

# Lab 1 Lexer

Sorin Macaluso

February 13, 2024

## 1 Test Cases

These are the test cases that I used for the program these are taken from Lexer without spaces and Gabriel Arnell Hall of Fame project. I have added changes to them to have unrecognized tokens and key words, as well a numbers in strings, pretty much anything that is not a lower case letter or space (since lowercase letters and space's are the only things allowed inside quotes).

I attempted to make it so that the tokens would match. Could not get all of them without ruining the document.

### 1.1 Test Case 1

---

```
1  /*Lexer without spaces*/
2  /*Long Test Case - Everything Except Boolean Declaration */
3  {
4      /* Int Declaration */
5      int a
6      int b
7      a = 0
8      b=0
9      /* While Loop */
10     while (a != 3) {
11         print(a)
12         while (b != 3) {
13             print(b)
14             b = 1 + b
15             if (b == 2) {
16                 /* Print Statement */
17                 print({"there is no spoon" /* This will do nothing */ )
18             }
19         }
20         b = 0
21         a = 1+a
22     }
23 }
24 $
```

```
26 Output:
27 Begin_Block [ { ] Found at line 2 : 1
28 TYPE [ int ] Found at line 4 : 11
29 ID [ a ] Found at line 4 : 13
30 TYPE [ int ] Found at line 5 : 11
```

```
31 ID [ b ] Found at line 5 : 13
32 ID [ a ] Found at line 6 : 9
33 AssignmentStatement [ = ] Found at line 6 : 11
34 DIGIT [ 0 ] Found at line 6 : 13
35 ID [ b ] Found at line 7 : 9
36 AssignmentStatement [ = ] Found at line 7 : 10
37 DIGIT [ 0 ] Found at line 7 : 11
38 While_Statment [ while ] Found at line 9 : 13
39 Open_Expression [ ( ] Found at line 9 : 15
40 ID [ a ] Found at line 9 : 16
41 Not_Equal [ != ] Found at line 9 : 19
42 DIGIT [ 3 ] Found at line 9 : 21
43 Close_Expression [ ) ] Found at line 9 : 22
44 Begin_Block [ { ] Found at line 9 : 24
45 Print_Statment [ print ] Found at line 10 : 17
46 Open_Expression [ ( ] Found at line 10 : 18
47 ID [ a ] Found at line 10 : 19
48 Close_Expression [ ) ] Found at line 10 : 20
49 While_Statment [ while ] Found at line 11 : 17
50 Open_Expression [ ( ] Found at line 11 : 19
51 ID [ b ] Found at line 11 : 20
52 Not_Equal [ != ] Found at line 11 : 23
53 DIGIT [ 3 ] Found at line 11 : 25
54 Close_Expression [ ) ] Found at line 11 : 26
55 Begin_Block [ { ] Found at line 11 : 28
56 Print_Statment [ print ] Found at line 12 : 21
57 Open_Expression [ ( ] Found at line 12 : 22
58 ID [ b ] Found at line 12 : 23
59 Close_Expression [ ) ] Found at line 12 : 24
60 ID [ b ] Found at line 13 : 17
61 AssignmentStatement [ = ] Found at line 13 : 19
62 DIGIT [ 1 ] Found at line 13 : 21
63 InTop [ + ] Found at line 13 : 23
64 ID [ b ] Found at line 13 : 25
65 If_Statment [ if ] Found at line 14 : 18
66 Open_Expression [ ( ] Found at line 14 : 20
67 ID [ b ] Found at line 14 : 21
68 Equal [ == ] Found at line 14 : 24
69 DIGIT [ 2 ] Found at line 14 : 26
70 Close_Expression [ ) ] Found at line 14 : 27
71 Begin_Block [ { ] Found at line 14 : 29
72 Print_Statment [ print ] Found at line 16 : 25
73 Open_Expression [ ( ] Found at line 16 : 26
74 QUOTE [ " ] Found at line 16 : 27
75 CHAR [ t ] Found at line 16 : 28
76 CHAR [ h ] Found at line 16 : 29
77 CHAR [ e ] Found at line 16 : 30
78 CHAR [ r ] Found at line 16 : 31
79 CHAR [ e ] Found at line 16 : 32
80 SPACE [   ] Found at line 16 : 33
81 CHAR [ i ] Found at line 16 : 34
82 CHAR [ s ] Found at line 16 : 35
83 SPACE [   ] Found at line 16 : 36
84 CHAR [ n ] Found at line 16 : 37
```

```

85 CHAR [ o ] Found at line 16 : 38
86 SPACE [   ] Found at line 16 : 39
87 CHAR [ s ] Found at line 16 : 40
88 CHAR [ p ] Found at line 16 : 41
89 CHAR [ o ] Found at line 16 : 42
90 CHAR [ o ] Found at line 16 : 43
91 CHAR [ n ] Found at line 16 : 44
92 QUOTE [ " ] Found at line 16 : 45
93 Close_Expression [ ) ] Found at line 16 : 74
94 End_Block [ } ] Found at line 17 : 17
95 End_Block [ } ] Found at line 18 : 13
96 ID [ b ] Found at line 19 : 13
97 AssignmentStatement [ = ] Found at line 19 : 15
98 DIGIT [ 0 ] Found at line 19 : 17
99 ID [ a ] Found at line 20 : 13
100 AssignmentStatement [ = ] Found at line 20 : 15
101 DIGIT [ 1 ] Found at line 20 : 17
102 InTop [ + ] Found at line 20 : 18
103 ID [ a ] Found at line 20 : 19
104 End_Block [ } ] Found at line 21 : 9
105 End_Block [ } ] Found at line 22 : 5
106 END_OF_PROGRAM [ $ ] Found at line 23 : 1
107 Number of Errors is 0 :(
108 End of program 1

