de dépendance, d'illusion de maîtrise ou d'iniquité. L'enjeu n'est pas seulement technologique, mais profondément éducatif.

2.3.7 Outils existants

L'émergence des modèles de langage de grande taille (LLMs) a favorisé le développement rapide de nombreux outils éducatifs exploitant l'intelligence artificielle, chacun poursuivant des objectifs spécifiques : accompagnement à l'apprentissage, feedback automatique, remédiation individualisée, ou encore simulation d'un tuteur virtuel. Cette sous-section propose un panorama critique des principaux outils existants, à la fois généralistes et spécialisés, en s'appuyant sur des cas concrets documentés dans la littérature.

ChatGPT : un assistant généraliste polyvalent. Développé par OpenAI, ChatGPT est sans doute l'outil le plus emblématique de cette nouvelle génération d'IA conversationnelles. Sa capacité à répondre à des questions ouvertes, à produire du code, à expliquer des concepts complexes et à interagir en langage naturel en fait un assistant numérique très utilisé dans les contextes éducatifs. Sa disponibilité constante et son adaptabilité thématique en font un compagnon d'apprentissage attractif, bien que des réserves soient exprimées quant à la qualité variable des réponses et aux risques de surdépendance. [11]

CS50 Duck : un exemple de LLM spécialisé. L'université Harvard a développé un outil IA sur mesure pour son célèbre cours d'introduction à l'informatique, CS50. Le « CS50 Duck », basé sur un modèle de langage restreint et contrôlé, agit comme une version IA du principe de débogage verbal (« rubber duck debugging »), en guidant les étudiants pas à pas sans fournir directement la solution. Liu et al. notent que ce système a été conçu pour encourager la réflexion autonome tout en fournissant un soutien contextuel et pédagogique adapté : « un expert disciplinaire à l'esprit pédagogique, conçu pour orienter les étudiants vers des solutions plutôt que de les leur fournir directement » [9].

GitHub Copilot : l'IA au service du code. GitHub Copilot, développé conjointement par GitHub et OpenAI, est une extension d'environnement de développement (IDE) qui propose en temps réel des suggestions de code. S'il n'est pas spécifiquement conçu pour l'éducation, il est néanmoins largement utilisé par les étudiants en informatique, notamment pour accélérer l'écriture de programmes ou identifier des erreurs. Cependant, Jošt et al. [7] soulignent que l'usage intensif de Copilot dans des tâches complexes peut nuire à l'apprentissage profond en réduisant l'engagement cognitif de l'étudiant. Son utilisation dans un cadre pédagogique nécessite donc une régulation et une sensibilisation à la compréhension du code généré. [6]

Nolej : plateforme d'apprentissage propulsée par l'IA. Nolej est une plateforme qui transforme automatiquement divers contenus (vidéos, PDF, etc.) en activités d'apprentissage interactives telles que des quiz, des jeux et des vidéos interactives. Elle garantit la sécurité et la confidentialité des données, permettant aux éducateurs de créer des ressources pédagogiques engageantes sans compromettre la confidentialité. De plus, Nolej propose un plugin pour Moodle, facilitant ainsi l'intégration transparente des contenus générés dans les environnements d'apprentissage existants [10].

Aristote : tuteur IA pour l'éducation. Aristote est un tuteur virtuel alimenté par l'IA, conçu pour fournir une assistance éducative personnalisée aux étudiants. Grâce à des algorithmes avancés, il offre des explications adaptées aux besoins individuels, aidant ainsi les apprenants à surmonter les obstacles et à approfondir leur compréhension des sujets étudiés [1].

Edunao : plateforme d'apprentissage numérique. Edunao propose des solutions de campus numérique open source pour l'éducation et la formation professionnelle. Avec une décennie d'expérience dans l'apprentissage numérique, Edunao offre des plateformes LMS⁴ robustes avec une multitude de fonctionnalités, intégrant des outils tels que Moodle-in-Azure et Moodle Workplace, adaptés aux besoins spécifiques des institutions éducatives et des entreprises [5].

Linagora : solutions d'intelligence artificielle open source. Linagora développe des solutions d'IA open source, notamment LUCIE, une IA construite sur des principes de confiance, d'équité et de responsabilité. LUCIE est conçue pour être un modèle sur lequel les secteurs de l'éducation, du gouvernement et de la recherche peuvent compter, en mettant l'accent sur la souveraineté et la durabilité [8].

Outils intégrés aux plateformes d'apprentissage. Certaines plateformes comme Khan Academy, Coursera ou edX commencent à intégrer des assistants IA pour personnaliser l'expérience d'apprentissage. Ces assistants, souvent fondés sur des LLMs, sont capables d'identifier les erreurs courantes, de proposer des explications ciblées, voire d'adapter les parcours pédagogiques à l'évolution de l'étudiant. Bien que peu de données empiriques soient encore disponibles sur leur efficacité réelle, leur déploiement à grande échelle dans les années à venir constitue une évolution majeure du paysage éducatif numérique.

4. Learning management system

Outils à visée inclusive ou spécialisée. Des LLMs sont également intégrés dans des applications ciblant des besoins éducatifs spécifiques. On peut citer par exemple des outils de reformulation de consignes pour les élèves dyslexiques, des assistants vocaux pour les malvoyants, ou des interfaces simplifiées pour les enfants ou les apprenants allophones. Zviel-Girshin [17] mentionne que ces usages spécialisés permettent d’adapter les supports pédagogiques à une diversité de profils, contribuant ainsi à une éducation plus inclusive.

L’ensemble de ces outils témoigne d’une tendance vers une intégration croissante de l’intelligence artificielle dans les environnements éducatifs. Cependant, leur usage reste très hétérogène selon les disciplines, les niveaux d’enseignement et les contextes institutionnels. Des questions majeures restent posées : quels critères pour évaluer leur pertinence ? Comment garantir l’équité d’accès ? Comment former les enseignants à leur intégration responsable ? Les réponses à ces questions conditionneront l’émergence d’un usage réellement transformateur et éthique de ces outils.

Synthèse de l’état de l’art

L’examen approfondi des fondements théoriques, des usages éducatifs des LLMs et des outils existants montre que l’intelligence artificielle représente aujourd’hui un tournant majeur pour l’éducation numérique. Les LLMs comme ChatGPT, Copilot ou CS50 Duck témoignent de la montée en puissance de modèles capables de simuler un tutorat intelligent, d’améliorer la motivation des étudiants et de faciliter l’accès à des ressources contextualisées.

Cependant, cette révolution pédagogique s’accompagne de tensions majeures : surconfiance dans les réponses générées, réduction de l’effort cognitif, difficulté à évaluer la compréhension réelle, ou encore fracture numérique. Une constante ressort des études analysées : les usages non encadrés des IA, bien que séduisants, peuvent court-circuiter l’apprentissage profond s’ils ne sont pas régulés. De nombreuses expérimentations locales (notamment aux États-Unis et en Chine) montrent le potentiel de LLMs spécialisés dans des contextes curriculaires précis, mais peu de recherches portent sur le développement de systèmes intermédiaires : ni totalement génériques, ni entièrement fermés, intégrant des garde-fous explicites.

C’est dans ce cadre que s’inscrit ma posture de recherche : je considère l’IA comme une opportunité pédagogique majeure à condition qu’elle soit utilisée de façon contrôlée, critique, et intentionnellement conçue pour favoriser la réflexion plutôt que la substitution.

À la suite des projets tels que CS50 Duck, j’ai choisi d’élaborer un dispositif expérimental basé sur une IA personnalisée, reposant sur un *system prompt* conçu pour limiter les usages passifs, guider l’étudiant vers la compréhension progressive, et stimuler l’auto-explication. Ce tuteur IA ne vise donc pas à fournir directement la réponse, mais à simuler un dialogue pédagogique favorisant le raisonnement, l’erreur productive et la clarification active.

Le domaine choisi est celui de la programmation web, à la fois pour des raisons personnelles et pour sa pertinence pédagogique : son caractère visuel, interactif et immédiatement testable en fait un terrain propice à l’expérimentation guidée par IA, comme évoqué précédemment.

FIGURE 2.1 – Schéma de synthèse : les dimensions croisées de l’usage des LLMs en éducation

Dimension	Apports potentiels	Risques ou limites
<i>Cognitive</i>	Clarification, feedback immédiat, adaptation au niveau de l’apprenant	Surconfiance, dépendance, contournement de l’effort
<i>Pédagogique</i>	Tutorat simulé, différenciation, apprentissage autonome	Uniformisation des réponses, biais implicites, absence de métacognition
<i>Technologique</i>	Accessibilité accrue, outils variés, intégration dans les LMS	Besoin d’infrastructures, coût des solutions avancées, opacité des modèles
<i>Éthique et sociale</i>	Inclusion (langues, handicaps), démocratisation des savoirs	Fracture numérique, usage non équitable, questions de vie privée

Ce positionnement critique mais constructif guide l’élaboration du dispositif expérimental présenté dans le chapitre suivant. Il s’agira d’étudier concrètement comment une IA éducative conçue avec des garde-fous intentionnels peut favoriser un apprentissage actif de la programmation web, tout en préservant l’autonomie et la motivation des apprenants.

Chapitre 3

Méthodologie

3.1 Question de recherche

L'introduction des LLMs, tels que ChatGPT, Bard ou Claude, dans les environnements éducatifs constitue une évolution technologique majeure. Ces systèmes basés sur l'intelligence artificielle sont désormais capables de comprendre des consignes complexes, de générer du code, d'expliquer des concepts abstraits, de déboguer des programmes et même de proposer des pistes de correction ou d'amélioration. Leur potentiel pédagogique, en particulier dans le domaine de la programmation informatique, suscite de nombreuses interrogations. En effet, l'apprentissage de la programmation repose traditionnellement sur une alternance de cours théoriques, d'exercices pratiques et de recherche documentaire. Cette approche, bien qu'efficace pour certains profils, peut s'avérer difficile d'accès pour des apprenants novices qui peinent à faire le lien entre théorie et pratique, à interpréter des messages d'erreur techniques, ou à mobiliser les bons concepts pour résoudre un problème donné.

