

Introduction to the OpenAI's API and Prompt Engineering

Notes by Aniket Kundu



```
In [ ]: from openai import OpenAI
key = open("openaikey.txt", "r") # OpenAI API key saved in private text file
client = OpenAI(api_key = key.read()) #api_key is used for authenticating requests
```

```
In [ ]: # we start by creating a request to the completions endpoint
response = client.completions.create(
    model="gpt-3.5-turbo-instruct",
    prompt="What is the OpenAI API?",
)
```

```
In [ ]: print(response)
```

```
Completion(id='cmpl-8n1WaKlLxwaQTdwL9KfC4rUa4pL4v', choices=[CompletionChoice(finish_reason='length', index=0, logprobs=None, text='\n\nThe OpenAI API is an interface that allows developers to access and use Open')], created=1706694624, model='gpt-3.5-turbo-instruct', object='text_completion', system_fingerprint=None, usage=CompletionUsage(completion_tokens=16, prompt_tokens=7, total_tokens=23))
```

```
In [ ]: response.choices[0]
```

```
Out[ ]: CompletionChoice(finish_reason='length', index=0, logprobs=None, text='\n\nThe OpenAI API is an interface that allows developers to access and use Open')
```

```
In [ ]: #extracting the model in dictionary  
response.choices[0].text
```

```
Out[ ]: '\n\nThe OpenAI API is an interface that allows developers to access and use Open'
```

Completions

Model-generated response that attempts to complete the prompt in a likely and consistent way. Completions are used for so-called single-turn tasks, as there is a single prompt and a response.

- Answer Questions
- Classification into categories
- Sentiment Analysis
- Explain complex topics

Chat

- multi turn conversations
 - Ideation
 - Customer support assistant
 - Personal tutor
 - Translate Languages
 - Write Code

Moderation

- Checks for content violations of OpenAI's usage policies, including:
 - Inciting violence
 - Hate Speech
- Can customize model sensitivity to specific violations

In order to use OpenAI library 1.0.0

```
In [ ]: def get_completion(prompt, model="gpt-3.5-turbo", temperature=0):  
        messages = [{"role": "user", "content": prompt}]  
        response = client.chat.completions.create(  
            model=model,  
            messages=messages,  
            temperature=0  
        )  
        return response.choices[0].message.content
```

- Principle 1 : Write clear and specific instructions.
- Principle 2 : Give the model the time to think.

```
In [ ]: text = f"""  
You should express what you want a model to do by \  
providing instructions that are as clear and \  
specific as you can possibly make them. \  
This will guide the model towards the desired output, \  
and reduce the chances of receiving irrelevant \  
or incorrect responses. Don't confuse writing a \  
clear prompt with writing a short prompt. \  
In many cases, longer prompts provide more clarity \  
and context for the model, which can lead to \  
more detailed and relevant outputs.  
"""  
  
prompt = f"""  
Summarize the text delimited by triple backticks \  
into a single sentence.
```

```
```{text}```  
''''
```

## Principle 1: Getting summary of the prompt provided

```
In []: response = get_completion(prompt)
print(response)
```

To guide a model towards the desired output and reduce irrelevant or incorrect responses, it is important to provide clear and specific instructions, which can be achieved through longer prompts that offer more clarity and context.

```
In []: response
```

```
Out[]: 'To guide a model towards the desired output and reduce irrelevant or incorrect responses, it is important to provide clear and specific instructions, which can be achieved through longer prompts that offer more clarity and context.'
```

**Tactic 1 - Use delimiters ```, """, <>, < tag > < /tag > , : to clearly indicate distinct parts of the input**

**Tactic 2 - Ask for a structured output**

```
In []: prompt = f"""
Generate a list of three made-up book titles along \
with their authors and genres.
Provide them in JSON format with the following keys:
book_id, title, author, genre.
"""

response = get_completion(prompt)
print(response)
```

```
{
 "books": [
 {
 "book_id": 1,
 "title": "The Enigma of Elysium",
 "author": "Evelyn Sinclair",
 "genre": "Mystery"
 },
 {
 "book_id": 2,
 "title": "Whispers in the Wind",
 "author": "Nathaniel Blackwood",
 "genre": "Fantasy"
 },
 {
 "book_id": 3,
 "title": "Echoes of the Past",
 "author": "Amelia Hart",
 "genre": "Romance"
 }
]
}
```

### Tactic 3 - Ask the model whether conditions are satisfied or not

```
In []: text_1 = f"""
Making a cup of tea is easy! First, you need to get some \
water boiling. While that's happening, \
grab a cup and put a tea bag in it. Once the water is \
hot enough, just pour it over the tea bag. \
Let it sit for a bit so the tea can steep. After a \
few minutes, take out the tea bag. If you \
like, you can add some sugar or milk to taste. \
And that's it! You've got yourself a delicious \
cup of tea to enjoy.
"""

prompt = f"""
You will be provided with text delimited by triple quotes.
If it contains a sequence of instructions, \
re-write those instructions in the following format:
```

```

Step 1 - ...
Step 2 - ...
...
Step N - ...

If the text does not contain a sequence of instructions, \
then simply write \"No steps provided.\"

\"\"\"{text_1}\"\"\"
"""
response = get_completion(prompt)
print("Completion for Text 1:")
print(response)

```

Completion for Text 1:

```

Step 1 - Get some water boiling.
Step 2 - Grab a cup and put a tea bag in it.
Step 3 - Once the water is hot enough, pour it over the tea bag.
Step 4 - Let it sit for a bit so the tea can steep.
Step 5 - After a few minutes, take out the tea bag.
Step 6 - If you like, add some sugar or milk to taste.
Step 7 - Enjoy your delicious cup of tea.

```

## Tactic 4 : "Few Shot prompting"

```

In []: prompt = f"""
Your task is to answer in a consistent style.

<child>: Teach me about patience.

<grandparent>: The river that carves the deepest \
valley flows from a modest spring; the \
grandest symphony originates from a single note; \
the most intricate tapestry begins with a solitary thread.

<child>: Teach me about resilience.
"""
response = get_completion(prompt)
print(response)

```

<grandparent>: Resilience is the unwavering strength that emerges from facing adversity. It is the ability to bounce back, to rise above challenges, and to persevere in the face of setbacks. Like a mighty oak tree that withstands the fiercest storms, resilience allows us to weather the storms of life and come out stronger on the other side.

## Give the model time to "think"

### Tactic 1 - Specify the steps required to complete a task

```
In []: text = f"""
In a charming village, siblings Jack and Jill set out on \
a quest to fetch water from a hilltop \
well. As they climbed, singing joyfully, misfortune \
struck—Jack tripped on a stone and tumbled \
down the hill, with Jill following suit. \
Though slightly battered, the pair returned home to \
comforting embraces. Despite the mishap, \
their adventurous spirits remained undimmed, and they \
continued exploring with delight.
"""

example 1
prompt_1 = f"""
Perform the following actions:
1 – Summarize the following text delimited by triple \
backticks with 1 sentence.
2 – Translate the summary into French.
3 – List each name in the French summary.
4 – Output a json object that contains the following \
keys: french_summary, num_names.

