

Prog. Avanzada – Listas, Diccionarios, Tuplas, Funciones, Conjuntos

Las listas, tuplas y diccionarios son objetos iterables.

Crear lista vacía lista = [] lista = list()	Crear diccionario vacío dic = {} dic = dict()	Crear tupla vacía - Inmutable tupla = () tupla = tuple()
Lista con elementos lista = [1,2,0,3j,'Hola',[1,2],[3,4]]	Diccionario con llaves y valores dic = {1: 20,'a': 'Hola', 2.0:{9:1}}	Tupla con valores tupla = (1,2,3)
Acceder elemento de la lista lista = [1,2,0,3j,'Hola',[1,2],[3,4]] a = lista[0] #retorna 1, a=1. b = lista[-1] #retorna (3,4)	Acceder elemento del diccionario dic = {1: 20,'a': 'Hola', 2.0:{9:1}} a = dic[1] #retorna 20, a = 20 b = dic[2.0] #retorna diccio. {9: 1}	Acceder elementos de tupla tupla = ([1,2],3.0,'4') a = tupla[0] #retorna [1,2] b = tupla[-1] #retorna '4'
Rango de índices en una lista lista = [1,2,0,3j,'Hola',[1,2],[3,4]] lista[2:] #retorna ['Hola',[1,2],[3,4]] lista[inicio:fin:paso]	Iterar sobre las llaves dic={1: 20,'a': 'Hola', 2.0:{9:1}} for llave in dic.keys(): print(llave) #Imprime las llaves: 1, 'a', 2.0	Rango de índices en una tupla tupla = ([1,2],3.0,'4') tupla[1:] #retorna (3.0,'4') tupla[inicio:fin:paso]
Iterar sobre elementos de la lista lista = [1,2,3,4] for elem in lista: print(elem) #Imprime 1 2 3 4	Iterar sobre los valores dic={1: 20,'a': 'Hola', 2.0:{9:1}} for val in dic.values(): print(val) #Imprime valores: 20,'Hola',{9:1}	Iterar sobre elementos de la tupla tupla = (1,2,3,4) for elem in tupla: print(elem) #Imprime 1 2 3 4
Método append de las listas lista = [] lista.append(1) #lista = [1] lista.append([2,3]) #lista = [1,[2,3]]	Iterar sobre las llaves y valores dic={1: 20,'a': 'Hola', 2.0:{9:1}} for key,val in dic.items(): print(key,' ',val) #Imprime llaves y valores: 1 20, # 'a' 'Hola', 2.0 {9:1}	Método count de las tuplas: cuenta la cantidad de veces que aparece un valor tupla = (1,2,3,1,1) tupla.count(1) #retorna 3 tupla.count(2) #retorna 1
Método index de las listas: retorna el índice donde está el numero: lista = [5,3,3,2,1] lista.index(3) #retorna 1	Crear una función def nombreFun(par1,par2): return par1+par2	Método index de las tuplas: retorna el índice donde está el numero: tupla = (5,4,3,2,1) tupla.index(3) #retorna 2
Método fromkeys de diccionario Crea un nuevo diccionario con las llaves de un iterable y a los valores les asigna un valor llaves = ['1',2,3.0] valor = 0 dic = {} dic.fromkeys(llave,valor) dic => {'1': 0,2: 0,3.0: 0}	Método get de los diccionarios: retorna el valor de la llave si la llave está en el diccionario. De lo contrario, devuelve el segundo argumento o None por defecto dic={1: 20,'a': 'Hola', 2.0:{9:1}} dic.get(3) #retornar None dic.get('b',0) #retorna 0 dic.get(1,0) #retorna 20	Función que recibe cantidad finita de argumentos def argsFinito(*params): for par in params: print(par) argsFinito(1,2,3,4,5) #Imprime 1 2 3 4 5
Función que recibe otra función def llamaFun(fun,a,b): return fun(a,b) llamaFun(lambda x,y: x*y,1,2) #retorna 1*2=2	Conjuntos: set Pueden operar sobre listas y tuplas. Retorna los elementos sin repetir. set([1,2,2,3,1,3]) #retornar {1,2,3} set((2,1,2,1,2,1)) #retornar {1,2}	Ayuda en Python Para ver todos los métodos de un objeto usar: dir(objeto) Para pedir ayuda: help(objeto.metodo)