```

---

## 1.2 Test Case 2

```

1  /*Lexer "without any spaces"*/
2  /*LongTestCase-EverythingExceptBooleanDeclaration*/{/*IntDeclaration*/intaintba=0b=0/
3  *WhileLoop*/while(a!=3){print(a)while(b!=3){print(b)b=1+bif(b==2)
4  {/*PrintStatement*/print"there is no 1spoon"/*Thiswillldonothing*/)}}b=-0a=1+a}}$
5
6  Output:
7  Begin_Block [ { ] Found at line 25 : 52
8  TYPE [ int ] Found at line 25 : 73
9  ID [ a ] Found at line 25 : 74
10 TYPE [ int ] Found at line 25 : 77
11 ID [ b ] Found at line 25 : 78
12 ID [ a ] Found at line 25 : 79
13 AssignmentStatement [ = ] Found at line 25 : 80
14 DIGIT [ 0 ] Found at line 25 : 81
15 ID [ b ] Found at line 25 : 82
16 AssignmentStatement [ = ] Found at line 25 : 83
17 DIGIT [ 0 ] Found at line 25 : 84
18 While_Statment [ while ] Found at line 25 : 102
19 Open_Expression [ ( ] Found at line 25 : 103
20 ID [ a ] Found at line 25 : 104
21 Not_Equal [ != ] Found at line 25 : 106
22 DIGIT [ 3 ] Found at line 25 : 107
23 Close_Expression [ ) ] Found at line 25 : 108
24 Begin_Block [ { ] Found at line 25 : 109
25 Print_Statment [ print ] Found at line 25 : 114
26 Open_Expression [ ( ] Found at line 25 : 115

