

DESARROLLO DE SOFTWARE DE BACKEND I

ROMERO CARRASCO SAMANTHA BELEN

Aarón Hernández García

Actividad. Operaciones y Strings

Operaciones Básicas y Avanzadas con Tipos Numéricos

Operaciones Básicas:

- **Aritmética:** Operaciones fundamentales como suma (+), resta (-), multiplicación (*), división (/), módulo (%), y exponente (**).

```
a = 5
b = 2
print(a + b) # 7
print(a - b) # 3
print(a * b) # 10
print(a / b) # 2.5
print(a % b) # 1
print(a ** b) # 25
```

- **División Entera:** La división que descarta la parte decimal y devuelve solo la parte entera.

```
print(a // b) # 2
```

Operaciones Básicas y Avanzadas con Tipos Numéricos

Operaciones Avanzadas:

- **Operaciones Bit a Bit:** Manipulación de bits dentro de los números

```
print(a & b) # AND bit a bit
print(a | b) # OR bit a bit
print(a ^ b) # XOR bit a bit
print(a << 1) # Desplazamiento a la izquierda
print(a >> 1) # Desplazamiento a la derecha
```

- **Funciones Matemáticas:** Uso de la librería math para operaciones complejas como raíces, logaritmos, trigonometría, etc.

```
import math
print(math.sqrt(16)) # 4.0
print(math.log(100, 10)) # 2.0
print(math.sin(math.pi/2)) # 1.0
```

Strings y Textos

- **Definición:** Una cadena (string) es una secuencia de caracteres. En Python, se definen usando comillas simples (') o dobles (").

```
s = 'Hello, World!'
```

- **Concatenación:** Unión de dos o más cadenas usando el operador +.

```
s1 = 'Hello'  
s2 = 'World'  
print(s1 + ', ' + s2 + '!') # Hello, World!
```

- **Repetición:** Repetir una cadena un número específico de veces usando el operador *.

```
print(s1 * 3) # HelloHelloHello
```

- **Indexación y Slicing:** Acceder a caracteres específicos o subsecciones de la cadena usando índices y rangos.

```
print(s[0])    # 'H'  
print(s[-1])   # '!'  
print(s[7:12]) # 'World'
```

Strings y Textos

- **Métodos de Cadenas:** Funciones incorporadas para manipular cadenas, como `upper()`, `lower()`, `strip()`, `split()`, `replace()`, etc.

```
print(s.upper())    # 'HELLO, WORLD!'
print(s.lower())    # 'hello, world!'
print(s.strip('!')) # 'Hello, World'
print(s.split(', '))# ['Hello', 'World!']
print(s.replace('World', 'Python')) # 'Hello, Python!'
```

Strings Literals y Expresiones

- **String Literals:** Representaciones textuales de datos de cadena en el código fuente. Pueden definirse usando comillas simples, dobles, triples (para multi-línea), y prefijos especiales (r, f, b).

```
single_quote = 'Hello'
double_quote = "Hello"
triple_quote = '''Hello
World'''
raw_string = r'C:\Users\Name'
formatted_string = f'Hello, {s2}!' # 'Hello, World!'
byte_string = b'Hello'
|
```

- **Expresiones de Formato:** Uso de la función format() o f-strings para insertar valores dentro de cadenas.

```
name = 'Aaron'
age = 28
print('My name is {} and I am {} years old'.format(name, age)) # 'My name is Aaron and I am 28 years old'
print(f'My name is {name} and I am {age} years old')           # 'My name is Aaron and I am 28 years old'
```

Referencias

Capítulo 7. Cadenas y archivos de texto. (s/f). Upv.Es. Recuperado el 31 de mayo de 2024, de <https://hedero.webs.upv.es/julia-basico/7-strings/>

Matemáticas básicas en JavaScript — números y operadores. (s/f). MDN Web Docs. Recuperado el 31 de mayo de 2024, de https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/Math

Matemáticas básicas en JavaScript — números y operadores. (s/f). MDN Web Docs. Recuperado el 31 de mayo de 2024, de https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/Math