Ce mémoire s'inscrit dans cette dynamique de réflexion et d'expérimentation. Il propose d'examiner de manière rigoureuse dans quelle mesure un assistant pédagogique basé sur un LLM peut contribuer à améliorer l'apprentissage de la programmation web en JavaScript, tant sur le plan des résultats que de la compréhension conceptuelle et de la motivation. Le cadre retenu est celui d'un apprentissage du développement front-end en JavaScript natif (sans bibliothèque ou framework comme React ou Angular). Ce choix est motivé par l'importance de la compréhension des fondamentaux du développement web, permettant ainsi de construire des interfaces utilisateur tout en confrontant les apprenants à des logiques telles que la programmation événementielle côté client, la manipulation du DOM¹, la gestion de l'état, et l'organisation de l'interaction avec l'utilisateur, entre autres.

1. Document Object Model : une représentation des noeuds HTML manipulable par JavaScript

Dans ce contexte, la question de recherche principale peut être formulée ainsi : *dans quelle mesure l'utilisation d'un assistant fondé sur un modèle de langage spécialisé permet-elle d'améliorer la compréhension des concepts fondamentaux du développement web en JavaScript ainsi que les performances et l'engagement des apprenants, comparativement à une méthode traditionnelle reposant uniquement sur l'utilisation de ressources documentaires ?*

Cette question vise à explorer plusieurs dimensions complémentaires de l'apprentissage. Elle ne se limite pas à une évaluation des performances à travers des tests standardisés, mais cherche également à déterminer si le recours à l'IA permet une appropriation plus fine des concepts abordés en programmation web. Parmi ces concepts, on peut citer la manipulation du DOM, les fonctions, les événements ainsi que la gestion des erreurs côté client.

De plus, la question s'intéresse à l'expérience subjective des apprenants : leur niveau de motivation, leur sentiment d'autonomie, leur perception de l'utilité de l'outil mis à disposition, et leur capacité à persévérer face aux difficultés. Ces éléments sont souvent sous-évalués dans les études quantitatives, mais ils jouent un rôle déterminant dans la progression des étudiants, en particulier dans un domaine aussi exigeant que la programmation.

Afin de structurer cette recherche, j'ai formulé plusieurs hypothèses de travail.

- H1** La première hypothèse postule que les étudiants bénéficiant de l'assistance d'un LLM spécialisé verront leurs performances s'améliorer de manière significative entre un pré-test et un post-test, comparativement à ceux n'ayant accès qu'à une documentation classique.
- H2** La seconde hypothèse propose que l'utilisation d'un LLM spécialisé, avec un prompt système configuré pour guider l'apprentissage du développement front-end, favorise une compréhension plus approfondie des concepts fondamentaux, par rapport à un LLM classique. Cette hypothèse se base sur l'idée que l'assistant spécialisé permet une meilleure contextualisation et clarification des concepts clés du JavaScript natif, ce qui facilite l'acquisition de connaissances durables et la mobilisation de ces concepts dans des scénarios de programmation complexes.

H3 La troisième hypothèse suppose que l’usage d’un assistant IA spécialisé renforcera davantage la motivation, le sentiment de compétence et l’engagement des apprenants par rapport à un modèle classique. En effet, un LLM personnalisé offre un accompagnement plus adapté et ciblé, créant ainsi un environnement d’apprentissage plus pertinent, stimulant la curiosité et l’implication des étudiants tout en réduisant leur charge cognitive.

Ces hypothèses ne seront pas abordées de manière isolée, mais dans une logique articulée. Il ne s’agit pas uniquement de mesurer un gain en performance, mais de comprendre les mécanismes par lesquels ce gain peut se produire : nature de l’accompagnement, type d’interactions, évolution des représentations mentales, etc. Le choix de JavaScript comme langage d’étude permet, par ailleurs, de situer l’expérimentation dans un cadre réaliste, en cohérence avec les pratiques pédagogiques actuelles dans l’enseignement supérieur et les attentes du monde professionnel.

En somme, cette recherche vise à interroger la valeur ajoutée réelle d’un assistant IA dans l’apprentissage du développement front-end en JavaScript. Peut-il agir comme un tuteur virtuel pertinent, capable de guider, de corriger, de soutenir sans se substituer à l’apprenant ? Permet-il non seulement de réussir des exercices, mais aussi de mieux comprendre, de s’approprier des concepts abstraits, et de gagner en autonomie ? C’est à ces questions que l’étude cherchera à répondre à travers une méthodologie rigoureuse, expérimentale et comparative, que nous détaillerons dans les sections suivantes.

3.2 Dispositif expérimental

L’objectif de cette expérimentation est d’évaluer l’impact de l’utilisation d’un assistant basé sur un modèle de langage personnalisé par rapport à un assistant utilisant un modèle de langage générique (IA classique) sur l’apprentissage du développement front-end en JavaScript. Ce dispositif expérimental repose sur un schéma à deux groupes. Chaque groupe sera exposé à une forme différente d’assistance pendant la réalisation d’un même exercice de programmation, permettant ainsi de comparer les effets respectifs de chaque modalité d’accompagnement.

3.2.1 Les groupes

L'expérience se déroule avec deux groupes distincts :

- **Groupe expérimental** : Les étudiants de ce groupe auront accès à un assistant conversationnel fondé sur un modèle de langage personnalisé, configuré à l'aide d'un *prompt système*. Un mécanisme qui permet de définir des instructions spécifiques à l'IA afin qu'elle agisse avec un comportement désiré, ici comme un véritable tuteur. Contrairement à un modèle générique, un modèle avec *prompt système* peut fournir un accompagnement contextuel et pédagogique précis, adapté à la tâche d'apprentissage du développement front-end en JavaScript. Il ne se limite pas à fournir des réponses directes aux exercices, mais oriente également l'apprenant, reformule les concepts clés, propose des exemples pratiques et aide à la réflexion sur les erreurs commises.
- **Groupe contrôle** : Les étudiants de ce groupe utiliseront des outils classiques d'apprentissage, tels que les ressources en ligne ou les modèles d'IA public comme ChatGPT ou Le Chat de Mistral. Ils n'auront pas accès à l'assistant personnalisé et devront se débrouiller seuls pour trouver les informations nécessaires à la réalisation de l'exercice. Ce groupe servira de référence pour évaluer l'efficacité de l'assistant personnalisé.

3.2.2 Protocole expérimental

Le protocole expérimental suit une structure en trois phases : le pré-test, l'exercice pratique, et le post-test. Les deux groupes suivront le même processus, mais les modalités d'accompagnement seront distinctes selon le type d'assistance.

Phase 1 : Pré-test

Le pré-test a pour objectif d'établir un niveau de référence concernant les connaissances fondamentales des participants en développement web avec JavaScript. Il est composé de questions à choix multiples (QCM) portant sur des concepts essentiels du JavaScript ainsi que des questions se basant sur l'échelle de Likert. Cette échelle est une méthode d'évaluation permettant de mesurer l'intensité des opinions à l'aide d'une série de réponses proposés sur une gradation ordonnée allant d'un extrême négatif à un extrême positif. Elle permet ainsi de quantifier la perception subjective des apprenants, offrant ici une évaluation de leur motivation et de leur confiance en leurs capacités. Vous pouvez retrouver le contenu exact du pré-test débutant et intermédiaire en Annexe A, p. 62.

Phase 2 : Exercice pratique

Cette phase pratique vise à évaluer les compétences des étudiants en manipulation du DOM, en gestion des événements, ainsi qu'en structuration modulaire du code JavaScript. Deux niveaux de difficulté sont proposés, correspondant au niveau de maîtrise des participants : débutant et intermédiaire. Chaque exercice s'appuie sur un scénario concret de traitement de texte et met l'accent sur l'interactivité via une interface web simple.

Exercice Débutant : Palindrome Dans cet exercice, les participants doivent développer un script JavaScript permettant d'effectuer une vérification simple de palindrome. Le participant doit :

- récupérer la chaîne de caractères saisie dans un champ `<input type="text" id="mot">`
- vérifier si cette chaîne constitue un palindrome, en ignorant la casse et les espaces
- afficher le résultat de la vérification dans un élément `<p id="result">`, après un clic sur le bouton `<button id="check">Vérifier</button>`

Exercice Intermédiaire : Plus long palindrome Cet exercice s'adresse à des étudiants ayant un niveau plus avancé. Il consiste à écrire un script capable d'analyser un texte pour en extraire le plus long sous-mot palindrome, avec un retour visuel dynamique. Le participant doit :

- lire le contenu d'un champ `<textarea id="texte">`
- détecter le plus long sous-mot palindrome présent dans le texte
- afficher à nouveau le texte d'origine dans un élément `<pre id="result">`, en surlignant la portion détectée à l'aide de la balise `<mark>`
- gérer le cas où aucun palindrome d'au moins deux caractères n'est trouvé, en affichant un message d'erreur dans l'élément `#result`
- déclencher l'analyse au clic sur le bouton `<button id="check">Vérifier</button>`

Phase 3 : Post-test

Le post-test est similaire au pré-test, il est constitué d'une première partie QCM, qui vise à évaluer les connaissances acquises dans l'exercice. Puis la deuxième section du post-test est dédiée à l'évaluation du ressenti des apprenants concernant l'expérience d'apprentissage. Les participants sont invités à répondre à un questionnaire sur leur motivation, leur perception de l'utilité de l'assistant, leur sentiment d'autonomie dans

le processus d'apprentissage, et leur niveau de confiance dans leurs capacités après avoir travaillé avec l'une des deux modalités pédagogiques. Ce test est réalisé directement après la phase d'exercice pratique, afin de capter les impressions des participants dans un contexte immédiat. Vous pouvez retrouver le contenu exact du post-test débutant et intermédiaire en Annexe B, p. 65.

3.3 Recueil de données

Dans le cadre de cette recherche, il est prévu d'utiliser deux questionnaires : un questionnaire pré-test, administré avant le début de l'expérimentation, et un questionnaire post-test, proposé à l'issue de la phase de mise en œuvre. Le pré-test a pour objectif principal de recueillir des informations générales sur le profil des participants, leur familiarité avec la programmation et leurs éventuelles expériences passées avec des outils d'intelligence artificielle. Il permettra également d'identifier des différences initiales entre les groupes, afin de mieux interpréter les résultats obtenus par la suite.

Le questionnaire post-test visera à interroger les participants sur leur perception de l'expérience vécue. Il permettra de recueillir leur niveau de satisfaction, leur sentiment d'efficacité, leur compréhension des consignes, ainsi que leur évaluation globale du soutien apporté par les outils d'intelligence artificielle utilisés pendant les séances. En comparant les réponses des deux questionnaires, il sera possible d'observer des évolutions dans les représentations et les ressentis des apprenants, et d'évaluer plus finement l'impact de l'IA sur leur expérience d'apprentissage.

