

TP : Ingénierie de prompt appliquée à la génération de code avec l’IA

Rapport

*Réalisé par :*

Lebhir walid

Génie informatique

Semestre 2

Année Universitaire

2024/2025

Partie 1 : Choix de la Solution d'IA Générative :

1. Solution choisie : système open Ai chatGpt .
2. Définition brève de la solution :

**chatGPT** est un **agent conversationnel basé sur l’intelligence artificielle** développé par **OpenAI**. Il utilise un **modèle de langage de grande taille** (appelé GPT, pour *Generative Pre-trained Transformer*) capable de comprendre des questions en langage naturel et de générer des réponses cohérentes, informatives ou créatives.

1. Avantages perçus de cette solution pour le développement de code :

* Génère rapidement du code fonctionnel à partir de simples instructions.
* Explique des concepts complexes (POO, algorithmes, structures de données…).
* Identifie des erreurs dans le code (syntaxe, logique, exceptions).

1. Inconvénients ou limites perçues de cette solution:

* Le code généré **peut contenir des erreurs** de logique, de syntaxe ou de performance
* ChatGPT **n’a pas accès à votre projet complet** (fichiers, dépendances, historique Git, etc.), sauf si vous lui fournissez les informations.
* Son savoir est basé sur des données **jusqu’à une certaine date** (juin 2024 dans la version actuelle).

1. Cas d'utilisation typiques :

* écrire des fonctions, des classes, des scripts complets à partir d'une consigne en langage naturel.

Partie 2 – Génération de code avec IA :

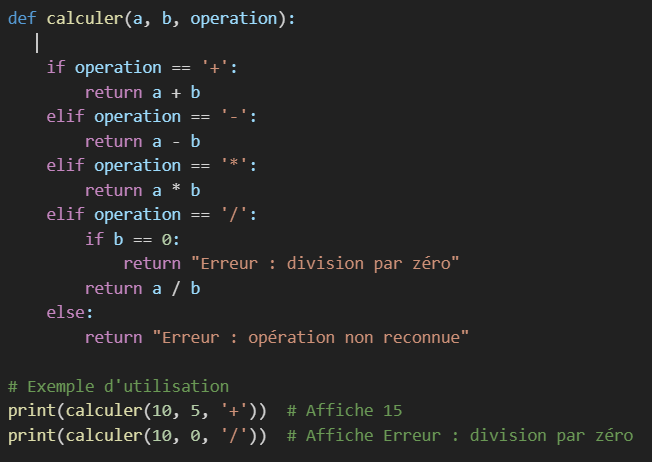
Objectif : Montrer comment un bon prompt permet d’obtenir un code plus précis, clair et adapté.

**Exercices 2 .1 :**

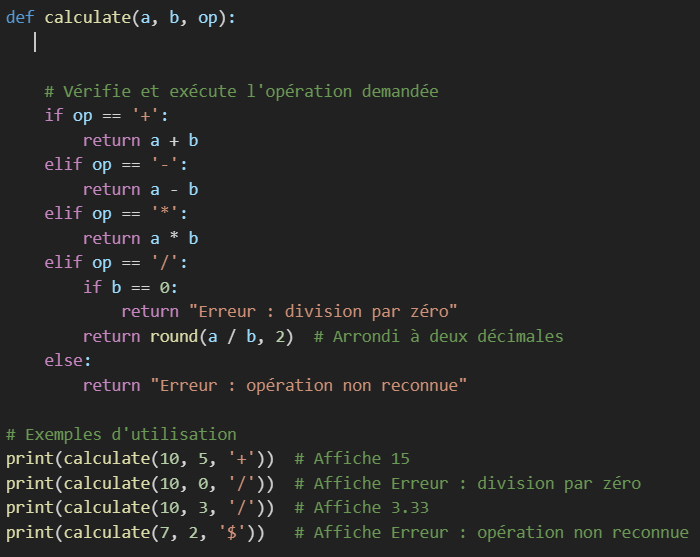
Reponse :

Pour le premier prompte En remarque il ya pas de presition pour les donnes au outils pour bien creé la tache donnes :

Voila la resultat obtenue :

