

Atelier Pratique : Automatisation avec un modèle LLM

Exercice 1 : Génération de contenu marketing

Demandez à l'IA de créer une description de produit ou un texte publicitaire attractif pour un produit donné.

Exemple : Générer un texte pour un smartphone ou un ordinateur portable.

Exercice 2 : Résumé des articles scientifiques

Téléchargez un fichier texte contenant un article et demandez à l'IA de générer un résumé en moins de 150 mots.

Exercice 3 : Conversion texte → liste d'actions

Transformez un long texte descriptif en une liste d'étapes ou d'actions. Par exemple, donnez-lui une recette complexe et demandez d'en extraire des étapes simples.

Exercice 4 : Extraction de mots-clés d'un document

Demandez à l'IA d'extraire les mots-clés les plus pertinents d'un fichier texte ou PDF pour faciliter la recherche de contenu.

Exercice 5 : Génération de quiz

À partir d'un texte donné (exemple : le contenu d'un cours), demandez à l'IA de générer 5 questions à choix multiple basées sur ce contenu.

Exercice 6 : Traduction multilingue

Automatisez la traduction d'un texte en plusieurs langues à l'aide de l'IA. Donnez un texte en français et demandez des traductions en anglais, espagnol et allemand.

Exercice 7 : Création de lettres formelles

Générez automatiquement une lettre de motivation ou une lettre formelle (par exemple, pour demander un remboursement) en fournissant des informations de base.

Exercice 8 : Réécriture et simplification

Prenez un texte complexe et demandez à l'IA de le réécrire de manière simplifiée, pour un public d'enfants ou de non-spécialistes.

Exercice 9 : Résolution de problèmes de programmation

Donnez une description de problème de codage, et demandez à l'IA de proposer une solution en Python ou dans un autre langage de programmation.

Exercice 10 : Analyse de sentiment

Demandez à l'IA d'analyser le sentiment d'un texte donné, par exemple, un avis client ou un commentaire sur un réseau social (positif, neutre ou négatif).

Exercice 11 : Classement de documents

Donnez une série de descriptions de documents (par exemple, des articles de presse) et demandez à l'IA de les classer par thématique (économie, politique, sport, etc.).

Exercice 12 : Génération d'idées de projet

Proposez un thème ou un domaine (par exemple, "intelligence artificielle dans la santé") et demandez à l'IA de générer 5 idées de projets innovants dans ce domaine.

Exercice 13. Résolution de problèmes mathématiques

Donnez une équation ou un problème mathématique, et demandez à l'IA de fournir une solution pas à pas (par exemple, résoudre une équation du second degré).

Exercice 14 : Correction grammaticale et orthographique

Donnez un texte contenant des erreurs de grammaire ou d'orthographe, et demandez à l'IA de le corriger.

Exercice 15 : Réponses contextuelles aux PDF

Chargez un fichier PDF et demandez à l'IA de répondre à des questions basées sur son contenu. Par exemple : "Quelle est la date mentionnée dans le deuxième paragraphe du document ?".

Solution

1. Génération de contenu marketing

Objectif : Générer une description de produit pour un smartphone.

```
import g4f

def generate_product_description(product_name):

    prompt = f"Écrire une description marketing attrayante pour un produit nommé {product_name}. Inclure ses caractéristiques principales et ses avantages."

    response = g4f.Completion.create(prompt=prompt)

    return response

product_name = "Smartphone Galaxy X"

description = generate_product_description(product_name)

print(description)
```

2. Résumer des articles scientifiques

```
import g4f

def summarize_scientific_article(article_text):

    prompt = f"Résumer cet article scientifique en moins de 150 mots : {article_text}"

    response = g4f.Completion.create(prompt=prompt)

    return response

article_text = "L'article scientifique sur les applications de l'IA dans la médecine décrit de nombreuses études récentes..."

summary = summarize_scientific_article(article_text)

print(summary)
```

3. Conversion texte → liste d'actions

```
import g4f

def extract_steps_from_recipe(recipe_text):
    prompt = f"Extraire les étapes de la recette suivante : {recipe_text}"
    response = g4f.Completion.create(prompt=prompt)
    return response

recipe_text = "1. Préchauffez le four à 180°C. 2. Mélangez les œufs et la farine..."

steps = extract_steps_from_recipe(recipe_text)

print(steps)
```

4. Extraction de mots-clés d'un document

```
import g4f

def extract_keywords(document_text):
    prompt = f"Extraire les mots-clés importants du texte suivant : {document_text}"
    response = g4f.Completion.create(prompt=prompt)
    return response

document_text = "L'intelligence artificielle et ses applications en médecine, finance, etc."

keywords = extract_keywords(document_text)

print(keywords)
```

