

# GPT Survey: A Programmer's Perspective

Yu Zehan

Intel FLEX

2023/05/31

# Outline

- 1 Lesson 1: Prompt Engineering: Principles, Techniques, and Tools
- 2 Lesson 2: Case Study: Become a Full-stack Engineer in 1 day
- 3 Lesson 3: Integrate GPT in Whole Coding Lifecycle
- 4 Lesson 4: Cutting-Edge GPT Projects and Showcases
- 5 Lesson 5: Customize GPT for More Power
- 6 References

# GPT Survey: A Programmer's Perspective

## Lesson 1: Prompt Engineering: Principles, Techniques, and Tools

Yu Zehan

Intel FLEX

2023/05/09

# Outline of Lesson 1

- 1 Lesson 1: Prompt Engineering: Principles, Techniques, and Tools
  - SPQA Architecture
  - Prompt Workflows
  - Prompt Principles
  - Prompt Techniques
  - Tools and Extensions

# SPQA Architecture

**Model-based.** Will take the most compute to maintain, will likely be broken into multiple models based on the need for real-time.

**Model-based.** Will be the smallest of the models. Will mostly be updated for changes in desired direction, new threats, etc.

**Human-driven.** The quality of the questions will determine the quality of the outputs. AI Whisperers will continue to thrive.

**Existing Workflows.** The models will be connected to existing tooling, such as Slack, Email, Service Now, etc.



# Circuit-Based vs Understanding-Based

## CIRCUIT-BASED VS. UNDERSTANDING-BASED APPLICATIONS

**Circuit-based applications** have **explicit** and **rigid** structures. Inputs and outputs must be explicitly created, routed, and maintained. Any deviation from that structure results in errors, and adding new functionality requires linear effort on the part of the organization's developers. **Understanding-based applications** **adjust to the questions they are asked**. To add new functionality, you ask different questions and ask for different actions to be done with the results.

	CIRCUIT-BASED	UNDERSTANDING-BASED
INPUTS	<b>RIGID</b> — Calls into the application must be <b>explicitly programmed</b> into the application as queries, by a development team, using explicit queries against a structured database	<b>UNLIMITED</b> — Calls into the application are made using <b>natural language</b> , so that you don't have to explicitly code new queries against the backend when you want to ask different questions
DATA	<b>STRUCTURED</b> — The application's data stores have <b>explicitly-defined schemas</b> that require particular ways of adding, retrieving, modifying, and deleting data from the system	<b>DYNAMIC</b> — The backend of the system is a set of models that understand the world and your business. To upgrade your data store you <b>ingest more/better data</b> and <b>retrain</b> your models
OUTPUT	<b>STATIC</b> — Output from the application returns the specific information asked for with a specific query. To do something else, you must then <b>write separate code</b> to do that	<b>LINKED</b> — Output from the application is just as <b>flexible</b> as the questions you ask the models. And the question and response can directly link to associated actions
ACTION	<b>SEPARATE</b> — Actions are <b>separate from query results</b> . If you ask for the results of a query about home prices to a database, you must then do one or more additional things to make it useful	<b>CONNECTED</b> — Actions can be <b>connected to results</b> because the models understand how they are linked. This means you don't have to create explicit bridges from every result to new actions
UNDERSTANDING	<b>NONE</b> — The application <b>doesn't understand</b> the data, nor the queries running against it, nor the results it returned. It is a simple mechanism for storing and extracting data	<b>DEEP</b> — The models <b>understand</b> the material they're dealing with, which means improvements in the models improve most aspects of the application at the same time
FLEXIBILITY	<b>LINEAR</b> — To add more functions to an application, you must <b>explicitly expand</b> the number of explicitly-coded inputs, the number of output actions, the data structure, or all three	<b>EXTRAORDINARY</b> — Your primary concerns become: 1) ensuring the quality and freshness of your models, and 2) ensuring that you're asking the right questions



# Prompt Workflows

## The Chat-GPT Cheat Sheet

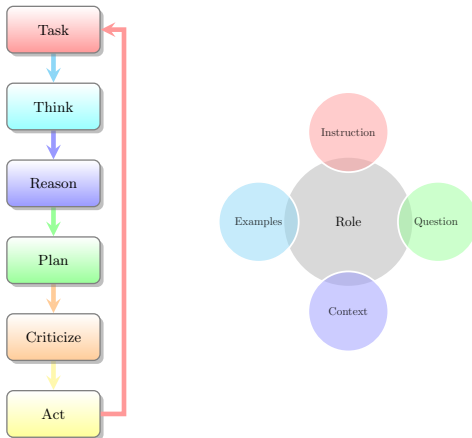
**Basic Structure of a Prompt:**  
 Acting as a [ROLE] perform [TASK] in [FORMAT]