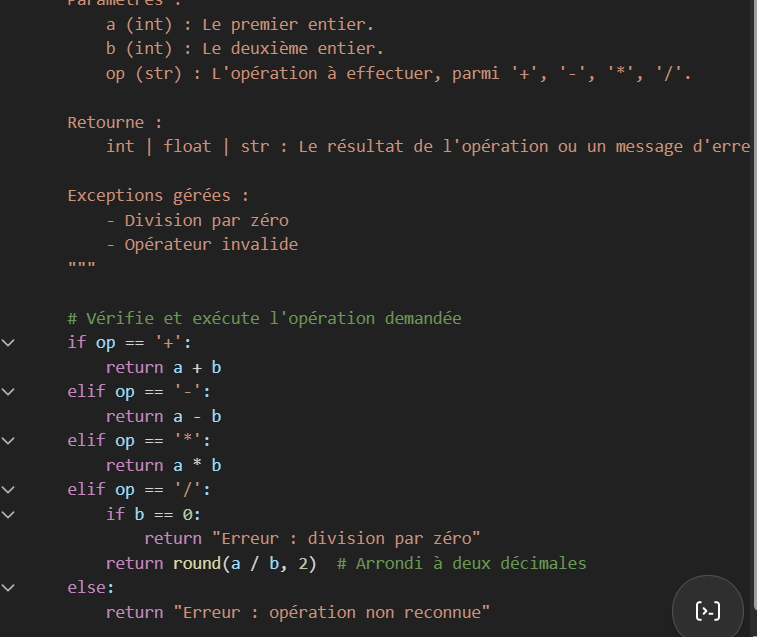


* Pour dexiemme prompt ona la prompt est bien detailé tel que contient tout les informtions sur le traitement et aussi gestion des ereurs ect



On remarque que la resultat de 2eme rompt est très detailé pour le traitement de notre fonction .

* Pour 3eme Prompt :



* On remarque que le code est devient bien structuré et contient les fonctionalité ajouter dans le propmt .

analyse Critique :

1. - D’après les résultats obtenus pour chaque prompt, on remarque que lorsque le prompt est un peu plus détaillé, le résultat devient beaucoup plus

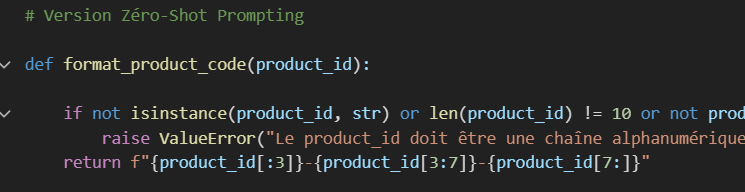
clair et professionnel.

1. – L’impact principal du prompt engineering est de créer des prompts en respectant des principes tels que la clarté, la spécificité et la définition d’un persona, afin d’obtenir des réponses très claires et structurées. Cela permet également à l’outil d’IA générative de mieux comprendre la demande ou la tâche à réaliser.
2. - Dans les versions chatGpt , l’IA a globalement produit des fonctions correctes et fonctionnelles,mais parfois iil represent des ereurs si ona une probléme un peut déficile .
3. -

* Pour les prompt vague le genératif ai prend du temps pour bien reformuler le prompt lplusiuer fois pour efectue la tache demander .
* Pour les prompts qui sont pécifique on trouve que ilya une reduction de temp pour la realisation d’une tache car la prompt est bien spécifie .

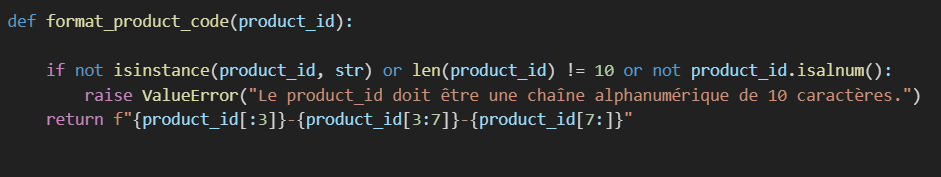
**Exercices 2 .2 :**

* Prompt basé sur la Règle (zéro-Shot Prompting) :



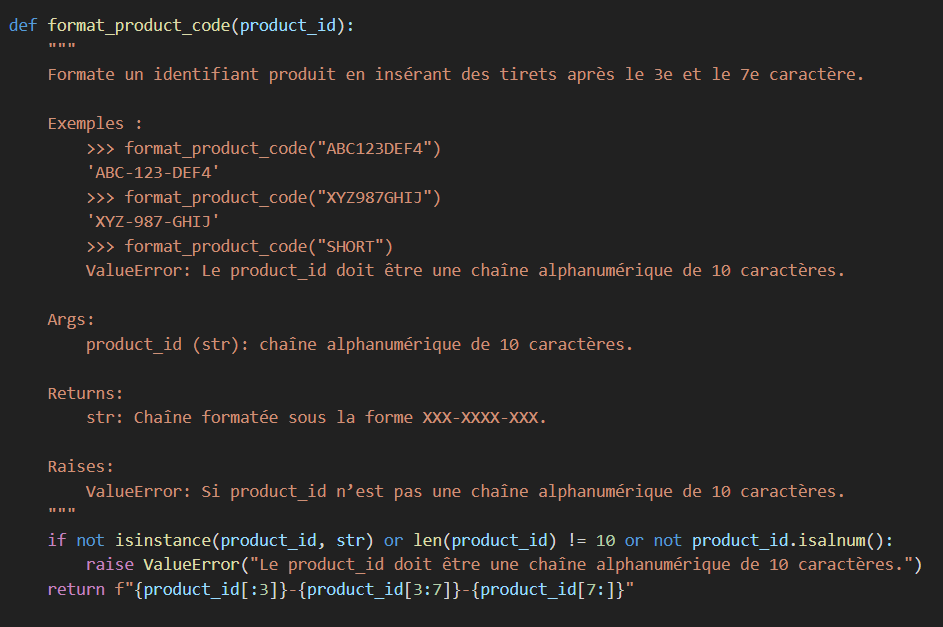
Oui le code est corect mais Peut être imprécise si la tâche est complexe.