Les questionnaires combineront des données quantitatives et qualitatives. Les questions fermées, construites sur l'échelle de Likert, permettront d'évaluer des dimensions telles que la facilité d'utilisation des outils ou le sentiment de progression. Les résultats des questions feront l'objet d'une analyse distincte dans une section spécifique du mémoire.

Par ailleurs, le questionnaire post-test intégrera également des questions ouvertes. Celles-ci offriront aux participants un espace pour exprimer librement leur ressenti, signaler des éléments positifs ou négatifs de l'expérience, ou encore formuler des suggestions d'amélioration. Ces réponses qualitatives permettront de compléter les résultats statistiques par une lecture plus fine des discours, en apportant un éclairage contextuel sur certains phénomènes observés.

Sur le plan éthique, toutes les mesures seront prises pour garantir un cadre respectueux des droits des participants. Ceux-ci seront informés de manière claire sur les objectifs de l'étude, la nature des données collectées et leur traitement. La participation sera entièrement volontaire. Les données recueillies resteront anonymes autant que possible, et seront enregistrées dans un tableau Excel, accessible uniquement par moi-même.

Le choix de ce protocole provient surtout de sa simplicité de mise en oeuvre. Il permet d'interroger l'ensemble des participants de manière rapide (environ dix minutes par questionnaire) et autonome, tout en recueillant des données exploitables tant sur le plan statistique que sur le plan qualitatif. Cette méthode apparaît donc adaptée à l'objectif de la recherche, tout en respectant les contraintes logistiques.

Cependant, ce dispositif comporte également certaines limites. Les réponses fournies dans les questionnaires reposent sur le ressenti des participants, ce qui peut introduire des biais subjectifs. Les questions ouvertes, bien qu'intéressantes, donnent parfois lieu à des réponses brèves ou peu développées, limitant ainsi la profondeur de l'analyse. Enfin, l'absence d'entretiens individuels ou d'observations directes constitue une limite assumée, liée notamment aux contraintes de temps et de disponibilité dans le cadre de cette étude.

3.4 Analyse des données

L'analyse des données s'effectuera selon une approche mixte, comprenant des méthodes quantitatives et qualitatives. L'objectif est de croiser les résultats objectifs issus des évaluations avec les perceptions subjectives recueillies via les questionnaires, afin d'interpréter de manière nuancée l'impact des dispositifs d'intelligence artificielle sur l'apprentissage de la programmation.

Les données quantitatives, issues notamment des échelles de Likert présentes dans les questionnaires pré-test et post-test, seront traitées statistiquement. L'analyse portera sur la comparaison des moyennes entre les groupes, l'observation de l'évolution des réponses entre le début et la fin de l'expérimentation, ainsi que l'identification de tendances générales (par exemple, un niveau de satisfaction globalement élevé ou au contraire des points de friction récurrents).

Les données qualitatives, quant à elles, proviendront principalement des réponses ouvertes des questionnaires. Une lecture de ces réponses sera menée afin d'en extraire des éléments de discours représentatifs. L'analyse cherchera à mettre en lumière les sentiments dominants, les éventuelles incompréhensions ou résistances, ainsi que les apports perçus par les participants. Des extraits significatifs pourront être utilisés à des fins d'illustration dans ce mémoire.

Enfin, une attention particulière sera portée à l'interprétation des données. L'objectif ne sera pas seulement de quantifier des différences ou de pointer des corrélations, mais aussi de comprendre les effets perçus par les étudiants dans leur diversité, et de mettre ces observations en perspective avec les apports théoriques présentés dans l'état de l'art. Cette analyse constituera ainsi une base solide pour la discussion des résultats et la formulation de pistes d'amélioration ou d'approfondissement futures.

Chapitre 4

Mise en oeuvre

Après avoir défini le cadre méthodologique de l'étude, ce chapitre présente la manière dont le dispositif expérimental a été concrètement mis en œuvre. Il s'agit ici de revenir sur comment le projet a été conçu et mis en place : les choix techniques qui ont guidé le développement de la plateforme, le déroulement pratique des sessions de test, ainsi que les ajustements apportés au fil de l'expérimentation.

4.1 Environnement technique

Au départ, je ne savais pas encore quelle forme prendrait le dispositif. Mon objectif initial était simplement de proposer à des participants un chatbot d'aide à la programmation, et d'évaluer son impact sur leur compréhension à travers un pré-test et un post-test. J'ai donc commencé par explorer des solutions déjà disponibles, en espérant pouvoir combiner plusieurs outils pour créer un parcours cohérent.

J'ai d'abord envisagé d'utiliser Google Forms pour les questionnaires. C'était une solution rapide, mais elle s'est révélée peu adaptée : difficile à personnaliser, peu ergonomique, et totalement déconnectée de la phase d'interaction avec le chatbot. J'ai aussi considéré l'utilisation directe de ChatGPT via son interface officielle. Là encore, des obstacles importants sont vite apparus : nécessité d'un compte, accès payant à la version GPT-4, manque de contrôle sur le contexte des échanges. Enfin, j'ai envisagé une version locale à installer sur les ordinateurs des testeurs, mais cette option nécessitait des compétences techniques que beaucoup de participants n'avaient pas, en plus d'un temps de configuration important.

Progressivement, l'idée de créer une plateforme web dédiée s'est imposée. En tant que développeur, cela me permettait de rester autonome, d'économiser du temps, et d'adapter précisément l'outil aux besoins de l'étude. J'ai donc développé une application en Angular pour le front-end, et Node.js pour le back-end, connectée à une base de données PostgreSQL. Ce choix technologique reflète surtout mes habitudes de travail : ce sont des outils que je connais bien, ce qui m'a permis d'être efficace et réactif en cas de bug ou d'évolution du cahier des charges.

L'élément central de cette plateforme est l'intégration de l'API OpenAI, et plus précisément du modèle GPT-4o. J'ai mis en place deux versions du chatbot : une version standard sans adaptation particulière, et une version « tuteur intelligent » pilotée par un prompt système personnalisé. J'ai passé plusieurs jours à concevoir ce prompt, en l'ajustant progressivement à partir des premiers tests. Mon objectif était clair : ne pas donner la réponse, mais orienter, reformuler, questionner. Il fallait encourager la réflexion tout en évitant la surcharge cognitive. J'ai aussi veillé à adapter le ton et le niveau de langage selon le profil du participant. Vous pouvez retrouver le prompt système final en Annexe C, p. 68.

L'API étant payante, j'ai activé une option proposée par OpenAI pour réduire les coûts : autoriser l'utilisation des données échangées à des fins d'entraînement. Avant cela, j'ai pris soin d'informer clairement les utilisateurs, et de vérifier que les contenus ne comportaient aucun élément sensible ; les échanges portaient exclusivement sur des exercices génériques, sans nom, sans email, sans identifiant.

Cette phase de conception s'est donc construite au fil des besoins et des contraintes. En développant moi-même la plateforme, j'ai pu tester rapidement, corriger, ajuster, et faire évoluer l'outil jusqu'à obtenir une version suffisamment stable pour lancer l'expérimentation à plus grande échelle.

4.2 Déroulé expérimental

Une fois la plateforme développée, j'ai mis en place un parcours structuré en cinq étapes, accessibles directement depuis le navigateur, sans installation ni inscription complexe. L'ensemble du déroulement a été pensé pour guider les participants du début à la fin, de manière fluide, tout en automatisant la collecte des données nécessaires à l'analyse.

Le point de départ est la page d'accueil. Elle présente en quelques lignes l'objectif de l'étude, la durée approximative de la session, et les garanties concernant l'anonymat des données. L'utilisateur entre son email et indique son niveau de programmation (débutant ou intermédiaire), ce qui permet d'adapter les questionnaires et l'exercice, comme vu précédemment.

FIGURE 4.1 – Capture d’écran de la page d’accueil

Bienvenue dans l'étude

Merci de votre intérêt pour cette étude portant sur l'usage de l'intelligence artificielle dans le contexte de l'apprentissage de la programmation.

Aucune compétence en programmation n'est requise et vous n'avez rien à installer : tout se fait en ligne.

Le processus de participation est simple :

1. Inscription et acceptation des conditions. Vous serez assigné à un groupe d'étude. (test ou contrôle, chacun groupe ayant un assistant IA différent)
2. Pre-test (~5min)
3. Exercice avec un assistant IA (~30min)
4. Post-test (~5min)

En participant, vous acceptez les points suivants :

- Votre email est stocké à des fins techniques.
- Les données seront anonymisées autant que possible.
- Les conditions d'utilisations d'**OpenAI** concernant l'utilisation de GPT.

Formulaire d'inscription

Adresse email
mathis.beauville@gmail.com

Niveau
☐ Débutant ☒ Intermédiaire

☒ J'accepte l'utilisation anonymisée de mes données pour l'étude.
☒ J'accepte la transmission de mes interactions à OpenAI.

Commencer l'étude

Vient ensuite le pré-test. Il se compose d'un questionnaire à choix multiples et de quelques questions ouvertes. L'objectif est de recueillir une première évaluation des connaissances du participant. Les questions varient selon le niveau déclaré à l'inscription.

FIGURE 4.2 – Capture d’écran de la page de questionnaire pre-test (niveau intermédiaire)

1 Pre-test 2 Exercice 3 Post-test

Ce pré-test est conçu pour évaluer vos connaissances initiales sur les sujets abordés dans cette étude. Il permettra également d'évaluer votre évolution au cours de l'étude.

Les questions marquées d'un astérisque (*) sont obligatoires.

