

Untitled

June 29, 2025



1 01PAO25-25 - Python, Data Types



Cristopher Santander

1.1 EJERCICIOS EN CLASE 27/06/2025

1. Escribe un programa que muestre por pantalla la concatenación de un número y una cadena de caracteres. Para obtener esta concatenación puedes usar uno de los operadores explicados en este tema. Ejemplo: dado el número 3 y la cadena 'abc', el programa mostrará la cadena '3abc'.
2. Escribe un programa que muestre por pantalla un valor booleano que indique si un número entero N está contenido en un intervalo semiabierto $[a,b)$, el cual establece una cota inferior a (inclusive) y una cota superior b (exclusive) para N .
3. Escribe un programa que, dado dos strings $S1$ y $S2$ y dos números enteros $N1$ y $N2$, determine si el substring que en $S1$ se extiende desde la posición $N1$ a la $N2$ (ambos inclusive) está contenido en $S2$.
4. Dada una *lista* con elementos duplicados, escribir un programa que muestre una nueva *lista* con el mismo contenido que la primera pero sin elementos duplicados.
5. Escribe un programa que, dada una *lista* de strings L , un string s perteneciente a L y un string t , reemplace s por t en L . El programa debe mostrar la lista resultante por pantalla.

6. Escribe un programa que defina una *tupla* con elementos numéricos, reemplace el valor del último por un valor diferente y muestre la *tupla* por pantalla. Recuerda que las *tuplas* son inmutables. Tendrás que usar objetos intermedios.
7. Dada la lista [1,2,3,4,5,6,7,8] escribe un programa que, a partir de esta lista, obtenga la lista [8,6,4,2] y la muestre por pantalla.
8. Escribe un programa que, dada una tupla y un índice válido i , elimine el elemento de la tupla que se encuentra en la posición i . Para este ejercicio sólo puedes usar objetos de tipo tupla. No puedes convertir la *tupla* a una *lista*, por ejemplo.
9. Escribe un programa que obtenga la mediana de una *lista* de números. Recuerda que la mediana M de una lista de números L es el número que cumple la siguiente propiedad: la mitad de los números de L son superiores a M y la otra mitad son inferiores. Cuando el número de elementos de L es par, se puede considerar que hay dos medianas. No obstante, en este ejercicio consideraremos que únicamente existe una mediana.

```
[4]: #ejercicio 1
a = 3
b = 'abc'
resultado = str(a)+b
print(resultado)
```

3abc

```
[6]: #Ejercicio 2
a = 1
b = 10
n = 11
resultado = a <= n < b
print(resultado)
```

False

```
[14]: #Ejercicio 3
s1 = "klkl"
s2 = "qlqj"
n1 = 1
n2 = 2
subcadena = s1[n1:n2+1]
print(subcadena in s2)
```

False

```
[15]: #Ejercicio 4
lista = [1, 2, 2, 3, 4, 4, 5]
lista_sin_duplicados = list(set(lista))
print(lista_sin_duplicados) # Salida (orden puede variar): [1, 2, 3, 4, 5]
```

[1, 2, 3, 4, 5]

```
[16]: #Ejercicio 5
L = ['manzana', 'pera', 'manzana', 'uva']
s = 'manzana'
t = 'banana'
L_resultado = [t if x == s else x for x in L]
print(L_resultado) # Salida: ['banana', 'pera', 'banana', 'uva']
```

['banana', 'pera', 'banana', 'uva']

```
[17]: #Ejercicio 6
tupla = (1, 2, 3, 4)
nueva_tupla = tupla[:-1] + (99,)
print(nueva_tupla) # Salida: (1, 2, 3, 99)
```

(1, 2, 3, 99)

```
[18]: #Ejercicio 7
lista = [1, 2, 3, 4, 5, 6, 7, 8]
nueva_lista = lista[::-1][::2]
print(nueva_lista) # Salida: [8, 6, 4, 2]
```

[8, 6, 4, 2]

```
[21]: #Ejercicio 8
def eliminar_elemento(tupla, i):
    return tupla[:i] + tupla[i+1:]

tupla = (10, 20, 30, 40, 50)
i = 2
nueva_tupla = eliminar_elemento(tupla, i)
print(nueva_tupla)
```

(10, 20, 40, 50)

```
[22]: #Ejercicio 9
def obtener_mediana(lista):
    lista_ordenada = sorted(lista)
    n = len(lista_ordenada)
    medio = n // 2
    if n % 2 == 1:
        return lista_ordenada[medio]
    else:
        return (lista_ordenada[medio - 1] + lista_ordenada[medio]) / 2

numeros = [7, 1, 3, 5, 9]
mediana = obtener_mediana(numeros)
print(mediana)
```