



# Certificación PCAP

## Lección 4: Cadenas y Listas II

# Actividad 1

## ACTIVIDAD LECCIÓN 4 (50%)

### Objetivos

- | Familiarizarse con el uso de las cadenas
- | Familiarizarse con los nuevos conceptos aprendidos de las listas

### Contenido correspondiente a lección 4:

2. Cadenas

3. Listas II

### Actividad relacionada con la lección 4:

El objetivo de esta actividad es crear un pequeño programa que dada una cadena de texto que represente una lista de números separada con espacios devuelva la suma total de los números de dicha lista y devuelva una cadena de texto por pantalla con la misma lista de números ordenada de forma numérica. Si la línea de números contiene la palabra "STOP" independientemente de que esté en mayúsculas o minúsculas o su combinación los números a la derecha de esta palabra se deben despreciar tanto para la suma como para la ordenación.

Se valorará que el código haga lo solicitado, la utilización de los conceptos y recursos vistos en la lección y la calidad del código.

**TIP:** Se recomienda leer los diferentes métodos que ofrecen los objetos de tipo cadena ya que podría haber alguno o algunos que os podrían facilitar la actividad:

<https://docs.python.org/3/library/stdtypes.html#string-methods>

A continuación, se muestra dos ejemplos de cómo debe funcionar el programa realizado y la salida por pantalla que debe mostrar:

```
Introduzca una línea con los números separados por espacios: 23 4 5.4 8.5 9.0
La suma total es: 49.9
La lista ordenada es: [4.0, 5.4, 8.5, 9.0, 23.0]
```

```
Introduzca una línea con los números separados por espacios: 23 4 5.4 8.5 STOP 9.0
La suma total es: 40.9
La lista ordenada es: [4.0, 5.4, 8.5, 23.0]
```

## Actividad 2

### ACTIVIDAD LECCIÓN 4 (50%)

#### Objetivos

- | Profundizar en los conceptos vistos en esta lección

#### Contenido correspondiente a lección 4:

4. Cadenas

5. Listas II

#### Actividad relacionada con la lección 4:

Las preguntas sobre listas, tuplas y diccionarios suelen ser muy comunes en el examen de certificación PCAP, por lo que con esta actividad se pretende trabajar en los diferentes conceptos vistos en esta lección. En esta actividad debes enviar para cada pregunta la opción u opciones correctas, así como una breve explicación de tu elección. **Es importante que no te olvides de enviar la breve explicación de tu elección ya que si no lo envías se te valorará la pregunta como incorrecta.**

#### Pregunta 1:

¿Cuál es la salida del siguiente fragmento de código? (1 opción)

```
print('abc' + 'xy' * 2 + 'a' * 3)
```

- A) abcxyabcxyaabcxyabcxyaabcxyabcxya
- B) abcxyabcxyaaa
- C) abcxyxyaaa
- D) El programa dará un error.

**Pregunta 2:**

¿Cuál es la salida del siguiente fragmento de código? (1 opción)

```
salida = "-"  
for ch in "ZYX":  
    salida = salida + chr(ord(ch) - 1) * 2  
print(salida)
```

- A) -WW
- B) -YYXXWW
- C) --YYXXWW
- D) -YYXXYXW-YYXXYXW
- E) -
- F) El programa dará un error

**Pregunta 3:**

¿Cuál o cuáles de las siguientes afirmaciones son ciertas? (una o varias opciones)

```
a = "2.1"  
b = float(a)  
c = int(b)  
d = int(a)  
idc = id(c)  
idd = id(d)
```

- A) Al finalizar la ejecución el valor de c y d será 2
- B) Al finalizar la ejecución idc e idd contendrán el mismo valor.
- C) Al finalizar la ejecución idc e idd contendrán distinto valor.
- D) El programa dará un error.

**Pregunta 4:**

¿Cuál o cuáles de las siguientes afirmaciones son ciertas? (una o varias opciones)

```
mi_lista = ["aa", "AA", "a"]
mi_lista1 = mi_lista[:]
mi_lista.sort()
for el in mi_lista:
    print(el, end="-")
for el in mi_lista1:
    print(el, end="-")
```

- A) El programa imprimirá por pantalla "AA-a-aa-AA-a-aa-"
- B) El programa imprimirá por pantalla "AA-a-aa-aa-AA-a-"
- C) Si ejecuto después "mi\_lista == mi\_lista1" me devolverá True.
- D) Si ejecuto después "mi\_lista == mi\_lista1" me devolverá False.
- E) El programa dará un error.

**Pregunta 5:**

¿Cuál es la salida del siguiente fragmento de código? (1 opción)

```
mi_lista = [[i if j > 1 else -1 for i in range(4)]
             for j in range(0,4) if j % 2 == 0]
print(mi_lista)
```

- A) [[0, 1, 2, 3], [0, 1, 2, 3]]
- B) [[-1, -1, -1, -1], [0, 1, 2, 3]]
- C) [[-1, -1, 2, 3], [-1, -1, 2, 3]]
- D) El programa dará un error.