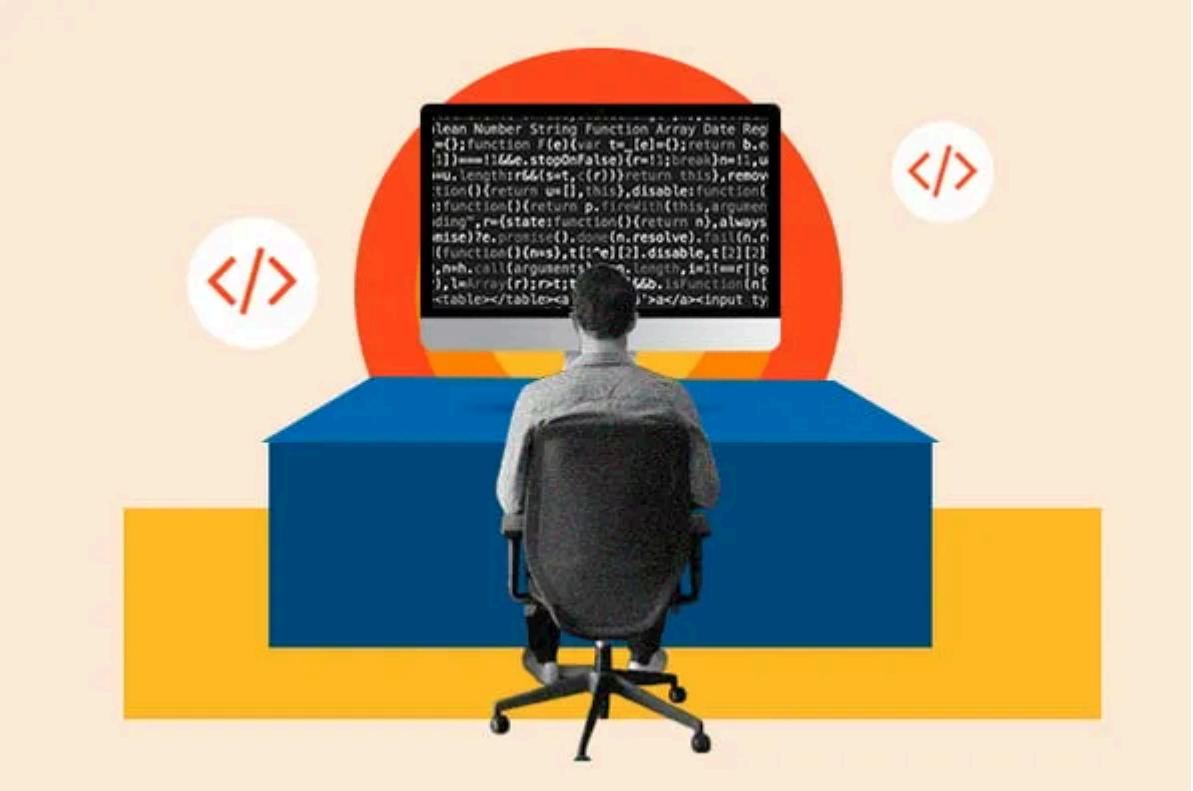
UNIVERSIDAD AUTÓNOMA DE CHIAPAS LICENCIATURA EN INGENIERÍA EN DESARROLLO Y TECNOLOGÍAS DE SOFTWARE

