Retrieval Augmented Generation (RAG)

AP SCI ACCELERATE PROGRAMME FOR SCIENTIFIC DISCOVERY

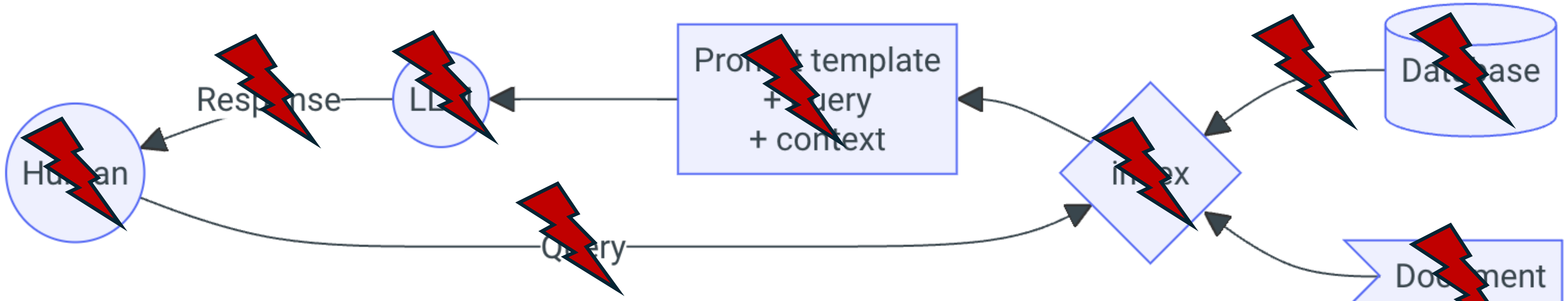
SCHMIDT FUTURES

RAG

60% of the time, it works every time.

- RAG is by far the most popular application of LLMs

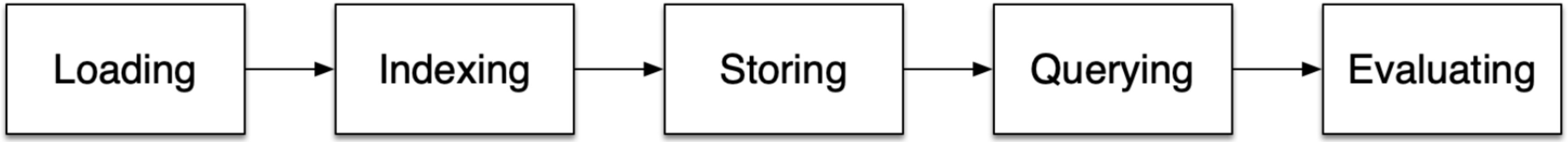
- Entire ecosystems like LangChain and LlamaIndex are built around it



RAG

- There are probably hundreds of tutorials online that will get you up and running with a trivial RAG applications in a few lines of code
- The reality is…

…RAG is HARD!



