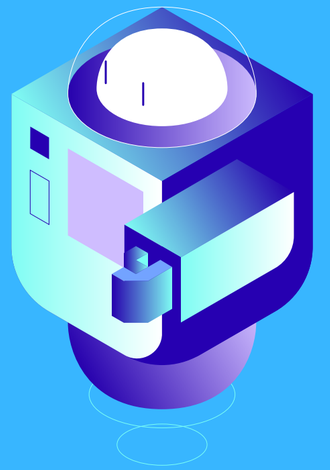
PROMPT ENGINEERING FOR CODE DEBUGGING

Unlocking the Power of Language Models

Presenter: Jesiah Martin &
Kesnel Mezinord



SOCIALS

INSTAGRAM

DISCORD

WHAT IS PROMPT ENGINEERING

In this workshop, we will explore the fundamentals of prompt engineering for large language models. Understanding how to craft effective prompts is essential for maximizing model performance and achieving desired outcomes. Join us as we delve into techniques, examples, and best practices.

WHAT ARE LLM

Large language models like GPT-3 and others are designed to understand and generate human-like text. This slide will cover the architecture of these models, their training processes, and how they interpret prompts to produce relevant responses.

WORKSHOP AGENDA

1. Techniques of Prompt Engineering
2. Engineer a Prompt Activity
3. Outlining Project Creation and Debugging with ChatGPT
4. Wrap-Up and Q&A

PROMPT CRAFTING TECHNIQUES

What is Prompt Crafting?

The process of designing clear, structured input queries to guide large language models (LLMs) for better, more accurate outputs. A well-structured prompt can lead to more accurate and contextually relevant responses.

FEW-SHOT PROMPTING

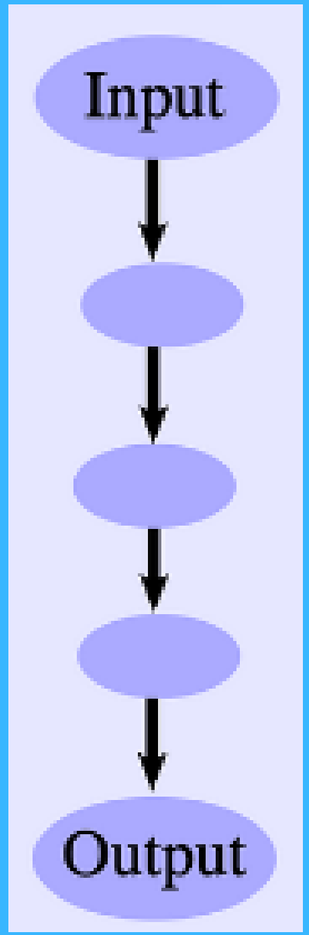
Demonstrative teaching where you provide the AI with a small number of examples (or "shots") within the prompt to guide its response.

TRANSLATE THE FOLLOWING SENTENCES FROM ENGLISH TO SPANISH:
1. ENGLISH: ”HELLO, HOW ARE YOU?” SPANISH: ”HOLA, ¿CÓMO ESTÁS?”
2. ENGLISH: ”WHAT TIME IS IT?” SPANISH: ”¿QUÉ HORA ES?”

NOW, TRANSLATE THIS SENTENCE: ”I LOVE PROGRAMMING.”

CHAIN OF THOUGHT REASONING

Prompt your model to "think out loud" by breaking down its reasoning step by step, leading to a more transparent and logical response.

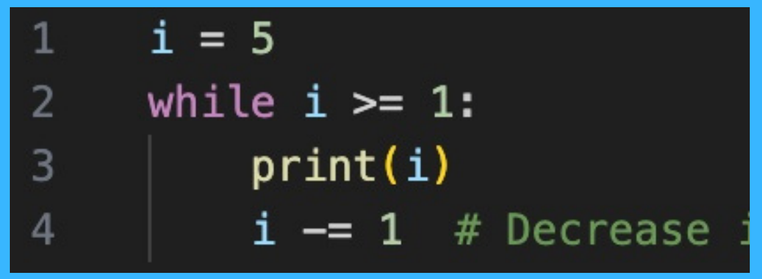


Q: IF ONE PAINTER TAKES 4 HOURS TO PAINT 1 ROOM, HOW LONG WILL 4 PAINTERS TAKE TO PAINT 4 ROOMS?

1. ONE PAINTER TAKES 4 HOURS FOR 1 ROOM.
2. 4 PAINTERS CAN WORK AT THE SAME TIME, EACH PAINTING 1 ROOM.
3. THEREFORE, IT WILL STILL TAKE 4 HOURS TO PAINT ALL 4 ROOMS.

DOCUMENT MIMICRY

Directly cite or show the source material you need help with—essentially, providing examples or references in your prompts that the AI can mimic or work from.