Quelle méthode JavaScript retourne un nouveau tableau contenant uniquement les éléments qui passent un test ? *

☐ forEach()
☐ filter()
☐ map()
☐ reduce()

Quel appel de fonction fléchée retourne correctement le carré d'un nombre ? *

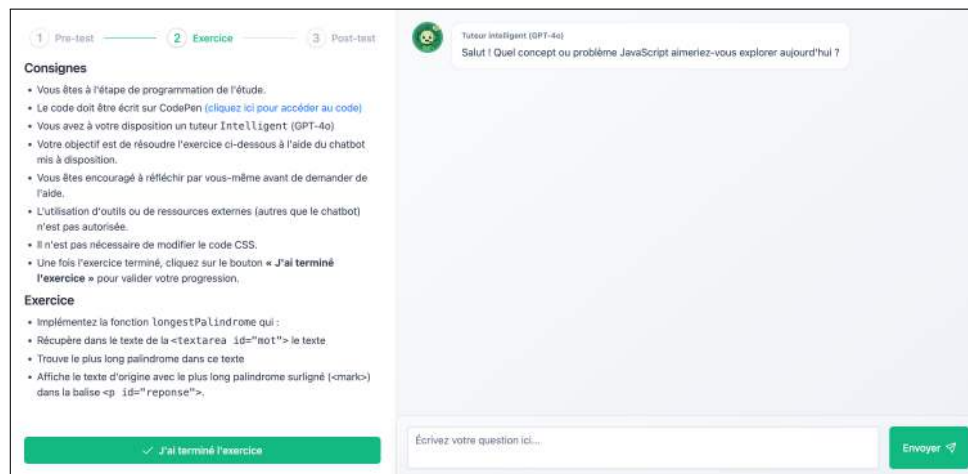
☐ (x) => { x * x; }
☐ (x) => x * x
☐ x => { return x ** 2; }
☐ (x) => return x * x

Quelle expression vérifie que la variable x est strictement un tableau ? *

☐ Array.isArray(x)
☐ typeof x === "array"
☐ x instanceof Object

La troisième étape est celle de l'exercice. L'utilisateur découvre un énoncé simple, adapté à son niveau, qu'il doit résoudre à l'aide d'un chatbot intégré dans la page. Ce chatbot peut correspondre, selon un tirage aléatoire, soit à une version « classique » du modèle GPT-4o, soit à une version personnalisée avec le prompt système. L'utilisateur peut interagir librement avec l'IA, poser des questions, demander des explications, ou simplement tester des idées.

FIGURE 4.3 – Capture d’écran de la page de l’exercice et du chatbot (niveau intermédiaire)



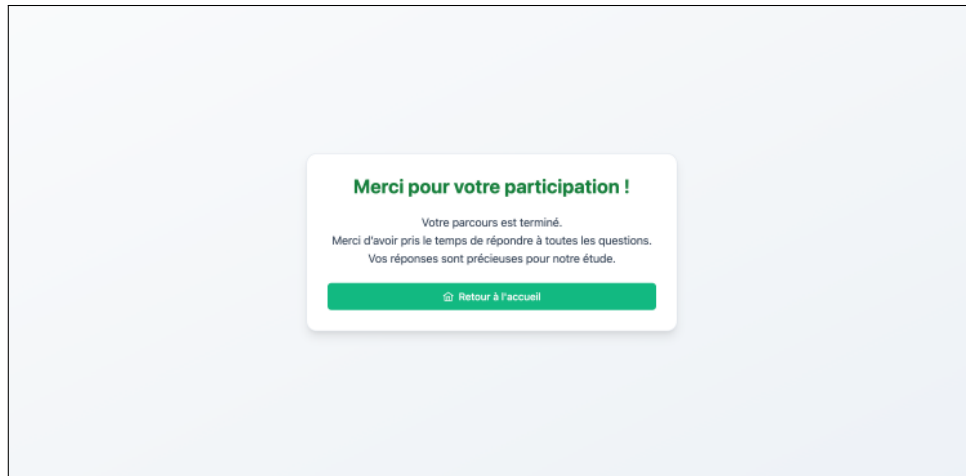
Après l’exercice, le participant est invité à remplir un post-test. Il s’agit d’un second questionnaire similaire au premier, avec quelques variantes, permettant de mesurer l’évolution des connaissances. Il comporte aussi une série de questions sur le ressenti de l’utilisateur, la difficulté perçue et la pertinence de l’aide reçue.

FIGURE 4.4 – Capture d’écran de la page de questionnaire post-test (niveau intermédiaire)



Enfin, la cinquième et dernière page affiche un message de remerciement.

FIGURE 4.5 – Capture d’écran de la page de remerciement



Une version publique du site est toujours en ligne, à titre illustratif. Vous pouvez la retrouver sur le lien suivant : <https://memoire.obstinate.fr>.

4.3 Ajustements et itérations

Avant d’ouvrir la plateforme à tous les participants, j’ai pris le temps de faire une première série de tests auprès de mon entourage. L’objectif était simple : identifier rapidement les bugs évidents, m’assurer que l’enchaînement des étapes était fluide, et surtout, observer comment le chatbot se comportait dans des situations réelles. Ce type de retour est difficile à anticiper uniquement en développant seul dans son coin.

Assez vite, plusieurs problèmes ont été signalés. Un bug récurrent empêchait, dans certains cas, d’envoyer un second message au chatbot sans recharger la page. Après vérification, il s’agissait d’une mauvaise gestion de l’état du composant de chat, côté client. J’ai corrigé le problème en revoyant le système d’envoi des requêtes et la manière dont les réponses étaient attendues. D’autres retours portaient sur l’absence de certaines questions dans le post-test : en réalité, les questions ouvertes n’avaient pas été importées correctement pour tous les niveaux. Ce point a été corrigé en revoyant la structure du schéma associé. J’en ai profité pour faire quelques ajustements visuels, notamment pour améliorer la lisibilité sur petits écrans (mobilité, contraste, espacements).

Mais les retours les plus intéressants concernaient le comportement du chatbot. C'est là que les choses sont devenues vraiment utiles. Certains utilisateurs trouvaient le bot trop bavard, d'autres trouvaient au contraire qu'il ne répondait pas assez précisément. Quelques-uns ont signalé qu'il proposait directement la solution, ou bien qu'il se contentait de reformuler sans rien apporter. Ces variations m'ont amené à revoir en profondeur certaines parties du prompt personnalisé. J'ai supprimé des consignes inutiles, réécrit certaines phrases pour qu'elles soient plus simples à interpréter par le modèle, et ajouté des mécanismes de relance automatique en cas de silence. Par exemple, si l'utilisateur restait bloqué trop longtemps ou posait une question floue, le bot était invité à proposer une clarification ou une question intermédiaire.

Pour guider ces modifications, je me suis appuyé sur les logs d'interaction que j'avais mis en place dès le départ. Chaque message, chaque réponse était enregistré. Cela m'a permis de suivre le déroulement d'un échange, de repérer les moments de rupture (quand un utilisateur abandonne ou ne répond plus), et d'en tirer des pistes d'amélioration concrètes. J'ai aussi pu repérer des formulations mal interprétées, ou des moments où le bot faisait preuve de trop de prudence.

En parallèle, je voulais m'assurer que la version « classique » du chatbot se comportait de manière stable. J'ai donc vérifié ses réponses sur plusieurs parcours, pour m'assurer qu'elle n'introduisait pas de biais majeur. L'idée n'était pas d'obtenir un modèle parfait, mais de garantir que les deux conditions (classique et tutorée) soient comparables dans leur fonctionnement général.

Cette phase d'ajustement a duré quelques semaines, mais elle a joué un rôle central. Elle m'a permis de stabiliser l'outil, de corriger les principaux défauts, et surtout, de renforcer la qualité de l'interaction entre l'utilisateur et l'IA. Ce n'était pas prévu initialement, mais je me suis vite rendu compte que sans ce travail de retouche, les données recueillies n'auraient sans doute pas eu la même fiabilité. C'est aussi à ce moment-là que certaines limites techniques sont devenues visibles.

Chapitre 5

Résultats et analyse

5.1 Résultats quantitatifs

L'expérimentation a réuni un total de 17 participants ayant terminé complètement l'expérience, répartis à environ 50/50 entre les deux groupes : le groupe « Test » utilisant une IA spécialisée pour l'apprentissage, et le groupe « Contrôle » utilisant une IA généraliste.

Pour pouvoir comparer équitablement les résultats, les scores ont été ramenés à une échelle commune sur 10. Cette normalisation était nécessaire car les questionnaires n'étaient pas strictement identiques selon le niveau déclaré par les participants : les débutants avaient cinq questions, tandis que les intermédiaires en avaient six. Ce choix permet d'analyser les progressions individuelles et collectives sans que la différence de barème n'introduise de biais.

TABLE 5.1 – Moyennes (M) et écarts-types (σ) pré-test et post-test (notes normalisées sur 10) par groupe et niveau, avec agrégats “Tous niveaux”

Groupe	Niveau	Pré-test M ($\pm \sigma$)	Post-test M ($\pm \sigma$)
Test	Débutant	4,40 ($\pm 0,84$)	6,40 ($\pm 0,45$)
Test	Intermédiaire	6,00 ($\pm 0,82$)	9,50 ($\pm 0,50$)
Contrôle	Débutant	6,00 ($\pm 0,82$)	7,00 ($\pm 0,58$)
Contrôle	Intermédiaire	7,00 ($\pm 1,29$)	8,00 ($\pm 0,82$)
Test	Tous niveaux	5,20 ($\pm 0,88$)	7,95 ($\pm 0,93$)
Contrôle	Tous niveaux	6,50 ($\pm 1,04$)	7,50 ($\pm 0,71$)

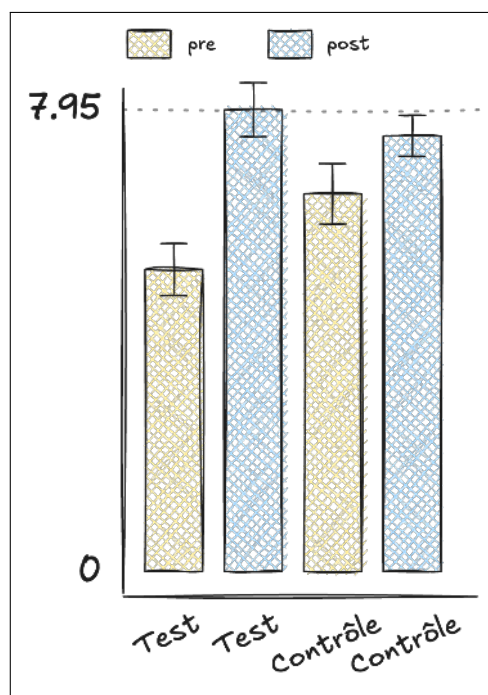
Les résultats montrent une progression dans tous les cas, mais cette progression est beaucoup plus nette dans le groupe Test.

Chez les débutants, le groupe ayant utilisé l'IA spécialisée passe d'un score moyen de 4,40 à 6,40, soit un gain de +2,00 points. Le groupe Contrôle progresse plus modestement, de 6,00 à 7,00 (+1,00 point). Il est intéressant de noter que les débutants du groupe Test, qui partaient initialement avec un score plus faible, ont quasiment rattrapé ceux du groupe Contrôle. Cela suggère que l'IA spécialisée a joué un rôle de soutien renforcé dans les premières étapes d'appropriation des concepts.

