🧠 Jupyter Notebook Exercise: Build a Multi-Agent Content Creation Pipeline

This document outlines the structure of a multi-agent content creation pipeline exercise using **CrewAI in a Jupyter Notebook environment**. The goal is to help students understand how to design, define, and execute a team of AI agents through code.

💮 Problem Statement

You're building a multi-agent system for a real-world content creation pipeline. The objective is to:

- Research a domain or topic
- Formulate a marketing or communication strategy
- Generate creative campaign ideas
- Write content or copy
- Review and approve outputs

You will implement this using CrewAI and Python.

🧪 Exercise 1: Think Through Agents and Tasks

Before coding, students should:

Define Agents

- Agent name
- Role
- Goal
- Backstory

Define Tasks

- Task name
- Assigned agent
- Description
- Expected output

This can be done on paper, Google Docs, or in a Markdown cell.

✍️ Exercise 2: Define Agents in Code

Students will:

1. Import the `Agent` class from CrewAI
2. Create 3-5 agents using their plan from Exercise 1
3. Include a unique role, goal, and backstory for each

Example Template:

```
from crewai import Agent

lead_market_analyst = Agent(
    role="Lead Market Analyst",
    goal="Conduct market and competitor research to support strategic
planning.",
    backstory="You are an experienced market analyst...",
    verbose=True
)
```

📋 Exercise 3: Define Tasks in Code

Students will:

1. Import the `Task` class from CrewAI
2. Define each task with:
    - `description`
    - `expected_output`
    - `agent` (must match an agent defined earlier)
3. Use f-strings to embed shared context
   (like `{project_description}` and `{customer_domain}`)

Example Template:

```
research_task = Task(
    agent=lead_market_analyst,
    description=f"Research the customer and competitors for
{project_description} in {customer_domain}...",
    expected_output="A detailed summary report..."
)
```

🏁 Exercise 4: Run the Crew

Students will:

1. Import and instantiate the `Crew` class
2. Add all tasks to the crew
3. Run the crew using `.run()`

Example:

```
crew = Crew(tasks=[research_task, ...], verbose=True)
```

```
result = crew.run()
print(result)
```

💬 Discuss:

- What worked well?
- What could be improved?

📚 Tips

- Make sure agent objects are not `None`
- Ensure task descriptions are clear and tied to project goals
- Use `verbose=True` to trace agent reasoning
- Add `Markdown` cells between code blocks for explanations

https://github.com/abhinav-kimothi/Agent-Exercise