The Astro Documentation

by Lotus

General Information

Astro's Concept

Astro's Concept is pretty simple and easy to summarize, Astro is designed for fast-paced and efficient coding, the level of abstraction should guarantee a fast and comfortable coding session. A programming language can only be used as good as the coder using it is. Theory says, that if you learn Astro the right way there should be no unnecessary lines, that only applies if you got the skill to back it up. This project got a lot of inspiration from Python, a fairly modern programming language still widely used, despite it's release date.

Compiled or Interpreted?

Astro is indeed interpreted, we decided to design it interpreted because to be honest, we don't have enough experience to make it as low-level as a compiled language. Despite that, Astro would probably take way too long to develop for if we decided to make it a compiled language simply because of it's low-level aspect and the size of the development team.

General Explanation

Astro's Compilation & Interpretation Process

(Compilation) (Python)

- File Handle gets established (*cls: AstroFile*)
- Tokenization of file content, char by char. (*cls: Tokenization*)
- “Compression” of tokens into smaller sub-tokens for compilation (*cls: Tokenization*)
- Post-compression token sets get passed to the syntax tree (*cls: Parser*)
- compilation of bytecode based on generated tree from the Parser (=> *.abc) (*cls: Bytecode*)

(Byte Code Interpretation) (C)

- Read byte code from *.abc file
- Interpret byte code in Astro's byte code format

Low-Level Explanation

Astro's Low-Level Interpretation Process

The AstroFile class is initialized, and AstroFile.content (*str*) gets assigned to the given files content given in AstroFile.file_name (*str*).

The Tokenizer class is initialized, each line from AstroFile.content will now get tokenized with the Token class, the result of the tokenized lines is stored in Tokenizer.tokens (format: [[token_line...], [token_line]...]).
Now the tokens get compressed for a ready-to-compile format with the Tokenizer.compress(token_id) method.
The compressed tokens get passed into the now overwritten Tokenizer.tokens.

This is then ready to be passed to the Parser class, which checks for syntax and generates a tree to maket the file easily compilable to Astro's byte code format (**.abc*).