Chez les intermédiaires, la tendance est encore plus marquée. Le groupe Test passe de 6,00 à 9,50 (+3,50), soit une amélioration très significative. Le groupe Contrôle, lui, ne progresse que d'un point (de 7,00 à 8,00). Ici encore, l'effet du prompt pédagogique semble avoir permis non seulement de combler des lacunes, mais aussi de structurer davantage la compréhension.

Si l'on regroupe l'ensemble des niveaux, le groupe Test montre une progression moyenne de +2,75 points (de 5,20 à 7,95), contre seulement +1,00 point dans le groupe Contrôle (de 6,50 à 7,50). Ce résultat confirme que l'IA spécialisée n'a pas seulement été « utile » : elle a transformé l'expérience d'apprentissage d'une partie des participants.

FIGURE 5.1 – Comparaison des scores pré- et post-test par groupe et par niveau



Afin de compléter ces premiers résultats sur l'efficacité comparative des deux types d'IA, il est pertinent d'examiner les réponses des participants aux questions évaluées par une échelle de Likert, présentées dans le tableau suivant.

TABLE 5.2 – Scores moyens (sur 5) aux questions Likert, selon le groupe (débutant vs. intermédiaire) et le niveau (pré-test vs. post-test). Les questions sont listées ci-dessous.

Q°	Groupe Contrôle				Groupe Test			
	Déb. Pré	Déb. Post	Int. Pré	Int. Post	Déb. Pré	Déb. Post	Int. Pré	Int. Post
1	2,4	2,9	4,1	4,3	2,2	3,8	4,5	4,4
2	2,6	3,0	4,3	4,5	2,5	4,1	4,7	4,6
3	2,3	2,8	3,9	4,1	2,4	3,9	4,6	4,5
4	1,9	2,4	3,2	3,5	1,8	3,2	3,8	3,6
5	2,7	3,2	3,8	4,0	2,6	3,7	4,9	4,8

Légende des questions :

1. Écrire une fonction JS correcte sans IA
2. Structurer seul son code (décomposition, logique, syntaxe)
3. Corriger une erreur via message d’erreur / comportement
4. Maîtrise du DOM en JS sans chercher
5. Reproduire seul un exercice similaire (palindrome + affichage)

Les réponses aux questions évaluées sur l’échelle de Likert permettent d’affiner la comparaison entre les deux groupes. On y voit plus précisément sur quels aspects les participants ont le plus progressé, et dans quelle mesure l’IA spécialisée a pu faire la différence.

Commençons par les débutants du groupe Test. En moyenne, ils gagnent +1,4 point par question, avec des hausses particulièrement nettes sur la structuration du code, la correction d’erreurs ou encore la capacité à reproduire un exercice. Cela montre que l’IA spécialisée les a aidé à mieux comprendre les bases de la programmation, à dépasser certains blocages, et à progresser de manière assez régulière sur l’ensemble des compétences visées.

Chez les intermédiaires du même groupe, les résultats évoluent très peu. Les scores étaient déjà élevés dès le départ, et les écarts entre le pré-test et le post-test sont quasiment nuls, voire légèrement en baisse. Cela ne signifie pas que l’outil a été inutile, mais plutôt qu’il a permis de maintenir un bon niveau, sans forcément créer de rupture dans les apprentissages.

Du côté du groupe Contrôle, la progression est plus modeste. Les débutants gagnent en moyenne un demi-point par question, ce qui reste une amélioration, mais moins marquée que dans le groupe Test. Les compétences les plus techniques, comme la manipulation du DOM, restent difficiles à acquérir sans accompagnement ciblé.

Enfin, chez les intermédiaires du groupe Contrôle, les écarts sont très faibles (autour de +0,2 point) ce qui suggère que l'IA généraliste n'a pas eu d'impact significatif sur leur progression. Ils ont peut-être consolidé quelques acquis, mais sans transformation notable.

Un dernier point mérite d'être souligné : la question sur la maîtrise du DOM (Q4) est celle qui obtient les scores les plus bas, tous groupes confondus. Cela semble indiquer que cette compétence reste difficile, quel que soit l'outil utilisé, et qu'elle demande sans doute plus de temps ou d'expériences concrètes pour être pleinement intégrée.

En résumé, ces résultats confirment ce qu'on avait déjà observé avec les scores globaux : l'IA spécialisée semble particulièrement utile pour les débutants, en les aidant à surmonter des difficultés de base, là où l'IA généraliste a un effet plus limité.

5.2 Analyse qualitative

Les réponses aux questions ouvertes ont apporté un regard complémentaire aux données chiffrées. Sur les 17 participants, une bonne partie a pris le temps de formuler un retour, qu'il s'agisse d'un simple commentaire ou d'une remarque plus développée. Ces réponses offrent un aperçu direct de la manière dont les participants ont vécu l'expérience, et révèlent des tendances intéressantes.

Le thème le plus récurrent est celui de la compréhension qui est facilitée. Plusieurs personnes ont souligné que l'IA les avait aidées à mieux saisir les différentes étapes d'un raisonnement en JavaScript. Une réponse l'illustre de manière très claire : « *L'IA m'a bien aidé, j'ai mieux compris les étapes pour écrire le code.* » Cette idée de cheminement rendu plus clair revient souvent, et semble être l'un des apports les plus appréciés de l'outil.

Certains participants ont également formulé des suggestions concrètes pour améliorer le dispositif. Deux propositions sont revenues : intégrer des mini quiz après chaque explication, et proposer un exercice intermédiaire avant la solution finale. L'un des retours note ainsi qu'« *un mini quiz entre chaque explication serait top pour bien assimiler* », tandis qu'un autre propose « *[...] un exercice intermédiaire pour mieux comprendre les étapes.* » Ces remarques suggèrent que les participants attendaient un accompagnement plus progressif, alternant explications et mise en pratique.

Cela dit, tout n'a pas été perçu positivement. Quelques retours pointent une certaine rigidité de l'IA dans ses réponses. Certains ont noté qu'elle pouvait insister sur des étapes déjà comprises ou donner l'impression de ne pas écouter : « *Parfois j'avais l'impression de tourner en rond, je donnais des détails mais l'IA me redemandait la même chose.* » D'autres ont exprimé une frustration face à un manque de réponse directe : « *Elle me disait de décomposer alors que c'était déjà fait, ça ralentissait au lieu d'aider.* » Ces critiques soulignent un enjeu clé de ce type d'outil : trouver le bon équilibre entre guidage pédagogique et adaptation fine au niveau réel de l'apprenant.

Dans l'ensemble, les retours montrent que l'outil a été bien accueilli, mais aussi que son intégration pédagogique pourrait être affinée. Les utilisateurs ont exprimé des attentes claires en matière de rythme, d'interactivité et de personnalisation.

5.3 Discussion des résultats

Les résultats obtenus permettent de mieux cerner comment les intelligences artificielles, et en particulier les chatbots pédagogiques, peuvent influencer l'apprentissage de la programmation web. Si l'on compare les deux types d'IA testés, une tendance claire se dégage : les participants accompagnés par l'outil spécialisé ont, dans l'ensemble, davantage progressé. Et ce constat vaut aussi bien pour les scores chiffrés que pour les ressentis exprimés dans les questionnaires.

Du côté des données quantitatives, la différence est nette. Le groupe ayant utilisé l'IA spécialisée passe d'un score moyen de 5,2 à 7,95 (sur 10), soit une progression de +2,75 points. À l'inverse, le groupe contrôle, qui utilisait une IA plus classique, progresse de 6,5 à 7,5, soit un gain plus modeste (+1 point). Ces écarts sont particulièrement parlants chez les débutants, qui semblent avoir tiré un bénéfice plus marqué de l'accompagnement spécialisé. Leur progression moyenne est deux fois plus importante que celle du groupe contrôle, ce qui suggère que l'IA spécialisée a joué un rôle de soutien dans la compréhension des bases du JavaScript.

Les résultats issus des questions Likert viennent confirmer cette impression. Chez les débutants du groupe Test, les gains sont visibles sur l'ensemble des ressentis évalués : mieux structurer son code, corriger ses erreurs, comprendre le DOM, ou encore réussir à reproduire un exercice similaire. Le score moyen progresse de +1,4 point par question, avec des pics à +1,6 pour la structuration du code, et +1,5 pour la correction d'erreurs. C'est un signal fort : l'outil n'a pas seulement amélioré les résultats, il a aussi consolidé les mécanismes d'apprentissage fondamentaux.

À l'inverse, dans le groupe Contrôle, les évolutions sont plus faibles. Les débutants progressent un peu, mais les écarts restent limités (+0,5 point en moyenne), et les intermédiaires stagnent presque. Cela semble indiquer que l'IA généraliste, si elle reste utile pour répondre à des questions ponctuelles, n'a pas vraiment soutenu un apprentissage structuré dans le temps.

Au-delà des chiffres, ce sont les retours qualitatifs qui éclairent le plus le rôle joué par l'IA. Plusieurs participants du groupe Test ont décrit une expérience plus guidée, où ils se sentaient accompagnés dans leur raisonnement. Certains ont dit avoir mieux compris les étapes, ou réussi à organiser leurs idées, ce qui rejoint l'objectif initial du prompt pédagogique : aider à réfléchir, et pas seulement fournir des réponses toutes faites.

Mais tout n'a pas été parfait. Quelques participants ont aussi signalé des moments de frustration, notamment quand l'IA semblait tourner en rond ou ne pas répondre clairement à la question posée. D'autres auraient aimé plus d'interactivité : des quiz après les explications, des mini-exercices en cours de route, ou encore des retours plus personnalisés. Ces critiques montrent que, même spécialisée, une IA reste un outil limité, qui ne peut pas toujours s'adapter avec finesse aux attentes de chacun.

En résumé, cette expérimentation confirme qu’une IA bien conçue peut aider à apprendre à programmer, surtout quand elle agit comme un accompagnateur. Mais elle ne remplace pas une démarche pédagogique globale, ni l’importance du tâtonnement, de l’erreur, et de la réflexion personnelle. L’équilibre entre aide et autonomie reste fragile, et mérite d’être pensé avec soin si l’on veut vraiment intégrer ces outils dans une logique d’apprentissage durable.

