Parser Examples

- Grammar

Program → StatementList EOF

StatementList → Statement StatementList | ε

Note: there is a recursion here we need to define at the parser when will this stop.

Statement →

Declaration

| Assignment

| IfStmt

| WhileStmt

| ForStmt

| FunctionStmt

| OutputStmt

| InputStmt

| ReturnStmt

| Comment

| ε

Declaration → (int | float | string | bool | double) Identifier = Expr ;

Assignment → Identifier = Expr ;

IfStmt → if (Condition) { StatementList } |

if (Condition) { StatementList } else { StatementList }

WhileStmt → while (Condition) { StatementList }

ForStmt → for (Declaration ; Condition ; Assignment) { StatementList }

FunctionStmt → def (Identifier | main)( ) { StatementList }

ReturnStmt → return Expr ;

Condition → Expr RelOp Expr

Expr → Term | Term (+ | -) Term

Term → Factor | Factor (* | /) Factor

Factor → Number | Identifier | String | (Expr)

Note: we split these To make sure we succeed in achieving the precedence of the mathematical operations

RelOp → == | != | < | > | <= | >=

OutputStmt → print (Expr) ;

InputStmt → input (Identifier) ;

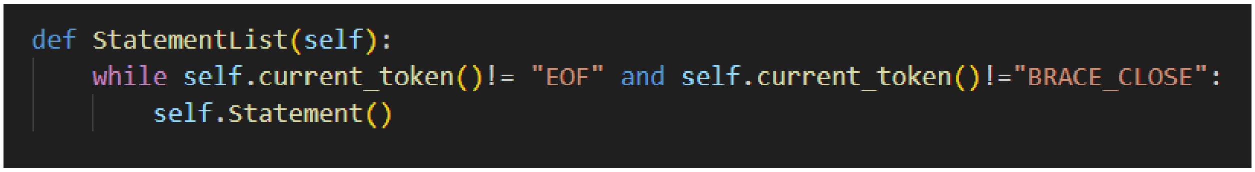
Comment → // any_characters_until_end_of_line

Note: When ever we got OR | between multiple choices we need to check on the current Token to find out which choice will we choose

Things would make it Easier on you :

- Function to Show the current Token
- Function to check if the expected token match the current one and move to next
- Function for each NonTerminal

Example for the Statement List , How to Handle Recursion call



```
def StatementList(self):
    while self.current_token()!= "EOF" and self.current_token()!="BRACE_CLOSE":
        self.Statement()
```

"A StatementList is made up of zero or more Statements, until a stopping point (either end of file or a closing brace)."

There will be no more statements at the end of the File

Also If statement ends and there is a closing bracket after

this means the statements List here is done

If u r having python structure in your language u will check the indentation beside the EOF

Final Notes:

1. Make sure that you are handling the errors when ever a statement error appears.
2. No need to find errors in the whole program at once.
3. Modify the grammar to reach your own grammar that satisfy your programming language.
4. Make sure **you and all of your Team** Understand the Code very well and the grammar of the Language.
5. If you have any concerns reach me out on Teams.