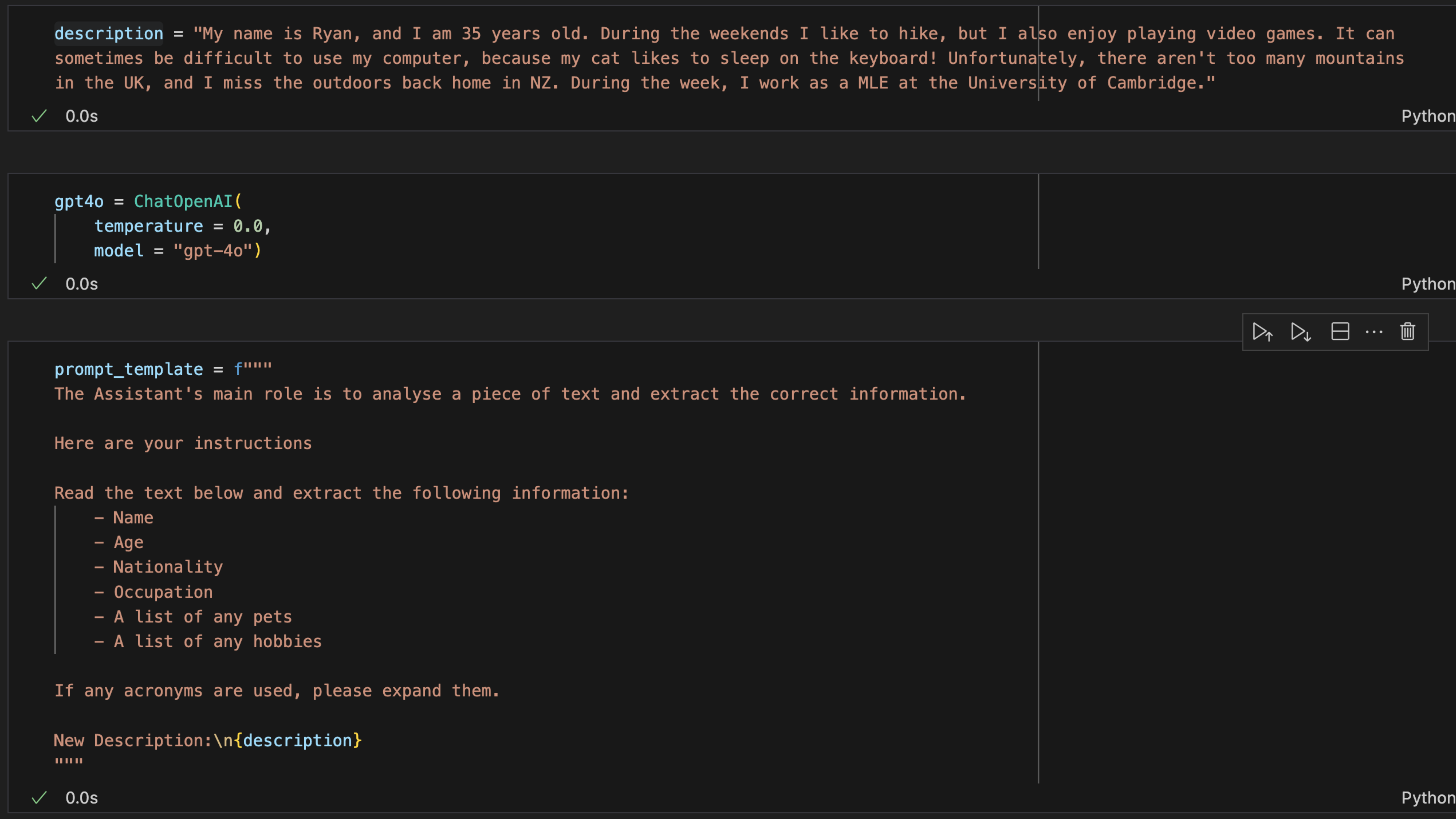
Start being Pydantic…

Suppose I have the following task:

Analyse a text description…

“My name is Ryan, and I am 35 years old. During the weekends I like to hike, but I also enjoy playing video games. It can sometimes be difficult to use my computer, because my cat likes to sleep on the keyboard! Unfortunately, there aren't too many mountains in the UK, and I miss the outdoors back home in NZ. During the week, I work as a MLE at the University of Cambridge.”

