

In [3]:

```

1 rock=["cancion1","cancion2","cancion3"]
2 bachata=["cancion4","cancion5","cancion6"]
3 pop=["cancion7","cancion8"]
4
5 playlist=rock+bachata
6 print(playlist)
7 print("_____")
8 playlist.append("ejem_append") #append agrega un nuevo item al final de la l
9 #playlist.append(pop)
10 print(playlist)
11 print("_____")
12
13 playlist.extend(pop)
14 print(playlist)
15 print("_____")
16
17 playlist.extend("cancion8")# cuando agrego una sola cosa con extend, la agre
18 print(playlist)
19
20 #Resumen:
21 #si quiero agregar un solo elemento a la lista lo agrego con append
22 #si quiero agregar otra lista a mi lista lo hago con extend (porque asi me u

```

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

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

```
['cancion1', 'cancion2', 'cancion3', 'cancion4', 'cancion5', 'cancion6', 'ejem_
append', 'cancion7', 'cancion8']
```

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

In [4]:

```

1 rock=["cancion1","cancion2","cancion3"]
2 del rock[1]#delete-> borrar desde la coordenada
3 print(rock)

```

```
['cancion1', 'cancion3']
```

In [6]:

```

1 rock=["cancion1","cancion2","cancion3"]
2 #.pop() -> toma el ultimo valor, lo elimina y lo retorna
3 ultima_cancion=rock.pop() #deshacer -> ctrl+z -> para listas
4 print(ultima_cancion)
5 print(rock)

```

```
cancion3
```

```
['cancion1', 'cancion2']
```

```
In [11]: 1 rock=["cancion1","cancion2","cancion3", "cancion2"]
2         #.remove() -> remover un elemento especifico,OJO! solo elimina el primero de
3         rock.remove("cancion2") #
4         print(rock)

['cancion1', 'cancion3', 'cancion2']
```

```
In [18]: 1 #slicing -> es recortar una lista en rangos definidos
2         print(list(range(0,10,1)))
3
4         a_list=[1,2,3,4,5,6,7,8,9]
5         lst_recortada=a_list[1:5]
6         print(lst_recortada)
7
8         lst_recortada2=a_list[::5]# nomenclatura diferente por ejemplo a_list[0:9:5]
9         print(lst_recortada2)

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

```
In [30]: 1 #saber si la lista contiene un elemento
2         a_list=[1,2,3,4,5,6,7,8,9]
3         pregunta1 = 8 in a_list
4         pregunta2 = 12 in a_list
5         print(pregunta1,pregunta2)
6
7         for n in a_list:
8             print(n, end=" ")
9
10        if (pregunta2 == False ):
11            a_list.append(12)
12
13        if (pregunta1 == True ):
14            a_list.remove(1)
15
16        print()
17        print(a_list)
```

```
True False
1 2 3 4 5 6 7 8 9
[2, 3, 4, 5, 6, 7, 8, 9, 12]
```

```
In [40]: 1 person1=["jose",28,0,170.00,234]
2 person2=["luis",25,0,175.00,435]
3 person3=["yudivith",18,1,165.00,123]
4 person4=["sara",21,1,165.00,789]
5
6 asistentes=person1+person2+person3+person4
7
8 n_person=(len(asistentes)//5)
9 age_average=0
10 age_sum=0
11 for age in asistentes[1::5]:
12     age_sum+=age
13 age_average= age_sum // n_person
14 print(n_person)
15 print("la edad promedio es : {}".format(age_average))
```

4
la edad promedio es : 23

```
In [ ]: 1 a_list=[1,2,3,4,5,6,7,8,9]
2 print(len(a_list)) #me retorna La Longitud -> La cantidad de elementos
3 print(max(a_list))
4 print(min(a_list))
5 print(sum(a_list))
6
7
8 hay_cosas_en_a=any(a_list) # pregunta si hay elementos en la lista
9 print(hay_cosas_en_a)
10 b_list=[0,''] # creo una lista vacia, porque se necesitara mas adelante
11 print(b_list)
12 any(b_list)
13
14
15 #for n in b_list:
16 #     b_list.append("sgXXX") # se que est lo necesitara mas adelante
17 #print(n, end=" ")
18
19 c.append(7) # cuando la lista no existe, append la crea
```

```
In [56]: 1 a_list=[1,2,3,4,6,7,8,9]
2 a_list.insert(4,5) # inserta en insert(coordena,valor)
3 print(a_list)
```

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

```
In [58]: 1 #sort organiza
2 a_list=[5,6,2,8,1,3,4,9,7]
3 a_list.sort() # ascendente
4 print(a_list)
5 a_list.sort(reverse = True) #descendente
6 print(a_list)
7
```

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

```
In [59]: 1 tup=(5,3,2,6,1,4)
2 #si yo tengo una tupla, y la quiero editar OBLIGATORIAMENTE debo convertirla
3 aux_list=list(tup)
4 aux_list.sort()
5 print(aux_list)
```

```
[1, 2, 3, 4, 5, 6]
```

```
In [67]: 1 l=[1,2,3,1,1,1,2,3,1,3]
2 print(l.count(1)) # count cuenta cuantas veces esta el elemento
3 print(l.count(2))
4 l.clear() # borra todo
5 print(l)
```

```
5
2
[]
```

```
In [75]: 1 profe1={"nombre":"jose","edad":"28","codigo":"3"}
2 print(profe1["nombre"])
3 np1=profe1.get("nombre") #obtiene el valor correspondiente a la clave
4 profe1.items() #obtiene todo el diccionario
5
```

```
jose
```

```
Out[75]: dict_items([('nombre', 'jose'), ('edad', '28'), ('codigo', '3')])
```

```
In [ ]: 1
```