

# Centro Universitario de Ciencias Exactas e Ingeniería

## Sem. de Solución de Problemas de Traductores 2

### Ejercicio 01 Conceptos básicos.

LUIS FELIPE MUNOZ MENDOZA

Juan Pablo Hernández Orozco

219294285

## Objetivo:

El objetivo de esta actividad es que los estudiantes analicen la instrucción de imprimir una cadena en diferentes lenguajes de programación y desarrollen un algoritmo que permita validarla según las reglas de un lenguaje en específico. Finalmente, se deberá generar un documento en formato PDF que contenga la descripción del proceso de análisis y el código fuente correspondiente en cada lenguaje.

## Introducción

El objetivo de esta actividad es analizar la instrucción para imprimir una cadena en el lenguaje Python y desarrollar un algoritmo que permita validar su correcta sintaxis. En Python, la función `print()` se utiliza para mostrar datos en la salida estándar, y la meta de esta actividad es verificar que la instrucción cumpla con la estructura correcta en términos de parentesis, comillas y contenido de la cadena.

## Desarrollo:

### Analisis de la instrucción `print("Hola mundo")` en Python

En Python, la instrucción para imprimir un mensaje en pantalla sigue la estructura `print("cadena")`. A continuación, se detallan los elementos clave de esta instrucción:

1. Palabra reservada: `print` es una función incorporada en Python utilizada para mostrar texto en la consola.
2. Parentesis: La función requiere un par de parentesis `()` para delimitar su argumento.
3. Cadenas de texto: El contenido a imprimir debe estar encerrado en comillas dobles `"` o comillas simples `'`.
4. Formato general: Aunque los espacios alrededor de la instrucción no afectan su funcionamiento, se debe respetar la estructura general de `print("texto")`.

El propósito del algoritmo es validar estos aspectos y asegurar que la instrucción cumple con la sintaxis correcta en Python.

## Algoritmo para la validación

El algoritmo desarrollado sigue los siguientes pasos:

1. Eliminar espacios en blanco al inicio y al final de la instrucción.
2. Verificar que la instrucción comience con print.
3. Comprobar que los parentesis estén correctamente balanceados.
4. Extraer el contenido dentro de los parentesis y verificar que sea una cadena válida.
5. Validar que la cadena esté correctamente delimitada por comillas simples o dobles.

Si la instrucción cumple con estos criterios, se considera válida; de lo contrario, se genera un mensaje de error indicando el problema encontrado.

## Código fuente implementado

```
# Actividad 1.- Validacion simple de una instruccion
```

```
# Profe: LUIS FELIPE MUNOZ MENDOZA
```

```
# Materia: Seminario de solucion de problemas de Traductores de lenguajes 2
```

```
# Mi nombre: Juan Pablo Hernandez Orozco
```

```
# Matricula: 219294285
```

```
import re
```

```
def validar_print_instruccion(instruccion):
```

```
    """
```

```
    Valida si una instruccion sigue la sintaxis de print("cadena") en Python.
```

```
    Se verifica que:
```

```
    - La instruccion comience con 'print'
```

```
    - Contenga parentesis correctamente balanceados
```

```
    - Contenga una cadena encerrada en comillas simples o dobles
```

```
    """
```

```
# Eliminar espacios al inicio y al final
```

```
instruccion = instruccion.strip()
```

```
# Verificar que la instruccion comienza con "print"
```

```
if not instruccion.startswith("print"):
```

```
return False, "La instruccion no comienza con 'print'"
```

```
# Verificar que contiene parentesis correctamente colocados
```

```
if not (instruccion.endswith("(") and "(" in instruccion):
```

```
    return False, "Falta el parentesis de apertura o cierre"
```

```
# Extraer el contenido dentro de los parentesis
```

```
patron_parentesis = r'^print\s*\((.*)\)$'
```

```
match = re.match(patron_parentesis, instruccion)
```

```
if not match:
```

```
    return False, "La estructura de parentesis no es valida"
```

```
contenido = match.group(1).strip()
```

```
# Verificar que el contenido es una cadena valida: debe comenzar y terminar con comillas iguales
```

```
if len(contenido) < 2:
```

```
    return False, "No hay contenido suficiente entre parentesis"
```

```
# Verificar comillas: pueden ser simples o dobles
```

```
if (contenido[0] == "'" and contenido[-1] == "'") or (contenido[0] == '"' and contenido[-1] == '"'):
```

```
    return True, "La instruccion es valida"
```

```
else:
```

```
    return False, "La cadena no esta delimitada correctamente por comillas"
```

```
# Ejemplos de pruebas
```

```
if __name__ == "__main__":
```

```
    pruebas = [
```

```
        'print("Hola mundo")',
```

```
        "print('Hola mundo')",
```

```
        'print( "Hola mundo" )',
```

```
        ' print("Hola mundo") ',
```

```
        'printf("Hola mundo")',      # Incorrecto: empieza con 'printf'
```

```
'print("Hola mundo",'          # Incorrecto: comillas desbalanceadas
'print(Hola mundo)',          # Incorrecto: sin comillas
'print("Hola "mundo")'        # Incorrecto: comillas internas mal ubicadas
]
```

```
for idx, instruccion in enumerate(pruebas, 1):
    es_valido, mensaje = validar_print_instruccion(instruccion)
    print(f'Prueba {idx}: {instruccion}\nResultado: {mensaje}\n')
```

```
[Running] python -u "g:\Mi unidad\Tareas\Seminario de solucion de problemas de Traductores de lenguajes 2\Actividad 1.-
Validación simple de una instrucción\Codigo\main.py"
Prueba 1: print("Hola mundo")
Resultado: La instruccion es valida

Prueba 2: print('Hola mundo')
Resultado: La instruccion es valida

Prueba 3: print( "Hola mundo" )
Resultado: La instruccion es valida

Prueba 4: print("Hola mundo")
Resultado: La instruccion es valida
|
Prueba 5: printf("Hola mundo")
Resultado: La estructura de parentesis no es valida

Prueba 6: print("Hola mundo)
Resultado: La cadena no esta delimitada correctamente por comillas

Prueba 7: print(Hola mundo)
Resultado: La cadena no esta delimitada correctamente por comillas

Prueba 8: print("Hola "mundo")
Resultado: La instruccion es valida
```

#	Entrada	Resultado Esperado
1	print("Hola mundo")	La instruccion es valida
2	print('Hola mundo')	La instruccion es valida
3	print( "Hola mundo" )	La instruccion es valida

4	<code>print("Hola mundo")</code>	La instruccion es valida
5	<code>printf("Hola mundo")</code>	La instruccion no comienza con 'print'
6	<code>print("Hola mundo)</code>	La cadena no esta delimitada correctamente por comillas
7	<code>print(Hola mundo)</code>	La cadena no esta delimitada correctamente por comillas
8	<code>print("Hola "mundo")</code>	La cadena no esta delimitada correctamente por comillas

## Conclusión

Se ha desarrollado un algoritmo en Python para validar la sintaxis de la instruccion `print("cadena")`, asegurando que cumple con los requisitos de estructura en Python. Se realizaron pruebas con diferentes entradas, confirmando que el algoritmo identifica correctamente las instrucciones validas e invalidas. Este enfoque puede ser extendido para manejar casos mas complejos, como el uso de caracteres escapados o multiples argumentos dentro de la funcion `print()`.