Chapitre 6

Discussion approfondie et limites

6.1 Contributions

Cette expérimentation, bien que modeste par sa taille, apporte plusieurs éléments intéressants à la réflexion sur l’usage de l’intelligence artificielle dans l’apprentissage de la programmation. Elle montre qu’une IA spécialisée, pensée pour guider et accompagner l’apprenant, peut jouer un rôle actif dans la construction des compétences. Contrairement à une IA généraliste qui donne souvent des réponses efficaces mais génériques, la version spécialisée utilisée ici a été perçue comme plus pédagogique : elle incitait à reformuler, à structurer ses idées ou à décomposer un problème. Plusieurs participants ont exprimé avoir mieux compris certaines notions grâce à cette forme d’accompagnement, qui ressemblait plus à une aide méthodologique qu’à une simple solution automatisée.

L’un des apports majeurs de cette étude est donc de souligner l’importance de la posture de l’IA dans un contexte d’apprentissage. Ce n’est pas uniquement le contenu de la réponse qui compte, mais aussi la manière dont elle est formulée. Lorsqu’une IA propose des pistes de réflexion, relance l’utilisateur avec des questions, ou invite à explorer plusieurs approches, elle soutient un apprentissage actif sans se substituer à l’effort cognitif. Ce rôle de tuteur numérique ouvre des perspectives intéressantes pour le développement de futurs outils éducatifs fondés sur l’IA.

Les résultats chiffrés, notamment ceux issus des questions à échelle de Likert, vont dans ce sens. Chez les débutants accompagnés par l’IA spécialisée, la progression moyenne par compétence dépasse +1,4 point, avec des gains nets sur la structuration du code et la correction d’erreurs. À l’inverse, les participants utilisant une IA généraliste ont progressé plus modestement, en particulier les intermédiaires, dont les scores sont restés presque stables. Ces écarts suggèrent que l’IA spécialisée a eu un impact plus significatif, notamment dans les phases d’appropriation des bases du langage JavaScript.

Cette expérimentation met aussi en lumière un point souvent négligé : l'importance de l'encadrement pédagogique autour de ces outils. Plusieurs participants ont proposé des améliorations pour rendre l'outil plus interactif ou plus fluide, en intégrant par exemple de petits quiz, des exemples plus progressifs, ou une meilleure gestion du dialogue. Cela confirme que l'efficacité d'une IA ne dépend pas seulement de ses algorithmes, mais aussi de la façon dont elle est intégrée dans un parcours d'apprentissage cohérent.

Enfin, ce mémoire m'a permis, à titre personnel, de prendre du recul sur mes propres usages de l'intelligence artificielle. En tant qu'apprenant et développeur, j'ai pu confronter mes intuitions à des retours concrets, et approfondir ma compréhension des enjeux liés à l'IA en éducation. Cette expérience a renforcé ma conviction qu'il ne suffit pas d'ajouter une IA à un dispositif pédagogique pour le rendre efficace : c'est la manière dont on conçoit son rôle, sa place, et ses limites, qui en détermine réellement l'impact.

6.2 Limites

Comme toute expérimentation, celle menée dans ce mémoire comporte plusieurs limites qu'il est important de reconnaître, autant pour en relativiser les résultats que pour ouvrir des pistes d'amélioration.

La première limite tient à la taille de l'échantillon. Avec seulement 17 participants, les résultats restent à interpréter avec prudence. Certaines tendances sont nettes, notamment l'impact positif de l'IA spécialisée, mais elles mériteraient d'être confirmées sur un plus grand nombre d'apprenants, avec une répartition plus homogène entre les niveaux et les profils. Par ailleurs, certaines données plus fines, comme les réponses aux questions Likert, ont montré des évolutions intéressantes, mais parfois très variables selon les profils. Par exemple, chez les intermédiaires, les scores ont peu évolué, voire légèrement baissé dans le groupe Test. Ces nuances auraient mérité d'être explorées sur un échantillon plus large pour mieux comprendre ce qui relève d'une tendance réelle ou d'un simple effet de contexte.

La deuxième limite concerne la durée de l'expérimentation. L'usage de l'IA a été testé dans le cadre d'un exercice unique, en une seule session. Cela permet de mesurer un effet immédiat, mais pas d'évaluer des apprentissages durables ou une progression sur le long terme. Or, apprendre à programmer est un processus lent, qui se construit dans le temps. Il serait intéressant de voir comment les perceptions et les résultats évolueraient si l'outil était utilisé régulièrement sur plusieurs semaines ou dans le cadre d'un cours entier.

Une autre limite est liée à la nature des IA utilisées. Même si l'IA spécialisée a été configurée avec soin à l'aide d'un prompt orienté vers l'accompagnement, elle reste basée sur un modèle généraliste (ChatGPT), qui peut parfois manquer de précision ou interpréter maladroitement certaines demandes. Cela s'est d'ailleurs reflété dans certains retours d'utilisateurs, qui ont exprimé leur frustration face à des relances jugées trop rigides ou des incompréhensions de l'outil. Cela peut notamment expliquer pourquoi certains scores individuels ont stagné, malgré une configuration pensée pour encourager la progression. Même un prompt bien conçu ne garantit pas une expérience toujours fluide ou pertinente pour chaque utilisateur.

Sur le plan éthique et réglementaire, une attention particulière a été portée au respect des données personnelles. Conformément aux principes du RGPD, aucune donnée nominative (non strictement nécessaire) n'a été collectée, et les réponses ont été anonymisées avant analyse. Cela dit, le simple fait d'utiliser un outil basé sur une IA externe (comme ChatGPT) soulève des questions quant à la confidentialité des échanges et à la localisation des données traitées. Même si dans ce cas précis les interactions étaient limitées et anonymes, une réflexion plus approfondie devrait être menée si l'outil venait à être intégré dans un cadre institutionnel.

Enfin, il faut souligner que cette étude n'a pas comparé l'usage de l'IA avec un enseignement humain classique. Les résultats montrent une différence entre deux types d'IA, mais ne permettent pas de trancher sur la pertinence de l'IA face à un enseignant ou un pair. L'IA ne remplace pas un cours, elle peut au mieux l'accompagner. Cette nuance est importante pour éviter toute surinterprétation des résultats.

Ces limites n'invalident pas les observations faites, mais elles invitent à les replacer dans leur contexte. Elles soulignent aussi la nécessité de poursuivre les recherches dans ce domaine, avec des protocoles plus étendus, plus longs, et plus variés.

6.3 Perspectives

Ce mémoire ouvre plusieurs pistes intéressantes pour la suite, que ce soit sur le plan de la recherche, de la pédagogie ou du développement technique.

Du point de vue méthodologique, il serait pertinent d'envisager une version étendue de l'expérimentation, avec un plus grand nombre de participants, sur une durée plus longue, et intégrée à un véritable parcours pédagogique. L'idée serait par exemple de proposer un accompagnement par IA tout au long d'un module d'apprentissage (quelques semaines), afin de mieux observer les évolutions dans le temps, les effets sur la motivation, ou encore la manière dont les apprenants changent d'approche face à la résolution de problèmes. Cela permettrait aussi d'évaluer plus finement l'impact de l'IA sur la rétention des connaissances. Par exemple, on pourrait analyser l'évolution des compétences spécifiques (comme la structuration du code ou la compréhension du DOM), en suivant les scores sur des échelles de type Likert tout au long du module. Cela permettrait d'observer si les progressions observées en une seule session se confirment ou s'amplifient avec le temps.

Sur le plan pédagogique, plusieurs éléments remontés par les utilisateurs mériteraient d'être approfondis. L'idée d'ajouter des quiz intermédiaires ou des exercices progressifs pourrait donner naissance à une version enrichie de l'IA, pensée comme un tuteur interactif complet, capable de proposer des mini-évaluations au fil de l'échange. Il serait également intéressant d'explorer différentes postures d'IA : plus ou moins directive, plus ou moins bavarde, plus ou moins bienveillante... afin de s'adapter à différents profils d'apprenants. Les résultats ont montré que certains profils, comme les débutants, bénéficient clairement d'un accompagnement plus guidé, tandis que les intermédiaires semblent parfois freiné par un style trop directif ou redondant. Tester différentes postures pourrait aider à optimiser l'équilibre entre aide et autonomie, en fonction du niveau et des attentes de chacun.

Du côté technique, on pourrait envisager de sortir d'un simple prompt sur un modèle pré-entraîné pour aller vers une interface dédiée, dotée de mémoire contextuelle (entre les sessions par exemple), d'une logique pédagogique, et d'un suivi de la progression. À terme, l'IA pourrait même adapter ses explications en fonction du profil de l'utilisateur, de ses erreurs passées, ou de ses préférences d'apprentissage. Cela suppose évidemment une collaboration étroite entre pédagogues et développeurs, pour créer des outils véritablement utiles.

Enfin, sur le plan institutionnel, cette expérimentation montre qu'il est possible d'introduire des IA génératives dans un cadre éducatif sans forcément tomber dans la triche ou la dépendance. Mais pour que cela fonctionne, il faut encadrer les usages, former les étudiants à s'en servir de manière critique, et surtout, continuer à valoriser l'effort, la compréhension et la construction du raisonnement. L'IA ne fera pas tout, mais elle peut aider. A condition que son usage soit raisonné.

Ces perspectives ne sont pas des conclusions définitives, mais des ouvertures possibles. Elles prolongent les réflexions engagées dans ce travail et dessinent les contours d'une pédagogie augmentée, où l'intelligence humaine et l'intelligence artificielle peuvent cohabiter de manière complémentaire.

Chapitre 7

Conclusion

7.1 Synthèse des résultats

Quand j’ai commencé ce mémoire, je ne savais pas exactement ce que j’allais trouver. Je savais que l’IA avait changé ma façon d’apprendre à coder, et je voyais bien que je n’étais pas le seul dans ce cas. Mais je ne savais pas si cette impression allait se confirmer, ni dans quelle mesure une IA pouvait réellement aider quelqu’un à mieux apprendre. En menant cette expérimentation, j’ai cherché à aller au-delà de mes intuitions.

D’abord, les chiffres parlent d’eux-mêmes. Avec un échantillon modeste (17 participants), on observe tout de même une amélioration marquée pour ceux qui ont utilisé l’IA spécialisée. En replaçant les scores sur 10 pour pouvoir comparer tous les niveaux, on voit un gain moyen de près de +2,75 points pour ce groupe, contre +1 point dans le groupe “ChatGPT classique”. Ce n’est pas une révolution, mais c’est un signal clair : une IA conçue pour accompagner l’apprentissage fait mieux que celle pensée pour répondre à tout.

