



Nombre: Eduardo Quetzal Delgado Pimentel

Fecha: 21/01/2024

Código: 217239716

Materia: Traductores de lenguaje II

Mini Analizador léxico

Objetivo:

Genera un pequeño analizador léxico, que identifique los siguientes tokens (identificadores y números reales) construidos de la siguiente manera.

identificadores = letra(letra|digito)*

Real = entero.entero+

Desarrollo:

```
# Importamos la biblioteca re
import re

# Definimos las expresiones regulares para los tokens
identificador = re.compile(r'\A[a-zA-Z][a-zA-Z0-9]*')
igual = re.compile(r'\A=')
mas = re.compile(r'\A\+')
real = re.compile(r'\A\d+\.\d+')

# Definimos una función que recibe una cadena y devuelve una lista de tokens
def analizar_lexico(cadena):
    # Inicializamos una lista vacía para almacenar los tokens
    tokens = []
    # Mientras la cadena no esté vacía
    while cadena:
        # Ignoramos los espacios en blanco al principio de la cadena
        cadena = cadena.lstrip()
        # Intentamos hacer un match con el token de igual
        m = igual.match(cadena)
        if m:
            # Si hay un match, añadimos el token a la lista con su tipo
            tokens.append(('IGUAL', m.group()))
            # Actualizamos la cadena quitando el token
            cadena = cadena[m.end():]
            # Continuamos con el siguiente token
            continue
        # Intentamos hacer un match con el token de más
        m = mas.match(cadena)
        if m:
            # Si hay un match, añadimos el token a la lista con su tipo
            tokens.append(('MAS', m.group()))
            # Actualizamos la cadena quitando el token
```

```

        cadena = cadena[m.end():]
        # Continuamos con el siguiente token
        continue
    # Intentamos hacer un match con el token de identificador
    m = identificador.match(cadena)
    if m:
        # Si hay un match, añadimos el token a la lista con su tipo
        tokens.append(('IDENTIFICADOR', m.group()))
        # Actualizamos la cadena quitando el token
        cadena = cadena[m.end():]
        # Continuamos con el siguiente token
        continue
    # Intentamos hacer un match con el token de real
    m = real.match(cadena)
    if m:
        # Si hay un match, añadimos el token a la lista con su tipo
        tokens.append(('REAL', m.group()))
        # Actualizamos la cadena quitando el token
        cadena = cadena[m.end():]
        # Continuamos con el siguiente token
        continue
    # Si no hay match con ningún token, lanzamos una excepción
    raise ValueError(f'Carácter no reconocido: {cadena[0]}')
# Devolvemos la lista de tokens
return tokens

# Probamos la función con un ejemplo
ejemplo = 'x = 3.14 +'
print(analizar_lexico(ejemplo))

```

Resultados

```

PS C:\Users\siete> & C:/Users/siete/AppData/Local/Programs/Python/Python310/python.exe "c:/Users/siete/OneDrive/Desktop/lexico.py"
[('IDENTIFICADOR', 'x'), ('IGUAL', '='), ('REAL', '3.14'), ('MAS', '+')]
PS C:\Users\siete>

```

```

PS C:\Users\siete> & C:/Users/siete/AppData/Local/Programs/Python/Python310/python.exe "c:/Users/siete/OneDrive/Desktop/lexico.py"
[('IDENTIFICADOR', 'x'), ('IGUAL', '='), ('REAL', '3.14'), ('MAS', '+'), ('IDENTIFICADOR', 'ya')]
PS C:\Users\siete>

```