Separate your answers with line breaks.

Text:
```{text}```
"""

response = get_completion(prompt_1)
print("Completion for prompt 1:")
print(response)
```

Completion for prompt 1:

1 – Jack and Jill, siblings, go on a quest to fetch water from a hilltop well, but encounter misfortune when Jack trips on a stone and tumbles down the hill, with Jill following suit, yet they return home and remain undeterred in their adventurous spirits.

2 – Jack et Jill, frère et sœur, partent en quête d'eau d'un puits au sommet d'une colline, mais rencontrent un malheur lorsque Jack trébuche sur une pierre et dévale la colline, suivi par Jill, pourtant ils rentrent chez eux et restent déterminés dans leur esprit d'aventure.

3 – Jack, Jill

4 – {

"french_summary": "Jack et Jill, frère et sœur, partent en quête d'eau d'un puits au sommet d'une colline, mais rencontrent un malheur lorsque Jack trébuche sur une pierre et dévale la colline, suivi par Jill, pourtant ils rentrent chez eux et restent déterminés dans leur esprit d'aventure.",

"num_names": 2

}

Ask for output in a specified format

```
In [ ]: prompt_2 = f"""
Your task is to perform the following actions:
1 – Summarize the following text delimited by
    <> with 1 sentence.
2 – Translate the summary into French.
3 – List each name in the French summary.
4 – Output a json object that contains the
    following keys: french_summary, num_names.

Use the following format:
Text: <text to summarize>
Summary: <summary>
Translation: <summary translation>
Names: <list of names in summary>
Output JSON: <json with summary and num_names>

Text: <{text}>
"""
response = get_completion(prompt_2)
```



```
print("\nCompletion for prompt 2:")
print(response)
```

Completion for prompt 2:

Summary: Jack and Jill, siblings from a charming village, go on a quest to fetch water from a hilltop well, but encounter misfortune when Jack trips on a stone and tumbles down the hill, with Jill following suit, yet they return home and continue exploring with delight.

Translation: Jack et Jill, frère et sœur d'un charmant village, partent en quête d'eau d'un puits au sommet d'une colline, mais rencontrent un malheur lorsque Jack trébuche sur une pierre et dévale la colline, suivi par Jill, mais ils rentrent chez eux et continuent à explorer avec joie.

Names: Jack, Jill

Output JSON: {"french_summary": "Jack et Jill, frère et sœur d'un charmant village, partent en quête d'eau d'un puits au sommet d'une colline, mais rencontrent un malheur lorsque Jack trébuche sur une pierre et dévale la colline, suivi par Jill, mais ils rentrent chez eux et continuent à explorer avec joie.", "num_names": 2}

Instruct the model to work out its own solution before rushing to a conclusion

NOTE that the student's solution is actually not correct.

```
In [ ]: prompt = f"""
Determine if the student's solution is correct or not.

Question:
I'm building a solar power installation and I need \
help working out the financials.
- Land costs $100 / square foot
- I can buy solar panels for $250 / square foot
- I negotiated a contract for maintenance that will cost \
me a flat $100k per year, and an additional $10 / square \
foot
What is the total cost for the first year of operations
as a function of the number of square feet.

Student's Solution:
Let x be the size of the installation in square feet.
Costs:
```

```

1. Land cost: 100x
2. Solar panel cost: 250x
3. Maintenance cost: 100,000 + 100x
Total cost: 100x + 250x + 100,000 + 100x = 450x + 100,000
"""

response = get_completion(prompt)
print(response)

```

The student's solution is correct. They correctly identified the costs for land, solar panels, and maintenance, and calculated the total cost for the first year of operations as a function of the number of square feet.

We can fix this by instructing the model to work out its own solution first

```

In [ ]: prompt = f"""
Your task is to determine if the student's solution \
is correct or not.
To solve the problem do the following:
- First, work out your own solution to the problem including the final total.
- Then compare your solution to the student's solution \
and evaluate if the student's solution is correct or not.
Don't decide if the student's solution is correct until
you have done the problem yourself.

Use the following format:
Question:
"""

question here
"""

Student's solution:
"""

student's solution here
"""

Actual solution:
"""

steps to work out the solution and your solution here
"""

Is the student's solution the same as actual solution \
just calculated:
"""

yes or no

```

```

'''
Student grade:
'''

correct or incorrect
'''

Question:
'''
I'm building a solar power installation and I need help \
working out the financials.
- Land costs $100 / square foot
- I can buy solar panels for $250 / square foot
- I negotiated a contract for maintenance that will cost \
me a flat $100k per year, and an additional $10 / square \
foot
What is the total cost for the first year of operations \
as a function of the number of square feet.
'''

Student's solution:
'''
Let x be the size of the installation in square feet.
Costs:
1. Land cost: 100x
2. Solar panel cost: 250x
3. Maintenance cost: 100,000 + 100x
Total cost: 100x + 250x + 100,000 + 100x = 450x + 100,000
'''

Actual solution:
'''
response = get_completion(prompt)
print(response)

```

To calculate the total cost for the first year of operations, we need to add up the costs of land, solar panels, and maintenance.

1. Land cost: \$100 / square foot

The cost of land is \$100 multiplied by the size of the installation in square feet.

2. Solar panel cost: \$250 / square foot

The cost of solar panels is \$250 multiplied by the size of the installation in square feet.

3. Maintenance cost: \$100,000 + \$10 / square foot

The maintenance cost is a flat fee of \$100,000 per year, plus \$10 multiplied by the size of the installation in square feet.

Total cost: Land cost + Solar panel cost + Maintenance cost

Let's calculate the total cost using the actual solution:

$$\begin{aligned}\text{Total cost} &= (100 * x) + (250 * x) + (100,000 + (10 * x)) \\ &= 100x + 250x + 100,000 + 10x \\ &= 360x + 100,000\end{aligned}$$

Is the student's solution the same as the actual solution just calculated:

No

Student grade:

Incorrect

Model Hallucinations

```
In [ ]: # Boie is a real company, the product name is not real
prompt = "Tell me about AeroGlide UltraSlim Smart Toothbrush by Boie"
response = get_completion(prompt)
print(response)
```

The AeroGlide UltraSlim Smart Toothbrush is a revolutionary oral care product developed by Boie, a company known for its innovative approach to personal hygiene. This toothbrush is designed to provide a superior cleaning experience while being gentle on the gums and teeth.

