



universidad  
cenfotec\_  
La U de la informática

# Principios de Programación.

Operadores y expresiones: primera parte.

# Objetivos

- Repasar los conceptos de declaración, inicialización y asignación.
- Comprender la funcionalidad de los operadores aritméticos en Python.

# Operadores aritméticos

En cualquier programa es importante declarar e inicializar las variables con los valores iniciales, ya que sobre ellos se fundamentará la solución a un problema. Es necesario procesar y modificar esta información para llegar a la resolución del problema.

Para lograrlo se utilizan los **operadores aritméticos**, los cuales son símbolos reservados por el lenguaje que permiten realizar operaciones sobre las variables de los diferentes tipos de datos, tales como suma, resta, multiplicación, división y módulo.

# Operadores aritméticos en Python

En la siguiente tabla se muestran los diferentes operadores aritméticos utilizados en Python:

Operador	Descripción	Ejemplos
+   -   *   /	Operadores aritméticos básicos.	<pre>cant_numeros = 3 sumatoria = 2 + 3 + 7 #Resultado 12 promedio = sumatoria / cant_numeros #Resultado 4</pre>
**	Exponente	<pre>valor = 2 ** 3 #Resultado 8</pre>
%	Módulo o residuo de la división.	<pre>residuo1 = 8 % 2 #Resultado: 0 residuo2 = 11 % 5 #Resultado: 1</pre>
=	Operador de asignación.	<pre>numero = 71</pre>

# Operadores aritméticos en Python

- Los operadores de **suma**, **resta**, **multiplicación** y **división** permiten realizar las operaciones matemáticas básicas sobre los valores dados de una variable.
- El operador **módulo** le permite obtener el residuo de una división de la división entera entre dos valores numéricos dados.
  - Por ejemplo, en *8 módulo 2* ( $8 \% 2$ ) se obtiene como resultado 0, ya que al realizar la operación *8 dividido entre 2* se genera el valor de residuo 0.
  - En cambio, en la operación *11 módulo 5* ( $11 \% 5$ ), se obtiene un resultado de 1, ya que al dividir *11 entre 5* genera 1 como residuo.
- El operador de **asignación** le permite asignarle un valor a una variable, este puede ser un valor específico o el resultado de una operación. Es importante que el valor que se asigne concuerde con el tipo de dato de la variable.

# Expresiones en Python

Se pueden escribir expresiones aritméticas complejas en Python utilizando la combinación de operadores aritméticos, que pueden ser acompañados de **paréntesis**, para asegurar el orden de la ejecución de las operaciones.

También se puede aprovechar la precedencia de los operadores y utilizar los paréntesis solo cuando sea necesario. En el primer ejemplo a continuación se puede escribir sin paréntesis usando la precedencia de los operadores y en el segundo ejemplo es necesario utilizarlos para obtener el resultado esperado.

Tipo	Operadores	Ejemplos
Aritméticas	+ - * / %	<code>b = x * x + 2 * x</code>
		<code>b = (x - 7) / (y + 9 * z)</code>

# Operadores aritméticos en String

El tipo de dato **String** permite almacenar texto en una variable. En ocasiones es necesario agregar nuevo contenido a esa variable, por ejemplo, si en una variable se almacena únicamente un nombre, se pueden agregar posteriormente los apellidos.

Esto se logra utilizando el operador aritmético de **suma**, que en este caso tomará la función de agregar nuevo contenido o **concatenar**. Por ejemplo:

```
nombre = "Alan"  
nombre = nombre + " Mathison Turing"  
print(nombre);
```

En este caso, se declara la variable de tipo **String** **nombre** y se inicializa con el valor “*Alan*”. En la siguiente línea al valor inicial de **nombre** se le agrega nuevo texto, en este caso “*Mathison Turing*” y se asigna nuevamente a la variable **nombre**.

Por último, se imprime el resultado final y se obtiene en consola el texto completo “*Alan Mathison Turing*”.





universidad  
cenfotec\_  
La U de la informática