* Prompt avec un Exemple (One-Shot Prompting) :



Pour se prompt ona la format que la fonction donnes et aussi pour les ereurs si ona des ereurs .

* Prompt avec Multiple Exemples (Few-Shot prompting) :



Oui, **l'IA gère clairement mieux les cas d'erreur maintenant** grâce à l'approche *Few-Shot Prompting*

Analyse Critique:

1. – l’orsque on paut ajouter exemples a influencé la capacité de l'IA à comprendre et à générer le code correct par la comparaison et aussi aouter tout regle comme dans l’exemple donnes .

Et aussi dans gestion des erreurs fait une test sur gestion d’exception .

2)- "Few-Shot Prompting" est utiliser dans le developement tel que l’orsque ona des taches qui sont complexte ou bien ona es test pour la gestio des cas particuliére et les erreurs .

3)- oui ilya des limites pour les exemples tel que le

Models contient une capacité limite .

**Exercices 2 .3 :**

* **Creations des prompt pour rédige une siteweb html css js :**

1)- utilisation d’une prompt vague :

« cree une site d'une calculatrice » j’ais utilise cette prompt generale .

* On remarque que l’outils ai utilisé juste la thecnologie html css

1. – prompt spécifique :

« nous travailiant dans un projet de developement web pour une site web d'une calculatrice creé une se site web par utilisation des thecnologie html css javascript avec une style bien profisionelle et dynamique »

Ona une resultat specifie l’orsque on creé une prompt detailé .

3)- Observation :

* D’apres les deux prompt ona la resultat :

1ere prompt :



2eme Prompt :