5. Génération de quiz

```
import g4f

def generate_quiz_from_text(text):

    prompt = f"Créer un quiz de 5 questions à partir du texte suivant : {text}"

    response = g4f.Completion.create(prompt=prompt)

    return response

text = "L'intelligence artificielle est un domaine de l'informatique qui simule des processus humains."

quiz = generate_quiz_from_text(text)

print(quiz)
```

6. Traduction multilingue

```
import g4f

def translate_text(text, target_languages=["en", "es", "de"]):

    translations = {}

    for lang in target_languages:

        prompt = f"Traduire ce texte en {lang} : {text}"

        response = g4f.Completion.create(prompt=prompt)

        translations[lang] = response

    return translations

text = "Bonjour, comment ça va ?"

translations = translate_text(text)

print(translations)
```

7. Création de lettres formelles

```
import g4f

def generate_cover_letter(name, job_title, company_name):

    prompt = f"Écrire une lettre de motivation pour un candidat nommé {name},  
postulant pour le poste de {job_title} chez {company_name}."

    response = g4f.Completion.create(prompt=prompt)

    return response

letter = generate_cover_letter("Alice Dupont", "développeur web", "XYZ Corp")

print(letter)
```

8. Réécriture et simplification

```
import g4f

def simplify_text(text):

    prompt = f"Réécrire ce texte de manière plus simple et compréhensible  
pour un enfant : {text}"

    response = g4f.Completion.create(prompt=prompt)

    return response

complex_text = "L'intelligence artificielle utilise des algorithmes complexes  
pour analyser de grandes quantités de données."

simplified_text = simplify_text(complex_text)

print(simplified_text)
```

9. Résolution de problèmes de programmation

```
import g4f

def solve_programming_problem(problem_description):

    prompt = f"Résoudre ce problème de programmation en Python :  
{problem_description}"

    response = g4f.Completion.create(prompt=prompt)

    return response

problem = "Écrire une fonction Python qui renvoie la somme des éléments  
d'une liste."

solution = solve_programming_problem(problem)

print(solution)
```

10. Analyse de sentiment

```
import g4f

def analyze_sentiment(text):

    prompt = f"Analysez le sentiment de ce texte : {text}"

    response = g4f.Completion.create(prompt=prompt)

    return response

text = "J'adore ce produit, il est vraiment génial !"

sentiment = analyze_sentiment(text)

print(sentiment)
```

11. Classement de documents

```
import g4f

def classify_document(text, categories=["économie", "politique", "sport"]):

    prompt = f"Classer ce document dans l'une des catégories suivantes :  
{categories}. Voici le texte : {text}"

    response = g4f.Completion.create(prompt=prompt)

    return response

text = "Le marché des actions a connu une hausse significative cette  
semaine."

classification = classify_document(text)

print(classification)
```

12. Génération d'idées de projet

```
import g4f

def generate_project_ideas(domain):

    prompt = f"Proposer 5 idées de projet innovants dans le domaine suivant :  
{domain}"

    response = g4f.Completion.create(prompt=prompt)

    return response

ideas = generate_project_ideas("intelligence artificielle dans la santé")

print(ideas)
```

13. Résolution de problèmes mathématiques

```
import g4f

def solve_math_problem(problem):
    prompt = f"Résoudre ce problème mathématique : {problem}"
    response = g4f.Completion.create(prompt=prompt)
    return response

problem = "Résoudre l'équation  $2x + 5 = 15$ "
solution = solve_math_problem(problem)
print(solution)
```

14. Correction grammaticale et orthographique

```
import g4f

def correct_text(text):
    prompt = f"Corriger les erreurs grammaticales et orthographiques dans ce texte : {text}"
    response = g4f.Completion.create(prompt=prompt)
    return response

text = "J'ai allez au parc avec mes amis"
corrected_text = correct_text(text)
print(corrected_text)
```

15. Réponses contextuelles aux PDF

```
import g4f

from PyPDF2 import PdfReader

def extract_pdf_text(pdf_path):
    reader = PdfReader(pdf_path)

    text = ""

    for page in reader.pages:
        text += page.extract_text()

    return text

def answer_pdf_question(pdf_path, question):
    text = extract_pdf_text(pdf_path)

    prompt = f"Répondre à la question suivante à partir du texte du PDF :  
{question} Voici le texte : {text}"

    response = g4f.Completion.create(prompt=prompt)

    return response

pdf_path = "chemin/vers/le/pdf.pdf"

question = "Quel est le sujet principal de ce document ?"

answer = answer_pdf_question(pdf_path, question)

print(answer)
```