

In [7]:

```

1  #añadir datos
2  rock=["cancion1","cancion2","cancion3"]
3  bachata=["cancion4","cancion5","cancion6"]
4  jazz=["cancion7","cancion8"]
5
6  playlist=rock+bachata
7
8  print(playlist)
9  print("_____")
10
11 playlist.append("ejm_append") #append agrega un nuevo item al final de la Li
12 print(playlist)
13 print("_____")
14
15 #playlist.append(jazz) #pasa de ser lista a ser matriz porque append agrega
16 #otra lista
17 #print(playlist)
18 #print("_____")
19
20 playlist.extend(jazz) # extend me agrega los elementos uno por uno
21 print(playlist)
22 print("_____")
23
24 playlist.extend("cancion8")# cuando agrego una sola cosa con extend, la agre
25 print(playlist)
26
27
28
29

```

```
['cancion1', 'cancion2', 'cancion3', 'cancion4', 'cancion5', 'cancion6']
```

```
['cancion1', 'cancion2', 'cancion3', 'cancion4', 'cancion5', 'cancion6', 'ejm_a
ppend']
```

```
['cancion1', 'cancion2', 'cancion3', 'cancion4', 'cancion5', 'cancion6', 'ejm_a
ppend', 'cancion7', 'cancion8']
```

```
['cancion1', 'cancion2', 'cancion3', 'cancion4', 'cancion5', 'cancion6', 'ejm_a
ppend', 'cancion7', 'cancion8', 'c', 'a', 'n', 'c', 'i', 'o', 'n', '8']
```

In [20]:

```

1  #Eliminar datos de La lista
2  rock=["cancion1","cancion2","cancion3","cancion4","cancion5"]
3  print(rock[1])
4  del rock[1] # del -> delete -> eliminar
5  print(rock[1])
6  print(rock)
7  # .pop() -> pop, revisa cual es el ultimo valor, lo retorna y lo elimina
8  ultimo_item=rock.pop() # ctrl-z toma el ultimo elemento
9  print(ultimo_item)
10 print(rock)
11 print("_____")
12 ultimo_item=rock.pop(0) # el argumento nos dice a que posicion hacer el pop
13 print(ultimo_item)
14 print(rock)
15 print("_____")
16 rock.remove("cancion4") # Elimina haciendo una busqueda por nombre o element
17 print(rock)
18 print("_____")
19 aux_list=[8,5,6,8,7,4,8,6]
20 rock.extend(aux_list*3) # podemos utilizar mutiplicadores para repetir un p
21 print(rock)
22 print("_____")
23 rock.remove(8) # si existen elementos repetidos solo borra el primero
24 print(rock)
25 print("_____")

```

cancion2

cancion3

['cancion1', 'cancion3', 'cancion4', 'cancion5']

cancion5

['cancion1', 'cancion3', 'cancion4']

cancion1

['cancion3', 'cancion4']

['cancion3']

['cancion3', 8, 5, 6, 8, 7, 4, 8, 6, 8, 5, 6, 8, 7, 4, 8, 6, 8, 5, 6, 8, 7, 4, 8, 6]

['cancion3', 5, 6, 8, 7, 4, 8, 6, 8, 5, 6, 8, 7, 4, 8, 6, 8, 5, 6, 8, 7, 4, 8, 6]

```
In [22]: 1 # slicing -> particionar, es recortar una lista en rangos definidos
2 print(list(range(0,10,1))) # range(inicio,final,paso)
3
4
5 aux_list=[8,5,6,8,7,4,8,6]
6 lst_recortada=aux_list[1:5]# [inicio:final:paso]
7 print(lst_recortada)
8 print("_____")
9
10 lst_recortada=aux_list[::3]# [inicio:final:paso] # nomenclatura diferente
11 print(lst_recortada)
12 print("_____")
13
```

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

[5, 6, 8, 7]