…and extract some important information



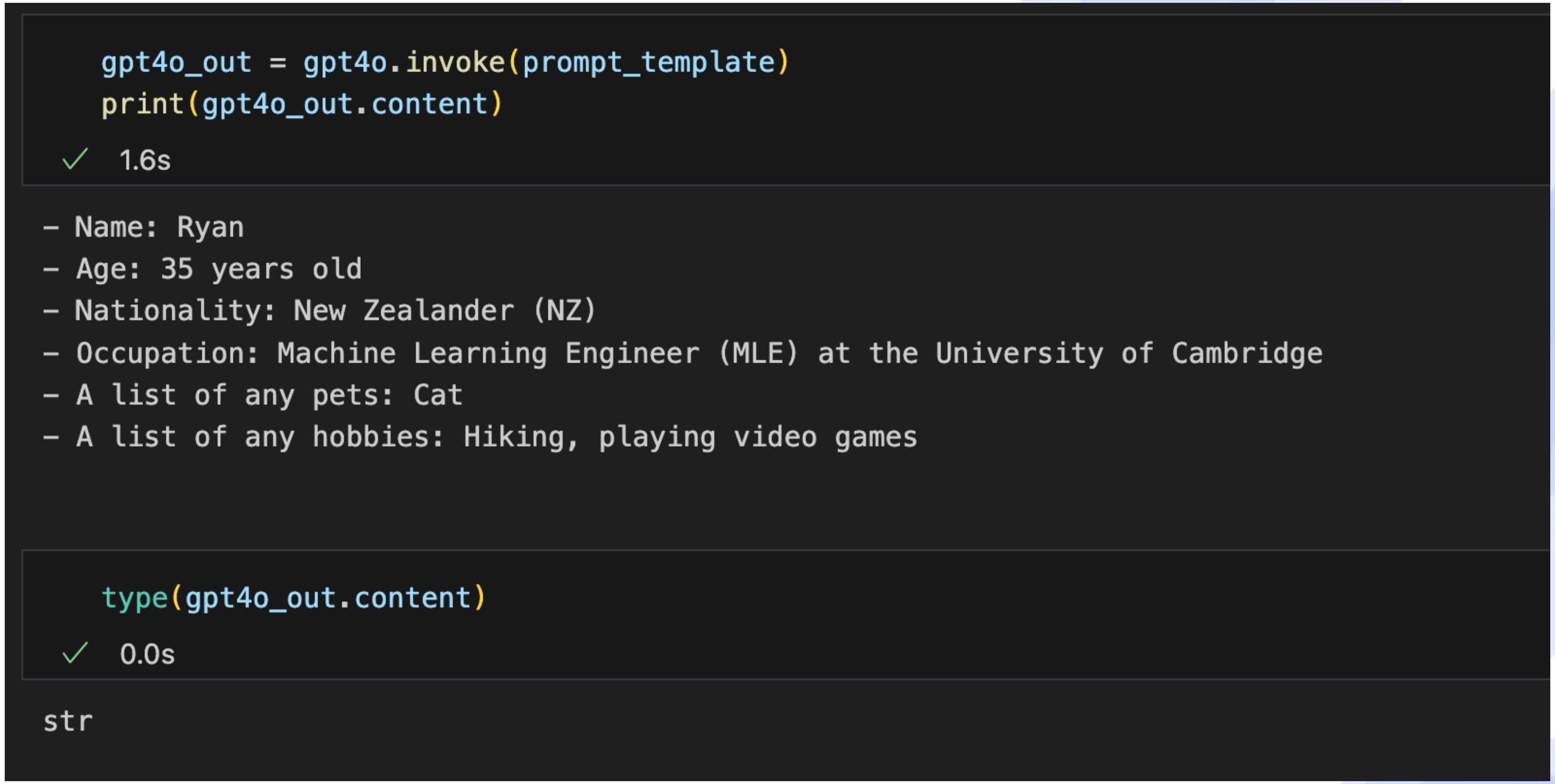
```python
description = "My name is Ryan, and I am 35 years old. During the weekends I like to hike, but I also enjoy playing video games. It can sometimes be difficult to use my computer, because my cat likes to sleep on the keyboard! Unfortunately, there aren't too many mountains in the UK, and I miss the outdoors back home in NZ. During the week, I work as a MLE at the University of Cambridge."
```

```python
gpt4o = ChatOpenAI(
    temperature = 0.0,
    model = "gpt-4o")
```

```python
prompt_template = f"""
The Assistant's main role is to analyse a piece of text and extract the correct information.

Here are your instructions

Read the text below and extract the following information:
    - Name
    - Age
    - Nationality
    - Occupation
    - A list of any pets
    - A list of any hobbies

If any acronyms are used, please expand them.

New Description:\n{description}
"""
```

Start being Pydantic…

Great, so this is easy to do, but this is the output from GPT-4o:



```
gpt4o_out = gpt4o.invoke(prompt_template)
print(gpt4o_out.content)
```

