JAITHON

A Simple Introduction to Math and Programming

We suffer so that you don't

Problem Statement/Intro

When I started developing Jaithon, my motivation was simple yet profound. It all started with my younger brother who had outgrown Scratch, a block-based programming language. He wanted to learn a Text-Based language, and I decided to teach him Java. My brother's struggle with the Java syntax quickly became evident, prompting me to seek a more beginner-friendly language. We quickly transitioned to Python, with its English-like syntax, but even this posed challenges for my brother who had questions. For example, he would ask me “Why are there so many parentheses?”, and I was short of an answer, for the simple fact that I couldn't introduce him to the concepts of objects and functions so early. I immediately saw a problem that needed a solution and embarked on the journey to design a programming language that would allow young learners to code. Thus, Jaithon was born, with its core principle being simplicity. By introducing an intuitive syntax and minimizing complex concepts, Jaithon strives to provide a smoother pathway for beginners to explore the world of programming. In essence, Jaithon emerged as a response to a young learner's queries, paving the way for a more approachable and enjoyable programming experience.

Solution Overview

At the centre of Jaithon lies the principle of simplicity. This programming language utilizes an intuitive syntax that mirrors spoken English, effectively acting as a stepping stone between block-based and text-based languages for users with minimal if any technical skill set.

This is important for the following reasons:

- Development of critical text-based skills
- Enhancement of pre developed abilities
- Creating a problem solving mindset
- A smooth transition to languages like Python
- Easy-to-grasp syntax

Syntax and Accessibility

Jaithon’s syntax stands as a testament to its commitment to accessibility. Unveiling a programming language that echoes simplicity. Learners can get a feel for the fundamentals of programming through its syntax, which as stated before, mirrors the English language.

The benefits of this approach vs a language with more of a technical reputation, such as Python, is stated below:

- Increased learning
- Longer attention span (with regards to coding)
- Less stress
- Happier experience

Theoretical Benefits

Jaithon's potential benefits have theoretical advantages to it's counterpart Python, which contribute to its value as a programming language for learners:

- Reduced Learning Barriers:
  - The english like syntax minimizes the initial learning curve.
  - Intuitive code readability lowers barriers for newcomers.
- Engaging Exploration:
  - With Jaithon, you can be certain that you won't get bored
  - There is always something to learn when you are talking about jaithon
- Stress Free and Better Experience:
  - Jaithons main quote is *“Bhai, tension nahin lene ka bhai!”* meaning *Have no stress.* Jaithon has made it a priority to make errors very easy to take care of. Our error handling is non technical, and will be relatively easy to solve, reducing overall stress.
  - You just don’t have to think about programming when you come over to Jaithon. It's like a new toy.

Conclusion

Jaithon is a new project, and like all new projects it has to prove itself. I hope you now have a deeper understanding of the issues in the current programming system, and how Jaithon can stop these issues. If you are convinced, please Star my GitHub repo Repo and feel free to reach out to me on linkedin Linkedin.