Act. 1.1 Investigar analizador Léxico y Lenguajes Regulares

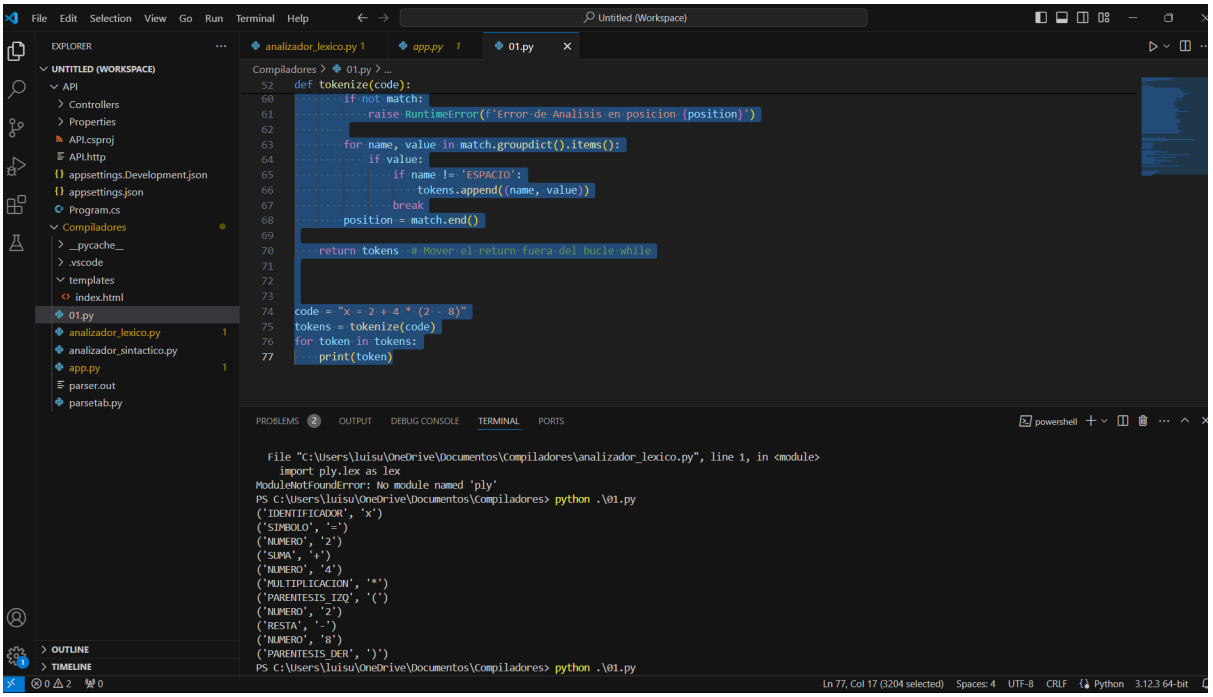


Unidad académica: Compiladores.
Docente: Dr. Gutierrez Alfaro Luis
Nombre del alumno: García Gonzaléz Luis Angel
Grado y grupo: 6M