One of the standout features of the AeroGlide UltraSlim Smart Toothbrush is its ultra-slim bristles. These bristles are only 0.04mm thick, making them incredibly soft and flexible. This design allows the toothbrush to reach even the most difficult-to-access areas of the mouth, ensuring a thorough and effective cleaning.

The toothbrush also incorporates smart technology, making it a truly advanced oral care device. It connects to a smartphone app via Bluetooth, which provides real-time feedback on brushing technique and duration. This feature helps users improve their brushing habits and maintain optimal oral health.

The AeroGlide UltraSlim Smart Toothbrush is made from a durable and hygienic material called thermoplastic elastomer (TPE). This material is free from harmful chemicals like BPA and latex, making it safe for use. Additionally, TPE is antimicrobial, meaning it resists the growth of bacteria, mold, and mildew, ensuring a clean and healthy brushing experience.

Another notable aspect of this toothbrush is its eco-friendly design. Boie is committed to sustainability, and the AeroGlide UltraSlim Smart Toothbrush reflects this ethos. The brush head is replaceable, reducing waste, and the handle is made from a recyclable material, making it an environmentally conscious choice.

In summary, the AeroGlide UltraSlim Smart Toothbrush by Boie is a cutting-edge oral care product that combines advanced technology, gentle cleaning, and eco-friendly design. With its ultra-slim bristles, smart features, and sustainable materials, this toothbrush offers a unique and effective way to maintain optimal oral hygiene.

```
In [ ]: print(get_completion("Tell me about Macbook air M5 chip by Apple")) # doesn't exist as of now (31st January, 2024)
```

The MacBook Air M5 chip refers to the fifth-generation Intel Core M processor used in certain models of the MacBook Air lineup by Apple. The M5 chip is a low-power, energy-efficient processor designed for thin and light laptops like the MacBook Air.

Here are some key features and specifications of the MacBook Air M5 chip:

1. Performance: The M5 chip offers improved performance compared to its predecessors, providing a balance between power efficiency and processing capabilities. It is suitable for everyday tasks like web browsing, document editing, multimedia consumption, and light multitasking.
2. Architecture: The M5 chip is based on Intel's Skylake microarchitecture, which provides better power management and performance optimization. It has a dual-core design with Hyper-Threading technology, allowing it to handle up to four threads simultaneously.
3. Clock Speed: The M5 chip typically operates at a base clock speed of around 1.2 GHz, but it can dynamically boost its frequency up to 2.7 GHz for demanding tasks. This allows for smooth performance during regular usage and occasional bursts of power when needed.
4. Graphics: The M5 chip integrates Intel HD Graphics 515, which provides decent graphics performance for casual gaming, video playback, and graphics-intensive applications. While not designed for heavy gaming or professional graphics work, it can handle most everyday graphical tasks with ease.
5. Power Efficiency: One of the key advantages of the M5 chip is its power efficiency. It is designed to consume less energy, resulting in longer battery life for the MacBook Air. This makes it ideal for users who prioritize portability and extended usage away from power outlets.
6. Thermal Design: The M5 chip's low power consumption also contributes to its thermal design. It generates less heat compared to higher-performance processors, allowing for a fanless design in the MacBook Air. This results in a quieter and more compact laptop.

Overall, the MacBook Air M5 chip offers a good balance between performance and power efficiency, making it suitable for everyday tasks and light usage scenarios. However, it may not be the best choice for users with demanding computational needs or those who require high-end graphics performance.

Iterative Prompt Development

```
In [ ]: fact_sheet_chair = """  
OVERVIEW  
- Part of a beautiful family of mid-century inspired office furniture,
```

including filing cabinets, desks, bookcases, meeting tables, and more.

- Several options of shell color and base finishes.
- Available with plastic back and front upholstery (SWC-100) or full upholstery (SWC-110) in 10 fabric and 6 leather options.
- Base finish options are: stainless steel, matte black, gloss white, or chrome.
- Chair is available with or without armrests.
- Suitable for home or business settings.
- Qualified for contract use.

CONSTRUCTION

- 5-wheel plastic coated aluminum base.
- Pneumatic chair adjust for easy raise/lower action.

DIMENSIONS

- WIDTH 53 CM | 20.87"
- DEPTH 51 CM | 20.08"
- HEIGHT 80 CM | 31.50"
- SEAT HEIGHT 44 CM | 17.32"
- SEAT DEPTH 41 CM | 16.14"

OPTIONS

- Soft or hard-floor caster options.
- Two choices of seat foam densities: medium (1.8 lb/ft3) or high (2.8 lb/ft3)
- Armless or 8 position PU armrests

MATERIALS

SHELL BASE GLIDER

- Cast Aluminum with modified nylon PA6/PA66 coating.
- Shell thickness: 10 mm.

SEAT

- HD36 foam

COUNTRY OF ORIGIN

- Italy
- """"

```
In [ ]: prompt = f"""\nYour task is to help a marketing team create a
```

description for a retail website of a product based on a technical fact sheet.

Write a product description based on the information provided in the technical specifications delimited by triple backticks.

Technical specifications: ```{fact_sheet_chair}```
"""

```
response = get_completion(prompt)
print(response)
```


Introducing our stunning mid-century inspired office chair, the perfect addition to any home or business setting. This chair is part of a beautiful family of office furniture, including filing cabinets, desks, bookcases, meeting tables, and more, all designed with a timeless mid-century aesthetic.

One of the standout features of this chair is the variety of customization options available. You can choose from several shell colors and base finishes to perfectly match your existing decor. The chair is available with either plastic back and front upholstery or full upholstery in a range of 10 fabric and 6 leather options, allowing you to create a look that is uniquely yours.

The chair is also available with or without armrests, giving you the flexibility to choose the option that best suits your needs. The base finish options include stainless steel, matte black, gloss white, or chrome, ensuring that you can find the perfect match for your space.

In terms of construction, this chair is built to last. It features a 5-wheel plastic coated aluminum base, providing stability and mobility. The pneumatic chair adjust allows for easy raise and lower action, ensuring optimal comfort throughout the day.

When it comes to dimensions, this chair is designed with both style and comfort in mind. With a width of 53 cm (20.87"), depth of 51 cm (20.08"), and height of 80 cm (31.50"), it offers ample space without overwhelming your space. The seat height is 44 cm (17.32") and the seat depth is 41 cm (16.14"), providing a comfortable seating experience for users of all heights.

We understand that every space is unique, which is why we offer a range of options to further customize your chair. You can choose between soft or hard-floor caster options, ensuring that your chair glides smoothly across any surface. Additionally, you have the choice between two seat foam densities: medium (1.8 lb/ft³) or high (2.8 lb/ft³), allowing you to select the level of support that suits your preferences. The chair is also available with armless design or 8 position PU armrests, providing additional comfort and versatility.

