

# Design and implementation of a simple compiler in Python for generating personalized avatars.

Juan David Guevara García – 20202020001  
Javier Alejandro Alvarez Marin – 20202020028



Computer Science III | tSemester 2025-I  
Eng. Carlos Andrés Sierra, M.Sc.  
Computer Engineering  
Universidad Distrital Francisco José de Caldas

## INTRODUCTION

Understanding how compilers work is a complex task, often limited to abstract concepts hidden behind lines of code. This presents a learning challenge, especially for students with limited exposure to systems-level programming.[2] While tools like compiler visualizers or syntax trees exist, they rarely offer interactive or intuitive representations. This project addresses that gap by proposing a visual and tangible simulation of a compiler, using natural language commands to generate digital avatars. The main challenge lies in translating simplified human input into structured representations that reflect key compiler processes, such as lexical analysis, parsing, and code generation—without sacrificing usability or clarity.

## GOAL

**Objective:** Develop a Python-based command interpreter that acts as a visual compiler[3], allowing users to create digital avatars through written instructions and better understand compiler principles.

**Research Question:** How can compiler concepts be taught through visual and interactive representations?

**Expected Outcome:** An educational system that translates natural language commands into customized SVG avatar images.[2]

## EXPERIMENTS

To validate the system, we conducted unit tests on the lexical, syntactic, and semantic analyzers. Tests used both valid and invalid user inputs to evaluate the accuracy and robustness of each module.

### Case 1: Valid Instructions

inicio  
teñir cabello castaño\_oscuro;  
ajustar ropa hoodie;  
añadir accesorio gafas\_redondas;  
expresar boca sonriente;  
expresar ojos feliz;  
final

**Result:** All analyzers accepted the input. The semantic analyzer correctly mapped attributes, and the avatar was rendered using the py-avataaars library.[1]

### Case 2: Invalid Semantic Instruction

inicio  
teñir ropa afro;  
final

#### Result:

Lexical and syntactic analyzers passed.

Semantic analyzer returned the error:

"Invalid value "afro" for attribue 'ropa'

## RESULTS

**Full Pipeline Integration**  
A comprehensive input was tested, combining multiple instructions. The avatar was generated with:

Input:  
inicio  
ajustar piel trigueña;  
teñir cabello rojo;  
ajustar cabello afro;  
teñir ropa pastel\_azul ;  
ajustar ropa hoodie;  
añadir accesorio gafas\_sol;  
expresar cejas fruncidas;  
expresar ojos sorprendidos;  
expresar boca gritando  
final



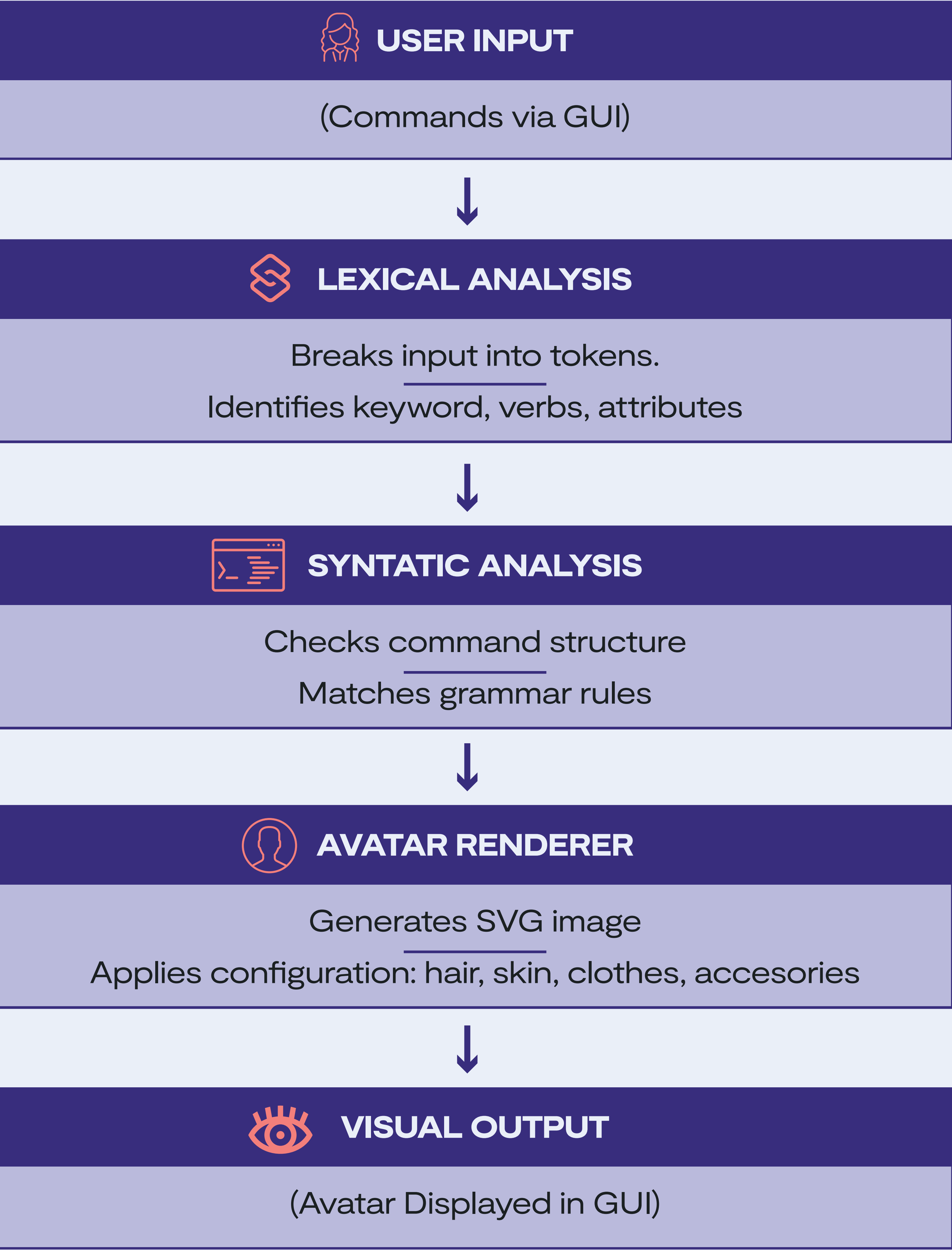
## BIBLIOGRAPHY

[1] K. Ebert, "py-avataaars – Python component for Avataaars", PyPI. [Online]. Available: PyPI. [Accessed: Jul. 2025].

[2] S.Stamenković and N.M.Jovanović, "A Web-Based Educational System for Teaching Compilers," IEEE Trans. Learn. Technol., vol. 17, Jul. 2024.

[3] M. Lutz, Learning Python, 5th ed. O'Reilly Media, 2013. [Online]. Available: <https://learning.oreilly.com/library/view/learning-python-5th/9781449355722/>

## PROPOSED SOLUTION



## CONCLUSION

The project successfully achieved its goal of simulating a visual compiler using Python.[3] By translating simplified natural language commands into avatar components[1], the system demonstrated key compilation stages—such as parsing and semantic analysis—through visual feedback. This approach effectively answered the research question, showing that compiler principles can be taught using interactive and visual representations, making them more accessible for learners with limited technical backgrounds.[2]