Paltario Certamen2 v2

Referencia

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

1 Cosas para Tuplas y Listas

Sea una lista L y una tupla T

```
q,w,e = T                       #Ahora q=1 , w=2 ,e=3   "Desempañetar" tupla o lista
q,w,e = L
sum(L) o sum(T)                 #Entrega la suma
len(L) o len(T)                 #Entrega el largo
max(L) min(L) o max(T) min(T)   #Entrega... eso
T[0], L[0]                      #Primera posicion de la tupla o lista
"hola" in T                     #Ve si "hola" es un elemento de la tupla
"chao" in L                     #Ve si "chao" es un elemento de la lista
```

2 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)         #Define una tupla
b = ("hola",5,1)    #Se pueden mezclar datos
```

3 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 = list(otralista)             #MANERA CORRECTA DE COPIAR UNA LISTA
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()                        #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]
```

4 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]
```

4.1 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
```

5 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
```

6 Todo el poder del for,listas y tuplas

6.1 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
```

6.2 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"
```

6.3 Ejemplo3 EL SUPER FOR

```
    a = [(0,1,2),(3,4,5),(5,6,9)]   #una lista de tuplas
    for a,b,c in a:                 #dentro del mismo for pueden descomponer
        print a,b,c
    #abajo esta mas explicado
    for a,b,c in [(0,1,2),(3,4,5),(5,6,9)]:
        print a,b,c
    #printeara 0,1,2 luego 3,4,5 y luego 5,6,9
    #Es bastante agradable para leer tuplas en una lista
```

7 Diccionarios

Diccionarios es una forma de organizar información mediante una llave

```
dix = dict()                                #crea un diccionario vacio
dix = {'carl':123,'rick':3,'fabian':343}    #define un diccionario
dix[1] = "uno"                              #Llave = 1 ; contenido = "uno"
dix["uno"] = 1                              #llave = "uno" ; contenido = 1
dix.values()                                # retorna ['uno', 343, 3, 123, 1]
dix.items()                                 #[(1, 'uno'), ('fabian', 343),
                                            ('rick', 3), ('carl', 123), ('uno', 1)]
dix.keys()                                  #retorna [1, 'fabian', 'rick', 'carl', 'uno']
list(dix)                                   #Lista de keys igual a .keys()
'rick' in dix                               #retorna True
len(dix)                                    #retorna cantidad de elementos en el diccionario
```

Por ejemplo imagina que tienes un diccionario lleno de numeros telefonicos donde la llave es el nombre de la persona, podemos iterar dentro del diccionario (SIN NINGUN ORDEN) y tomaremos el valor de cada llave

```
dix = {'carl':1111,'fabian':2222,'nombre2':333,'mauro':45454}
for k in dix:
    print "La llave es",k,"y su valor es:",dix[k]
```

8 Otros

En caso de que no puedan usar la funcion count(x) pueden implementarlo y es muy facil lul

```
def contar(L,x):
    c = 0
    for k in L:
        if x==k:
            c+=1
    return c
a = [1,2,3,4,5,5,6,7,7,7,8,8,9]
print contar(a,2)       #retorna 1
print contar(a,7)       #retorna 3
```

Los superpoderes del min y el max

Creais que el MIX Y MAX solo sería para encontrar el mayor entre muchos enteros?! PUES YA NO MAS IT WAS ME DIO. Al igual que como haces las comparaciones entre numeros 8¿9 es como se busca el max/min, y al igual que como comparas tuplas y listas [1,2]¿[5,6] o bien (10,4) ¡ (10,5) tambien se puede buscar el maximo o el minimo entre ellos ¿Para que me puede servir?. Pos para encontrar una fecha mas reciente o mas antigua

```
#Deben estar en formato anio-mes-dias (tal y como si las compararas
fechas = [(2018,10,8),(2012,8,5),(2020,2,1),(2008,2,25)]
print max(fechas) #A que no me crees que realmente funciona
```