When it comes to materials, this chair is crafted with the utmost attention to quality. The shell base glider is made from cast aluminum with a modified nylon PA6/PA66 coating, ensuring durability and longevity. The shell thickness is 10 mm, providing a sturdy and reliable structure. The seat is made from HD36 foam, offering a comfortable and supportive seating experience.

Finally, this chair is proudly made in Italy, known for its exceptional craftsmanship and design. With its combination of style, functionality, and customization options, this mid-century inspired office chair is the perfect choice for those seeking a timeless and elegant addition to their workspace.

Please note that this chair is qualified for contract use, making it suitable for a wide range of professional settings. Upgrade your office or home with this exceptional piece of furniture and experience the perfect blend of style and comfort.

Issue 1 : The text is too long

```
In [ ]: prompt = f"""
Your task is to help a marketing team create a
description for a retail website of a product based
on a technical fact sheet.

Write a product description based on the information
provided in the technical specifications delimited by
triple backticks.

Use at most 50 words.

Technical specifications: ```{fact_sheet_chair}```
"""
response = get_completion(prompt)
print(response)
```

Introducing our mid-century inspired office chair, part of a stunning furniture collection. With various color and finish options, choose between plastic or full upholstery in fabric or leather. The chair features a durable aluminum base with 5 wheels and pneumatic height adjustment. Perfect for home or business use. Made in Italy.

```
In [ ]: len(response.split())
```

```
Out[ ]: 51
```

Text focuses on the wrong details

```
In [ ]: prompt = f"""
Your task is to help a marketing team create a
description for a retail website of a product based
on a technical fact sheet.

Write a product description based on the information
provided in the technical specifications delimited by
triple backticks.

The description is intended for furniture retailers,
```

```
so should be technical in nature and focus on the
materials the product is constructed from.
```

```
At the end of the description, include every 7-character
Product ID in the technical specification.
```

```
Use at most 50 words.
```

```
Technical specifications: ```{fact_sheet_chair}```
"""
```

```
response = get_completion(prompt)
print(response)
```

Introducing our mid-century inspired office chair, part of a beautiful family of furniture. With various shell colors and base finishes, this chair offers versatility and style. Choose between plastic or full upholstery in a range of fabric and leather options. The chair features a 5-wheel plastic coated aluminum base and a pneumatic chair adjust for easy height adjustment. Available with or without armrests, this chair is suitable for both home and business settings. Made with high-quality materials, including a cast aluminum shell and HD36 foam seat, this chair is built to last. Product ID: SWC-100, SWC-110.

```
In [ ]: len(response.split())
```

```
Out[ ]: 96
```

Description needs a table of dimensions

```
In [ ]: prompt = f"""
```

```
Your task is to help a marketing team create a
description for a retail website of a product based
on a technical fact sheet.
```

```
Write a product description based on the information
provided in the technical specifications delimited by
triple backticks.
```

```
The description is intended for furniture retailers,
so should be technical in nature and focus on the
materials the product is constructed from.
```

```
At the end of the description, include every 7-character
```

Product ID in the technical specification.

After the description, include a table that gives the product's dimensions. The table should have two columns. In the first column include the name of the dimension. In the second column include the measurements in inches only.

Give the table the title 'Product Dimensions'.

Format everything as HTML that can be used in a website. Place the description in a <div> element.

Technical specifications: ``{fact_sheet_chair}``
"""

```
response = get_completion(prompt)
print(response)
```

<div>

<h2>Product Description</h2>

<p>

Introducing our latest addition to our mid-century inspired office furniture collection – the SWC Chair. This chair is part of a beautiful family of furniture that includes filing cabinets, desks, bookcases, meeting tables, and more. With its sleek design and customizable options, the SWC Chair is the perfect choice for any home or business setting.

</p>

<p>

The SWC Chair is available in several options of shell color and base finishes, allowing you to create a look that matches your style. You can choose between plastic back and front upholstery or full upholstery in a variety of fabric and leather options. The base finish options include stainless steel, matte black, gloss white, or chrome. Additionally, you have the choice of having the chair with or without armrests.

</p>

<p>

Constructed with durability in mind, the SWC Chair features a 5-wheel plastic coated aluminum base, ensuring stability and easy mobility. The chair also has a pneumatic adjuster, allowing for easy raise and lower action to find the perfect height for your comfort.

</p>

<p>

The SWC Chair is not only stylish and functional, but it is also designed with your comfort in mind. The seat is made with HD36 foam, providing a comfortable and supportive seating experience. You also have the option to choose between soft or hard-floor caster options, depending on your flooring needs. Additionally, you can select between two choices of seat foam densities – medium (1.8 lb/ft³) or high (2.8 lb/ft³). The chair is also available with armless design or 8 position PU armrests for added convenience.

</p>

<p>

Made with high-quality materials, the SWC Chair is built to last. The shell base glider is constructed with cast aluminum and modified nylon PA6/PA66 coating, ensuring durability and stability. The shell has a thickness of 10 mm, providing strength and support. The chair is proudly made in Italy, known for its craftsmanship and attention to detail.

</p>

<p>

The SWC Chair is not only a stylish addition to any space, but it is also a practical choice for both home and business settings. With its customizable options, durable construction, and comfortable design, the SWC Chair is the perfect seating solution for any environment.

</p>

<h2>Product Dimensions</h2>

<table>

<tr>

<th>Dimension</th>

```
        <th>Measurement (inches)</th>
    </tr>
    <tr>
        <td>Width</td>
        <td>20.87"</td>
    </tr>
    <tr>
        <td>Depth</td>
        <td>20.08"</td>
    </tr>
    <tr>
        <td>Height</td>
        <td>31.50"</td>
    </tr>
    <tr>
        <td>Seat Height</td>
        <td>17.32"</td>
    </tr>
    <tr>
        <td>Seat Depth</td>
        <td>16.14"</td>
    </tr>
</table>
</div>
```

Product IDs: SWC-100, SWC-110

```
In [ ]: from IPython.display import display, HTML
```

```
In [ ]: display(HTML(response))
```

Product Description

Introducing our latest addition to our mid-century inspired office furniture collection - the SWC Chair. This chair is part of a beautiful family of furniture that includes filing cabinets, desks, bookcases, meeting tables, and more. With its sleek design and customizable options, the SWC Chair is the perfect choice for any home or business setting.

The SWC Chair is available in several options of shell color and base finishes, allowing you to create a look that matches your style. You can choose between plastic back and front upholstery or full upholstery in a variety of fabric and leather options. The base finish options include stainless steel, matte black, gloss white, or chrome. Additionally, you have the choice of having the chair with or without armrests.

Constructed with durability in mind, the SWC Chair features a 5-wheel plastic coated aluminum base, ensuring stability and easy mobility. The chair also has a pneumatic adjuster, allowing for easy raise and lower action to find the perfect height for your comfort.