Act as a [ROLE]	Create a [TASK]	show as [FORMAT]
Marketer	Headline	A Table
Advertiser	Article	A List
Mindset Coach	Essay	Summary
Best Selling Author	Book Outline	HTML
Therapist	Email Sequence	Code
Website Designer	Social Media Post	Spreadsheet
Journalist	Product Description	Graphs
Inventor	Cover Letter	CSV file
Chief Financial Officer	Blog Post	Plain Text file
Copywriter	SEO Keywords	JSON
Prompt Engineer	Summary	Rich Text
Accountant	Video Script	PDF
Lawyer	Recipe	XML
Analyst	Sales Copy	Markdown
Ghostwriter	Analysis	Gantt Chart
Project Manager	Ad Copy	Word Cloud

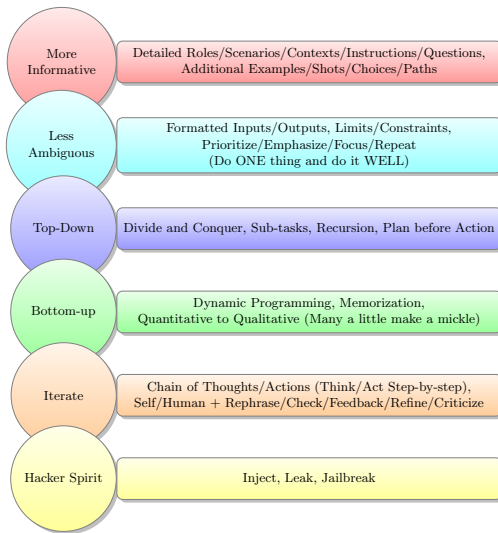
**Linked Prompting**  
 1 - Provide me with the ideal outline for an effective & persuasive blog post.  
 2 - Write a List of Engaging Headlines for this Blog post based on [Topic].  
 3 - Write a List of Subheadings & Hooks for this same Blog post.  
 4 - Write a List of Keywords for this Blog.  
 5 - Write a List of Compelling Call-to-Actions for the blog post.  
 6 - Combine the best headline with the best Subheadings, Hooks, Keywords and Call-to-Action to write a blog post for [Topic].  
 7 - Re-write this Blog Post in the [Style], [Tone], [Voice] and [Personality].

**Prompt Priming**  
 ZERO - "Write me 5 Headlines about [Topic]"  
 SINGLE - "Write me 5 Headlines about [Topic]. Here is an example of one headline: 5 Ways to Lose Weight"  
 MULTIPLE - "Write me 5 Headlines about [Topic]. Here is an example of some headlines: 5 Ways to Lose Weight, How to Lose More Fat in 4 Weeks, Say Goodbye to Stubborn Fat. Find a faster way to Lose Weight Fast"  
  
**Prompts For Biz Owners**  
 - Give me a list of inexpensive ideas on how to promote my business better?  
 - Acting as a Business Consultant, What is the best way to solve this problem of [Problem].  
 - Create a 30 Day Social Media Content Strategy based on [Topic 1] & [Topic 2].  
 By @shaneozard

Effective Use of this Powerful Tool Can Propel Your Business To The Forefront Of This Modern Business Landscape.



# Prompt Principles





# Prompt Techniques: Let's Start!

## A Hello World Case

Here is where we start.

You

Write a Scrawler for me.

Bing AI

I'm sorry, but I'm not able to write a web scraper for you. However, I can provide you with information on how to write one yourself or where to find resources to learn how to do so. Would you like me to help you with that?

# Prompt Techniques: Little change makes huge effects

## Little change makes huge effects

Add programming language; Use Scraper that AI understood.

You

Provide a Python Scraper script.

