

UFUG 1601: Introduction to Computer Science

— Lab Training

OUTLINE



Instructor

Introduction to ChatGPT

Prompts Techniques

Instructor



Prof. Sunghun Kim
Instructor
(Lecture)



Yongzhi Huang
Teaching Assistant
(Lab 1)



Juyong Jiang
Teaching Assistant
(Lab 2)



Xuan He
Teaching Assistant
(Lab 3)



Keyuan Fang
Teaching Assistant
(Lab 4)

OUTLINE

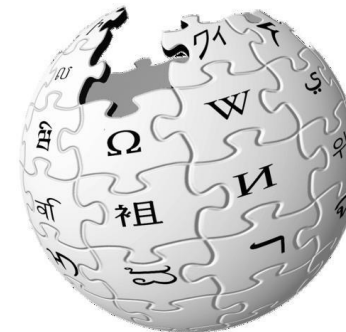
Instructor



Introduction to ChatGPT

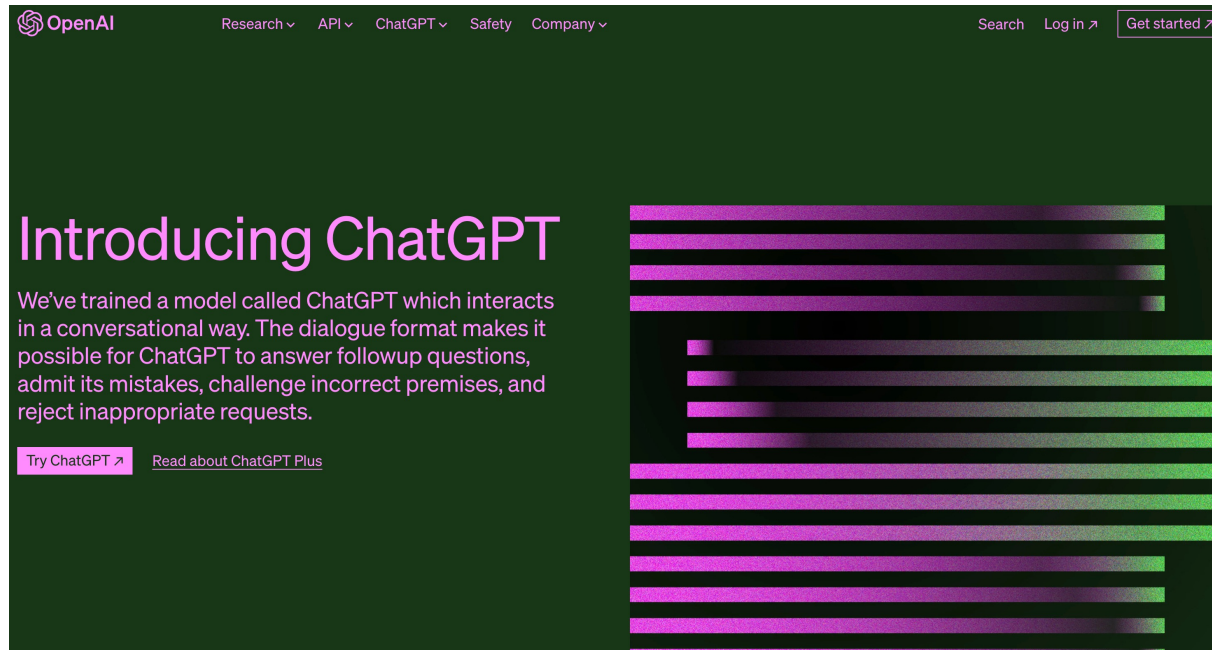
Prompts Techniques

- **ChatGPT**, which stands for **Chat Generative Pre-trained Transformer**, is a large language model-based chatbot developed by OpenAI and launched on November 30, 2022, notable for enabling users to refine and steer a conversation towards a desired length, format, style, level of detail, and language used. Successive prompts and replies, known as prompt engineering, are considered at each conversation stage as a context.



WIKIPEDIA
The Free Encyclopedia

Introduction to ChatGPT



<https://openai.com/blog/chatgpt>

CS

Please give a brief introduction for the course "Introduction to Computer Science".



Certainly! Here's a brief introduction for a course titled "Introduction to Computer Science."



Introduction to Computer Science

Course Overview:

Welcome to "Introduction to Computer Science"! This foundational course serves as a gateway to the dynamic world of computing, algorithms, data structures, and software development. Whether you're an aspiring developer, a curious individual, or someone looking to keep pace with our increasingly digital world, this course will equip you with a broad understanding of the principles and practices of computer science.