The SWC Chair is not only stylish and functional, but it is also designed with your comfort in mind. The seat is made with HD36 foam, providing a comfortable and supportive seating experience. You also have the option to choose between soft or hard-floor caster options, depending on your flooring needs. Additionally, you can select between two choices of seat foam densities - medium (1.8 lb/ft³) or high (2.8 lb/ft³). The chair is also available with armless design or 8 position PU armrests for added convenience.

Made with high-quality materials, the SWC Chair is built to last. The shell base glider is constructed with cast aluminum and modified nylon PA6/PA66 coating, ensuring durability and stability. The shell has a thickness of 10 mm, providing strength and support. The chair is proudly made in Italy, known for its craftsmanship and attention to detail.

The SWC Chair is not only a stylish addition to any space, but it is also a practical choice for both home and business settings. With its customizable options, durable construction, and comfortable design, the SWC Chair is the perfect seating solution for any environment.

Product Dimensions

Dimension	Measurement (inches)
Width	20.87"
Depth	20.08"
Height	31.50"

Seat Height	17.32"
Seat Depth	16.14"

Product IDs: SWC-100, SWC-110

Summarizing

```
In [ ]: prod_review = """
Got this panda plush toy for my daughter's birthday, \
who loves it and takes it everywhere. It's soft and \
super cute, and its face has a friendly look. It's \
a bit small for what I paid though. I think there \
might be other options that are bigger for the \
same price. It arrived a day earlier than expected, \
so I got to play with it myself before I gave it \
to her.
"""
```

```
In [ ]: prompt = f"""
Your task is to generate a short summary of a product \
review from an ecommerce site.

Summarize the review below, delimited by triple
backticks, in at most 30 words.

Review: ```{prod_review}```
"""

response = get_completion(prompt)
print(response)
```

This panda plush toy is loved by the reviewer's daughter, but they feel it is a bit small for the price.

Summarize with focus on some aspect


```
In [ ]: prompt = f"""
Your task is to generate a short summary of a product \
review from an ecommerce site to give feedback to the \
pricing department, responsible for determining the \
price of the product.

Summarize the review below, delimited by triple
backticks, in at most 30 words, and focusing on any aspects \
that are relevant to the price and perceived value.

Review: ```{prod_review}```
"""

response = get_completion(prompt)
print(response)
```

The customer loves the panda plush toy for its softness and cuteness, but feels it is overpriced compared to other options available.

Try "extract" instead of "summarize"

```
In [ ]: prompt = f"""
Your task is to extract relevant information from \
a product review from an ecommerce site to give \
feedback to the Shipping department.

From the review below, delimited by triple quotes \
extract the information relevant to shipping and \
delivery. Limit to 30 words.

Review: ```{prod_review}```
"""

response = get_completion(prompt)
print(response)
```

The shipping department should take note that the product arrived a day earlier than expected.

Summarize multiple product reviews

```
In [ ]: review_1 = prod_review

# review for a standing lamp
review_2 = """
Needed a nice lamp for my bedroom, and this one \
had additional storage and not too high of a price \
point. Got it fast – arrived in 2 days. The string \
to the lamp broke during the transit and the company \
happily sent over a new one. Came within a few days \
as well. It was easy to put together. Then I had a \
missing part, so I contacted their support and they \
very quickly got me the missing piece! Seems to me \
to be a great company that cares about their customers \
and products.
"""

# review for an electric toothbrush
review_3 = """
My dental hygienist recommended an electric toothbrush, \
which is why I got this. The battery life seems to be \
pretty impressive so far. After initial charging and \
leaving the charger plugged in for the first week to \
condition the battery, I've unplugged the charger and \
been using it for twice daily brushing for the last \
3 weeks all on the same charge. But the toothbrush head \
is too small. I've seen baby toothbrushes bigger than \
this one. I wish the head was bigger with different \
length bristles to get between teeth better because \
this one doesn't. Overall if you can get this one \
around the $50 mark, it's a good deal. The manufactuer's \
replacements heads are pretty expensive, but you can \
get generic ones that're more reasonably priced. This \
toothbrush makes me feel like I've been to the dentist \
every day. My teeth feel sparkly clean!
"""

# review for a blender
```

```
review_4 = """
So, they still had the 17 piece system on seasonal \
sale for around $49 in the month of November, about \
half off, but for some reason (call it price gouging) \
around the second week of December the prices all went \
up to about anywhere from between $70-$89 for the same \
system. And the 11 piece system went up around $10 or \
so in price also from the earlier sale price of $29. \
So it looks okay, but if you look at the base, the part \
where the blade locks into place doesn't look as good \
as in previous editions from a few years ago, but I \
plan to be very gentle with it (example, I crush \
very hard items like beans, ice, rice, etc. in the \
blender first then pulverize them in the serving size \
I want in the blender then switch to the whipping \
blade for a finer flour, and use the cross cutting blade \
first when making smoothies, then use the flat blade \
if I need them finer/less pulpy). Special tip when making \
smoothies, finely cut and freeze the fruits and \
vegetables (if using spinach-lightly stew soften the \
spinach then freeze until ready for use-and if making \
sorbet, use a small to medium sized food processor) \
that you plan to use that way you can avoid adding so \
much ice if at all-when making your smoothie. \
After about a year, the motor was making a funny noise. \
I called customer service but the warranty expired \
already, so I had to buy another one. FYI: The overall \
quality has gone down in these types of products, so \
they are kind of counting on brand recognition and \
consumer loyalty to maintain sales. Got it in about \
two days.
"""

reviews = [review_1, review_2, review_3, review_4]
```

```
In [ ]: reviews
```

```
Out[ ]: ["\nGot this panda plush toy for my daughter's birthday, who loves it and takes it everywhere. It's soft and \n
super cute, and its face has a friendly look. It's \nna bit small for what I paid though. I think there \n
might be other options that are bigger for the \n
same price. It arrived a day earlier than expected, \nso I got
to play with it myself before I gave it \n
to her.\n",
'\nNeeded a nice lamp for my bedroom, and this one had additional storage and not too high of a price point. Got
it fast - arrived in 2 days. The string to the lamp broke during the transit and the company happily sent over a n
ew one. Came within a few days as well. It was easy to put together. Then I had a missing part, so I contacted the
ir support and they very quickly got me the missing piece! Seems to me to be a great company that cares about thei
r customers and products. \n',
"\nMy dental hygienist recommended an electric toothbrush, which is why I got this. The battery life seems to be
pretty impressive so far. After initial charging and leaving the charger plugged in for the first week to conditio
n the battery, I've unplugged the charger and been using it for twice daily brushing for the last 3 weeks all on t
he same charge. But the toothbrush head is too small. I've seen baby toothbrushes bigger than this one. I wish the
head was bigger with different length bristles to get between teeth better because this one doesn't. Overall if y
ou can get this one around the $50 mark, it's a good deal. The manufactuer's replacements heads are pretty expensi
ve, but you can get generic ones that're more reasonably priced. This toothbrush makes me feel like I've been to t
he dentist every day. My teeth feel sparkly clean! \n",
'\nSo, they still had the 17 piece system on seasonal sale for around $49 in the month of November, about half of
f, but for some reason (call it price gouging) around the second week of December the prices all went up to about
anywhere from between $70-$89 for the same system. And the 11 piece system went up around $10 or so in price also
from the earlier sale price of $29. So it looks okay, but if you look at the base, the part where the blade locks
into place doesn't look as good as in previous editions from a few years ago, but I plan to be very gentle with it
(example, I crush very hard items like beans, ice, rice, etc. in the \n
blender first then pulverize them in the
serving size I want in the blender then switch to the whipping blade for a finer flour, and use the cross cutting
blade first when making smoothies, then use the flat blade if I need them finer/less pulpy). Special tip when maki
ng smoothies, finely cut and freeze the fruits and vegetables (if using spinach-lightly stew soften the \n
spinach then freeze until ready for use-and if making sorbet, use a small to medium sized food processor) \n
that you
plan to use that way you can avoid adding so much ice if at all-when making your smoothie. After about a year, the
motor was making a funny noise. I called customer service but the warranty expired already, so I had to buy anothe
r one. FYI: The overall quality has gone done in these types of products, so they are kind of counting on brand re
cognition and consumer loyalty to maintain sales. Got it in about two days.\n']
```

```
In [ ]: for i in range(len(reviews)):
        prompt = f"""
        Your task is to generate a short summary of a product \
        review from an ecommerce site.

        Summarize the review below, delimited by triple \
        backticks in at most 20 words.
```

```
Review: ```{reviews[i]}```  

  

response = get_completion(prompt)  

print(i, response, "\n")
```