```

```
27 ID [ a ] Found at line 25 : 116
28 Close_Expression [ ) ] Found at line 25 : 117
29 While_Statment [ while ] Found at line 25 : 122
30 Open_Expression [ ( ] Found at line 25 : 123
31 ID [ b ] Found at line 25 : 124
32 Not_Equal [ != ] Found at line 25 : 126
33 DIGIT [ 3 ] Found at line 25 : 127
34 Close_Expression [ ) ] Found at line 25 : 128
35 Begin_Block [ { ] Found at line 25 : 129
36 Print_Statment [ print ] Found at line 25 : 134
37 Open_Expression [ ( ] Found at line 25 : 135
38 ID [ b ] Found at line 25 : 136
39 Close_Expression [ ) ] Found at line 25 : 137
40 ID [ b ] Found at line 25 : 138
41 AssignmentStatement [ = ] Found at line 25 : 139
42 DIGIT [ 1 ] Found at line 25 : 140
43 InTop [ + ] Found at line 25 : 141
44 ID [ b ] Found at line 25 : 142
45 If_Statment [ if ] Found at line 25 : 144
46 Open_Expression [ ( ] Found at line 25 : 145
47 ID [ b ] Found at line 25 : 146
48 Equal [ == ] Found at line 25 : 148
49 DIGIT [ 2 ] Found at line 25 : 149
50 Close_Expression [ ) ] Found at line 25 : 150
51 Begin_Block [ { ] Found at line 25 : 151
52 Print_Statment [ print ] Found at line 25 : 174
53 Open_Expression [ ( ] Found at line 25 : 175
54 QUOTE [ " ] Found at line 25 : 176
55 CHAR [ t ] Found at line 25 : 177
56 CHAR [ h ] Found at line 25 : 178
57 CHAR [ e ] Found at line 25 : 179
58 CHAR [ r ] Found at line 25 : 180
59 CHAR [ e ] Found at line 25 : 181
60 SPACE [   ] Found at line 25 : 182
61 CHAR [ i ] Found at line 25 : 183
62 CHAR [ s ] Found at line 25 : 184
63 SPACE [   ] Found at line 25 : 185
64 CHAR [ n ] Found at line 25 : 186
65 CHAR [ o ] Found at line 25 : 187
66 SPACE [   ] Found at line 25 : 188
67 Error: int not allowed in string [ 1 ] Found at line 25 : 189
68 CHAR [ s ] Found at line 25 : 190
69 CHAR [ p ] Found at line 25 : 191
70 CHAR [ o ] Found at line 25 : 192
71 CHAR [ o ] Found at line 25 : 193
72 CHAR [ n ] Found at line 25 : 194
73 QUOTE [ " ] Found at line 25 : 195
74 Close_Expression [ ) ] Found at line 25 : 217
75 End_Block [ } ] Found at line 25 : 218
76 End_Block [ } ] Found at line 25 : 219
77 ID [ b ] Found at line 25 : 220
78 AssignmentStatement [ = ] Found at line 25 : 221
79 Error: non recognized symbol [ - ] Found at line 25 : 222
80 DIGIT [ 0 ] Found at line 25 : 223
```

```
81 ID [ a ] Found at line 25 : 224
82 AssignmentStatement [ = ] Found at line 25 : 225
83 DIGIT [ 1 ] Found at line 25 : 226
84 InTop [ + ] Found at line 25 : 227
85 ID [ a ] Found at line 25 : 228
86 End_Block [ } ] Found at line 25 : 229
87 End_Block [ } ] Found at line 25 : 230
88 END_OF_PROGRAM [ $ ] Found at line 25 : 231
89 Number of Errors is 2 :(
90 Lexer failed :(
91 End of program 2
```

---

### 1.3 Test Case 3

---

```
1 /*keywords in strings*/
2 {/* Comment!!!*/ () print=whileif"teststring" intstringbooleanfalse true
3 ==!=+ a 0123456789}$
4
5 Output:
6 Begin_Block [ { ] Found at line 27 : 1
7 Open_Expression [ ( ] Found at line 27 : 18
8 Close_Expression [ ) ] Found at line 27 : 19
9 Print_Statment [ print ] Found at line 27 : 25
10 AssignmentStatement [ = ] Found at line 27 : 26
11 While_Statment [ while ] Found at line 27 : 31
12 If_Statment [ if ] Found at line 27 : 33
13 QUOTE [ " ] Found at line 27 : 34
14 CHAR [ t ] Found at line 27 : 35
15 CHAR [ e ] Found at line 27 : 36
16 CHAR [ s ] Found at line 27 : 37
17 CHAR [ t ] Found at line 27 : 38
18 CHAR [ s ] Found at line 27 : 44
19 CHAR [ t ] Found at line 27 : 44
20 CHAR [ r ] Found at line 27 : 44
21 CHAR [ i ] Found at line 27 : 44
22 CHAR [ n ] Found at line 27 : 44
23 CHAR [ g ] Found at line 27 : 44
24 QUOTE [ " ] Found at line 27 : 45
25 TYPE [ int ] Found at line 27 : 49
26 TYPE [ string ] Found at line 27 : 55
27 TYPE [ boolean ] Found at line 27 : 62
28 Boolean_Value [ false ] Found at line 27 : 67
29 Boolean_Value [ true ] Found at line 27 : 72
30 Equal [ == ] Found at line 27 : 75
31 Not_Equal [ != ] Found at line 27 : 77
32 InTop [ + ] Found at line 27 : 78
33 ID [ a ] Found at line 27 : 80
34 DIGIT [ 0 ] Found at line 27 : 91
35 DIGIT [ 1 ] Found at line 27 : 91
36 DIGIT [ 2 ] Found at line 27 : 91
37 DIGIT [ 3 ] Found at line 27 : 91
38 DIGIT [ 4 ] Found at line 27 : 91
39 DIGIT [ 5 ] Found at line 27 : 91
```

```
40 DIGIT [ 6 ] Found at line 27 : 91
41 DIGIT [ 7 ] Found at line 27 : 91
42 DIGIT [ 8 ] Found at line 27 : 91
43 DIGIT [ 9 ] Found at line 27 : 91
44 End_Block [ } ] Found at line 27 : 92
45 END_OF_PROGRAM [ $ ] Found at line 27 : 93
46 Number of Errors is 0 :(
47 End of program 3
48 Number of Errors is 0 :(
49 End of program 3
```

