

UNIVERSIDAD EXTERNADO DE COLOMBIA
INTELIGENCIA ARTIFICIAL CON APLICACIONES EN ECONOMÍA I
TALLER TIPOS DE OBJETOS Y ESTRUCTURAS DE DATOS SECUENCIALES Y NO SECUENCIALES EN
PYTHON

Docente: Lina María Castro
17 de agosto de 2025

- Revisa el material de las clases 4 y 5 para responder.
- Utiliza un Jupyter Notebook, sea en Google Colab o en VS Code, para comprobar tus respuestas.
- La entrega de este taller es **OPCIONAL** y, en caso de ser entregado, entra en las notas de ejercicios y quices.
- **Plazo máximo de entrega:** martes, 26 de agosto de 2025 antes de las 11:59 p.m.
- **Forma de entrega:** Jupyter Notebook publicado en su cuenta de Github con el nombre "Taller_objetos_Python.ipynb". Ser ordenado: debe aparecer el número del ejercicio, el código, el resultado del código y escribir en una celda de markdown cuál es la respuesta correcta (a, b, c, d). Enviar link del notebook al correo lmcastroco@gmail.com antes del plazo máximo de entrega, de lo contrario, no será tenido en cuenta.

1.

¿Cuál de los siguientes NO es un tipo de objeto básico en Python?

- a) Números
- b) Cadenas de texto
- c) Conjuntos
- d) Funciones matemáticas

2.

En Python, los valores True y False pertenecen al tipo:

- a) Entero
- b) Booleano
- c) Cadena
- d) Conjunto

3.

¿Cuál de los siguientes objetos se considera un tipo numérico en Python?

a) int, float y complex

b) str, int y tuple

c) list, set y dict

d) bool, tuple y set

4.

¿Qué imprime este código?

```
x = (1, 2, 3)
```

```
print(x[1])
```

a) 1

b) 2

c) 3

d) Error

5.

¿Cuál es el error en este código?

```
t = (1, 2, 3)
```

```
t[0] = 10
```

a) No hay error

b) Las tuplas son mutables

c) Las tuplas son inmutables

d) Error de sintaxis

6.

¿Qué resultado produce?

```
s = {1, 2, 2, 3, 3}
```

```
print(s)
```

- a) {1, 2, 2, 3, 3}
- b) {1, 2, 3}
- c) [1, 2, 3]
- d) Error

7.

¿Cuál es la salida?

```
palabra = "Mundo"
```

```
print(palabra[1:4])
```

- a) "Mund"
- b) "M"
- c) "und"
- d) "ndo"

8.

¿Qué imprime?

```
lista = [10, 20, 30]
```

```
lista.append(40)
```

```
print(lista)
```

- a) [10, 20, 30]
- b) [10, 20, 30, 40]
- c) [40, 10, 20, 30]
- d) Error

9.

¿Qué ocurre?

```
dic = {"a": 1, "b": 2}
```

```
print(dic["c"])
```

- a) 0
- b) None
- c) Error
- d) "c"

10.

Elige la opción correcta para obtener el valor 3 de este diccionario:

```
d = {"x": 1, "y": 2, "z": 3}
```

- a) `d["z"]`
- b) `d[3]`
- c) `d.z`
- d) `d.get("a")`

11.

¿Qué devuelve este código?

```
print(type(3.0))
```

- a) `int`
- b) `float`
- c) `str`
- d) `complex`

12.

¿Cuál es el resultado?

```
x = [1, 2, 3]
```

```
y = x
```

```
y.append(4)
```

```
print(x)
```

- a) `[1, 2, 3]`
- b) `[1, 2, 3, 4]`
- c) `Error`
- d) `[4]`

13.

```
numeros = ["10", "20", "30"]
```

```
resultado = ",".join(numeros)
```

El valor de resultado será:

- a) 10,20,30 (como enteros)

b) "10,20,30"

c) ["10,20,30"]

d) (10,20,30)

14.

Completa el código para convertir un conjunto a lista:

s = {1, 2, 3}

a) list(s)

b) tuple(s)

c) dict(s)

d) set(s)

15.

¿Cuál es la salida?

cadena = "Hola"

print(cadena * 3)

a) "Hola Hola Hola"

b) "HolaHolaHola"

c) [Hola, Hola, Hola]

d) Error

16.