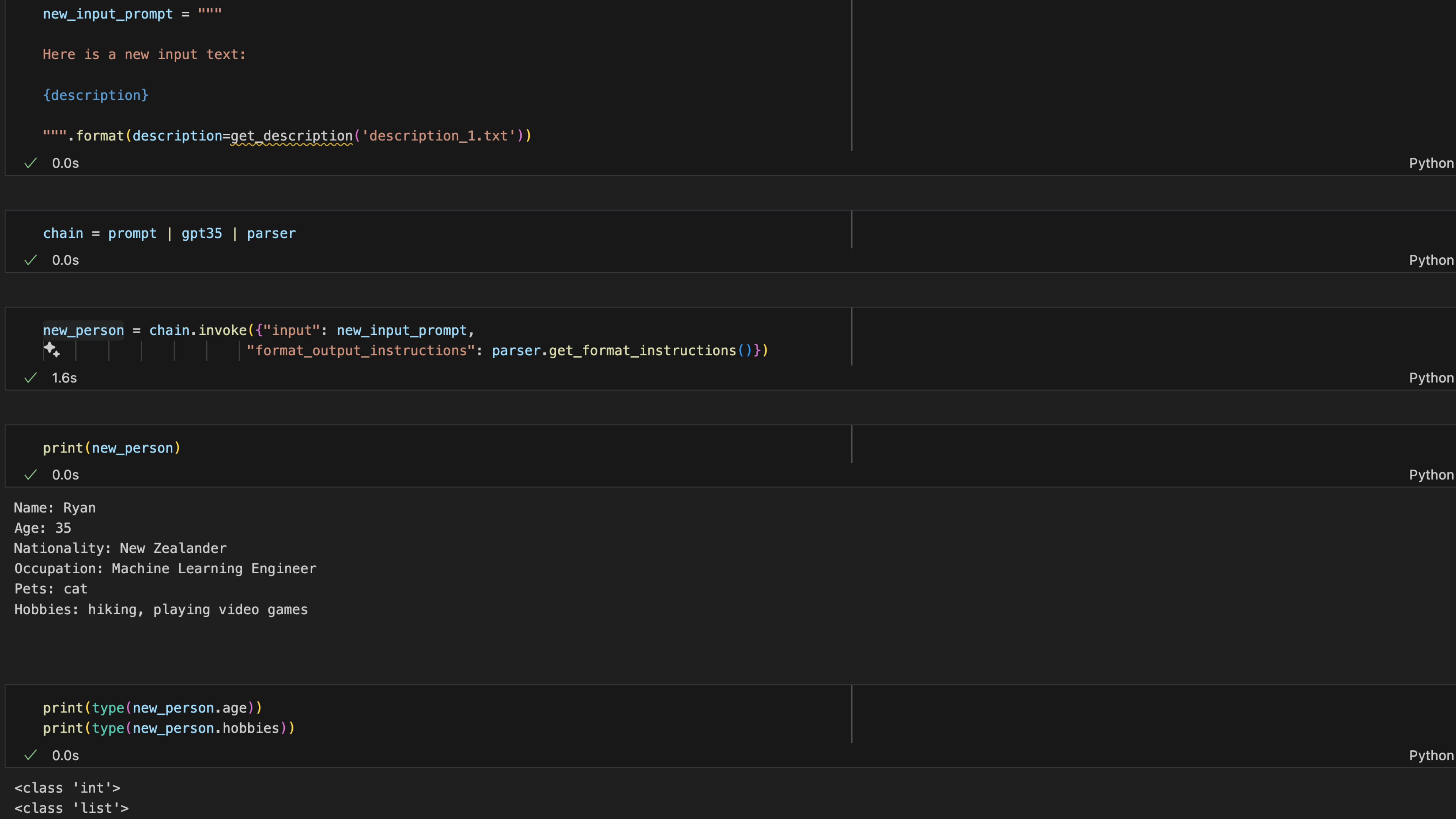
✓ 1.6s

```
- Name: Ryan
- Age: 35 years old
- Nationality: New Zealander (NZ)
- Occupation: Machine Learning Engineer (MLE) at the University of Cambridge
- A list of any pets: Cat
- A list of any hobbies: Hiking, playing video games
```

```
type(gpt4o_out.content)
```

✓ 0.0s

```
str
```



```
new_input_prompt = """

Here is a new input text:

{description}

""".format(description=get_description('description_1.txt'))
```

```
chain = prompt | gpt35 | parser
```

```
new_person = chain.invoke({"input": new_input_prompt,
                         "format_output_instructions": parser.get_format_instructions()})
```

```
print(new_person)
```

```
Name: Ryan
Age: 35
Nationality: New Zealander
Occupation: Machine Learning Engineer
Pets: cat
Hobbies: hiking, playing video games
```

```
print(type(new_person.age))
print(type(new_person.hobbies))
```

```
<class 'int'>
<class 'list'>
```