Agile Development and Prompts

Blurring the Line Between Specification and Code
Administrative Details

● **Credits**
  ○ Anthropic accounts: confirming Today
  ○ OpenAI: confirming Tuesday
  ○ Baseten [https://app.baseten.co/signup/](https://app.baseten.co/signup/): $30 -> $130 (use Stanford email to get credit)
  ○ I can create a temporary API key for OpenAI if a project is in urgent need

● **Projects**
  ○ We have 20 project teams submitted, and 12 invitations to join a github repo

● **Attendance**
  ○ Visit website and click attendance link there, code phrase: suburban prompt engineers
  ○ [https://forms.gle/3jjPxDGW7wGZFAa8](https://forms.gle/3jjPxDGW7wGZFAa8)
Feb. 1, Sprint 1  3 Minute Mini-Presentations

- **Requirement:** include preliminary Model results for some part of the project
  - Use ChatGPT/GitHub copilot as additional teammates
  - See if they can help organizing the flow of the work ahead
  - Come discuss/brainstorm with us if you need project related applications

- **Show something to the class that you were able to accomplish**
  - An interesting output
  - An AI driven flow, or data organization

- **Give one direction you hope to expand on next sprint**
  - New capability
  - More interactions
Prompts and Scaling

- Timeline for LLMs and Copyright
- Review from Tuesday
- Agile approach for developing prompts
- A preliminary “clean” example
- Mitigating weaknesses
A Timeline for LLMs and Copyright

- Sept. 2023
  - Fiction authors incl. George R.R. Martin sue OpenAI

- Nov. 2023
  - Nonfiction authors sue OpenAI

- Dec. 2023
  - Axel Springer agreement with OpenAI
  - NYT sues OpenAI

- Jan. 2024
  - OpenAI indicates training without copyrighted material would not work
Review

- **Model Scaling**
  - Prompts can be much more elaborate than before
  - Token pricing is dropping fast allowing greater creativity
  - Provide exactly the data you want the model to discuss

- **Emergence**
  - Embrace the built in models of the world, of humans, and of communication
  - Expand prompts following the guidelines those internal models offer
Internal Models

People

Communication

People

People

World
An Anthropomorphic Scaffolding For Prompting

- Write a scenario for how you are going to solve your project
- Your project is a performance, and models are the actors
- The decor is the context and support code you provide
- The narrator provides the flow of data setting up the actors
- The actors perform the scenario and the curtain call is the successful outcome
An Agile Approach to Getting There

- Start with bare bones everything
- Build up from no decor and a single actor performing a soliloquy
- Iteratively build the story from the central theme
- Add other main characters, then bit players
- Add narration and decor to flesh out the story
A Preliminary Example

What are the five most popular programming languages of the 21st century in order

The popularity of programming languages can vary based on different criteria such as usage, community support, job market demand, etc. However, based on a combination of these factors, the five most popular programming languages of the 21st century, generally speaking, are:

1. **JavaScript**: Widely used for web development, JavaScript is essential for front-end development and has grown significantly with the advent of Node.js for server-side programming.

2. **Python**: Known for its simplicity and readability, Python has become a favorite for beginners and is widely used in scientific computing, data analysis, artificial intelligence, and web development.

3. **Java**: A staple in enterprise environments, Java is known for its portability across platforms, making it a popular choice for large-scale systems and Android app development.

4. **C#**: Developed by Microsoft, C# is a key language for Windows applications and has grown with the .NET framework. It’s widely used in game development, particularly with the Unity game engine.

5. **C++**: An extension of the C language, C++ is known for its performance and efficiency. It’s used in system/software development, game development, and applications requiring real-time processing.
How relevant is Tiobe's ranking of programming languages

The TIOBE Index is a widely recognized indicator of the popularity of programming languages. Its relevance is primarily due to several factors:

1. **Broad Data Sources**: TIOBE calculates its rankings based on a variety of data sources, including search engine queries, technical websites, and publications. This wide range of sources provides a comprehensive view of the popularity of programming languages across different platforms and communities.

2. **Consistency and Longevity**: TIOBE has been tracking programming language popularity for many years, providing a long-term view of trends in the programming world. This historical data is valuable for understanding how languages have evolved over time.

3. **Industry Recognition**: The index is frequently cited by industry professionals and media outlets, contributing to its status as a reputable source. Many organizations and developers look to the TIOBE Index for insights into current trends and to gauge the market demand for certain programming skills.

4. **Useful for Identifying Trends**: TIOBE is particularly useful for identifying long-term trends in programming language popularity. This can help in decision-making for educational purposes, career planning, and understanding the technology market.
Example - 3

However, it's important to consider the limitations of the TIOBE Index:

- **Focus on Web and Search Popularity**: TIOBE primarily relies on search engine data, which may not fully represent the actual usage of languages in industry projects.

- **Doesn't Measure Quality or Suitability**: The index doesn't assess the quality, performance, or suitability of a language for specific tasks. It's purely a measure of popularity.