Fecha: 18 Agosto del 2024.

```
#mi primer analizador lexico que solo me reconocer
# numeros enteros, identificadores, operadores
# librerias: lexico, re

import re

#crear tokens
token_patterns = [
    ('NUMERO', r'\d+'),                     # Número entero
    ('IDENTIFICADOR', r'[A-Za-z]\w*'),      # Identificador
    ('SUMA', r'\+'),                        # Operador de suma
    ('RESTA', r'-'),                        # Operador de resta
    ('MULTIPLICACION', r'\*'),              # Operador de
multiplicación
    ('DIVISION', r'/'),                     # Operador de división
    ('PARENTESIS_IZQ', r'\('),              # Paréntesis izquierdo
    ('PARENTESIS_DER', r'\)'),              # Paréntesis derecho
    ('ESPACIO', r'\s+'),                    # Espacios
    ('SIMBOLO', r'.'),                      # Otros caracteres
    ('FUNCION', r'function'),              #FUNCION
    ('RETORNO', r'return'),                  #Retono
    ('SI', r'if'),                           # SI
    ('SINO', r'else'),                       #SINO
    ('MIENTRAS', r'while'),                  #MIENTRAS
    ('PARA', r'for'),                        #PARA
    ('FIN_DE_LINEA', r';'),                  #FIN DE LINEA
    ('COMENTARIO', r'//.*'),                 #COMENTARIO
    ('COMENTARIO_MULTILINEA', r'/\*[\s\S]*?\*/'), #COMENTARIO
MULTILINEA
    ('CADENA', r'"[^"]*"'),                   #CADENA
    ('CARACTER', r"'[^']*'"),                 #CARACTER
    ('BOOLEANO', r'(true|false)'),            #BOOLEANO
    ('NULO', r'null'),                        #NULO
    ('ARROBA', r'@') ,                        #ARROBA
    ('DOS_PUNTOS', r':') ,                    #DOS PUNTOS
    ('PUNTO_Y_COMA', r';') ,                  #PUNTO Y COMA
    ('CORCHETE_IZQ', r'\[') ,                 #CORCHETE IZQ
    ('CORCHETE_DER', r'\]')  ,                #CORCHETE DER
    ('%', r'%')    ,                          #MODULO
    ('IGUALDAD', r'=='),                      #IGUALDAD
    ('DIFERENTE', r'!=') ,                    #DIFERENTE
    ('MAYOR', r'>') ,                         #MAYOR QUE
```

```
    ('MENOR', r'<') ,                           #MENOR QUE
    ('MAYOR_IGUAL', r'>=') ,                    #MAYOR O IGUAL
    ('MENOR_IGUAL', r'<=') ,                    #MENOR O IGUAL
    ('ASIGNACION', r'=')  ,                     #ASIGNACION

]

# token expresiones regulares patrones
token_regex = '|'.join(f'(?P<{name}>{pattern})' for name, pattern in
token_patterns)
get_token = re.compile(token_regex).match

def tokenize(code):
    line_number = 1
    line_start = 0
    position = 0
    tokens = []

    while position < len(code):
        match = get_token(code, position)
        if not match:
            raise RuntimeError(f'Error de Analisis en posicion
{position}')

        for name, value in match.groupdict().items():
            if value:
                if name != 'ESPACIO':
                    tokens.append((name, value))
                break
        position = match.end()

    return tokens  # Mover el return fuera del bucle while



code = "x = 2 + 4 * (2 - 8)"
tokens = tokenize(code)
for token in tokens:
    print(token)
```


File Edit Selection View Go Run Terminal Help
Untitled (Workspace)
EXPLORER
UNTITLED (WORKSPACE)
API
Controllers
Properties
API.csproj
API.http
appsettings.Development.json
appsettings.json
Program.cs
Compiladores
__pycache__
.vscode
templates
index.html
01.py
analizador_lexico.py 1
analizador_sintactico.py
app.py 1
parser.out
parsetab.py
OUTLINE
TIMELINE
analizador_lexico.py 1
app.py 1
01.py
Compiladores > 01.py > ...
52 def tokenize(code):
60 if not match:
61 raise RuntimeError(f'Error de Analisis en posicion {position}')
62
63 for name, value in match.groupdict().items():
64 if value:
65 if name != 'ESPACIO':
66 tokens.append((name, value))
67 break
68 position = match.end()
69
70 return tokens # Mover el return fuera del bucle while
71
72
73
74 code = "x = 2 + 4 * (2 - 8)"
75 tokens = tokenize(code)
76 for token in tokens:
77 print(token)
PROBLEMS 2
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
powershell
File "C:\Users\luisu\OneDrive\Documentos\Compiladores\analizador_lexico.py", line 1, in <module>
import ply.lex as lex
ModuleNotFoundError: No module named 'ply'
PS C:\Users\luisu\OneDrive\Documentos\Compiladores> python .\01.py
('IDENTIFICADOR', 'x')
('SIMBOLO', '=')
('NUMERO', '2')
('SUMA', '+')
('NUMERO', '4')
('MULTIPLICACION', '*')
('PARENTESIS_IZQ', '(')
('NUMERO', '2')
('RESTA', '-')
('NUMERO', '8')
('PARENTESIS_DER', ')')
PS C:\Users\luisu\OneDrive\Documentos\Compiladores> python .\01.py
0 2 0
Ln 77, Col 17 (3204 selected)
Spaces: 4
UTF-8
CRLF
Python
3.12.3 64-bit