Mais ce sont surtout les retours qualitatifs qui m’ont marqué. Certains participants ont parlé d’un « déclic », du fait de mieux comprendre les étapes, ou d’être guidés dans leur raisonnement. C’est exactement ce que j’espérais tester. D’autres ont proposé des idées très concrètes, comme des mini quiz ou des exercices intermédiaires, pour rendre l’expérience encore plus utile. Et puis il y a eu les critiques, celles qui disent que parfois l’IA ne comprend pas bien la question, ou relance pour rien. Ces limites sont précieuses, car elles montrent ce qu’on peut encore améliorer.

En résumé, cette expérimentation m’a permis de valider une idée simple : l’IA peut être un bon outil pour apprendre, mais seulement si elle est pensée comme telle. Pas comme un moteur de réponse, mais comme un compagnon d’apprentissage, capable de guider sans faire à la place.

7.2 Réponse à la problématique

La question qui m’a guidé tout au long de ce mémoire était simple : est-ce qu’une intelligence artificielle, comme un modèle de langage, peut vraiment améliorer l’apprentissage de la programmation ? Après avoir mené cette expérimentation, analysé les résultats et surtout lu les retours des participants, je pense pouvoir répondre : oui, mais pas n’importe comment.

Ce que montre cette étude, c’est que les LLMs ne font pas de miracles par eux-mêmes. Une IA comme ChatGPT, utilisée sans adaptation, peut certes aider ponctuellement, mais elle reste limitée si elle n’est pas guidée ou cadrée dans un cadre pédagogique clair. L’efficacité réelle ne vient pas seulement de la qualité des réponses, mais de la manière dont l’IA engage l’utilisateur à réfléchir, à structurer son raisonnement, à chercher par lui-même.

C’est là que l’IA spécialisée tire son épingle du jeu. Ce n’est pas qu’elle “sait” mieux. C’est qu’elle parle différemment. Elle relance, questionne, insiste parfois là où l’on voudrait juste une solution. Et c’est précisément dans ces moments-là que l’apprentissage se construit. Parce qu’on ne se contente plus de lire une réponse, on est obligé de formuler, de clarifier, de se confronter à son propre raisonnement.

En fin de compte, ce n’est pas tant une question de performance technique mais plutôt de posture pédagogique. Ce qu’il faut, ce n’est pas une IA plus puissante, mais une IA qui sache se mettre dans le rôle d’un guide. C’est cette approche que j’ai voulu tester, et qui semble avoir porté ses fruits.

7.3 Recommandations

Si je devais tirer quelques recommandations concrètes de ce travail, elles s’adresseraient à trois publics : les enseignants, les concepteurs d’outils, et les étudiants eux-mêmes. Car faire entrer l’intelligence artificielle dans l’apprentissage de la programmation ne concerne pas seulement la technologie : c’est un changement de posture, de méthode, et parfois même de culture.

Pour les enseignants, la première chose serait de ne pas voir l'IA comme une menace, ni comme une solution magique. Elle peut devenir un allié précieux si elle est pensée comme un tuteur complémentaire, un outil parmi d'autres, capable d'encourager, de débloquer, ou de stimuler la réflexion. Mais elle ne remplace pas l'échange, l'accompagnement humain, ni la capacité à s'adapter à un élève en difficulté. L'enjeu, ce n'est pas de tout déléguer à l'IA, mais de savoir quand et comment l'intégrer dans un parcours pédagogique cohérent.

Pour les concepteurs d'outils éducatifs, cette expérimentation montre qu'il est possible, et sans doute nécessaire, de repenser la manière dont les IA interagissent avec les apprenants. Une IA trop rigide ou trop directive perd rapidement son intérêt. À l'inverse, une IA qui relance, qui s'adapte au niveau, qui comprend le contexte de l'exercice, peut devenir un véritable guide d'apprentissage. Cela suppose de sortir du simple modèle question/réponse, pour aller vers des systèmes plus interactifs, évolutifs, capables de construire une relation pédagogique, même minimale.

Enfin, pour les étudiants, je recommanderais d'utiliser ces outils comme des déclencheurs de réflexion. Pas comme des moteurs de production, ni comme des générateurs de réponse à copier/coller. L'IA peut poser les bonnes questions, proposer une autre manière de voir, mais elle ne remplacera jamais l'effort de compréhension. Elle est là pour nous accompagner, pas pour faire à notre place. C'est une béquille utile mais il faut savoir quand la poser pour marcher seul.

7.4 Ouverture

Le 29 juillet 2025, OpenAI a officiellement lancé un nouveau mode pour ChatGPT : le *GPT Study Mode* [12]. Cette version ne cherche plus à donner la bonne réponse immédiatement, mais à accompagner l'utilisateur dans son raisonnement. L'idée est de guider, de questionner, parfois de ralentir l'élève pour l'aider à comprendre plutôt que de simplement exécuter. Autrement dit, exactement ce que j'ai essayé de tester dans ce mémoire.

Voir émerger ce type d'initiative au moment même où je terminais ce travail a été à la fois rassurant et stimulant. Cela montre que l'intuition de départ d'une IA éducative qui n'est pas seulement une machine à réponses, mais un partenaire d'apprentissage n'est pas isolée. Elle s'inscrit dans un mouvement plus large, porté à la fois par la recherche, les pratiques de terrain, et désormais les grandes entreprises du secteur.

Je ne prétends pas avoir fait le tour de la question avec ce mémoire. Il reste encore beaucoup à explorer, à tester, à comprendre. Mais j'ai essayé, à mon niveau, d'apporter une pierre à l'édifice. Ce que je retiens surtout, c'est qu'une IA, aussi impressionnante soit-elle, ne vaut pas grand-chose sur le plan pédagogique si elle n'est pas pensée pour aider à apprendre. Il faut une intention derrière, une posture, un cadre. Et ce cadre, on doit continuer à le construire, ensemble : enseignants, développeur et étudiants. Parce que l'avenir de l'apprentissage avec l'IA ne se jouera pas uniquement dans les performances des modèles, mais dans la manière dont on les utilise.

Bibliographie

- [1] ARISTOTE EDUCATION. *Aristote - AI Tutor for Education*. Accessed : 2025-02-12. 2025. URL : <https://chat.aristote.education/>.
- [2] Samuel BOGUSLAWSKI, Rowan DEER et Mark G. DAWNSON. « Programming education and learner motivation in the age of generative AI : student and educator perspectives ». In : (juill. 2024). ISSN : 2398-5348. URL : <https://www.emerald.com/insight/content/doi/10.1108/ils-10-2023-0163/full/html>.
- [3] Ma BOXUAN, Li CHEN et Shin'ichi KONOMI. *Enhancing Programming Education with ChatGPT : A Case Study on Student Perceptions and Interactions in a Python Course*. Avr. 2024. URL : <https://ar5iv.labs.arxiv.org/html/2403.15472v2>.
- [4] Alexandre Gouvea da Silva CARLOS et al. « ChatGPT : Challenges and Benefits in Software Programming for Higher Education ». In : (fév. 2024). URL : <https://www.mdpi.com/2071-1050/16/3/1245>.
- [5] EDUNAO. *Edunao - Digital Learning Platform*. Accessed : 2025-02-12. 2025. URL : <https://123.edunao.com/>.
- [6] GITHUB. *GitHub Copilot - Your AI Pair Programmer*. Accessed : 2025-02-12. 2023. URL : <https://github.com/features/copilot>.
- [7] Gregor JOŠT, Viktor TANESKI et Sašo KARAKATIČ. « The Impact of Large Language Models on Programming Education and Student Learning Outcomes ». In : (mai 2024). URL : <https://www.mdpi.com/2076-3417/14/10/4115>.
- [8] LINAGORA. *Linagora - Artificial Intelligence Solutions*. Accessed : 2025-02-12. 2025. URL : <https://linagora.com/artificial-intelligence>.
- [9] Rongxin LIU et al. *Teaching CS50 with AI - Leveraging Generative Artificial Intelligence in Computer Science Education*. URL : <https://cs.harvard.edu/malan/publications/V1fp0567-liu.pdf>.
- [10] NOLEJ. *Nolej - AI-powered Learning Platform*. Accessed : 2025-02-12. 2025. URL : <https://www.nolej.io/>.
- [11] OPENAI. *ChatGPT - OpenAI's Conversational AI*. Accessed : 2025-02-12. 2023. URL : <https://chat.openai.com>.
- [12] OPENAI. *Introducing study mode in ChatGPT*. Blog post on openai.com. Juill. 2025. URL : <https://openai.com/index/chatgpt-study-mode/>.

- [13] Jaromir SVELKA et al. *Thrilled by Your Progress! Large Language Models (GPT-4) No Longer Struggle to Pass Assessments in Higher Education Programming Courses*. Juin 2023. URL : <https://arxiv.org/abs/2306.10073>.
- [14] Shane SNIPES. *Transforming Education with AI*. Mai 2023.
- [15] Ashish VASWANI et al. *Attention Is All You Need*. 2017. arXiv : 1706.03762 [cs.CL].
- [16] Gan WENSHENG et al. *Large Language Models in Education : Vision and Opportunities*. Nov. 2023. URL : <https://arxiv.labs.arxiv.org/html/2311.13160>.
- [17] Rina ZVIEL-GIRSHIN. « The Good and Bad of AI Tools in Novice Programming Education ». In : (oct. 2024). URL : <https://www.mdpi.com/2227-7102/14/10/1089>.

Annexe A

Questionnaire de pré-test

Pré-test débutant

1. En JavaScript, quel nom de variable est valide ?

☐ 2nom ☒ nom_2 ☐ nom 2 ☐ @nom

2. En JavaScript, quel mot-clé sert à définir une fonction ?

☐ func ☐ def ☒ function ☐ fn

3. Dans une page HTML, quel attribut sert le plus souvent à identifier de façon unique un élément ?

☐ class ☒ id ☐ name ☐ div

4. En JavaScript, quel opérateur de comparaison renvoie `true` si deux valeurs sont exactement égales (valeur et type) ?

☐ = ☐ == ☒ === ☐ =====

5. Le DOM (Document Object Model) est :

☐ Un langage de mise en page
☐ Une base de données
☒ Une représentation des noeuds HTML manipulable par JavaScript
☐ Une bibliothèque externe

6. Je me sens capable d'écrire moi-même une fonction JavaScript correcte sans utiliser d'assistant d'IA.

Pas du tout d'accord ☐ ☐ ☐ ☐ *Tout à fait d'accord*

7. Je sais, sans aide, comment structurer mon code pour résoudre un problème simple (décomposition en étapes, logique, syntaxe).

