



## 01 PAO25-25 - PYTHON, Data Types



*Luis Pilaguano*

### Ejercicios

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'.

```
In [3]: numeros = 17
cadena = 'abc'
numero_Cadena = str(numeros) + cadena
print('Resultado de la concatenación: ' + numero_Cadena)
```

Resultado de la concatenación: 17abc

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.

```
In [13]: n = 7
a = 5
b = 10
comparacion = (n >= a) and (n < b)
print(comparacion)

n2 = 10
comparacion1 = (n2 >=a) and (n2 < b)
print(comparacion1)
```

True  
False

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.

```
In [23]: s1 = 'Machine Learning'
s2 = 'Desarrollo de Software'
n1 = 0
n2 = 10

substring = s1[n1:n2+1]
contiene = substring in s2
print(f"s1: '{s1}'")
print(f"s2: '{s2}'")
print(f"Substring de s1 desde {n1} hasta {n2}: '{substring}'")
print(f"El substring está contenido en S2:? {contiene}")

# -----
print()
S1 = "Desarrollo web"
S2 = "Aprender es clave"
N1 = 0
N2 = 9
substring_buscado = S1[N1 : N2 + 1]
esta_contenido = substring_buscado in S2
print(f"S1: '{S1}'")
print(f"S2: '{S2}'")
print(f"Substring de S1 desde {N1} hasta {N2}: '{substring_buscado}'")
print(f"El substring está contenido en S2:? {esta_contenido}")
```

```
s1: 'Machine Learning'
s2: 'Desarrollo de Software'
Substring de s1 desde 0 hasta 10: 'Machine Lea'
El substring está contenido en S2:? False
```

```
S1: 'Desarrollo web'
S2: 'Aprender es clave'
Substring de S1 desde 0 hasta 9: 'Desarrollo'
El substring está contenido en S2:? False
```

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.

```
In [25]: lista = [1,1,2,2,3,3,4,4,5,5,6,7,8,9]
# set para eliminar duplicados
lista_sin_duplicar = list(set(lista))
#imprimir datos
print("Lista original: ",lista)
print("lista sin duplicar datos: ",lista_sin_duplicar)
```

```
Lista original: [1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 7, 8, 9]
lista sin duplicar datos: [1, 2, 3, 4, 5, 6, 7, 8, 9]
```

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.

```
In [5]: L = ["Monitor", "Teclado", "Mause", "CPU"]
s = "Teclado"
t = "Parlantes"
#creo la nueva lista
L_nueva = []
for i in L:
    if i == s:
        L_nueva.append(t)
    else:
        L_nueva.append(i)
#mostrar los resultado
print("Lista original", L)
print("Nueva lista ", L_nueva)
```

Lista original ['Monitor', 'Teclado', 'Mause', 'CPU']  
 Nueva lista ['Monitor', 'Parlantes', 'Mause', 'CPU']

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.

```
In [8]: tupla = [5,10,15,20,25]
#convertir la tupla en ua lista
lista1 = list(tupla)
#Boy a reemplazar el ultimo valor
lista1[-1] = 99
#vuelvo a convertir la lista en una tupla
tupla2 = tuple(lista1)
#Muestro los resultados
print("Tupla Original", tupla)
print("Tupla Modificada", tupla2)
```

Tupla Original [5, 10, 15, 20, 25]  
 Tupla Modificada (5, 10, 15, 20, 99)

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.

```
In [14]: lista_original = [1, 2, 3, 4, 5, 6, 7, 8]

lista = lista_original[::-2]

print("Lista original:", lista_original)
print("Lista Modificada:", lista)
```

Lista original: [1, 2, 3, 4, 5, 6, 7, 8]  
 Lista Modificada: [8, 6, 4, 2]

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 convetir la tupla a una lista, por ejemplo.

```
In [17]: tupla = [1,2,3,4,5,6,7,8,9]
# creo el indice que quiero eiliminar en este caso 3
i = 3
#creo una nueva tupla sin el dato de la posicion i (3)
nueva_tupla = tupla[:i] + tupla[i+1:]
## imprimir resultado
```

```
print("Tupla original", tupla)
print("Tupla sin elemento indice ", i , " : ", nueva_tupla)
```

Tupla original [1, 2, 3, 4, 5, 6, 7, 8, 9]

Tupla sin elemento indice 3 : [1, 2, 3, 5, 6, 7, 8, 9]

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.

```
In [21]: L = [7,1,5,3,8,2]
#ordeno la lista
L_Ordenada = sorted(L)
#obtengo la cantidad de elementos
cantidad = len(L_Ordenada)
#busco calcular el indice ecntral
indice_M = cantidad // 2
mediana = L_Ordenada[indice_M]
#Mostrar resultado
print("lista ordenada: ", L_Ordenada)
print("Mediana. ", mediana)
```

lista ordenada: [1, 2, 3, 5, 7, 8]

Mediana. 5

[GitHub Actividad 3](#)

In [ ]: