

# tipos-datos

October 9, 2024

## 1 Ejercicio 1

Convierte  $y=1.0$  a entero, recuerda que para convertir a entero debes ocupar la función `int()`.

```
[ ]: # Definamos un flotante
y = 1.0
print(y)
print(type(y))

# Convertir el flotante a entero con la función int
y_int = int(y)
print(y_int)
print(type(y_int))
```

```
1.0
<class 'float'>
1
<class 'int'>
```

Descomentar las siguientes seldas de código para ver los diferentes resultados

```
[ ]: print(int(1.2321))
print(int(1.747))
print(int(-3.94535))
print(int(-2.19774))
```

```
1
1
-3
-2
```

## 2 Ejercicio 2

¿Qué sucede cuando multiplicas un número entero o flotante por `True`? ¿Qué pasa cuando los multiplicas por `False`? ¿Cómo cambia la respuesta si los números son positivos o negativos? ¿Qué sucede cuando multiplicas una cadena por `True`? por `False`?

Utilice la siguiente celda de código para responder las preguntas.

```
[ ]: #Prueba con entero
print(4 * True)
print(4 * False)
#Prueba con flotante
print(4.5 * True)
print(4.5 * False)
#prueba con string
print(type("cosi" * False))
print(len("cosi" * False))
print(type("cosi" * True))
print(len("cosi" * True))
```

```
4
0
4.5
0.0
<class 'str'>
0
<class 'str'>
4
```

### 3 Ejercicio 3

Usa la siguiente celda de código para crear una función `obt_cost_esp` que tome como entrada tres variables:

`camas` - número de habitaciones (tipo de dato flotante) `balneario` - número de baños (tipo de dato float) `sotano` - si la casa tiene o no un sótano (tipo de dato booleano)

Debería devolver el costo esperado de una vivienda de esas características.

```
[ ]: def obt_cost_esp(camas, balneario, sotano):
    value = 80000 + camas * 30000 + balneario * 10000 + sotano * 40000
    return value

print(obt_cost_esp(0, 0, False))
print(obt_cost_esp(1, 1, False))
print(obt_cost_esp(2, 1, True))
print(obt_cost_esp(1, 1, True))
```

```
80000
120000
190000
160000
```

## 4 Ejercicio 4

Continuaremos nuestro estudio de la aritmética booleana. Para esta pregunta, tu tarea es proporcionar una descripción de lo que sucede cuando agregas valores booleanos.

```
[ ]: print(False + False)
      print(True + False)
      print(False + True)
      print(True + True)
      print(False + True + True + True)
```

0  
1  
1  
2  
3

Los valores booleanos True y False toman 2 valores numéricos:

True = 1

False = 0

Entonces al hacer las sumas obtenemos los siguientes resultados:

False + False == 0 + 0 == 0

True + False == 1 + 0 == 1

False + True == 0 + 1 == 1

True + True == 1 + 1 == 2

False + True + True + True == 0 + 1 + 1 + 1 == 3

## 5 Ejercicio 5

Tienes una tienda en línea donde vendes anillos con grabados personalizados. Ofreces anillos de oro macizo y cubiertos en oro.

Los anillos chapados en oro tienen un costo base de 50 dolares y cobras 7 dolares por unidad grabada. Los anillos de oro macizo tienen un costo base de 100 dolares y cobras 10 dolares por unidad grabada.

Los espacios y la puntuación se cuentan como unidades grabadas.

Escriba una función `cost_de_proyect()` que tome dos argumentos:

**grabado** - una cadena de Python con el texto del grabado

**solid\_oro** - un valor booleano que indica si el anillo es de oro macizo

Se debe devolver el costo del proyecto

```
[12]: def cost_de_proyect(grabado, solid_oro):  
      grabado_len = len(grabado)  
      cost = (50 + grabado_len * 7) * (not solid_oro) + (100 + grabado_len * 10)  
      ↪* solid_oro  
      return cost
```

Ejecute la siguiente celda de código para calcular el costo de grabar tu nombre con tu primer apellido en un anillo de oro macizo e imprime el resultado.

```
[21]: proyect_uno = cost_de_proyect("Cosijoeza Melchor", True)  
      print(proyect_uno)
```

270

En una celda de código calcula el costo de grabar el 10/08/2000 en un anillo chapado en oro.

```
[20]: proyect_uno = cost_de_proyect("10/08/2000", False)  
      print(proyect_uno)
```

120