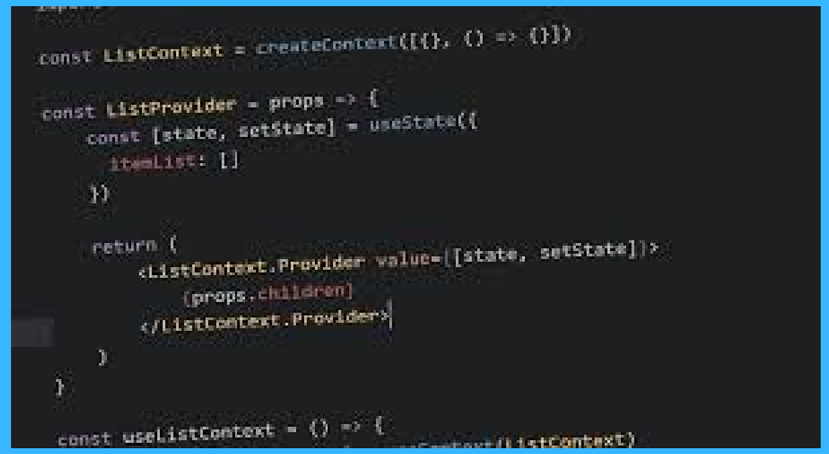
”I NEED HELP WITH MY ESSAY ON DEFORESTATION. HERE’S AN INTRODUCTION I LIKE FOR CLIMATE CHANGE:

’IN THE 21ST CENTURY, CLIMATE CHANGE IS ONE OF THE MOST PRESSING ISSUES, WITH GLOBAL TEMPERATURES RISING AND CONSEQUENCES FELT WORLDWIDE.’

CAN YOU WRITE SOMETHING SIMILAR FOR DEFORESTATION?”

CONTEXTUAL PROMPTS

Contextual prompts enhance LLM responses by offering clear information, resulting in more accurate and relevant answers while minimizing errors and improving the overall interaction.



WITHOUT CONTEXT:
”EXPLAIN PHOTOSYNTHESIS.”
WITH CONTEXT:
”EXPLAIN PHOTOSYNTHESIS TO A 10-YEAR-OLD USING SIMPLE LANGUAGE.”

ENGINEER A PROMPT

“WHY ISN’T MY IF STATEMENT WORKING?”

“My if statement isn’t printing ‘Even’ when the number is even. Here’s my code:”

```
def is_even(number):
    if number % 2 = 0:
        return True
    else:
        return False

result = is_even(10)
if result == True:
    print("Even")
else:
    print("Odd")
```

Response:

AI could respond with a clear explanation: "The issue is that = is used for assignment, not comparison. Use == for comparisons in the if statement."

“WHY DOESN’T MY SUM WORK?”

“I’m trying to add two numbers in Python, but I get a wrong result. Here’s my code:”

```python
num1 = 5
num2 = '10'
total = num1 + num2
print(total)
```

Response:

"Your code works as expected for integers. If you were expecting a different result, double-check the data types and how you want the values displayed."

”WHY ISN’T MY LOOP WORKING?”

"My for loop is not iterating over the list as expected. I want to print each element, but the last one is printed twice. Here’s the code snippet I’m using:"

```python
my_list = [1, 2, 3, 4, 5]
for i in my_list:
    print(i)
print(i)  # This prints the last element
```

Contextual Prompts

Example 2: Troubleshooting a Segmentation Fault

Inaccurate Prompt:

“Why isn’t my loop working?”

Response: The AI might offer generic advice about loops without addressing the specific problem

.

Precise Prompt:

“I’m trying to write a loop that counts down from 5 to 1, but it runs infinitely. Here’s my code:

```
1    i = 5
2    while i >= 1:
3        print(i)
4
5
6
```

Response:

AI might offer the precise help:

"You forgot to decrement i. Without i -= 1, the value of i never changes, causing an infinite loop.

"

Here’s the code after precise prompt:

```
1    i = 5
2    while i >= 1:
3        print(i)
4        i -= 1  # Decrease i to avoid infinite loop
```

To Be a Better Programmer

AI Augments, Not Replaces

- **Use AI to enhance your abilities, not as a complete solution.**

Automate Tedious Tasks

- **Leverage AI for repetitive work, but stay involved in problem-solving.**

Understand Code Solutions

- **Ensure you fully comprehend AI-suggested solutions.**

Active Engagement in Problem-Solving

- **Don’t rely solely on AI—stay engaged in the coding process.**

Stay Updated with Technology

- **Keep learning new tools and technologies to remain versatile.**

Use AI Wisely

- **Combine AI with your skills to become more effective and innovative.**