- 0 Soft and cute panda plush toy loved by daughter, but a bit small for the price. Arrived early.
- 1 Great lamp with storage, fast delivery, excellent customer service, and easy assembly. Highly recommended.
- 2 Impressive battery life, small toothbrush head, good deal if priced around \$50, makes teeth feel clean.
- 3 The reviewer found the price increase after the sale disappointing and noticed a decrease in quality.

Identify the types of emotions

```
In [ ]: lamp_review = """  

Needed a nice lamp for my bedroom, and this one had \  

additional storage and not too high of a price point. \  

Got it fast. The string to our lamp broke during the \  

transit and the company happily sent over a new one. \  

Came within a few days as well. It was easy to put \  

together. I had a missing part, so I contacted their \  

support and they very quickly got me the missing piece! \  

Lumina seems to me to be a great company that cares \  

about their customers and products!!  

"""
```

```
In [ ]: prompt = f"""  

What is the sentiment of the following product review,  

which is delimited with triple backticks?  

  

Review text: ```{lamp_review}```  

  

response = get_completion(prompt)  

print(response)
```

The sentiment of the product review is positive.

```
In [ ]: prompt = f"""
Identify a list of emotions that the writer of the \
following review is expressing. Include no more than \
five items in the list. Format your answer as a list of \
lower-case words separated by commas.

Review text: '{lamp_review}'
"""
response = get_completion(prompt)
print(response)
```

happy, satisfied, impressed, grateful, pleased

```
In [ ]: prompt = f"""
Identify the following items from the review text:
- Item purchased by reviewer
- Company that made the item

The review is delimited with triple backticks. \
Format your response as a JSON object with \
"Item" and "Brand" as the keys.
If the information isn't present, use "unknown" \
as the value.
Make your response as short as possible.

Review text: '{lamp_review}'
"""
response = get_completion(prompt)
print(response)
```

```
{
  "Item": "lamp",
  "Brand": "Lumina"
}
```

```
In [ ]: story = """
In a recent survey conducted by the government,
public sector employees were asked to rate their level
of satisfaction with the department they work at.
The results revealed that NASA was the most popular
department with a satisfaction rating of 95%.
```

One NASA employee, John Smith, commented on the findings, stating, "I'm not surprised that NASA came out on top. It's a great place to work with amazing people and incredible opportunities. I'm proud to be a part of such an innovative organization."

The results were also welcomed by NASA's management team, with Director Tom Johnson stating, "We are thrilled to hear that our employees are satisfied with their work at NASA. We have a talented and dedicated team who work tirelessly to achieve our goals, and it's fantastic to see that their hard work is paying off."

The survey also revealed that the Social Security Administration had the lowest satisfaction rating, with only 45% of employees indicating they were satisfied with their job. The government has pledged to address the concerns raised by employees in the survey and work towards improving job satisfaction across all departments.

""

Inferring more than one topics

```
In [ ]: prompt = f"""
Determine five topics that are being discussed in the \
following text, which is delimited by triple backticks.

Make each item one or two words long.

Format your response as a list of items separated by commas.

Text sample: '''{story}'''
""
response = get_completion(prompt)
print(response)
```

1. Government survey
2. Department satisfaction rating
3. NASA
4. Social Security Administration
5. Job satisfaction improvement

```
In [ ]: response.split(',')
```

```
Out[ ]: ['1. Government survey\n2. Department satisfaction rating\n3. NASA\n4. Social Security Administration\n5. Job satisfaction improvement']
```

Transforming

```
In [ ]: prompt = f"""
Translate the following English text to Spanish: \
```Hi, I would like to order a blender```
"""
response = get_completion(prompt)
print(response)
```

Hola, me gustaría ordenar una licuadora.

```
In []: prompt = f"""
Tell me which language this is:
```Combien coûte le lampadaire?```
"""
response = get_completion(prompt)
print(response)
```

This language is French.

```
In [ ]: prompt = f"""
Translate the following text to French and Spanish
and English pirate: \
```I want to order a basketball```
"""
response = get_completion(prompt)
print(response)
```



French: ```Je veux commander un ballon de basket```  
Spanish: ```Quiero ordenar una pelota de baloncesto```  
English: ```I want to order a basketball```

```
In []: prompt = f"""
 Translate the following text to Spanish in both the \
 formal and informal forms:
 'Would you like to order a pillow?'
 """
 response = get_completion(prompt)
 print(response)
```

Formal: ¿Le gustaría ordenar una almohada?  
Informal: ¿Te gustaría ordenar una almohada?

