

Cuestiones

- Suponga declaradas las siguientes variables: lista (de tipo list), tupla (de tipo tuple), conjunto (de tipo set) y diccionario (de tipo dict). Escriba una o dos instrucciones para realizar cada una de las siguientes acciones; si considera que no es posible realizar alguna de las acciones solicitadas, explique por qué.
 - Añadir el valor 10 a conjunto.
 - Sustituir el primer elemento de lista por el valor 20.
 - Sustituir el primer elemento de tupla por el valor 20.
 - Eliminar la clave 10 de diccionario.
 - Eliminar la primera posición de conjunto.
- Qué muestra en la consola esta secuencia

```
lista1=[i for i in range(0,11,2)]  
lista2=[(i,i+1) for i in lista1]  
print(sum((y for _,y in lista2))
```
- Dada la función

```
def funcion(x, y, z):  
    if y:  
        a=x+z  
    else:  
        a=x-z  
    return a
```


¿Qué muestra en pantalla la expresión?

```
print(funcion(5, False, -1)+funcion(3,True,2))
```
- ¿Cuál es la salida del siguiente trozo de código?

```
lista=[2,3,1,0]  
for i in lista:  
    print(sum(x for x in lista[i:]))
```
- Indique en cada caso cuál sería el valor devuelto por la función:

```
def funcion(lista, valor):  
    res = []  
    for e in lista:  
        if e[0] == valor:  
            res.append(e[1])  
        if e[1][0] == valor:  
            break  
    return res
```

 - `funcion([("a", "arriba"), ("b", "barco"), ("c", "casa")], "b")`
 - `funcion([("1", "arriba"), ("2", "barco"), ("1", "casa")], "1")`
 - `funcion([("arriba", "a"), ("barco", "b"), ("casa", "c")], "b")`
- Crea una variable llamada hoy de tipo datetime con la fecha y hora actual. Muestra en la consola el resultado de multiplicar el día de hoy por 24 más la hora.

- 7 Tenemos una tupla de tipo Persona con los siguientes atributos:
- # nombre: str con el nombre de la persona.
 - # edad: int con la edad de la persona en años.
 - # tiene_licencia: Booleano que indica si la persona tiene licencia de conducir.
 - # hobbies: Conjunto de str con los hobbies o actividades que le gustan a la persona.
 - # paises_visitados: Conjunto de str con los países que la persona ha visitado.
 - # libros_leidos: Lista de str con los libros que la persona ha leído en orden de lectura.
- Persona = namedtuple('Persona', ['nombre', 'edad', 'tiene_licencia', 'hobbies', 'paises_visitados', 'libros_leidos'])
- a) Si tenemos dos variables p1 y p2 de tipo Persona, escriba una expresión lógica que devuelva si es cierto que el último libro que ha leído p1 es “Romeo y Julieta” y p2 no lo ha leído aún.
- b) Si tenemos un conjunto de tuplas de tipo Persona, escriba la expresión que muestre en la consola la suma de las edades de las personas que tienen más de 3 hobbies y su nombre comienza por ‘A’
- c) Si p1 y p2 son de tipo Persona, escriba una expresión lógica que sea verdadera si tanto p1 como p2 han leído “Romeo y Julieta”, y han nacido en años distintos.

Funciones

Tenemos un fichero csv con la información de automóviles de segunda mano en venta. Como el siguiente

1234XYZ, Ford,23/10/2021,21620,34328.7
0987ABC, Seat,18/10/2022,12230,26245.9
...

Los datos son matrícula de tipo str, nombre de la marca de tipo str, fecha de tipo date, kilometraje de tipo int y precio de tipo float

1. Define un namedtuple Automovil y una función de lectura para dicho fichero que devuelva una lista de tuplas. Chequee que las matrículas son 4 dígitos seguidos de tres letras, que las fechas son posteriores al año 2020 incluido y que los kilómetros son positivos.
Nota: la invocación cadena.isdigit() de Python devuelve True si todos los caracteres de cadena son dígitos.
Igualmente, la función isalpha devuelve True si todos los caracteres de la cadena que invoca son alfabéticos.
2. Implemente una función que recibe una lista de tuplas de tipo Automovil y devuelva una lista ordenada de las marcas de la lista sin repetir. La lista estará ordenada por longitud de caracteres del nombre de la marca.
3. Implemente una función media_kilometros que reciba una lista de tuplas de tipo Automovil y un año y devuelva la media de kilómetros de los coches cuya fecha de matriculación coincide con el año indicado. Si no hay coches del año indicado, la función devuelve None.