[8, 8, 8]

```
In [31]: 1 # saber si la lista tiene un elemento
2
3 # for i in lstMercado:
4 aux_list=[8,5,6,8,7,4,8,6]
5
6 pregunta1= 8 in aux_list #tengo en stock / bodega / existencias del producto
7 pregunta2= 1 in aux_list
8
9 print(pregunta1,pregunta2)
10
11 print(aux_list)
12
13 while (pregunta1 == True):
14     pregunta1= 8 in aux_list
15     if (pregunta1 == True):
16         aux_list.remove(8)
17         print(aux_list)
18
```

True False

[8, 5, 6, 8, 7, 4, 8, 6]

[5, 6, 8, 7, 4, 8, 6]

[5, 6, 7, 4, 8, 6]

[5, 6, 7, 4, 6]

```
In [32]: 1 person1=["jose",28,0,170.00,234]
2 person2=["luis",25,0,175.00,435]
3 person3=["jorge",21,1,181.00,123]
4 person4=["laurent",17,1,160.00,789]
5 asistentes=person1+person2+person3+person4
6 print(asistentes)
7 n_person=(len(asistentes)//5) ### division exacta -> int
8 age_average=0
9 age_sum=0
10 for age in asistentes[1::5]:
11     age_sum+=age
12 age_average= age_sum // n_person
13 print(n_person)
14 print("la edad promedio es : {}".format(age_average))
```

```
['jose', 28, 0, 170.0, 234, 'luis', 25, 0, 175.0, 435, 'jorge', 21, 1, 181.0, 1
23, 'laurent', 17, 1, 160.0, 789]
4
la edad promedio es : 22
```

```
In [48]: 1 #Otras funciones utiles
2
3 a_list=[1,2,3,4,5,6,7,8,9]
4 print(len(a_list)) #me retorna la longitud -> La cantidad de elementos
5 print(max(a_list)) # retorna el maximo
6 print(min(a_list)) # retorna el min
7 print(sum(a_list)) # retorna la suma
8
9 b_list=["dr","ds","dc","da"] #toma los valores del codigo ASCII
10 # compara los valores de izq a derecha
11 print(len(b_list)) #me retorna la longitud -> La cantidad de elementos
12 print(max(b_list)) # retorna el maximo
13 print(min(b_list)) # retorna el min
14
15 b_list.sort()
16 print(b_list)
17 #100-d
18 #97-a
19 #114-r
20
```

```
9
9
1
45
4
ds
da
['da', 'dc', 'dr', 'ds']
```

```
In [56]: 1 #mas funciones utiles
2 l=[1,2,3,1,1,1,2,3,1,3]
3 print(l.count(1)) # count cuenta cuantas veces esta el elemento
4 print(l.count(2))
5 l.insert(4,"test") # inserta en insert(coordena,valor)
6 print(l)
7 l.clear() # borra todo
8 print(l)
```

```
5
2
[1, 2, 3, 1, 'test', 1, 1, 2, 3, 1, 3]
[]
```

```
In [40]: 1 # sort -> ordenar
2 a_list=[5,6,2,8,1,3,4,9,7]
3 b_list=["c","j","a","d","e"]
4 a_list.sort() # ascendente
5 print(a_list)
6 a_list.sort(reverse = True) #descendente
7 print(a_list)
8 print("_____")
9 b_list.sort() # ascendente
10 print(b_list)
11 b_list.sort(reverse = True) #descendente
12 print(b_list)
```

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

['a', 'c', 'd', 'e', 'j']
['j', 'e', 'd', 'c', 'a']
```

```
In [ ]: 1 # recordatorio
2 tup=(5,3,2,6,1,4)
3 #si yo tengo una tupla, y la quiero editar OBLIGATORIAMENTE debo convertirla
4 aux_list=list(tup)
5 aux_list.sort()
6 print(aux_list)
7
8
9
```

```
In [53]: 1 #Diccionarios
2 profe1={"nombre":"jose","edad":"28","codigo":"3"}
3 print(profe1["nombre"])
4 np1=profe1.get("nombre") #obtiene el valor correspondiente a la clave
5 profe1.items() #obtiene todo el diccionario
```

```
jose
```

```
Out[53]: dict_items([('nombre', 'jose'), ('edad', '28'), ('codigo', '3')])
```