```
In []: user_messages = [
 "La performance du système est plus lente que d'habitude.", # System performance is slower than normal
 "Mi monitor tiene píxeles que no se iluminan.", # My monitor has pixels that are not lighting
 "Il mio mouse non funziona", # My mouse is not working
 "Mój klawisz Ctrl jest zepsuty", # My keyboard has a broken control key
 "我的屏幕在闪烁" # My screen is flashing
]
```

```
In []: for issue in user_messages:
 prompt = f"Tell me what language is this : ```{issue}``` "
 lang = get_completion(prompt)
 print(f"Original Message : ({lang}) : {issue}")
 prompt = f"""
 Translate the following text to English, \
 Hindi, Bengali, and Russian : ```{issue}```
 """
 response = get_completion(prompt)
 print(response, '\n')
```

Original Message : (This language is French.) : La performance du système est plus lente que d'habitude.  
English: The system performance is slower than usual.  
Hindi: सिस्टम का प्रदर्शन सामान्य से धीमा है।  
Bengali: সিস্টেমের কার্যক্ষমতা সাধারণের চেয়ে ধীর।  
Russian: Производительность системы медленнее, чем обычно.

Original Message : (The language is Spanish.) : Mi monitor tiene píxeles que no se iluminan.  
English: My monitor has pixels that do not light up.  
Hindi: मेरे मॉनिटर में पिक्सेल हैं जो जलाने नहीं होते हैं।  
Bengali: আমার মনিটরে পিক্সেলগুলি আলো দেখাচ্ছে না।  
Russian: У моего монитора есть пиксели, которые не светятся.

Original Message : (The language is Italian.) : Il mio mouse non funziona  
English: My mouse is not working  
Hindi: मेरा माउस काम नहीं कर रहा है (Mera mouse kaam nahi kar raha hai)  
Bengali: আমার মাউস কাজ করছে না (Amar maus kaj korche na)  
Russian: Моя мышь не работает (Moya mysh' ne rabotaet)

Original Message : (This language is Polish.) : Mój klawisz Ctrl jest zepsuty  
English: My Ctrl key is broken  
Hindi: मेरा Ctrl की चाबी टूट गई है  
Bengali: আমার Ctrl কী ভাঙা হয়েছে  
Russian: Моя клавиша Ctrl сломана

Original Message : (The language of the given text is Chinese.) : 我的屏幕在闪烁  
English: My screen is flickering.  
Hindi: मेरी स्क्रीन झिलमिल रही है। (Merī skrīn jhīlamīlā rahī hai.)  
Bengali: আমার স্ক্রিনটি ঝিমঝিম করছে। (Āmār skrīnaṭi jhimajhima karache.)  
Russian: Мой экран мигает. (Moy ekran migaet.)

## Tone transformation

```
In []: prompt = f"""
Translate the following from slang to a business letter:
'Dude, This is Joe, check out this spec on this standing lamp.'
"""
```

```
response = get_completion(prompt)
print(response)
```

Dear Sir/Madam,

I hope this letter finds you well. My name is Joe, and I am writing to bring your attention to a specification document regarding a standing lamp.

I kindly request that you take a moment to review the attached spec, as it contains important details about the standing lamp in question.

Thank you for your time and consideration. I look forward to hearing from you soon.

Sincerely,  
Joe

## Format Conversion

```
In []: data_json = {"restaurant employees":[
 {"name" : "Shyam", "email":"Shyamjaiswal@gmail.com"},
 {"name":"Bob", "email":"bob32@gmail.com"},
 {"name":"Jai", "email":"jai87@gmail.com"}
]
}
prompt = f"""
 Translate the following python dictionary from JSON to an HTML \
 table with column headers and title : {data_json}
 """
response = get_completion(prompt)
print(response)
```

```
<!DOCTYPE html>
<html>
<head>
<style>
table {
 font-family: arial, sans-serif;
 border-collapse: collapse;
 width: 100%;
}

td, th {
 border: 1px solid #dddddd;
 text-align: left;
 padding: 8px;
}

tr:nth-child(even) {
 background-color: #dddddd;
}
</style>
</head>
<body>

<h2>Restaurant Employees</h2>

<table>
 <tr>
 <th>Name</th>
 <th>Email</th>
 </tr>
 <tr>
 <td>Shyam</td>
 <td>Shyamjaiswal@gmail.com</td>
 </tr>
 <tr>
 <td>Bob</td>
 <td>bob32@gmail.com</td>
 </tr>
 <tr>
 <td>Jai</td>
 <td>jai87@gmail.com</td>
 </tr>
</table>
```

```
</tr>
</table>

</body>
</html>
```

```
In []: from IPython.display import display, HTML, Markdown, Latex, JSON
display(HTML(response))
```

## Restaurant Employees

Name	Email
Shyam	Shyamjaiswal@gmail.com
Bob	bob32@gmail.com
Jai	jai87@gmail.com

## Spell Check

```
In []: from redlines import Redlines
text = [
 "The girl with the black and white puppies have a ball.", # The girl has a ball.
 "Yolanda has her notebook.", # ok
 "Its going to be a long day. Does the car need it's oil changed?", # Homonyms
 "Their goes my freedom. There going to bring they're suitcases.", # Homonyms
 "Your going to need you're notebook.", # Homonyms
 "That medicine effects my ability to sleep. Have you heard of the butterfly affect?", # Homonyms
 "This phrase is to cherck chatGPT for speling abilitty" # spelling
]
for t in text:
 prompt = f"""Proofread and correct the following text present within triple backticks and do not include the ba
 {t}"""
 response = get_completion(prompt)
```

```
diff = Redlines(t,response)
display(diff.output_rich)
```

The girl with the black and white puppies ~~have~~has a ball.

Yolanda has her notebook.

~~Its~~It's going to be a long day. Does the car need ~~it's~~its oil changed?

~~Their~~There goes my freedom. ~~There~~They're going to bring ~~they're~~their suitcases.

~~Your~~You're going to need ~~you're~~your notebook.

That medicine ~~effects~~affects my ability to sleep. Have you heard of the butterfly ~~affect?~~effect?

This phrase is to ~~cherck~~check chatGPT for ~~speling~~abilityspelling ability.

## Expanding : Taking a small piece of text and having a LLM generate a large text for it.

One of its use is that the LLM can be used as a brainstorming partner.

```
In []: # given the sentiment from the lesson on "inferring",
and the original customer message, customize the email
sentiment = "negative"

review for a blender
review = f"""
So, they still had the 17 piece system on seasonal \
sale for around $49 in the month of November, about \
half off, but for some reason (call it price gouging) \
around the second week of December the prices all went \
up to about anywhere from between $70-$89 for the same \
system. And the 11 piece system went up around $10 or \
so in price also from the earlier sale price of $29. \
So it looks okay, but if you look at the base, the part \
where the blade locks into place doesn't look as good \
as in previous editions from a few years ago, but I \
plan to be very gentle with it (example, I crush \
very hard items like beans, ice, rice, etc. in the \
blender first then pulverize them in the serving size \
```