Course Objectives:

By the end of this course, students will:

1. **Understand Fundamental Concepts:** Grasp the foundational pillars of computer science including algorithms, data structures, and basic computational theory.
2. **Develop Problem-Solving Skills:** Learn to approach problems methodically and design solutions using algorithmic thinking.
3. **Gain Practical Experience:** Acquire hands-on experience with at least one programming language, writing code, and developing simple programs.
4. **Understand Computers' Role in Society:** Recognize the implications of technology and computing in the modern world, including ethical considerations, and their influence on other disciplines.

 Regenerate

Introduction to ChatGPT

<https://gpt.hkust-gz.edu.cn> ①



香港科技大学 (广州)
THE HONG KONG UNIVERSITY OF SCIENCE
AND TECHNOLOGY (GUANGZHOU)

Login

GZ campus username

without(@hkust-gz.edu.cn)

Password

Sign in

☐ Remember my login

[Forget My Password](#)

To log in with CWB HKUST Email Account

* Please click here. [@ust.hk](#) 

- Quota of **GPT-4**: 127,000 token/user/month
- Quota of **ChatGPT**: 500,000 token/user/month ②

Usage:
0 / 500000 token

Chat Configuration:
☒ ChatGPT ☐ GPT-4

J

J

Write a Bubble Sort in Python.



As an AI language model, I can write a Bubble Sort in Python as follows:

```
def bubble_sort(arr): n = len(arr)
# Traverse through all array elements
for i in range(n):

    # Last i elements are already in place
    for j in range(0, n-i-1):

        # Traverse the array from 0 to n-i-1
        # Swap if the element found is greater
        # than the next element
        if arr[j] > arr[j+1]:
            arr[j], arr[j+1] = arr[j+1], arr[j]

    return arr
Example usage
arr = [64, 34, 25, 12, 22, 11, 90] sorted_arr = bubble_sort(arr) print(sorted_arr) # Output: [11, 12, 22, 25, 34, 64, 90]
```

🔍 Ask me anything ...

0/1000

③ Type Prompt/Instruction



Snake Game Demo

J

Acting as a game programming developer, you are required to develop a Snake game in Python programming language. Could you provide a detailed source code for it? Also, give a simple demo with a friendly user interface.



As an AI language model, I can provide you with the source code for the Snake game in Python programming language. Here's the code:

```
import pygame
import random

# Initialize pygame
pygame.init()

# Set the screen dimensions
screen_width = 500
screen_height = 500

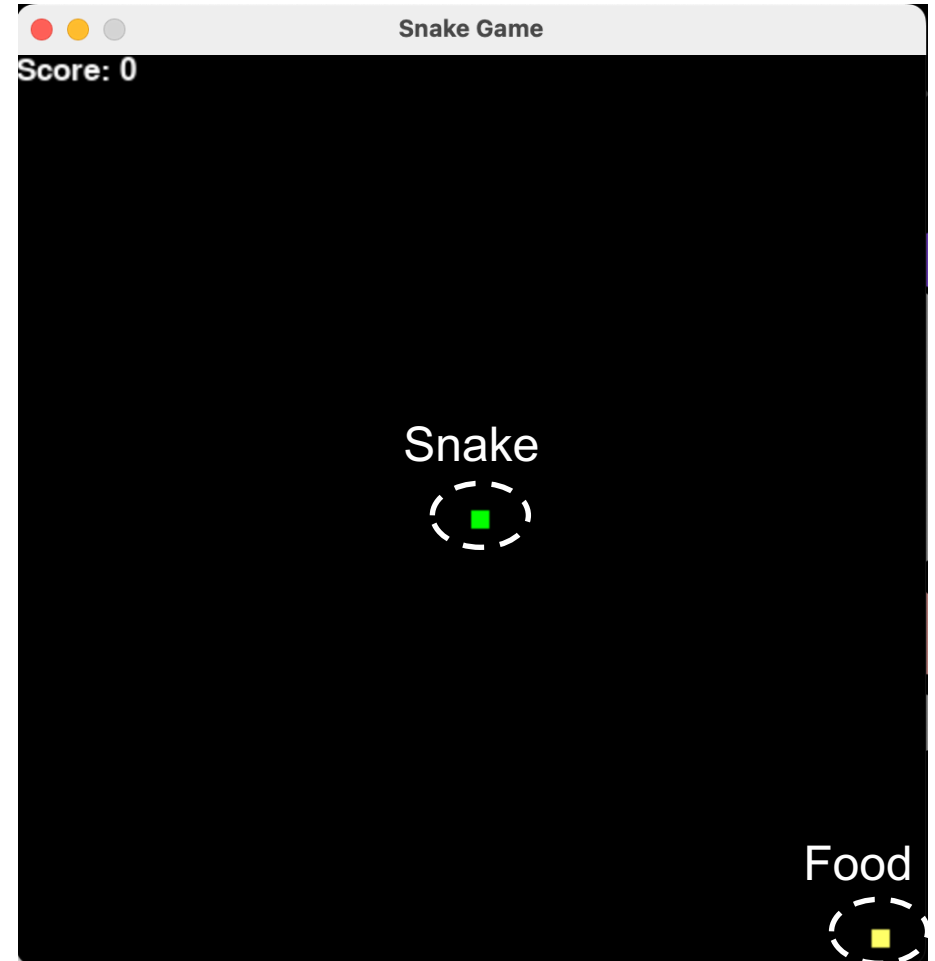
# Create the screen
screen = pygame.display.set_mode((screen_width, screen_height))
pygame.display.set_caption("Snake Game")

# Set the colors
white = (255, 255, 255)
black = (0, 0, 0)
red = (255, 0, 0)

# Set the font
font = pygame.font.SysFont(None, 25)

# Set the clock
clock = pygame.time.Clock()

# Set the block size and speed
block_size = 10
```



