SyntaxBlend [ReadMe]

We will be using Python3 for this project. The language that we will be creating will be our own implementation of SAAS.

In this module -- 1lexerSimple, we create the lexer, which breaks up our input into a list of what we call tokens. We can then use those tokens in the next module to further analyze the code.

To get the output of the first module -- 1lexerSimple Commands:

```
cd 1lexerSimple
```

```
python3 shell.py
```

In this second module -- 2parserSimple we create the parser, which takes in the tokens created by the lexer and builds up what we call an abstract syntax tree, which is basically a tree of the structure of the code. We can then traverse that tree in the next module and interpret it.

To get the output of the second module -- 1parserSimple Commands:

```
cd 2parserSimple
```

```
python3 shell.py
```

In this third module --3interpreterSimple we create the interpreter, which traverses the tree we built up in the last module and executes the appropriate code. By the end of this module our simple expression interpreter is complete, supporting ints, floats, simple operations and parentheses.

To get the output of the third module -- 3interpreterSimple Commands:

```
cd 3interpreterSimple
```

```
python3 shell.py
```

In this module -- 4interpreterSimple2 we will be adding a power operator the the language, and this involves updating the lexer, the parser and the interpreter.

To get the output of the fourth module -- 4interpreterSimple2 Commands:

```
cd 4interpreterSimple2
```

```
python3 shell.py
```

In this module --5variables we add support for variables in our language. This requires updating the lexer, parser and interpreter, along with adding a new SymbolTable class.

To get the output of the fifth module -- 5variables Commands:

```
cd 5variables
```

```
python3 shell.py
```

In this module -- 6comparisonOperators we add all the different comparison operators to our language: equals, not equals, less than, greater than, etc.

To get the output of the sixth module -- 6comparisonOperators Commands:

```
cd 6comparisonOperators
```

```
python3 shell.py
```

In this module -- 7ifStatement we add in the IF statement.

To get the output of the seventh module -- 7ifStatement Commands:

```
cd 7ifStatement
```

```
python3 shell.py
```

In this module -- 8forAndWhile we add in the FOR statement and WHILE statements.

To get the output of the eighth module -- 8forAndWhile Commands:

```
cd 8forAndWhile
```

```
python3 shell.py
```

In this module -- 9functions we add the Functions to our programming language.

To get the output of the ninth module -- 9functions Commands:

```
cd 9functions
```

```
python3 shell.py
```

In this module -- 10strings we add in the Strings.

To get the output of the 10th module -- 10strings Commands:

```
cd 10strings
```

```
python3 shell.py
```

In this module -- 11lists we add another new type to our language - lists.

To get the output of the 11th module -- 11lists Commands:

```
cd 11lists

python3 shell.py
```

In this module -- 12builtInFuncs we add quite a few built-in functions to our language, including print, input, clear, type checking, and list operations. You may add plenty more, such as string operations, math functions, random number generators and more!

To get the output of the 12th module -- 12builtInFuncs Commands:

```
cd 12builtInFuncs

python3 shell.py
```

In this module -- 13multiLineStatements we add multi-line statements to our language (for if, for, while and functions).

To get the output of the 13th module -- 13multiLineStatements Commands:

```
cd 13multiLineStatements

python3 shell.py
```

In this module -- 14returnContBreak we add the return, continue and break statements to functions and loops!

To get the output of the 14th module -- 14returnContBreak Commands:

```
cd 14returnContBreak

python3 shell.py
```

In this module 15fileInput we add run statement to run an external file and comments.

To get the output of the 15th module -- 15fileInput Commands:

```
cd 15fileInput

python3 shell.py
```

Documentation

IEEE Research Paper

Requirements

To run this project, you will need to add the following requirements

Python3 should be installed `python3 --version`

If not installed then, `brew install python3`

Installation

Install my-project with npm

npm should be installed

```
npm install my-project
cd my-project
```

For Users

Feel free to leave a comment or dm personally if you have any problems, and don't forget to like and fork if you enjoyed :)