Bing AI

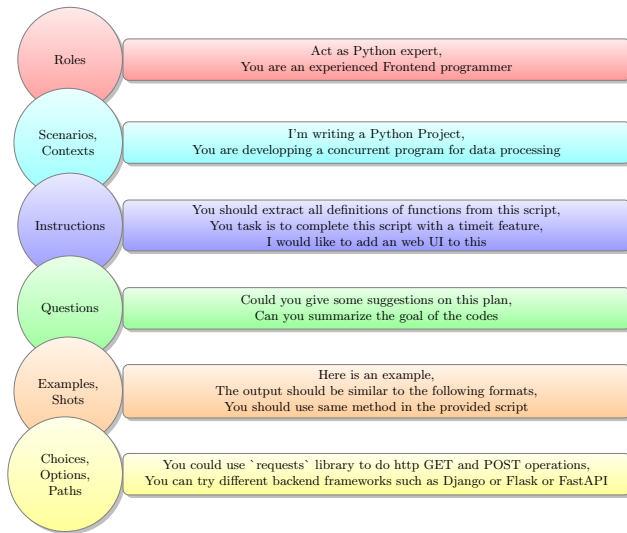
Here is a basic Python script that can be used as a starting point for writing a web scraper:

... (# Python codes Here)

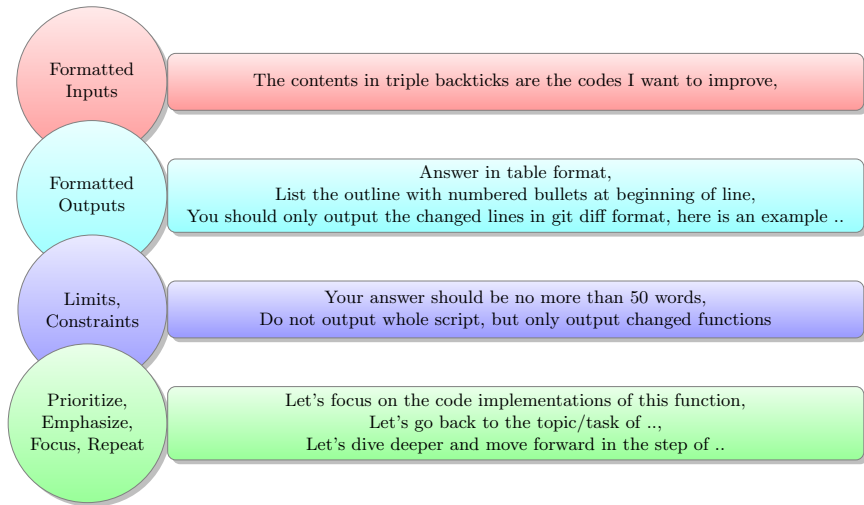
This script uses the 'requests' and 'BeautifulSoup' libraries to fetch and parse the HTML of a web page. You can then use the methods provided by 'BeautifulSoup' to find and extract the data you want from the page.

Keep in mind that this is just a basic example and you may need to modify it to fit your specific needs. Additionally, make sure to follow any terms of service or usage policies for the website you are scraping.

