

# Test compétence/motivation Data Science LangChain

## Contexte

Ce test évalue votre capacité à :

- Apprendre un nouvel outil en autonomie
- Suivre un tutoriel et faire fonctionner un exemple
- Adapter et améliorer du code existant

Durée estimée : 2-3 heures

## Prérequis

- Python installé (3.8+)
- Compte Anthropic (Claude) ou OpenAI
- Clé API configurée dans les variables d'environnement

## Partie 1 : Quickstart LangChain (obligatoire)

### Objectif

Réaliser le quickstart officiel de LangChain pour créer un agent météo.

### Instructions

1. Consultez le tutoriel : <https://docs.langchain.com/oss/python/langchain/quickstart>
2. Suivez les étapes pour créer l'agent météo complet :
  - Installation de LangChain
  - Création des outils (`get_weather_for_location`, `get_user_location`)
  - Configuration du modèle
  - Ajout de la mémoire
  - Création et exécution de l'agent
3. Vérifiez que le code fonctionne :

- L'agent répond aux questions météo
- La mémoire conversationnelle fonctionne
- Les outils sont appelés correctement

## Livrables

- Code fonctionnel du quickstart
  - Capture d'écran ou sortie console montrant l'exécution
  - Explication brève de ce que fait chaque composant
- 

# Partie 2 : Adaptation et amélioration (à choisir)

Choisissez une des trois options ci-dessous et implémentez-la.

## Option A : Ajouter un nouvel outil

Objectif : Ajouter un outil personnalisé à l'agent. Consigne :

- Créez un outil `get_weather_advice` qui donne un conseil vestimentaire basé sur la météo d'une ville
- Intégrez-le à l'agent existant
- Modifiez le prompt système pour que l'agent l'utilise quand c'est pertinent

Exemple d'utilisation attendue :

User: "What should I wear in Paris?"

Agent: [utilise `get_weather_for_location` puis `get_weather_advice`]

Critères d'évaluation :

- L'outil est correctement défini avec le décorateur `@tool`
  - L'outil est ajouté à la liste des outils de l'agent
  - Le prompt système guide l'utilisation de l'outil
  - L'agent appelle l'outil de manière pertinente
- 

## Option B : Enrichir les outils existants

Objectif : Améliorer les outils existants pour retourner plus d'informations. Consigne :

- Modifiez `get_weather_for_location` pour retourner plusieurs informations (température, conditions, humidité, etc.)
- Vous pouvez simuler ces données ou utiliser une API météo réelle (optionnel)
- Adaptez le format de réponse si nécessaire

Exemple de sortie attendue :

```
# Au lieu de : "It's always sunny in Paris!"  
  
# Retourner : "Paris: 22°C, Sunny, Humidity: 65%"
```

Critères d'évaluation :

- L'outil retourne des informations structurées
  - Le code est bien organisé (peut utiliser un dictionnaire ou dataclass)
  - L'agent utilise correctement ces nouvelles informations dans ses réponses
- 

## Option C : Modifier le comportement de l'agent

Objectif : Changer le style et les capacités de l'agent. Consigne :

- Modifiez le prompt système pour changer le style de réponse (ex: formel, technique, amical)
- Ajoutez une nouvelle capacité : l'agent peut aussi donner des recommandations d'activités selon la météo
- Implémentez cette logique directement dans le prompt ou via un nouvel outil

Exemple d'utilisation attendue :

User: "What can I do outside today?"

Agent: [donne des recommandations d'activités basées sur la météo]

Critères d'évaluation :

- Le prompt système est modifié de manière cohérente
  - La nouvelle capacité fonctionne correctement
  - Les réponses de l'agent reflètent le nouveau style/comportement
- 

## Livrables finaux

1. Code source complet (fichier Python ou repository github)
1. Document README.md contenant :
  - Instructions d'installation et de configuration

- Explication de ce qui a été fait
  - Exemples d'utilisation avec captures d'écran
  - Difficultés rencontrées et solutions trouvées
1. Option choisie (A, B ou C) et justification

## Critères d'évaluation

- Autonomie : capacité à trouver et utiliser la documentation
- Compréhension : explication claire des concepts LangChain utilisés
- Qualité du code : structure, lisibilité, commentaires
- Fonctionnalité : code qui fonctionne et répond aux consignes
- Créativité : améliorations ou optimisations personnelles (bonus)

## Note importante

Si vous ne parvenez pas à tout finaliser, ce n'est pas grave. L'objectif est d'évaluer votre démarche d'apprentissage et votre compréhension. Lors de l'entretien, nous discuterons de votre compréhension globale, de votre approche et de ce que vous avez tenté, même si cela n'a pas abouti. L'important est de montrer votre capacité à apprendre de manière autonome.

## Ressources utiles

- Documentation LangChain : <https://docs.langchain.com/>
- Quickstart : <https://docs.langchain.com/oss/python/langchain/quickstart>
- Documentation sur les outils : [https://docs.langchain.com/docs/use\\_cases/tool\\_use/](https://docs.langchain.com/docs/use_cases/tool_use/)

## Questions ?

- N'hésitez pas à poser des questions si vous êtes bloquée. L'objectif est d'évaluer votre capacité à apprendre et à vous adapter, pas de vous bloquer.