توضیحات برنامه:

ابتدا در خط اول و دوم برنامه کتابخانه regex برای جستجوی الگو و کتابخانه مربوط به کار کردن با فایل های اکسل را به برنامه اضافه می‌کنیم

```
import re
import xlsxwriter
```

در خط چهارم یک دیکشنری تعریف می‌کنیم . در این دیکشنری هر یک از الگوهای مورد نظر را با اسمی که باید در فایل اکسل اضافه شود را درج کرده ایم. این الگوها شامل عملگرها ، لیترال های عددی و رشته‌ای، و شناسه ها ، و همچنین علاعم مربوط به پرانتز ، آکولاد و ... هستند. در واقع هر یک از سطرها یک الگوی regex است که با تابع `finditer` در برنامه پایتون ورودی جستجو خواهد شد

برای مثال (`'STRING_LITRAL', r'".*"'`) در خط 5 تمام ثابت های رشته ای که بین دو دابل کوتیشن قرار دارند را برمی‌گرداند و در خط 42 عبارت (`'ID', r'[a-zA-Z_][0-9a-zA-Z_]*'`) تمام شناسه ها(که با یک حرف یا علامت زیرخط شروع و در ادامه هر تعداد کارکتر یا رقم یا علامت زیرخط می‌تواند داشته باشد) را توصیف می‌کند. طبعیتا می‌توان هر الگوی دیگری که در زبان پایتون معتبر است را به این دیکشنری اضافه کنیم.

```
opDic= {
    ('STRING_LITRAL', r'".*"'),
    ('COMMENT_STARTER','#'),

    ('LBRACKET', '\('),
    ('RBRACKET', '\)'),
    ('LBRACE', '\{'),
    ('RBRACE', '\}'),
    ('COMMA', ','),
    ('SEMICOLON', ';'),
    ('COLON',':'),
    ('EQ', '=='),
    ('NE', '!='),
    ('LE', '<='),
    ('GE', '>='),
    ('LT', '<'),
    ('GT', '>'),

    ('PLUS', '\+[\=]?'),
    ('MINUS', '-[\=]?'),
    ('MULT', '\*[\=]?'),
```

```
    ('DIV', '\/[\=]?'),
    ('MOD', '%[\=]?'),
    ('FLOORDIV', '\/\/[\=]?'),
    ('POw','\*\*[\=]?'),
    ('LEFTSHIFT',"<<[\=]?"),
    ('RIGHTSHIFT',">>[\=]?"),
    ('BitwiseXOR',"\^[\=]?"),
    ('BitwiseOR',"\|[\=]?"),
    ('BitwiseAND',"&[\=]?"),

    ('FLOAT_CONST', r'\d(\d)*\.\d(\d)*'),
    ('INTEGER_CONST', r'\d(\d)*'),

    ('NEWLINE', r'\n'),
    ('SKIP', r'[ \t]+'),

    ('ASSIGNMENT','\='),
    ('ID', r'[a-zA-Z_][0-9a-zA-Z_]*'),

    }
```

در خط 46 برنامه یک لیست تعریف کرده ایم که شامل تمام کلمات کلیدی در پایتون است تا بعدا بین شناسه ها و این کلمات تمایز قاعل شویم

```
keywordList = ['and', 'as', 'assert', 'break', 'class', 'def', 'del','elif',
'else', 'except', 'exec', 'finally', 'for', 'from', 'global', 'if','import','in',
'is', 'lambda', 'not', 'or', 'pass', 'print', 'raise', 'return', 'try', 'while',
'with', 'yield']
```

و تابع `isKeyword` به سادگی چک می‌کند که رشته ورودی جزع این کلمات کلیدی هست یا خیر.

```
def isKeyword(str):
    return str in keywordList
```

درادامه تابع اصلی برنامه تعریف و پیاده سازی کرده ایم که در خطهای 57 تا 68 آدرس فایل پایتون ورودی و اکسل خروجی را از کاربر دریافت می‌کند. سپس فایل ورودی را باز کرده در صورتی که محتوایی نداشته باشد پیغامی چاپ کرده و از برنامه خارج می‌شویم و در غیراینصورت خط به خط برنامه را خوانده و به رشته buf اضافه می‌کنیم

```
print("(STARTMARKER)")
path = input("Enter path of input scipt file:")
outpath = input("Enter path of output excel file:")
f = open(path, 'r')
lines = f.readlines()
```

```
if len(lines) <= 0:
   print("Input file %s is empty" % f)
   quit(0)
buf=''
for line in lines:
   buf+=line
```

در ادامه در خط 70 تمام الگوهایی که در دیکشنری آمده است را با هم در یک رشته بصورت OR ترکیب می‌کنیم تا تمام این الگوها در فایل جستجو شوند

```
tokens_join = '|'.join('(?P<%s>%s)' % x for x in opDic)
```

سپس یک فایل اکسل با ادرس دریافتی ایجاد می‌کنیم و در سطر اول آن عنوان ستون ها را وارد می‌کنیم و به دو سطر بعد می‌رویم

```
workbook = xlsxwriter.Workbook(outpath)
worksheet = workbook.add_worksheet()
worksheet.write(0, 0, "sentence")
worksheet.write(0, 1, "type")
row=2
commentFLAG = False
```

در خط 78 در حلقه for با تابع finditer تمام الگوهای regex را در فایل پیدا می‌کنیم و در دو خط بعد خود الگو و نام آن را در دو متغیر می‌ریزیم

```
for item in re.finditer(tokens_join, buf):
    tokenType = item.lastgroup
    token = item.group(tokenType)
```

حال با توجه به نوع الگو یکی از حالت های زیر دنبال می‌شود:

اگر الگو بیانگر newline(پایان خط) باشد و پرچم کامنت فعال باشد یعنی تا به حال عبارات یک خط که کامنت شده است را پردازش کرده ایم که اکنون خط به پایان رسده و بنابراین پرچم کامنت را false می‌کنیم.

```
 if ((tokenType == 'NEWLINE') and (commentFLAG==True)):
     commentFLAG = False
     continue;
```

اگر عبارت شامل space tab یا newline(فاصله، جهش، و پایان خط) باشد آنرا نادیده می‌گیریم.

```
elif ((tokenType == 'SKIP') or(tokenType == 'NEWLINE')):
     continue;
```

در صورتی که الگو برابر # باشد بیانگر شروع یک خط یا جزعی از یک خط که کامنت شده است و بنابراین پرچم کامنت را true می‌کنیم.

```
elif tokenType== "COMMENT_STARTER":
     commentFLAG = True
```

اگر پرچم کامنت true باشد بعنب قبلا # مشاده شده و بنابراین بدون توجه به نوع توکن آن را از نوع comment در نظر می‌گیریم.

```
elif(commentFLAG==True):
    tokenType="COMMENT"
```

اگر الگو بیانگر یک شناسه باشد چک می‌کنیم که جزع کلمات کلیدی هست یا نه اگر باشد نوع آن را از نوع keyword در نظر می‌گیریم

```
elif (tokenType=="ID" and isKeyword(token)):
     tokenType="KEYWORD"
```

در ادامه توکن و نوع آن را در فایل اکسل خروجی درج می‌کنیم.

```
worksheet.write(row, 0, repr(token))
worksheet.write(row, 1, tokenType)
row += 1
```

در پایان تعداد توکن های استخراج شده در چاپ و برنامه به پایان می‌رسد.

```
print(str(row-2) + " token extracted...")
print("(ENDMARKER)")
workbook.close()
```