- **Subject to Fluctuations**: Short-term changes in the rankings might be influenced by temporary factors, such as the release of a new language version or a trending technology topic.

In summary, while the TIOBE Index is a relevant and useful tool for gauging the popularity of programming languages, it should be considered alongside other factors like language capabilities, community support, and specific project or industry needs.
Example - 4

How would you contrast your ranking of the 5 top programming languages of the 21st century with the TIOBE Index?

To contrast the ranking I provided earlier with the TIOBE Index, it’s important to understand the basis of each ranking and the context in which they are used.

1. Criteria for Ranking:
   - My Ranking: The ranking I provided is based on a general assessment of popularity considering various factors such as usage in different domains (web development, data science, etc.), community support, educational adoption, and job market demand.
   - TIOBE Index: TIOBE focuses on the popularity of programming languages based on search engine queries, which include searches in popular search engines, technical websites, and specific language tutorials and documentation.

2. Time Sensitivity:
   - My Ranking: It’s a more static and holistic view, considering the impact and adoption of these languages throughout the 21st century.
   - TIOBE Index: TIOBE updates its rankings monthly, making it more sensitive to current trends, new language releases, and immediate shifts in interest within the programming community.
3. **Specificity of Use Cases:**
   - **My Ranking:** It takes into account specific use cases and domains where these languages excel, like Python in data science or JavaScript in web development.
   - **TIOBE Index:** It is more neutral in terms of use cases or domains. A language’s ranking in TIOBE doesn’t specify the areas where it’s most used or popular.

4. **Community and Educational Influence:**
   - **My Ranking:** It includes the influence of educational practices and community support, which is crucial for beginners and the growth of the language.
   - **TIOBE Index:** The focus is more on professional and commercial usage as reflected in search queries and technical discussions online.

5. **Industry Trends and Job Market:**
   - **My Ranking:** The selection is influenced by industry trends and job market demands.
   - **TIOBE Index:** While it reflects what languages programmers are interested in or querying about, it doesn’t directly correlate with job market demands.

In summary, while there are overlaps in the rankings (like the presence of languages such as JavaScript, Python, and Java in both lists), the TIOBE Index provides a dynamic, search-query-based popularity measure, whereas my ranking is a more static, multi-faceted assessment, considering a range of factors from industry use to educational impact.
Tiobe over Time

![Tiobe Programming Community Index](source: www.tiobe.com)
Example - 6

Which industry ranking of programming languages is most consistent with yours

The industry ranking of programming languages that most closely aligns with the ranking I provided earlier—emphasizing JavaScript, Python, Java, C#, and C++—is likely the **Stack Overflow Developer Survey**. This survey is conducted annually and gathers data from a large number of developers around the world. It's well-regarded for its comprehensive insights into the trends and preferences within the developer community. Here's why this ranking is consistent with the one I provided:

1. **Wide Developer Base**: Stack Overflow's survey collects responses from a diverse range of developers, from professionals to hobbyists, across various domains. This diversity offers a broad view of the programming landscape.

2. **Multiple Factors Considered**: The survey considers not just the popularity of languages but also aspects like developer preferences, job market demand, and community support. This aligns with the multi-faceted approach of my ranking.
1. **Focus on Practical Use and Trends**: Like my ranking, Stack Overflow's survey reflects practical use in the industry, current trends, and the evolving landscape of technology development.

2. **Educational and Community Influence**: The survey also captures insights into what languages are being learned by beginners and which ones have strong community support, which were factors in my ranking.

3. **Dynamic and Updated Annually**: Although not as frequently updated as the TIOBE Index, the annual nature of the survey keeps it relatively current and in tune with ongoing changes in the industry.

It's important to note that while the Stack Overflow Developer Survey is a good match, no industry ranking can be perfectly aligned with another due to differences in methodology, sample size, and the specific criteria used for ranking. Additionally, other surveys and indices like the GitHub Octoverse, RedMonk Programming Language Rankings, and the IEEE Spectrum ranking also provide valuable insights and might show similar trends.
Observed Strengths and Weaknesses

- Remarkable summarization and ranking capabilities
- Very weak temporal command
- Fairly consistent answering style
- Strong recall of highly ranked/relevant equivalents
- Makes up plausible data points, explanations
- Sometimes says the opposite of what it ought to mean
Mitigating Weaknesses

How do you flesh out a prompt to play to strengths and reduce weaknesses?

- Specify a style and format for answers
- Supply data for any non top ranked information
- Summarizations and short lists, good
- Elaborations and long lists, bad to ugly
- Limit any kind of justifications and qualitative explanations
- Time and math? Forget about it (for now)
Problem Solving Scenarios

- Design away temporal, mathematical, long tail aspects of your problem
- Make sure you focus on concrete, quantitative, factual interactions
- Whenever possible supplement prompts with relevant data
- Avoid elaborations that can not be constructed from supplied data
- Lean on summarization and synthesis capabilities
- Output specifications like JSON and code are quite accurate