

Operadores Aritméticos

1. ¿Cuáles son los operadores aritméticos?

A continuación se presentan los operadores aritméticos soportados por Python 3

Operator	Example	Meaning
<code>+</code> (unario)	<code>+a</code>	Unario Positivo
<code>+</code> (binario)	<code>a + b</code>	Suma
<code>-</code> (unario)	<code>-a</code>	Unario Negativo
<code>-</code> (binario)	<code>a - b</code>	Resta
<code>*</code>	<code>a * b</code>	Multiplicación
<code>/</code>	<code>a / b</code>	División
<code>%</code>	<code>a % b</code>	Módulo
<code>//</code>	<code>a // b</code>	División de enteros (también denominado Floor Division)
<code>**</code>	<code>a ** b</code>	Exponencial

2. Operadores Unarios

Los operadores unarios se caracterizan porque se aplican sobre un único operando. En Python se soportan el operador **Unario Positivo** y **Unario Negativo**. Este tipo de operadores se aplican sobre tipos numéricos en Python.

```
num = 10
```

```
# Operador Unario Positivo
+num
```

```
# Operador Unario Negativo
-num
```

```
texto = "Hola mundo"
```

```
-texto
```

3. Suma y Resta

Los operadores **Suma** y **Resta** son operadores binarios que pueden aplicarse sobre distintos tipos de datos.

3.1. Tipos de datos Numéricos

```
num1 = 10
num2 = 5
```

```
num1 + num2
```

```
num1 - num2
```

```
num2 - num1
```

```
num3 = 1.5
num4 = 0.5
```

```
num3 + num4
```

```
num3 - num4
```

```
x = 1
y = 1.0
x + y
```

3.2. Strings

```
text1 = "Hola"  
text2 = "mundo"
```

```
text1 + text2
```

```
text1 + " " + text2
```

```
text1 - text2
```

4. Multiplicación y División

Los operadores **Multiplicación** y **División** son operadores binarios que pueden aplicarse sobre distintos tipos de datos.

4.1. Tipos de datos numéricos

```
num1 = 10  
num2 = 5
```

```
num1 * num2
```

```
# IMPORTANTE: El resultado de una division siempre es un float  
num1 / num2
```

4.2. Strings

```
text1 = "Hola"  
text2 = "mundo"
```

```
text1 * text2
```

```
-----  
TypeError                                Traceback (most recent call last)  
Cell In[33], line 1  
----> 1 text1 * text2  
  
TypeError: can't multiply sequence by non-int of type 'str'
```

```
text1 / text2
```

```
-----  
TypeError                                Traceback (most recent call last)  
Cell In[37], line 1  
----> 1 text1 / text2  
  
TypeError: unsupported operand type(s) for /: 'str' and 'str'
```

```
text1 * 3
```

```
-----  
TypeError                                Traceback (most recent call last)  
Cell In[38], line 1  
----> 1 text1 * 3.2  
  
TypeError: can't multiply sequence by non-int of type 'float'
```

5. Módulo

El operador **Módulo** es un operador binario que devuelve el resto de una división entre tipos de datos numéricos.

```
num1 = 10  
num2 = 7
```

```
num1 % num2
```

```
3.3
```

```
num2 % num1
```

```
7
```

```
num1 = 10.5  
num2 = 7.2
```

```
num1 % num2
```

```
3.3
```

6. Exponencial

El operador **Exponencial** es un operador binario que se aplica sobre tipos de datos numéricos.

```
num1 = 10
```

```
num1 ** 2
```

```
100
```

```
num1 ** 3
```

```
1000
```

```
text1 = "Hola mundo"
```

```
text1 ** 2
```

```
-----  
TypeError                                Traceback (most recent call last)  
Cell In[45], line 1  
----> 1 text1 ** 2  
  
TypeError: unsupported operand type(s) for ** or pow(): 'str' and 'int'
```

7. Floor Division

El operador **Floor Division** es un operador binario que se aplica sobre tipos de datos numéricos y **devuelve la parte entera del resultado. Sin redondear**

```
num1 = 10  
num2 = 7
```

```
num1 / num2
```

```
num1 // num2
```

```
num1 = 13  
num2 = 7
```

```
num1 / num2
```

```
num1 // num2
```

```
num1 = 10.3  
num2 = 8.1
```

```
num1 // num2
```

```
1.0
```