---

## 1.4 Test Case 4

---

```
1  /*Testing for boolean statments*/
2  {
3  if true {
4      print("if statement worked")
5  }
6  if false{
7      print("if statement failed")
8  }
9  }$
10
11 Output:
12 Begin_Block [ { ] Found at line 30 : 1
13 If_Statment [ if ] Found at line 31 : 2
14 Boolean_Value [ true ] Found at line 31 : 7
15 Begin_Block [ { ] Found at line 31 : 9
16 Print_Statment [ print ] Found at line 32 : 9
17 Open_Expression [ ( ] Found at line 32 : 10
18 QUOTE [ " ] Found at line 32 : 11
19 CHAR [ i ] Found at line 32 : 13
20 CHAR [ f ] Found at line 32 : 13
21 SPACE [   ] Found at line 32 : 14
22 CHAR [ s ] Found at line 32 : 15
23 CHAR [ t ] Found at line 32 : 16
24 CHAR [ a ] Found at line 32 : 17
25 CHAR [ t ] Found at line 32 : 18
26 CHAR [ e ] Found at line 32 : 19
27 CHAR [ m ] Found at line 32 : 20
28 CHAR [ e ] Found at line 32 : 21
29 CHAR [ n ] Found at line 32 : 22
30 CHAR [ t ] Found at line 32 : 23
31 SPACE [   ] Found at line 32 : 24
32 CHAR [ w ] Found at line 32 : 25
33 CHAR [ o ] Found at line 32 : 26
34 CHAR [ r ] Found at line 32 : 27
35 CHAR [ k ] Found at line 32 : 28
36 CHAR [ e ] Found at line 32 : 29
37 CHAR [ d ] Found at line 32 : 30
38 QUOTE [ " ] Found at line 32 : 31
39 Close_Expression [ ) ] Found at line 32 : 32
40 End_Block [ } ] Found at line 33 : 1
```

```

41 If_Statment [ if ] Found at line 34 : 2
42 Boolean_Value [ false ] Found at line 34 : 8
43 Begin_Block [ { ] Found at line 34 : 9
44 Print_Statment [ print ] Found at line 35 : 9
45 Open_Expression [ ( ] Found at line 35 : 10
46 QUOTE [ " ] Found at line 35 : 11
47 CHAR [ i ] Found at line 35 : 13
48 CHAR [ f ] Found at line 35 : 13
49 SPACE [   ] Found at line 35 : 14
50 CHAR [ s ] Found at line 35 : 15
51 CHAR [ t ] Found at line 35 : 16
52 CHAR [ a ] Found at line 35 : 17
53 CHAR [ t ] Found at line 35 : 18
54 CHAR [ e ] Found at line 35 : 19
55 CHAR [ m ] Found at line 35 : 20
56 CHAR [ e ] Found at line 35 : 21
57 CHAR [ n ] Found at line 35 : 22
58 CHAR [ t ] Found at line 35 : 23
59 SPACE [   ] Found at line 35 : 24
60 CHAR [ f ] Found at line 35 : 25
61 CHAR [ a ] Found at line 35 : 26
62 CHAR [ i ] Found at line 35 : 27
63 CHAR [ l ] Found at line 35 : 28
64 CHAR [ e ] Found at line 35 : 29
65 CHAR [ d ] Found at line 35 : 30
66 QUOTE [ " ] Found at line 35 : 31
67 Close_Expression [ ) ] Found at line 35 : 32
68 End_Block [ } ] Found at line 36 : 1
69 End_Block [ } ] Found at line 37 : 1
70 END_OF_PROGRAM [ $ ] Found at line 37 : 2
71 Number of Errors is 0 :(
72 End of program 4

