

## Guía Teórica N°2

### Tipos de Datos Básicos

Python tiene por definición, cuatro tipos de datos básicos:

- **Números enteros:** Aquellos números negativos y positivos, incluyendo el número cero, que no poseen punto decimal. (**int()**)  
Por ejemplo: 3, -5, 145, 0.
- **Números flotantes:** Aquellos números negativos y positivos, incluyendo el número cero, que poseen punto decimal. (**float()**)  
Por ejemplo: 4.5, 2.9999, -3.67524, 0.0.
- **Cadenas de caracteres (Strings):** Secuencia de caracteres que se utiliza para guardar y representar texto. (**str()**)  
Por ejemplo: "Educa2", "Aprendiendo a programar", 'Hola'.
- **Booleanos:** Contienen únicamente los valores *True* y *False*, los cuales representan los valores lógicos verdadero y falso respectivamente. (**bool()**)

Es importante saber que los tipos de datos en un lenguaje de programación nos permiten conocer el conjunto de valores que podemos trabajar con un dato en terminado y dependerá del tipo actual que tenga dicho dato, es decir, por ejemplo, si nos encontramos trabajando con un número entero, entonces los valores posibles para ese dato serán exclusivamente valores negativos y positivos (incluyendo el cero) sin coma decimal.

También, los tipos de datos poseen un **Rango de Valores**, los invitamos a leer la documentación del lenguaje en su sitio web oficial para conocer los valores posibles que puede tomar cada tipo de dato básico.

## ¿Qué son las variables?

Una variable es un identificador ligado a algún valor.

### ***Reglas para nombrar variables:***

- Comienzan siempre por una letra, seguida de otras letras o números.
- No se pueden utilizarse palabras reservadas del lenguaje.

A diferencia de otros lenguajes no tienen asociado un tipo y no es necesario declararlas antes de usarlas (tipado dinámico).

Para asignar un valor a una variable se utiliza el operador **=** y para borrar una variable se utiliza la instrucción **del**.

Por ejemplo, declaremos unas cuantas variables que contengan diferentes valores a almacenar:

**miVariable = 5**

**numero = -5.4783787**

**texto = 'Hola, estoy aprendiendo a programar en Python'**

### **Operaciones Aritméticas en Python:**

- **Suma (+):** Realiza la suma entre dos números.  
Por ejemplo,  $5 + 5 = 10$ .
- **Resta (-):** Realiza la resta entre dos números.  
Por ejemplo,  $45 - 20 = 25$ .
- **Multiplicación (\*):** Realiza la multiplicación entre dos números.  
Por ejemplo,  $10 * 4 = 40$ .
- **División (/):** Realiza la división entre dos números.  
Por ejemplo,  $25 / 5 = 5.0$ . (Note que retorna un número flotante).
- **División modular (%):** Realiza la división entre dos números, pero en esta ocasión retorna el valor del residuo de dicha división.

Por ejemplo,  $25 \% 5 = 0$ .

- **Potencia (\*\*):** Realiza la potencia de un número dado otro número siguiendo el orden siguiente, base \*\* exponente.

Por ejemplo,  $4 ** 3 = 64$ .

**Regla de precedencia en las operaciones aritméticas:** Nos indica en qué orden se ejecutarán las operaciones aritméticas en una expresión matemáticas. Esta regla se encuentra estructurada en los siguientes niveles empezando por el de mayor prioridad y luego disminuyendo su prioridad:

1. Paréntesis
2. Potencia
3. Multiplicación, División y División modular
4. Suma y Resta

Si dos operaciones se encuentran al mismo nivel, se evaluará primero la que se encuentre a la izquierda y así sucesivamente.

Ejecute las siguientes instrucciones y observe las diferencias entre los resultados. Se le recomienda analizar el motivo de los diferentes resultados a obtener:

```
print(12 * 5 + 2 / 3 ** 2)
print((12 * 5) + (2 / (3 ** 2)))
print((12 * 5) + (2 / 3) ** 2)
```

### Operaciones Lógicas en Python:

- Y lógico (**b1 and b2**) Devuelve True si los datos booleanos b1 y b2 son True, y False en caso contrario.

- O lógico (**b1 or b2**): Devuelve True si alguno de los datos booleanos b1 o b2 son True, y False en caso contrario.
- Negación lógica (**not b**): Devuelve True si el dato booleano b es False, y False en caso contrario.

#### Operaciones básicas de relación:

- Igualdad (**==**): ¿a es igual a b? (**a==b**)
- Diferencia (**!=**): ¿a es diferente de b? (**a != b**)
- Mayor (**>**): ¿a es mayor a b? (**a > b**)
- Mayor o igual (**>=**): ¿a es mayor o igual a b? (**a >= b**)
- Menor (**<**): ¿a es menor a b? (**a < b**)
- Menor o igual (**<=**): ¿a es menor o igual a b? (**a <= b**)

#### Operaciones básicas con cadenas caracteres:

- Concatenación (**c1 + c2**): Devuelve la cadena resultado de concatenar las cadenas c1 y c2.
- Repetición (**c \* n**): Devuelve la cadena resultado de concatenar n copias de la cadena c.
- Verificar si hay una cadena dentro de otra cadena (**c1 in c2**): Devuelve True si c1 es una cadena contenida en c2 y False en caso contrario.

**Instrucción print( ):** La utilizamos para imprimir por pantalla (terminal) cadenas de caracteres los cuales pueden ser de información para el usuario.

**Instrucción input( ):** La utilizamos para pedirle al usuario el ingreso de dato a través de teclado.