Lexer

Class:
- Token → type, val, lineNum
  ↳ get_type(), get_val(), get_lineNum(), <<

- Symbol → type, scope, value, lineOfDeclaration
  ↳ <<

vector:
- Keywords
- booleans
- arithmetic_operators
- rational_operators
- assignment_operators
- logical_operators
- all_operators
- misc
- Keyword_statements
- valid_parenthesis
- Tokens

unordered_map:
- Sym_table

functions:
- Scan()
- is_keyword()
       identifier()
       operator()
       boolean()
       misc()
       Keystatement()
       parenthesis()
- operator_type()
- get_parenthesis_type()
- reset()
- tester()

Deconstructing Scan ( )

```
Scan
  main while loop
      if newline        → increment lineNum

      if statement       → store token for statement

      if " is found      → store everything until next "

      if keyword         → store token for keyword

      if identifier      → store token for identifier

      if parenthesis
              if '[' or ']'    → lex for an array

              if '(' or ')'    → lex for normal paran

      if misc
              if ';'           → lex for semi colon

              if ','           → lex for comma

              if '//'          → lex for comment

      is literal         → lex for literal

      is alphanum        → add char
```