

Paltario ayudantia4

Referencia

<https://docs.python.org/2/tutorial/datastructures.html>

1 Tuplas

Las tuplas no son editables en el tiempo, una vez definidas quedaron así para siempre :'. La verdad yo no las recomiendo mucho

```
a = (1,2,3)          #Siento "a" y "b" tuplas
q,w,e = a            #Define una tupla
a[0]                 #Ahora q=1 , w=2 ,e=3  "Desempaquetar" tupla
sum(a)               #Retorna lo que hay en la primera posicion
len(a)               #Suma los valores de la tupla
b = ("hola",5,1)     #Retorna el largo de la tupla
"hola" in b          #Se pueden mezclar datos
                    #Retorna verdadero si es que esta dentro de la tupla
```

2 Listas

```
#Siendo "l" una lista
l = [1,"asd"]        #También puedes mezclar datos
l = [ (1,3) , (5,6) , "hola", ] #Cualquier estructura es aceptada
l = []               #Crea una lista vacia
l.append(X)           #Agrega x a la lista (en ultima posicion)
l.insert(i,x)         #Inserta x en la posicion i
l.remove(x)           #Elimina el elemento x de la lista
l.pop()               #Remueve el último y lo retorna
l.pop(i)              #Remueve el item en la posición "i" y lo retorna
l.index(x)             #Busca X en la lista y retorna su posicion
l.count(x)             #Retorna la cantidad del elemento X
l.sort(x)              #Ordena la lista de menor a mayor
a = [1,2,3]
b = [-4,6,-7]
c = [-5] + a + b + [5,7,8] #Las listas se pueden sumar y se "concatenaran"
                        #C valdra [-5,1,2,3,-4,6,-7,5,7,8]
```

3 Operaciones utiles y ejemplos varios

```
(1,2,3)==(1,2,3)     #Las tuplas se pueden comparar
(1,2,3)>(1,2,4)        #Comparaciones posicion por posicion
[1,2,3]==[1,2,3]      #Las comparaciones con listas y tuplas son la misma
[1,2,3]>[1,2,4]
[3,2,1]>[3,3,0]
```

3.1 IMPORTANTE: PUEDEN COMPARAR FECHAS FACILMENTE ASI

```
a = [2018,08,13]    #Deben preocuparse que esté en formato anio-mes-dia
b = [2018,10,4]
a > b               #Retornara False
```

4 El intento mas detallado de hacer un ciclo for que pude hacer

El ciclo for es casi igual que un while solo que mas ordenado y poderoso

```
range(5)              #Retorna una lista [0,1,2,3,4]
range(1,5)            #Retorna una lista [1,2,3,4]
range(5,23,3)         #[5,8,11,14,17,20]
range(5,24,3)         #[5,8,11,14,17,20,23]
range(1,-1,-1)        #[1,0] desde el 1 hasta el -1 avanzando -1
range(a,b,c)          # Numeros entre [a,b[ de C en C
for numero in [0,1,2,3,4]: #Numero cada ciclo tomara los valores 0,1,2...
    print numero
for k in range(5):      #Exactamente lo mismo que antes
    print k
for k in ["me","echare","los","ramos"]: #Cualquier lista es valida
    print k
```

5 Todo el poder del for,listas y tuplas

5.1 Ejemplo1

```
mensajes = ["hola","adios"]
personas = ["mauro","jose","mario","ayudante"]
for m in mensajes:
    for p in personas:
        print m,p
```

El programa terminará imprimiendo

```
hola mauro
hola jose
hola mario
hola ayudante
adios mauro
adios jose
adios mario
adios ayudante
```

5.2 Ejemplo2

Imagina que necesitas sacar todas las posibles combinaciones de dados que la suma sea igual a 7

```
for i in range(1,7):
    for k in range(1,7):
        if (i+k==7):
            print i,k,"Suman 7"
```