



---

# EJERCICIO 1

---

Mercado Manzo Luis Alfonso



15 DE OCTUBRE DE 2023

```
# -*- coding: utf-8 -*-
```

```
"""
```

Created on Mon Sep 18 13:49:08 2023

```
@author: luis mercado
```

```
"""
```

```
import re
```

```
# Expresiones regulares para identificar tokens
```

```
token_patterns = [
```

```
    (r'int|float|char|void|string', 0),    # Tipo de dato
```

```
    (r'[a-zA-Z_][a-zA-Z0-9_]*', 1),        # Identificador(un numero o una letra)
```

```
    (r'(pi|\d+(\.\d+)?)', 2),             # Constante (número o pi)
```

```
    (r';', 3),                             # ; (punto y coma)
```

```
    (r',', 4),                             # , (coma)
```

```
    (r'\(', 5),                             # ( (parentesis izquierdo)
```

```
    (r'\)', 6),                             # ) (parentesis derecho)
```

```
    (r'\{', 7),                             # { (corchete izquierdo)
```

```
    (r'\}', 8),                             # } (corchete derecho)
```

```
    (r'=', 9),                             # =
```

```
    (r'if', 10),                           # if
```

```
    (r'while', 11),                        # while
```

```
    (r'return', 12),                      # return
```

```
    (r'else', 13),                        # else
```

```
    (r'for', 13),                         # for
```

```
    (r'[+|-]', 14),                       # Operador de adición
```

```
    (r'[*\V]|<<|>>', 15),               # Operador de multiplicación
```

```
    (r'&&|\|\|', 16),                   # Operador lógico
```

```

(r'<|>|<=|>=|==|!=', 17),          # Operador relacional

]

# Función para analizar una cadena y para encontrar los tokens y errores
def analizar_cadena(cadena):
    tokens = []
    errores = []

    while cadena:
        encontrado = False
        for patron, categoria in token_patterns:
            match = re.match(patron, cadena)
            if match:
                valor = match.group(0)
                tokens.append((valor, categoria))
                cadena = cadena[len(valor):]
                encontrado = True
                break

        if not encontrado:
            errores.append(cadena[0])
            cadena = cadena[1:]

    return tokens, errores

# el espacio es marcado como un error
if __name__ == "__main__":
    cadena = input("Ingrese la cadena a analizar: ")

```

```
tokens, errores = analizar_cadena(cadena)
```

```
# Categorías de tokens
```

```
categorias = {
```

```
    0: "Tipo de dato",
```

```
    1: "Identificador",
```

```
    2: "Constante",
```

```
    3: ";",
```

```
    4: ",",
```

```
    5: "(",
```

```
    6: ")",
```

```
    7: "{",
```

```
    8: "}",
```

```
    9: "=",
```

```
   10: "if",
```

```
   11: "while",
```

```
   12: "return",
```

```
   13: "else/for",
```

```
   14: "Operador de adición",
```

```
   15: "Operador de multiplicación",
```

```
   16: "Operador lógico",
```

```
   17: "Operador relacional",
```

```
}
```

```
# mostrar el resultado de los tokens
```

```
print("\nTokens encontrados:")
```

```
for token, categoria in tokens:
```

```
    print(f"{token}: {categorias[categoria]}")
```

```
if errores:
```

```

print("\nErrores encontrados:")

for error in errores:

    print(f"Error: Caracter '{error}' no reconocido.")


# Contar tokens de cada categoría
conteo_categorias = {categoria: 0 for categoria in categorias.values()}

for _, categoria in tokens:

    conteo_categorias[categorias[categoria]] += 1


print("\nCantidad de tokens por categoría:")

for categoria, cantidad in conteo_categorias.items():

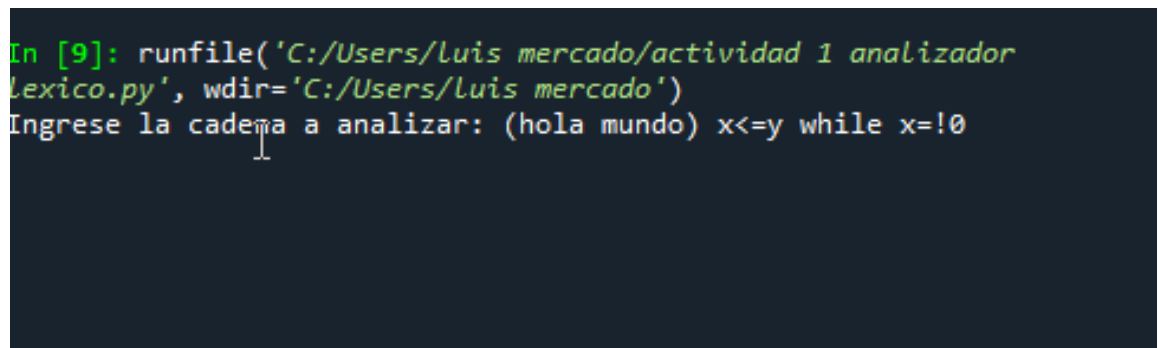
    print(f"{categoria}: {cantidad}")

```

descripción del código

similar a la práctica 1, este programa lee una frase y busca los identificadores y caracteres en la frase.

En lugar de leer un archivo, el programa permite escribir una frase.



```

In [9]: runfile('C:/Users/luis mercado/actividad 1 analizador
lexico.py', wdir='C:/Users/luis mercado')
Ingrese la cadena a analizar: (hola mundo) x<=y while x!=0

```

Resultados:

Tokens encontrados:

(: (

hola: Identificador

mundo: Identificador

): )

x: Identificador

<: Operador relacional

=: =

y: Identificador

while: Identificador

x: Identificador

=: =

0: Constante

Errores encontrados:

Error: Caracter ' ' no reconocido.

Error: Caracter ' ' no reconocido.

Error: Caracter ' ' no reconocido.

Error: Caracter ' ' no reconocido.

Error: Caracter '!' no reconocido.

Cantidad de tokens por categoría:

Tipo de dato: 0

Identificador: 6

Constante: 1

:: 0

,: 0

(: 1

): 1

{: 0

}: 0

=: 2

if: 0

while: 0

return: 0

else/for: 0

Operador de adición: 0

Operador de multiplicación: 0

Operador lógico: 0

Operador relacional: 1