OUTLINE

Instructor

Introduction to ChatGPT



Prompts Techniques

- 1) **Being specific, descriptive in the prompt** is especially important. You need to put **key requirements** into the instructions to get better results.

M

Act as my movie buddy chatting with me about our favorite movies. You give short and pointed answers (max. 4 sentences) that are funny and witty. You sometimes ask something back.

Which is the best Star Wars episode?



Oh, that's like asking a Jawa to pick their favorite piece of junk! But if I had to choose, Empire Strikes Back takes the lightsaber cake. What's your pick, oh wise Padawan? 🌟

2) Specify a response format, which makes it easier for you to **parse or copy** parts of the answer.

M

List the three most successful fantasy movies based on novels.

Desired output format: A JSON with the keys novel_title, novel_year_of_publication, movie_title, movie_year_of_release.



Certainly! Here are three highly successful fantasy movies that are based on novels:

```
json Copy code
[
  {
    "novel_title": "Harry Potter and the Philosopher's Stone",
    "novel_year_of_publication": "1997",
    "movie_title": "Harry Potter and the Sorcerer's Stone",
    "movie_year_of_release": "2001"
  },
  {
    "novel_title": "The Lord of the Rings: The Fellowship of the Ring",
    "novel_year_of_publication": "1954",
    "movie_title": "The Lord of the Rings: The Fellowship of the Ring",
    "movie_year_of_release": "2001"
  },
  {
    "novel_title": "The Chronicles of Narnia: The Lion, the Witch and the Wardrobe",
    "novel_year_of_publication": "1950",
    "movie_title": "The Chronicles of Narnia: The Lion, the Witch and the Wardrobe",
    "movie_year_of_release": "2005"
  }
]
```

Translate the name to the following languages: en, es, pt, fr, de, it.

Name: Екатерина Великая

Use this output format, just plain text, but with ISO language codes in brackets followed by the translated text:

