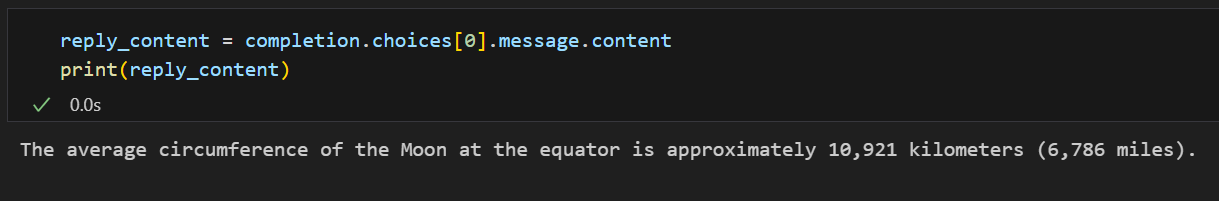
**Notes:**

While coding the simple chatbot from the OpenAI API tutorial, I came across some errors in the AI’s responses where it would hallucinate and mix up values from its answers.



At first it would give the correct answer, however when prompted to answer the same question again, it responded with this:

A computer screen with text and numbers

AI-generated content may be incorrect.

I believe that the AI model has taken the value in miles from the first answer, and used it as the value in kilometres for the second answer, showing how it can easily misconstrue the data.

**What is agentic AI:**

* Agentic workflow = iterative and makes the AI do some more thinking/research than usual
* Zero-shot prompting = not asking the AI for any revisions
* GPT-3.5 with an agentic workflow beats GPT-4 with zero-shot prompting

**Agentic Reasoning Design Patterns:**

1. **Reflection:**

Prompt the AI to solve a problem, feed the AI the solution to the problem and ask to fix any errors that emerge from unit tests

1. **Tool use:**

Using the AI to use external tools for research, analysis, and personal productivity

1. **Planning:**

The AI agent breaks the problem down into smaller problems, then plans multiple solutions to solve each problem

1. **Multi-agent collaboration:**

You can prompt the AI agents to have specific roles in a system to solve problems by using specialisation

**What are AI agents?**

* Shift in models from monolithic to compound systems
* Integrate the model into pre-existing processes
* Compound systems are based on the fact that some problems are better if the principles of system design are applied
* Systems are modular
* Compound systems are much faster and easier to adapt
* Compound systems have programmatic control logic (human, think fast) or agents (LLM, think slow)
* Agentic approach is when the agent takes control of the logic
* LLM agents have the ability to reason, act (via tools) and access memory
* ReAct agents – user query -> plan/think -> act (tools) -> observe -> answer
* Narrow problems -> programmatic route
* Complex problems -> agentic route