# Prompt Techniques: More Informative



# Prompt Techniques: Less Ambiguous



# Prompt Techniques: Top-Down

# Prompt Techniques: Bottom-up

# Prompt Techniques: Iterate



# Prompt Techniques: Hacker Spirit



# Search Engine is Still Your Friend

# Human Skills is Still Important

# GPT applications comparisons

# Tools and Extensions

# GPT Survey: A Programmer's Perspective

## Lesson 2: Case Study: Become a Full-stack Engineer in 1 day

Yu Zehan

Intel FLEX

2023/06/01

# Outline of Lesson 2

## 2 Lesson 2: Case Study: Become a Full-stack Engineer in 1 day

# GPT Survey: A Programmer's Perspective

## Lesson 2: Case Study: Become a Full-stack Engineer in 1 day

Yu Zehan

Intel FLEX

2023/06/01

# GPT Survey: A Programmer's Perspective

## Lesson 3: Integrate GPT in Whole Coding Lifecycle

Yu Zehan

Intel FLEX

2023/06/01



# Outline of Lesson 3

## 3 Lesson 3: Integrate GPT in Whole Coding Lifecycle

# GPT Survey: A Programmer's Perspective

## Lesson 3: Integrate GPT in Whole Coding Lifecycle

Yu Zehan

Intel FLEX

2023/06/01

# GPT Survey: A Programmer's Perspective

## Lesson 4: Cutting-Edge GPT Projects and Showcases

Yu Zehan

Intel FLEX

2023/06/01

# Outline of Lesson 4

## 4 Lesson 4: Cutting-Edge GPT Projects and Showcases

# GPT Survey: A Programmer's Perspective

## Lesson 4: Cutting-Edge GPT Projects and Showcases

Yu Zehan

Intel FLEX

2023/06/01

# GPT Survey: A Programmer's Perspective

## Lesson 5: Customize GPT for More Power

Yu Zehan

Intel FLEX

2023/06/01

# Outline of Lesson 5

## 5 Lesson 5: Customize GPT for More Power

# GPT Survey: A Programmer's Perspective

## Lesson 5: Customize GPT for More Power

Yu Zehan

Intel FLEX

2023/06/01





# References I

1. SPQA: The AI-based Architecture That'll Replace Most Existing Software.  
- <https://danielmiessler.com/blog/spqa-ai-architecture-replace-existing-software>
2. Introduction to prompt engineering - Azure OpenAI.  
- **Introduction:** <https://learn.microsoft.com/en-us/azure/cognitive-services/openai/concepts/prompt-engineering>  
- **Techniques:** <https://learn.microsoft.com/en-us/azure/cognitive-services/openai/concepts/advanced-prompt-engineering>
3. Azure OpenAI Service models.  
- <https://learn.microsoft.com/en-us/azure/cognitive-services/openai/concepts/models>
4. Chat GPT For Programming: 100+ Coding Prompts For Chat GPT.  
- <https://www.greataiprompts.com/chat-gpt/best-coding-prompts-for-chat-gpt>
5. Chat-GPT Cheat Sheet V2 : coolguides.  
- **Reddit:** [https://www.reddit.com/r/coolguides/comments/12ouw8o/chatgpt\\_cheat\\_sheet\\_v2](https://www.reddit.com/r/coolguides/comments/12ouw8o/chatgpt_cheat_sheet_v2)  
- **Cheatsheet:** <https://i.redd.it/eunvmybl9cua1.png>
6. ChatGPT Prompt Engineering Principles for Developers — Clarity and Specificity in Instructions.  
- <https://levelup.gitconnected.com/chatgpt-prompt-engineering-principles-for-developers-clarity-and-specificity-in-instructions-61f60ffd1eed>
7. How to Communicate with ChatGPT – A Guide to Prompt Engineering.  
- <https://www.freecodecamp.org/news/how-to-communicate-with-ai-tools-prompt-engineering>
8. Learn Prompting: Your Guide to Communicating with AI.  
- **Doc:** <https://learnprompting.org/docs/intro>  
- **GitHub:** [https://github.com/trigaten/Learn\\_Prompting](https://github.com/trigaten/Learn_Prompting)

# References II

9. Prompt Engineering Guide.
  - Doc: <https://www.promptingguide.ai/>
  - GitHub: <https://github.com/dair-ai/Prompt-Engineering-Guide>
10. StarCoder: A State-of-the-Art LLM for Code.
  - HuggingFace: <https://huggingface.co/bigcode/starcoder>
  - GitHub: <https://github.com/bigcode-project/starcoder>
  - Blog: <https://huggingface.co/blog/starcoder>
  - Paper: <https://drive.google.com/file/d/1cN-b9GnWtHzQRoE7M7gAEyivY0kl4BYs/view>
11. Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond.
  - arXiv: <https://arxiv.org/abs/2304.13712>
  - GitHub: <https://github.com/mooler0410/llmspracticalguide>
12. ChatGPT Prompt Engineering for Developers.
  - Official Course: <https://www.deeplearning.ai/short-courses/chatgpt-prompt-engineering-for-developers/>
  - Notebooks: <https://github.com/datawhalechina/prompt-engineering-for-developers/tree/main>
  - Bilibili: <https://www.bilibili.com/video/BV1No4y1t7Zn>
13. FlowGPT — the best ChatGPT prompts&AI prompts community.
  - <https://flowgpt.com/>
14. awesome-chatgpt-prompts: This repo includes ChatGPT prompt curation to use ChatGPT better.
  - GitHub: <https://github.com/f/awesome-chatgpt-prompts>
  - Webpage: <https://prompts.chat/>
  - HuggingFace: <https://huggingface.co/datasets/fka/awesome-chatgpt-prompts>
15. The Art of ChatGPT Prompting: A Guide to Crafting Clear and Effective Prompts.
  - <https://app.gumroad.com/d/e2e51298d50b8c1bd6c6950a6d11a772>

# References III

## 16. PGF/TikZ Manual.

- [PDF](http://mirrors.ctan.org/graphics/pgf/base/doc/pgfmanual.pdf): <http://mirrors.ctan.org/graphics/pgf/base/doc/pgfmanual.pdf>
- [Online Doc](https://tikz.dev/): <https://tikz.dev/>
- [GitHub](https://github.com/pgf-tikz/pgf): <https://github.com/pgf-tikz/pgf>

