RabbitLang documentation

# Code

%{

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int current\_line = 1;

%}

%option noyywrap

IDENTIFIER \^[a-zA-Z\_][a-zA-Z0-9\_]\*

NUMBER\_CONST 0|[+|-]?[1-9][0-9]\*([.][0-9]\*)?|[+|-]?0[.][0-9]\*

STRING\_CONST [\"][a-zA-Z0-9 ]+[\"]

CHAR\_CONST [\'][a-zA-Z0-9 ][\']

%%

"if"|"print"|"inputInt"|"inputString"|"else"|"elseif"|"for"|"-->"|"while"|"set\_nth"|"get\_nth"|"False"|"True"|"len" {printf("Reserved word: %s\n", yytext);}

"+"|"-"|"\*"|"/"|"="|">"|">="|"<"|"<="|"=="|"not"|"and"|"or" {printf("Operator: %s\n", yytext);}

"\n"|"("|")"|"\["|"\]"|"#"|","|"\|" {printf("Separator: %s\n", yytext == "\n" ? "\\n" : yytext);}

{IDENTIFIER} {printf("Identifier: %s\n", yytext);}

{NUMBER\_CONST} {printf("Number: %s\n", yytext);}

{STRING\_CONST} {printf("String: %s\n", yytext);}

{CHAR\_CONST} {printf("Character: %s\n", yytext);}

[ \t]+ {}

[\n]+ {current\_line++;}

[0-9][a-zA-Z0-9\_]\* {printf("Illegal identifier at line %d\n", current\_line);}

[+|-]0 {printf("Illegal numeric constant at line %d\n", current\_line);}

[+|-]?[0][0-9]\*([.][0-9]\*)? {printf("Illegal numeric constant at line %d\n", current\_line);}

[\'][a-zA-Z0-9 ]{2,}[\']|[\'][a-zA-Z0-9 ][a-zA-Z0-9 ][\'] {printf("Illegal character constant at line %d\n", current\_line);}

[\"][a-zA-Z0-9\_]+|[a-zA-Z0-9\_]+[\"] {printf("Illegal string constant at line %d\n", current\_line);}

%%

void main(argc, argv)

int argc;

char\*\* argv;

{

if (argc > 1)

{

FILE \*file;

file = fopen(argv[1], "r");

if (!file)

{

fprintf(stderr, "Could not open file %s\n", argv[1]);

exit(1);

}

yyin = file;

}

yylex();

}

## Documentation

To run the project, type:

flex scanner.l

gcc lex.yy.c -o lex

./lex <input\_file> (where input\_file is the file with the source code)