¿Qué valor imprime?

lista = [10, 20, 30]

print(lista[-1])

a) 10

b) 20

c) 30

d) Error

17.

```
precios = [1200, 500, 3400, 800]
```

```
precios.sort()
```

¿Qué contendrá precios después de ejecutar el código?

a) [500, 800, 1200, 3400]

b) [1200, 500, 3400, 800]

c) [3400, 1200, 800, 500]

d) (500, 800, 1200, 3400)

18.

¿Cuál es el error?

```
conjunto = {1, 2, 3}
```

```
print(conjunto[0])
```

a) Los conjuntos no son indexables

b) Falta una coma

c) Error de sintaxis

d) No hay error

19.

¿Qué devuelve?

```
len({"a": 1, "b": 2, "c": 3})
```

a) 6

b) 3

c) 2

d) 1

20.

¿Qué resultado da?

```
lista = [1, 2, 3]
```

```
print(lista * 2)
```

a) [1, 2, 3, 1, 2, 3]

b) [2, 4, 6]

c) [1, 2, 3, 2]

d) Error

21.

¿Qué ocurre?

```
d = {"a": 1}
```

```
d["a"] = 100
```

```
print(d)
```

a) {"a": 1}

b) {"a": 100}

c) {"a": 1, "a": 100}

d) Error

22.

¿Qué imprime?

```
print("python".upper())
```

a) PYTHON

b) Python

c) python

d) Error

23.

Completa el código para agregar el número 5 a un conjunto:

```
s = {1, 2, 3}
```

```
s.__(5)
```

a) add

b) append

c) insert

d) extend

24.

¿Cuál es la salida?

```
d = {"x": 10, "y": 20}
```

```
print("x" in d)
```

a) False

b) True

c) Error

d) None

25.

¿Qué imprime?

```
print([1, 2, 3] == [1, 2, 3])
```

a) True

b) False

c) None

d) Error

26.

```
for i in range(5):
```

```
    print(i)
```

¿Qué se imprimirá en pantalla?

a) 1 2 3 4 5

b) 0 1 2 3 4 5

c) 0 1 2 3 4

d) 5 4 3 2 1

27.

¿Qué ocurre?

```
s = {1, 2, 3}
```

```
s.remove(4)
```


- a) Borra 4 del conjunto
- b) No hace nada
- c) Lanza un error
- d) Convierte a lista

28.

Tienes el siguiente texto:

texto = "Colombia México Perú Chile"

países = texto.split()

¿Qué contendrá la variable países?

- a) "Colombia,México,Perú,Chile"
- b) ["Colombia", "México", "Perú", "Chile"]
- c) ["Colombia México Perú Chile"]
- d) ("Colombia", "México", "Perú", "Chile")

29.

ventas = "1000;2500;3200;4500"

valores = ventas.split(";")

¿Qué resultado produce valores?

- a) ["1000", "2500", "3200", "4500"]
- b) [1000, 2500, 3200, 4500]
- c) ("1000", "2500", "3200", "4500")
- d) "1000 2500 3200 4500"

30.

palabras = ["Economía", "Mercados", "Exportaciones"]

texto = " - ".join(palabras)

¿Cuál será el contenido de texto?

- a) ["Economía - Mercados - Exportaciones"]
- b) "Economía - Mercados - Exportaciones"

- c) "Economía Mercados Exportaciones"
- d) ("Economía - Mercados - Exportaciones")

31.

```
empresas = ["Amazon", "Google", "Microsoft", "Apple"]
```

```
empresas.sort()
```

¿Qué contendrá empresas después de ejecutar el código?

- a) ["Amazon", "Apple", "Google", "Microsoft"]
- b) ["Amazon", "Google", "Microsoft", "Apple"]
- c) ["Microsoft", "Google", "Apple", "Amazon"]
- d) ("Apple", "Amazon", "Google", "Microsoft")

32.

```
numeros = list(range(2, 10, 2))
```

¿Qué contendrá la lista numeros?

- a) [2, 4, 6, 8]
- b) [2, 3, 4, 5, 6, 7, 8, 9]
- c) [2, 6, 10]
- d) [2, 4, 6, 8, 10]