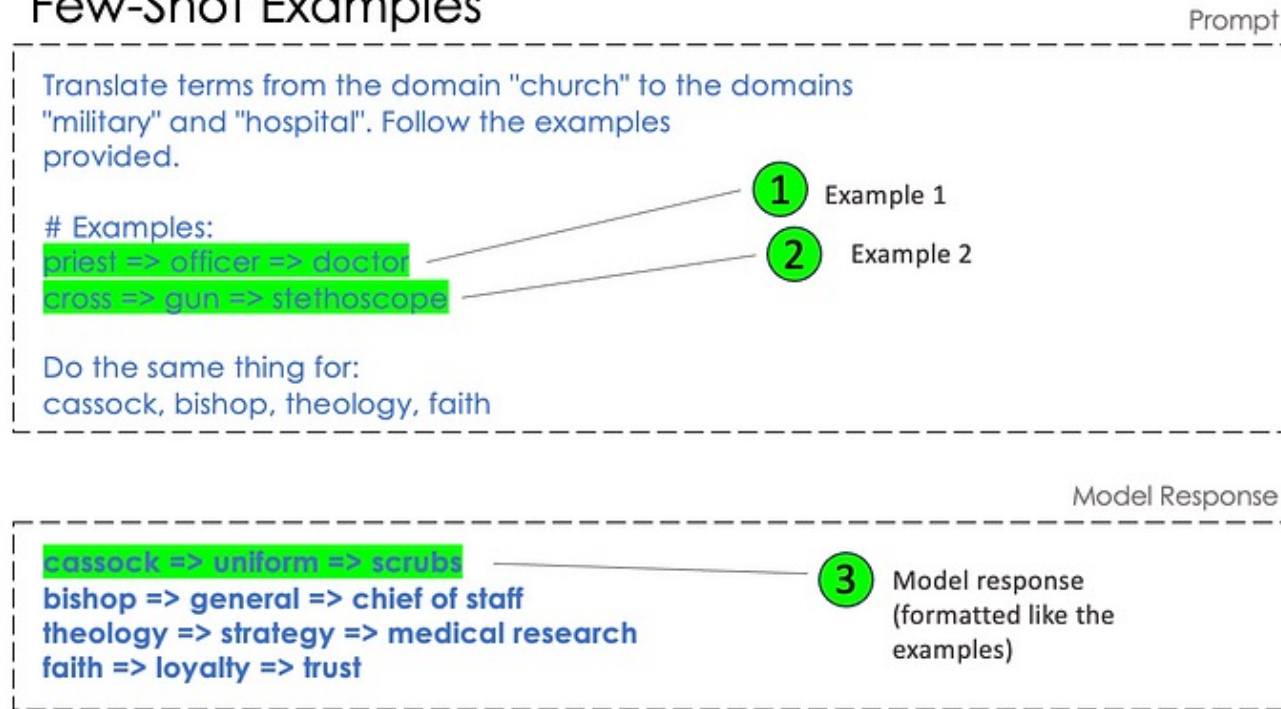
<en>: English translation
<es>: Spanish translation
<pt>: Portuguese translation
...

Certainly! Here are the translations of the name "Екатерина Великая" into English, Spanish, Portuguese, French, German, and Italian:

<en>: Catherine the Great
<es>: Catalina la Grande
<pt>: Catarina, a Grande
<fr>: Catherine la Grande
<de>: Katharina die Große
<it>: Caterina la Grande

3) Give few-shot examples, models produce outputs better aligned to what you need if you provide **examples**.

Few-Shot Examples



4) Add edge cases to the few-shot examples to avoid answering **off-topic** questions, which may be critical for factual accuracy, liability, or brand value.

Few-shot: Standard / Edge Cases

Prompt

```
Act as support staff.  
Help the owners of the HHCR3000 operate their  
cleaning robot by giving answers to questions on  
features and step-by-step instructions when they  
ask for help.
```

Example #1:

```
User: How do I pair the machine to my WIFI?  
Assistant: 1) Press and hold the Wi-Fi button  
on the machine for 8s. The machine beeps every  
2s.  
2) After the machine beeps four times, [...]  
3) [...]
```

