01 PAO25-25 - PYTHON, Data Types

Luis Pilaguano

Ejercicios

1.- Escribe un programa que calcule la suma de todos los elementos de una lista dada. La lista sólo puede contener elementos numéricos.

In [3]:
```
numeros = [1,2,3,4,5]
suma = sum(numeros)
print(f"Suma total {suma}")
```

```
Suma total 15
```

1.- 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. Para este ejercicio, no puedes hacer uso de objetos de tipo 'Set'.

In [5]:
```
lista = [1,2,2,3,4,5,5]
nueva_lista = []
for elemento in lista:
    if elemento not in nueva_lista:
        nueva_lista.append(elemento)
print(f"Lista sin duplicados {nueva_lista}")
```

```
Lista sin duplicados [1, 2, 3, 4, 5]
```

3.- Escribe un programa que construya un diccionario que contenga un número (entre 1 y n) de elementos de esta forma: (x, x*x). Ejemplo: para n = 5, el diccionario resultante sería {1: 1, 2: 4, 3: 9, 4: 16, 5: 25}

In [6]:
```
n = 5
num = {x: x * x for x in range(1, n + 1)}
print(num)
```

```
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25}
```

4.- Escribe un programa que, dada una lista de palabras, compruebe si alguna empieza por 'a' y tiene más de 9 caracteres. Si dicha palabra existe, el programa deberá terminar en el momento exacto de encontrarla. El programa también debe mostrar un mensaje apropiado por pantalla que indique el éxito o el fracaso de la búsqueda. En caso de éxito, también se mostrará por pantalla la palabra encontrada.

In [ ]:
```
##
```

5.- Dada una lista L de números positivos, escribir un programa que muestre otra lista (ordenada) que contenga todo índice i que cumpla la siguiente condición: L[i] es múltiplo de 3. Por ejemplo, dada la lista L = [3,5,13,12,1,9] el programa mostrará la lista [0,3,5] dado que L[0], L[3] y L[5] son, respectivamente, 3, 12 y 9, que son los únicos múltiplos de 3 que hay en L.

In [7]:
```
L = [3, 5, 13, 12, 1, 9]
indices = [i for i in range(len(L)) if L[i] % 3 == 0]
print("Índices con múltiplos de 3 ", sorted(indices))
```

```
Índices con múltiplos de 3: [0, 3, 5]
```

6.- Dado un diccionario cuyos elementos son pares de tipo string y numérico (es decir, las claves son de tipo 'str' y los valores son de tipo 'int' o 'float'), escribe un programa que muestre por pantalla la clave cuyo valor asociado representa el valor númerico más alto de todo el diccionario. Por ejemplo, para el diccionario {'a': 4.3, 'b': 1, 'c': 7.8, 'd': -5} la respuesta sería 'c', dado que 7.8 es el valor más alto de los números 4.3, 1, 7.8 y -5.

In [8]:
```
d = {'a': 4.3, 'b': 1, 'c': 7.8, 'd': -5}
clave_max = max(d, key=d.get)
print("Clave con el valor máximo ", clave_max)
```

```
Clave con el valor máximo: c
```

7.- Dada la lista a = [2, 4, 6, 8] y la lista b = [7, 11, 15, 22], escribe un programa que itere las listas a y b y multiplique cada elemento de a que sea mayor que 5 por cada elemento de b que sea menor que 14. El programa debe mostrar los resultados por pantalla.

In [9]:
```
a = [2, 4, 6, 8]
b = [7, 11, 15, 22]

for num_a in a:
    if num_a > 5:
        for num_b in b:
```

```
            if num_b < 14:
                print(f"{num_a} * {num_b} = {num_a * num_b}")
```

```
6 * 7 = 42
6 * 11 = 66
8 * 7 = 56
8 * 11 = 88
```

8.- Escribir un programa que pida un valor numérico X al usuario. Para ello podéis hacer uso de la función predefinida 'input'. El programa deberá mostrar por pantalla el resultado de la división 10/X. En caso de que el usuario introduzca valores no apropiados, el programa deberá gestionar correctamente las excepciones, por ejemplo, mostrando mensajes informativos por pantalla.

In [10]:
```
try:
    x = float(input("Ingresa un número"))
    resultado = 10 / x
    print("Resultado ", resultado)
except ValueError:
    print("Por favor ingresa un número válido")
except ZeroDivisionError:
    print("No se puede dividir por cero")
```

```
Resultado:  1.6666666666666667
```

9.- Escribir un programa que cree un diccionario cualquiera. Posteriormente, el programa pedirá al usuario (a través de la función predefinida 'input') que introduzca una clave del diccionario. Si la clave introducida es correcta (es decir, existe en el diccionario), el programa mostrará por pantalla el valor asociado a dicha clave. En caso de que la clave no exista, el programa gestionará de manera apropiada el error, por ejemplo, mostrando un mensaje informativo al usuario.

In [12]:
```
diccionario = {'nombre': 'Luis', 'edad': 26, 'pais': 'Ecuador'}
clave = input("Introduce una clave del diccionario ")

if clave in diccionario:
    print("Valor asociado ", diccionario[clave])
else:
    print("Esa clave no existe en el diccionario")
```

```
Valor asociado  Luis
```

10.- Escribe una list comprehension que construya una lista con los números enteros positivos de una lista de números dada. La lista original puede incluir números de tipo float, los cuales deben ser descartados.

In [16]:
```
numeros = [1, -2, 3.5, 4, -5.0, 6, 0, 7.2, 8]
positivos = [n for n in numeros if n > 0 and int(n) == n]
print(positivos)
```

```
[1, 4, 6, 8]
```

11.- Escribe una set comprehension que, dada una palabra, construya un conjunto que contenga las vocales de dicha palabra.

In [17]:
```
palabra = "Programacion"
vocales = {letra for letra in palabra if letra in 'aeiou'}
print(vocales)
```

```
{'o', 'a', 'i'}
```

12.- Escribe una list comprehension que construya una lista con todos los números del 0 al 50 que contengan el dígito 3. El resultado será: [3, 13, 23, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 43].

In [22]:
```
numeros = [x for x in range(51) if '3' in str(x)]
print(numeros)
```

```
[3, 13, 23, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 43]
```

13.- Escribe una dictionary comprehension que construya un diccionario que incluya los tamaños de cada palabra en una frase dada. Ejemplo: el resultado para la frase "Soy un ser humano" será {'Soy': 3, 'un': 2, 'ser': 3, 'humano': 6}

In [23]:
```
frase = "Soy un ser humano"
palabra_frase = {palabra: len(frase) for palabra in frase.split()}
print(palabra_frase)
```

```
{'Soy': 17, 'un': 17, 'ser': 17, 'humano': 17}
```

14.- Escribe una list comprehension que construya una lista que incluya todos los números del 1 al 10 en orden. La primera mitad se mostrarán en formato numérico; la segunda mitad en texto. Es decir, el resultado será: [1, 2, 3, 4, 5, 'seis', 'siete', 'ocho', 'nueve', 'diez'].

In [25]:
```
texto = ['seis', 'siete', 'ocho', 'nueve', 'diez']
resultado = [x for x in range(1, 6)] + texto
print(resultado)
```

```
[1, 2, 3, 4, 5, 'seis', 'siete', 'ocho', 'nueve', 'diez']
```

Git Hub Estructuras de Control

In [ ]:
```
```