```

---

## 1.5 Test Case 5

---

```

1
2 /*This is suppose to crash at the top since there is a space before the first line */
3 {
4     int a
5     a = 3
6     int b
7     b = 4
8     a = b
9     print(a)
10    if (a == b) {
11        print(a)
12    }
13 }$
14
15 Output:
16 Begin_Block [ { ] Found at line 40 : 2
17 TYPE [ int ] Found at line 41 : 10
18 ID [ a ] Found at line 41 : 12

```

```

19 ID [ a ] Found at line 42 : 8
20 AssignmentStatement [ = ] Found at line 42 : 10
21 DIGIT [ 3 ] Found at line 42 : 12
22 TYPE [ int ] Found at line 43 : 10
23 ID [ b ] Found at line 43 : 12
24 ID [ b ] Found at line 44 : 8
25 AssignmentStatement [ = ] Found at line 44 : 10
26 DIGIT [ 4 ] Found at line 44 : 12
27 ID [ a ] Found at line 45 : 8
28 AssignmentStatement [ = ] Found at line 45 : 10
29 ID [ b ] Found at line 45 : 12
30 Print_Statment [ print ] Found at line 46 : 12
31 Open_Expression [ ( ] Found at line 46 : 13
32 ID [ a ] Found at line 46 : 14
33 Close_Expression [ ) ] Found at line 46 : 15
34 If_Statment [ if ] Found at line 47 : 9
35 Open_Expression [ ( ] Found at line 47 : 11
36 ID [ a ] Found at line 47 : 12
37 Equal [ == ] Found at line 47 : 15
38 ID [ b ] Found at line 47 : 17
39 Close_Expression [ ) ] Found at line 47 : 18
40 Begin_Block [ { ] Found at line 47 : 20
41 Print_Statment [ print ] Found at line 48 : 15
42 Open_Expression [ ( ] Found at line 48 : 16
43 ID [ a ] Found at line 48 : 17
44 Close_Expression [ ) ] Found at line 48 : 18
45 End_Block [ } ] Found at line 49 : 8
46 End_Block [ } ] Found at line 50 : 6
47 END_OF_PROGRAM [ $ ] Found at line 50 : 7
48 Number of Errors is 0 :(
49 End of program 5

```

---

## 1.6 Test Case 6

---

```

1  /*General test case for stuff*/
2  {
3      int a
4      boolean b
5      string c
6      a = 9
7      b = true
8      {
9          print(a)
10         print(b)
11         b = false
12         c = "hello world"
13         int b
14         b = 0
15         {
16             print(c)
17             a = 1 + 2 + a
18             {
19                 print(b)

```



```

20         }
21     }
22     b = a
23     print(b)
24 }
25 print(b)
26
27
28
29
30 }$
31
32 Begin_Block [ { ] Found at line 52 : 1
33 TYPE [ int ] Found at line 53 : 11
34 ID [ a ] Found at line 53 : 13
35 TYPE [ boolean ] Found at line 54 : 15
36 ID [ b ] Found at line 54 : 17
37 TYPE [ string ] Found at line 55 : 14
38 ID [ c ] Found at line 55 : 16
39 ID [ a ] Found at line 56 : 8
40 AssignmentStatement [ = ] Found at line 56 : 10
41 DIGIT [ 9 ] Found at line 56 : 12
42 ID [ b ] Found at line 57 : 9
43 AssignmentStatement [ = ] Found at line 57 : 11
44 Boolean_Value [ true ] Found at line 57 : 16
45 Begin_Block [ { ] Found at line 58 : 9
46 Print_Statment [ print ] Found at line 59 : 16
47 Open_Expression [ ( ] Found at line 59 : 17
48 ID [ a ] Found at line 59 : 18
49 Close_Expression [ ) ] Found at line 59 : 19
50 Print_Statment [ print ] Found at line 60 : 16
51 Open_Expression [ ( ] Found at line 60 : 17
52 ID [ b ] Found at line 60 : 18
53 Close_Expression [ ) ] Found at line 60 : 19
54 ID [ b ] Found at line 61 : 12
55 AssignmentStatement [ = ] Found at line 61 : 14
56 Boolean_Value [ false ] Found at line 61 : 20
57 ID [ c ] Found at line 62 : 12
58 AssignmentStatement [ = ] Found at line 62 : 14
59 QUOTE [ " ] Found at line 62 : 16
60 CHAR [ h ] Found at line 62 : 17
61 CHAR [ e ] Found at line 62 : 18
62 CHAR [ l ] Found at line 62 : 19
63 CHAR [ l ] Found at line 62 : 20
64 CHAR [ o ] Found at line 62 : 21
65 SPACE [   ] Found at line 62 : 22
66 CHAR [ w ] Found at line 62 : 23
67 CHAR [ o ] Found at line 62 : 24
68 CHAR [ r ] Found at line 62 : 25
69 CHAR [ l ] Found at line 62 : 26
70 CHAR [ d ] Found at line 62 : 27
71 QUOTE [ " ] Found at line 62 : 28
72 TYPE [ int ] Found at line 63 : 14
73 ID [ b ] Found at line 63 : 16