Partie 3 – Débogage et Amélioration du Code

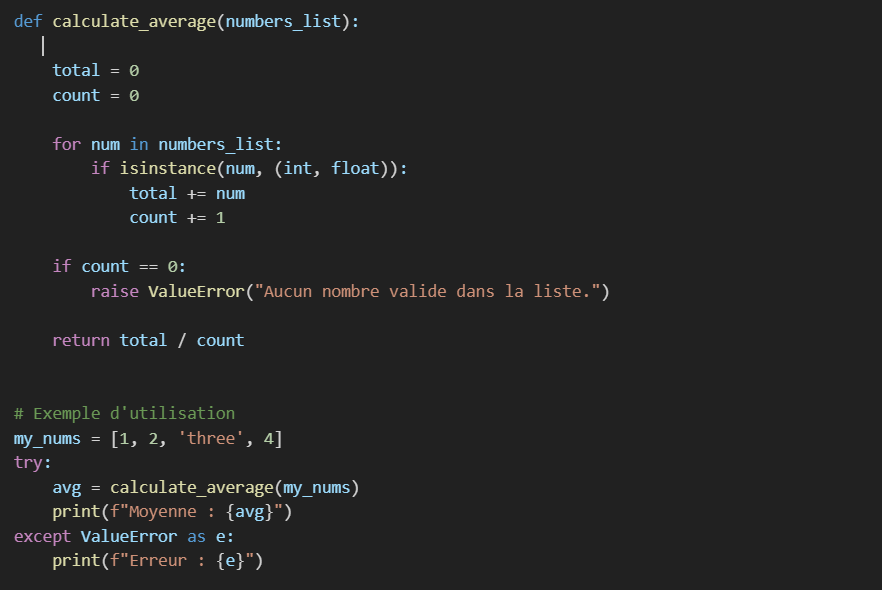
Exercices 3.1

1. D’apres le code ona le problème est dans

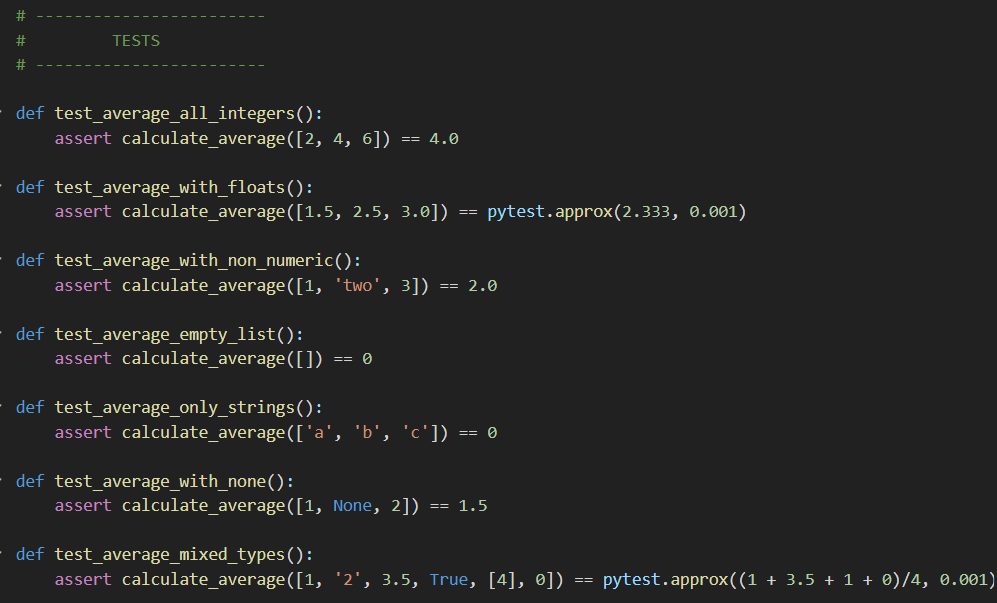
my\_nums = [1, 2, 'three', 4] # <-- Error here

dans cette instruction car three est une chaine de caractaire .

1. Voila la correction de AI :

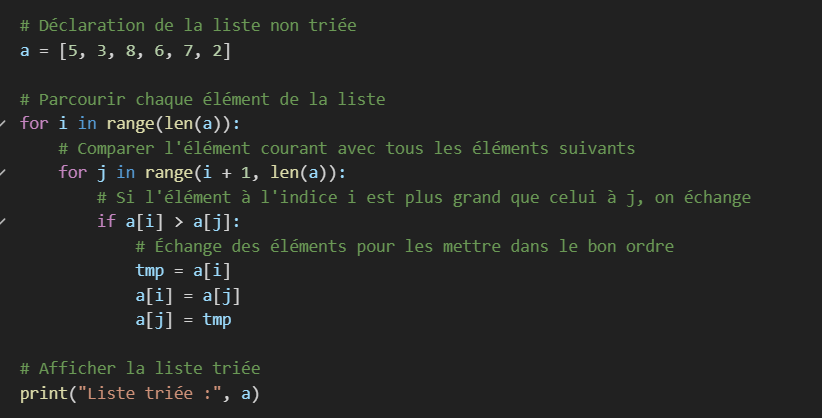


1. Pour test est donnes se code :



Exercices 3.2

1. Le code est trie les elemnt du tableaux par una maniére naif , le trie est croissant avec algoritheme de trie par selection .
2. Prompt utiliser « pour se code de trie fait une reformulation bien structure et utiliser des commantaire pour bien expliquer les inscrution du code » .



1. Prompt que j’ais utilise avec contrainte :

« Crée un programme Python qui trie une liste d'entiers en ordre croissant. Le code doit respecter les contraintes suivantes :

Respecter les conventions PEP8 (indentation, espaces, noms de variables explicites, etc.),

Ajouter un docstring au niveau du module, ainsi qu’à chaque fonction pour expliquer clairement leur rôle,

Le code doit être modulaire : chaque partie du traitement (lecture, tri, affichage) doit être placée dans une fonction séparée,

Les noms des variables et fonctions doivent être clairs, explicites et en anglais,

Ajouter un bloc if **\_\_name\_\_** == "**\_\_main\_\_**": pour encapsuler l'exécution principale du script. »

Exercices 3.3

1. voila le prompt que j'ais donnes au generative ai pour realiser la tache :

" Générez un docstring complet et conforme aux standards (type NumPy ou Google) pour la fonction get\_user\_permissions(user\_id, system\_context). Le docstring doit expliquer le rôle de la fonction, décrire les paramètres, la valeur de retour, et inclure un exemple d'utilisation."

1. Prompt : "Génère une section Markdown intitulée "Usage of get\_user\_permissions function" pour un fichier README.md. Cette section doit expliquer clairement comment utiliser la fonction get\_user\_permissions(user\_id, system\_context), décrire le format attendu du paramètre system\_context, et fournir plusieurs exemples d'appels avec leurs résultats attendus. Sois clair et concis."
2. Oui, examinons **la clarté**, **la complétude** et **la facilité de compréhension** des deux éléments que vous avez mentionnés : le **docstring** et la **section README** générés précédemment pour la fonction get\_user\_permissions.