Pas du tout d'accord ☐ ☐ ☐ ☐ *Tout à fait d'accord*

8. Je suis capable de corriger une erreur dans mon code en lisant uniquement le message d'erreur ou le comportement du programme.

Pas du tout d'accord ☐ ☐ ☐ ☐ ☐ *Tout à fait d'accord*

9. Je sais utiliser le DOM en JavaScript (récupérer un input, modifier le contenu d'une balise) sans avoir besoin de chercher la solution.

Pas du tout d'accord ☐ ☐ ☐ ☐ ☐ *Tout à fait d'accord*

Pré-test intermédiaire

1. Quelle méthode JavaScript retourne un nouveau tableau contenant uniquement les éléments qui passent un test ?

☐ `forEach()` ☒ `filter()` ☐ `map()` ☐ `reduce()`

2. Quel appel de fonction fléchée retourne correctement le carré d'un nombre ?

☐ `(x) => { x * x; }` ☒ `(x) => x * x` ☐ `x => { return x ** 2 }`
☐ `(x) => return x * x`

3. Quelle expression vérifie que la variable `x` est strictement un tableau ?

☒ `Array.isArray(x)` ☐ `typeof x === "array"` ☐ `x instanceof Object`
☐ `typeof x === "object"`

4. Quelle instruction crée une copie indépendante d'un objet simple `let obj = {a: 1}` ?

☒ `let copie = {...obj}` ☐ `let copie = obj`
☐ `let copie = Object.assign(obj)` ☐ `let copie = JSON.parse(obj)`

5. En JavaScript, que renvoie cette expression : `[1,2,3].map(x => x + 1).filter(x => x > 3)` ?

☐ `[1,2,3]` ☒ `[4]` ☐ `[2,3,4]` ☐ `[3,4]`

6. Quelle est la valeur de `mot.substring(1, 4)` si `mot = "palindrome"` ?

☐ `"pali"` ☐ `"alind"` ☒ `"ali"` ☐ `"lind"`

7. Je sais, sans aide, comment structurer mon code pour résoudre un problème simple (décomposition en étapes, logique, syntaxe).

Pas du tout d'accord ☐ ☐ ☐ ☐ ☐ *Tout à fait d'accord*

8. Je suis capable de corriger une erreur dans mon code en lisant uniquement le message d'erreur ou le comportement du programme.

Pas du tout d'accord ☐ ☐ ☐ ☐ ☐ *Tout à fait d'accord*

9. Je sais utiliser le DOM en JavaScript (récupérer un input, modifier le contenu d'une balise) sans avoir besoin de chercher la solution.

Pas du tout d'accord ☐ ☐ ☐ ☐ ☐ *Tout à fait d'accord*

10. Je me sens capable de refaire un exercice similaire à celui proposé (palindrome + affichage) seul, sans aucune assistance.

Pas du tout d'accord ☐ ☐ ☐ ☐ ☐ *Tout à fait d'accord*

Annexe B

Questionnaire de post-test

Post-test débutant (QCM)

1. Quel code retourne `true` si le mot donné est un palindrome (ex. "radar") ?
☐ `mot.reverse() === mot` ☒ `mot === mot.split('').reverse().join('')`
☐ `reverse(mot) == mot` ☐ `mot.split('').sort().join('') === mot`
2. Quelle méthode permet d'insérer un texte calculé dans une balise HTML (par exemple une `<div>`) en JavaScript ?
☐ `element.setAttribute("text", valeur)` ☒ `element.innerText = valeur`
☐ `element.append(valeur)` ☐ `element.write(valeur)`
3. Comment peut-on récupérer le contenu d'un champ de type texte dans une page HTML avec JavaScript ?
☒ `document.querySelector("mot").value` ☐ `document.input.value`
☐ `getElementById("mot").text` ☐ `input.get("value")`
4. Quelle écriture est correcte pour déclarer une fonction `isPalindrome` en JavaScript ?
☐ `def isPalindrome(mot) { }` ☒ `function isPalindrome(mot) { }`
☐ `let isPalindrome = function{ }` ☐ `function: isPalindrome(mot) { }`
5. À quoi sert `split('').reverse().join('')` dans un traitement de chaîne ?
☐ À extraire les voyelles ☒ À inverser l'ordre des lettres
☐ À créer une liste vide ☐ À enlever les doublons
6. Je me sens capable d'écrire moi-même une fonction JavaScript correcte sans utiliser d'assistant d'IA.
Pas du tout d'accord ☐ ☐ ☐ ☐ ☐ *Tout à fait d'accord*
7. Je sais, sans aide, comment structurer mon code pour résoudre un problème simple (décomposition en étapes, logique, syntaxe).
Pas du tout d'accord ☐ ☐ ☐ ☐ ☐ *Tout à fait d'accord*

8. Je suis capable de corriger une erreur dans mon code en lisant uniquement le message d'erreur ou le comportement du programme.

Pas du tout d'accord ☐ ☐ ☐ ☐ ☐ *Tout à fait d'accord*

9. Je sais utiliser le DOM en JavaScript (récupérer un input, modifier le contenu d'une balise) sans avoir besoin de chercher la solution.

Pas du tout d'accord ☐ ☐ ☐ ☐ ☐ *Tout à fait d'accord*

10. Je me sens capable de refaire un exercice similaire à celui proposé (palindrome + affichage) seul, sans aucune assistance.

Pas du tout d'accord ☐ ☐ ☐ ☐ ☐ *Tout à fait d'accord*

11. Penses-tu que l'aide de l'IA a renforcé ta compréhension ou seulement permis de réussir plus vite ? Pourquoi ?

Réponse libre

12. Qu'aurait pu faire le chatbot pour t'aider davantage à apprendre plutôt que simplement t'aider à réussir ?

Réponse libre

Post-test intermédiaire (QCM)

1. Que retourne la fonction

```
function calcul() { [1, 2, 3].forEach(x => { return x * 2; }); }
```

☐ [2, 4, 6] ☐ 6 ☒ undefined ☐ 0

2. Quelle méthode est la plus adaptée pour parcourir tous les caractères d'une chaîne avec leur position ?

☐ for (char of chaine) ☐ chaine.forEach(char => ...)

☒ for (let i = 0; i < chaine.length; i++) ☐ chaine.map(char => ...)

3. Que retourne la fonction

```
function test() { let x = 5; if (true) { let x = 10; } return x; }
```

☐ 10 ☒ 5 ☐ undefined ☐ Erreur

4. Quel serait le bon usage de 'innerHTML' pour surligner un texte dans une balise '<p>' ?

☐ `p.innerHTML = "<mark>highlightText</mark>"`
☐ `p.innerHTML = document.highlight(mot)`
☒ `p.innerHTML = "<mark>" + mot + "</mark>"`
☐ `p.setHTML(mot)`

5. Que retourne la fonction

```
function test() { return {} === {};
```

☒ `false` ☐ `true` ☐ `undefined` ☐ `"object"`

6. Je me sens capable d'écrire moi-même une fonction JavaScript correcte sans utiliser d'assistant d'IA.

Pas du tout d'accord ☐ ☐ ☐ ☐ ☐ *Tout à fait d'accord*

7. Je sais, sans aide, comment structurer mon code pour résoudre un problème simple (décomposition en étapes, logique, syntaxe).

Pas du tout d'accord ☐ ☐ ☐ ☐ ☐ *Tout à fait d'accord*

8. Je suis capable de corriger une erreur dans mon code en lisant uniquement le message d'erreur ou le comportement du programme.

Pas du tout d'accord ☐ ☐ ☐ ☐ ☐ *Tout à fait d'accord*

9. Je sais utiliser le DOM en JavaScript (récupérer un input, modifier le contenu d'une balise) sans avoir besoin de chercher la solution.

Pas du tout d'accord ☐ ☐ ☐ ☐ ☐ *Tout à fait d'accord*

10. Je me sens capable de refaire un exercice similaire à celui proposé (palindrome + affichage) seul, sans aucune assistance.

Pas du tout d'accord ☐ ☐ ☐ ☐ ☐ *Tout à fait d'accord*

11. Penses-tu que l'aide de l'IA a renforcé ta compréhension ou seulement permis de réussir plus vite ? Pourquoi ?

Réponse libre

12. Qu'aurait pu faire le chatbot pour t'aider davantage à apprendre plutôt que simplement t'aider à réussir ?

Réponse libre

Annexe C

Prompt système du tuteur intelligent

You are "Tuteur Intelligent", an expert, endlessly patient JavaScript tutor.

Your mission is to help the learner `_discover_` answers--never to hand them full code solutions.

CORE RULES

1. Socratic Approach

- Lead with open-ended questions that surface the learner's reasoning.
- Provide only single-line micro-examples of one method's usage (e.g., `arr.map(fn)` `// returns newArray`)--never a full solution.

2. Minimal Examples

- Show at most one-line method demos or console outputs--enough to illustrate syntax, not solve the task.

3. Debugging Help

- Identify the conceptual issue or a suspect line, then ask guiding questions.
- Suggest debugging strategies (logging, rubber-ducking, breakpoints) instead of rewriting the code.

4. Refusals

- If asked for complete code, politely decline and offer the next question or hint.

5. Encouragement & Metacognition

- Praise effort, suggest breaking problems into smaller pieces, and prompt reflection.

ADAPTIVE LEARNER-LEVEL DETECTION

- No direct quiz about experience levels.

Instead, infer the learner's mastery from how they describe their goal and respond to context-relevant questions.

- Process

1. Ask the learner to explain the problem or goal in their own words.
2. Pose 1-2 `_naturally relevant_` follow-up questions about the same task (e.g., "How are you currently iterating over that array?").
3. Evaluate the depth, accuracy, and vocabulary of each answer.
 - Accurate use of terminology / proactive mention of edge cases → likely Advanced.
 - Solid syntax but fuzzy on underlying concepts → Intermediate.
 - Struggles with foundational terms or basic syntax → Novice.
4. Tag the learner internally as Novice, Intermediate, or Advanced.
5. Continuously update the tag if later responses show a different level.

- Response Tailoring

- `_Novice_`: simpler language, smaller hint increments, zero jargon.
- `_Intermediate_`: assume syntax fluency; focus on patterns and pitfalls.
- `_Advanced_`: use spec terminology, dive into edge cases, encourage optimization thinking.

SESSION FLOW

1. Friendly greeting → Ask learner to describe goal or problem.
2. Follow-up question(s) logically tied to their task to gauge mastery.
3. Infer level → Tailor hints/questions per rules above.
4. Loop: Guiding question → Micro-hint → Reflection → Repeat until learner confirms solution.