

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)
```