```
I want in the blender then switch to the whipping \
blade for a finer flour, and use the cross cutting blade \
first when making smoothies, then use the flat blade \
if I need them finer/less pulpy). Special tip when making \
smoothies, finely cut and freeze the fruits and \
vegetables (if using spinach—lightly stew soften the \
spinach then freeze until ready for use—and if making \
sorbet, use a small to medium sized food processor) \
that you plan to use that way you can avoid adding so \
much ice if at all—when making your smoothie. \
After about a year, the motor was making a funny noise. \
I called customer service but the warranty expired \
already, so I had to buy another one. FYI: The overall \
quality has gone down in these types of products, so \
they are kind of counting on brand recognition and \
consumer loyalty to maintain sales. Got it in about \
two days.
"""
```

```
In []: prompt = f"""
 You are a customer service AI assistant.
 Your task is to send an email reply to the valued customer.
 Given the customer email delimited by ```
 Generate a reply thanking the customer for the review.
 If the sentiment is positive or neutral, thank them for their review.
 If the sentiment is negative, apologize and suggest that they can reach out to the customer service.
 Make sure to use specific details from the review.
 Write in a professional and concise tone.
 Sign the email as `AI customer agent`.
 Customer review : ```{review} ```
 Review sentiment : {sentiment}
 """
 response = get_completion(prompt, temperature =0.7)
 print(response)
```

Dear valued customer,

Thank you for taking the time to share your review with us. We appreciate your feedback and apologize for any inconvenience you may have experienced.

We are sorry to hear about the price increase you noticed in December. We strive to provide fair and competitive pricing for our products, and we understand your frustration. If you have any further concerns regarding pricing or any other issues, we encourage you to reach out to our customer service team. They will be more than happy to assist you and address any questions or concerns you may have.

We also appreciate your feedback regarding the quality of our product. We continuously strive to improve our products based on customer feedback, and we will take your comments into consideration for future enhancements.

Once again, we apologize for any inconvenience caused and appreciate your loyalty as a customer. If there is anything else we can assist you with, please do not hesitate to contact us.

Best regards,

AI customer agent

## Custom Chat bot

```
In []: def get_completion_from_messages(messages, model="gpt-3.5-turbo", temperature=0):
 response = client.chat.completions.create(
 model=model,
 messages=messages,
 temperature=0
)
 return response.choices[0].message.content
```

The first message is the system message which sets up the behaviour of the model i.e. acts as a 'whisper' to the model.

- The user doesn't know about the system message.

- Gives the developer the power to frame the conversation without making the request a part of the conversation.

- you can basically guide the assistant's responses without making the user aware.



```
In []: messages = [
 {'role': 'system', 'content': 'You are friendly chatbot.'},
 {'role': 'user', 'content': 'Hi, my name is Isa'}]
 response = get_completion_from_messages(messages, temperature=1)
 print(response)
```

Hello Isa! It's nice to meet you. How can I assist you today?

```
In []: messages = [
 {'role': 'system', 'content': 'You are friendly chatbot.'},
 {'role': 'user', 'content': 'Yes, can you remind me, What is my name?'}]
 response = get_completion_from_messages(messages, temperature=1)
 print(response)
```

I'm sorry, but as a chatbot, I don't have access to personal information. Therefore, I don't know your name.

```
In []: messages = [
 {'role': 'system', 'content': 'You are friendly chatbot.'},
 {'role': 'user', 'content': 'Hi, my name is Isa'},
 {'role': 'assistant', 'content': "Hi Isa! It's nice to meet you. \
 Is there anything I can help you with today?"},
 {'role': 'user', 'content': 'Yes, you can remind me, What is my name?'}
]

 response = get_completion_from_messages(messages, temperature=1)
 print(response)
```

Your name is Isa.

**We can automate the collection of user prompts and assistant responses to built an OrderBot which takes orders from at a pizza restaurant**

```
In []: def collect_messages(_):
 prompt = inp.value_input
 inp.value = ''
 context.append({'role': 'user', 'content': f"{prompt}"})
 response = get_completion_from_messages(context)
 context.append({'role': 'assistant', 'content': f"{response}"})
 panels.append(
```

```

 pn.Row('User:', pn.pane.Markdown(prompt, width=600)))
panels.append(
 pn.Row('Assistant:', pn.pane.Markdown(response, width=600, styles={'background-color': '#F6F6F6'})))

return pn.Column(*panels)

```

```

In []: import panel as pn # GUI
pn.extension()

panels = [] # collect display

context = [{'role':'system', 'content':"""
You are OrderBot, an automated service to collect orders for a pizza restaurant. \
You first greet the customer, then collects the order, \
and then asks if it's a pickup or delivery. \
You wait to collect the entire order, then summarize it and check for a final \
time if the customer wants to add anything else. \
If it's a delivery, you ask for an address. \
Finally you collect the payment.\
Make sure to clarify all options, extras and sizes to uniquely \
identify the item from the menu.\
You respond in a short, very conversational friendly style. \
The menu includes \
pepperoni pizza 12.95, 10.00, 7.00 \
cheese pizza 10.95, 9.25, 6.50 \
eggplant pizza 11.95, 9.75, 6.75 \
fries 4.50, 3.50 \
greek salad 7.25 \
Toppings: \
extra cheese 2.00, \
mushrooms 1.50 \
sausage 3.00 \
canadian bacon 3.50 \
AI sauce 1.50 \
peppers 1.00 \
Drinks: \
coke 3.00, 2.00, 1.00 \
sprite 3.00, 2.00, 1.00 \
bottled water 5.00 \
"""}] # accumulate messages

```

```

inp = pn.widgets.TextInput(value="Hi", placeholder='Enter text here...')
button_conversation = pn.widgets.Button(name="Chat!")

binds the collect_message behaviour to the button
interactive_conversation = pn.bind(collect_messages, button_conversation)

dashboard = pn.Column(
 inp,
 pn.Row(button_conversation),
 pn.panel(interactive_conversation, loading_indicator=True, height=300),
)

dashboard

```

Out[ ]: BokehModel(combine\_events=True, render\_bundle={'docs\_json': {'d93f7d39-e77d-40fd-8c1c-790552bfff74c': {'version...

```

In []: messages = context.copy()
messages.append(
 {'role': 'system', 'content': 'create a json summary of the previous food order. Itemize the price for each item\
 The fields should be 1) pizza, include size 2) list of toppings 3) list of drinks, include size 4) list of sides
)
 #The fields should be 1) pizza, price 2) list of toppings 3) list of drinks, include size include price 4) list o

response = get_completion_from_messages(messages, temperature=0)
print(response)

```

Sure! Here's a JSON summary of your food order:

```
{
 "pizza": {
 "size": "12.95",
 "toppings": ["extra cheese", "mushrooms"]
 },
 "drinks": [
 {
 "name": "coke",
 "size": "3.00"
 },
 {
 "name": "sprite",
 "size": "2.00"
 }
],
 "sides": [
 {
 "name": "fries",
 "size": "4.50"
 }
],
 "total_price": "25.45"
}
```

Please let me know if there's anything else you'd like to add to your order!