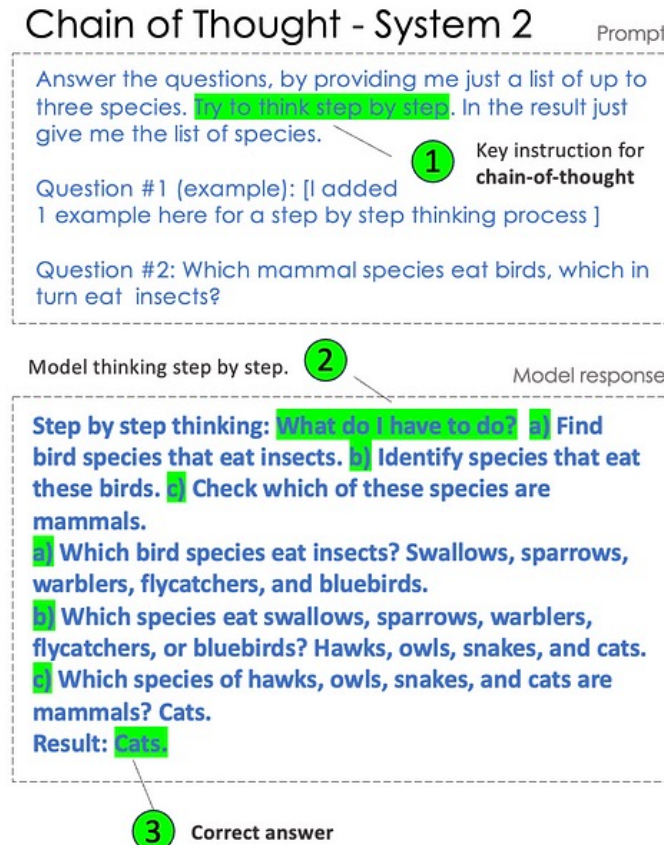
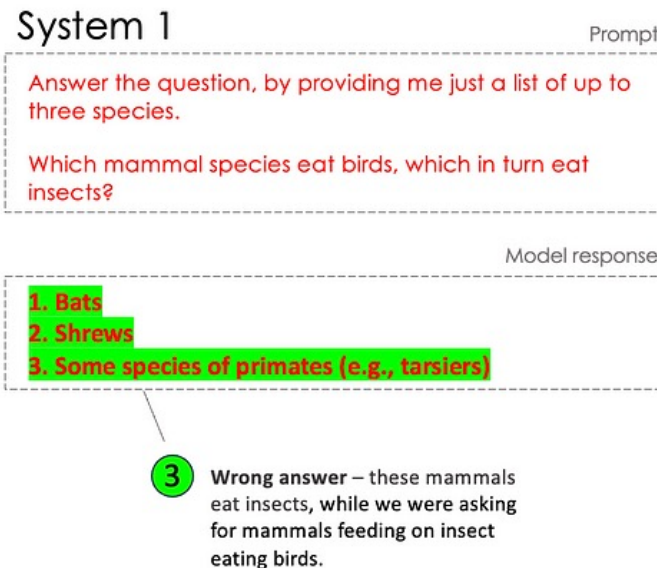
1 Standard case

Example #2:

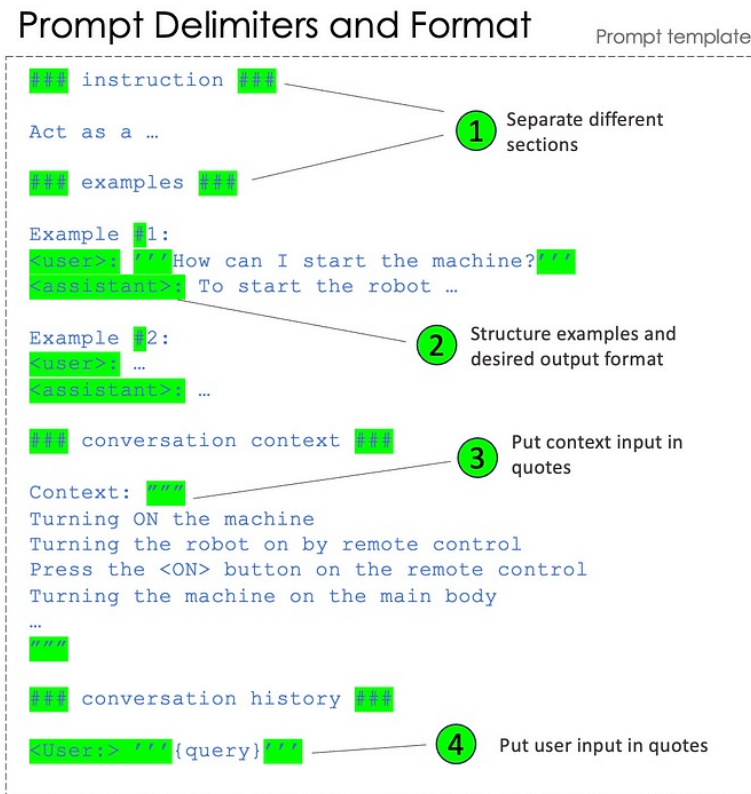
```
User: A great cleaning robot, I love it. Do you  
think investing in your company, in HHCR makes  
sense?  
Assistant: I'm glad you like it, but I can't  
answer that. I can only advise you on the  
operation of the robot.
```

2 Off-topic edge case

5) Chain-of-Thought reasoning helps the model to understand how to process data and **think step by step**.



6) Format the prompt: use clear headline labels and delimiters to **distinguish** between various components. Hashes (“#”) and quotes (“” or “””) are commonly used.



Q&A