```

```

74 ID [ b ] Found at line 64 : 12
75 AssignmentStatement [ = ] Found at line 64 : 14
76 DIGIT [ 0 ] Found at line 64 : 16
77 Begin_Block [ { ] Found at line 65 : 12
78 Print_Statment [ print ] Found at line 66 : 19
79 Open_Expression [ ( ] Found at line 66 : 20
80 ID [ c ] Found at line 66 : 21
81 Close_Expression [ ) ] Found at line 66 : 22
82 ID [ a ] Found at line 67 : 15
83 AssignmentStatement [ = ] Found at line 67 : 17
84 DIGIT [ 1 ] Found at line 67 : 19
85 InTop [ + ] Found at line 67 : 21
86 DIGIT [ 2 ] Found at line 67 : 23
87 InTop [ + ] Found at line 67 : 25
88 ID [ a ] Found at line 67 : 27
89 Begin_Block [ { ] Found at line 68 : 15
90 Print_Statment [ print ] Found at line 69 : 22
91 Open_Expression [ ( ] Found at line 69 : 23
92 ID [ b ] Found at line 69 : 24
93 Close_Expression [ ) ] Found at line 69 : 25
94 End_Block [ } ] Found at line 70 : 15
95 End_Block [ } ] Found at line 71 : 12
96 ID [ b ] Found at line 72 : 12
97 AssignmentStatement [ = ] Found at line 72 : 14
98 ID [ a ] Found at line 72 : 16
99 Print_Statment [ print ] Found at line 73 : 16
100 Open_Expression [ ( ] Found at line 73 : 17
101 ID [ b ] Found at line 73 : 18
102 Close_Expression [ ) ] Found at line 73 : 19
103 End_Block [ } ] Found at line 74 : 9
104 Print_Statment [ print ] Found at line 75 : 13
105 Open_Expression [ ( ] Found at line 75 : 14
106 ID [ b ] Found at line 75 : 15
107 Close_Expression [ ) ] Found at line 75 : 16
108 End_Block [ } ] Found at line 80 : 1
109 END_OF_PROGRAM [ $ ] Found at line 80 : 2
110 Number of Errors is 0 :(
111 End of program 6

```

---

## 1.7 Test Case 7

```

1  /*Testing for no new lines in strings*/
2  "a
3  b"$
4
5  QUOTE [ " ] Found at line 83 : 1
6  CHAR [ a ] Found at line 83 : 2
7  Error: new line not allowed in string [ ] Found at line 83 : 2
8  CHAR [ b ] Found at line 84 : 1
9  QUOTE [ " ] Found at line 84 : 2
10 END_OF_PROGRAM [ $ ] Found at line 84 : 3
11 Number of Errors is 1 :(
12 Lexer failed :(

```

## 2 References

[Java Regular Expression](#)  
[Enum](#)  
[Switch cases \(Was more of an idea\)](#)  
[Double Quote For Regular Expressions](#)  
[Groups in Regular Expressions](#)  
[More Group stuff](#)  
[Index for Regular Expressions](#)  
[Check for end of line](#)  
[Full Syntax Java Regx](#)  
[Quote In Regx](#)

I used these all to help me understand how to use Regular Expression and get the data that I need in order to make the tokens. Enum and Cases where not used but more of an ideas that I was thing about using but never did end up using them