

# Lexer

A presentation that I don't expect to present

By Ethan Smith

# It reads each line and breaks on spaces

There's a special case for quotes ( string) and pound ( comment)

Strips out newlinecharacters.

# Regex based on rules

Keywords are upper case

Identifiers are lower case, and can't start with number

Comments start with ( and can end with ) a # symbol

Lines end with semicolon

Space between any token

Numbers can have decimal

```
if not found: # if the token is an error/ invalid identifier... an example would be an unclosed string, or a number with multiple dots
    print(f"Type: ERROR \tValue{s}")
def parseLine(l):
    full = ""
    str = False
    for c in l: # step through each character in the line
        full += c # full sequence we're checking
        if c == "\"" or (c == "#" and str == False): # checks if we're leaving a string or comment, and checks its type
            if str:
                str = False
                callall(full)
                full=""
            else:
                str = True
        if c == " " and str == False: # hits a space, which is the delimiter between tokens
            a = full.rstrip() # strip newline character
            callall(a)
            full = ""
    a = full.strip() # run once more after the loop ends to clear the last thing in the buffer*
    callall(a)
    print()
```

# output

Here's some output....

There is an unshown error for when a  
type isn't recognized

"This can happen with an invalid  
number like 123.4.2.421 or a string that  
Doesn't have an ending quotation mark,  
like this entire paragraph.

```
Type: Comment    Value: # this is a comment and it is going to be ignored
```

```
Type: Identifier  Value: x
```

```
Type: Assignment Value: =
```

```
Type: Number     Value: 22
```

```
Type: Operator   Value: +
```

```
Type: Number     Value: 44
```

```
Type: Endline    Value: ;
```

```
Type: String     Value: "Hi there!!!"
```

```
Type: Endline    Value: ;
```

```
Type: Keyword    Value: DEFINE
```

```
Type: Identifier  Value: x1
```

```
Type: Assignment Value: =
```

```
Type: String     Value: "Hi there"
```

```
Type: Endline    Value: ;
```

```
Type: Keyword    Value: UPDATE
```

```
Type: Identifier  Value: x1
```

```
Type: Assignment Value: =
```

```
Type: String     Value: "see you later"
```

```
Type: Endline    Value: ;
```

```
Type: Keyword    Value: PRINT
```

```
Type: String     Value: "Hi"
```