



Clase 3 Listas I

≡ TP N°2: Listas (parte 1)



Listas. Empaquetado y desempaquetado. Operaciones. Métodos.
Números al azar

Teoria

Acceder a los elementos de la lista

Para acceder a los elementos de una lista se utiliza un subíndice:

```
a = numeros[0]
'''El primer elementos de la lista siempre
lleva el subindice 0'''

'''Usar un subindice negativo hace que la cuenta comience des
de atras'''
[5 , 7, 4, 4, 5, 6, 2, 6, 1]
 0  1  2  3  4  5  6  7  8
 -9 -8  -7 -6 -5 -4 -3 -2 -1

'''Si se utiliza subindice fuera de rango provocara
un error'''
```

Empaquetado

Empaquetar consiste en asignar un conjunto de constantes o variables a una lista:

```
n1 = 5  
n2 = 9  
n3 = 4  
numeros = [n1 , n2 , n3]
```

Desempaquetado

Desempaquetar consiste en asignar los elementos de una lista a un conjunto de variables:

```
dias = ["lunes","martes","jueves"]  
d1,d2,d3 = dias
```

Operaciones con listas

Concatenación

```
"Las listas puede concatenarse con el operador +:"  
lista1 = [1,2,3]  
lista2 = [4,5,6]  
lista3 = lista1 + lista2  
print(lista3) #[1,2,3,4,5,6]  
  
'''Concatenacion permite agregar nuevos elementos  
en una lista:'''  
lista = [3,4,5]  
lista = lista + [6] #[3,4,5,6]  
'''El elemento debe encerrarse entre corchetes para que sea c  
onsiderado como una lista'''
```

Replicación

```
'''Tambien pueden ser replicadas  
mediante el asterisco'''  
listal = [3,4,5]  
listal = listal*3  
print(listal) #[3,4,5,3,4,5,3,4,5]  
lista2 = [0] * 5 ---> para matrices  
print(lista2) #[0,0,0,0,0]
```

Agregar elementos

```
'''Otra manera de agregar elementos consiste en usar  
el metodo append:'''  
lista = [3,4,5]  
lista.append(6)  
print(lista) #[3,4,5,6]
```

| las listas son mutables las variables no

Modificar

```
'''Por ser secuencias mutables , los  
elementos de una lista puede ser  
modificados a traves del subdice:'''  
lista = [3,4,5]  
lista[1] = 7  
print(lista) #[3,7,5]
```

EJEMPLO 1

Leer un conjunto de números enteros, calcular promedio e imprimir valores mayores a promedio.

```

lista = []
suma = 0
cant = 0
n = int(input("Ingrese un numero entero o -1 para terminar"))
while n!= -1:
    lista.append(n)
    suma = suma + n
    cant = cant + 1
    n = int(input("Ingrese un numero entero o -1 para terminar"))
if cant == 0 :
    print("No se ingresaron valores")
else:
    prom = suma/cant
    print("Promedio" , prom)
    for i in range(cant):
        if lista[i] > prom:
            print(lista[i], end=" ")
    print()

```

Operaciones con listas

Función Len

```

'''La funcion len() devuelve la cantidad de
elementos de una lista'''
lista = [3,4,5,6]
print(len(lista)) #4

```

Función sum

```
'''La funcion sum() devuelve la suma de los elementos de la lista.'''
lista = [3,4,5,6]
print(sum(lista)) #18
#La lista debe contener numeros,
```

EJEMPLO 2

Crear grafico de barras con los porcentajes obtenidos por cada candidato en una elección

```
votos = []
n = int(input("Votos del candidato? -1 para terminar"))
while n!= -1:
    votos = votos + [n] #votos.append(n) seria lo mismo
    n = int(input("Votos del candidato? -1 para terminar"))
print()

#Calcular porcentajes e imprimir listado final
total = sum(votos)
for i in range(len(votos)):
    porcentaje = votos[i] * 100 /total
    print("Candidato %d: %d votos (%.2f%%)" %
          (i,votos[i] , porcentaje) , end = " ")
#Imprimir grafico de barras
for j in range(int(porcentaje/10)):
    print("*" , end = "")
print()
```

Función max y min

```
'''La funcion max() devuelve el mayor de los elementos de la lista'''
lista = [4,6,3,5]
```

```

print(max(lista)) #6
#La lista debe contener elementos homogeneos.
m = [2,5,89,9]
p = [56,362,4,5,1]
x = [23,36,5,2,4,8]
print(max(m,p,x)) #[56, 362, 4, 5, 1]
#de tres lsitas la que contiene el primer nro más grande

'''La funcion min() devuelve el menor de los elementos de la
lista.'''
lista = [4,6,3,5]
print(min(lista)) #3
#La lista debe contener elementos homogeneos.

'''Tambien pueden ser utilizadas con un conjunto de valores,
constantes o variables'''
mayor = max(4,1,7,2)
print(mayor) #7

```

Función List

```

'''La funcion list() convierte cualquier iterable en una list
a'''
lista = list(range(5))
print(lista) #[0,1,2,3,4]
'''Se puede usar con rangos , cadenas , tuplas , conjuntos ,
etc '''

```

Operador in

```

'''El operador in permite verificar la presencia de un elemen
to'''
'''Devuelve True o False'''
lista = [3,4,5,6]

```

```

if 4 in lista:
    print("Hay un 4 en la lista")

'''La ausencia de un elemento se comprueba con not in'''
'''Devuelve True o False'''
lista = [3,4,5,6]
if 7 not in lista:
    print("No hay un 7 en la lista")

```

EJEMPLO 3

Escribir función que reciba como parámetro dos números correspondientes al mes y año de una fecha y devuelva cuantos días tiene ese mes en ese año.

```

def obtenerdias(m,a):
    if m in [1,3,5,7,8,10,12]:
        dias = 31
    elif m in [4,6,9,11]:
        dias = 30
    elif m == 2:
        if (a %4 == 0 and a % 100 != 0) or a % 400 == 0:
            dias = 29
        else:
            dias = 28
    else:
        dias = -1 #inválido
    return dias

#pp---
mes = int(input("Ingrese el mes --- "))
anio = int(input("Ingrese el año --- "))
dact = int(input("La fecha actual del mes ingresado es: "))
diasm = obtenerdias(mes,anio)
print()
print("El mes %d tiene %d dias." %(mes,diasm))

```

```
faltan = lambda x,y: (x-y)
print("Días para fin de mes:", faltan(diasm,dact))
```

Métodos

▼ Métodos de agregamiento

```
'''El metodo append(<elem>) agregar un elemento al final de lista'''
lista = [3,4,5]
lista.append(6)
print(lista) #[3,4,5,6]
```

```
'''El metodo insert(<pos> , <elem>) inserta un elemento en la lista , en una posicion determinada'''
lista = [3,4,5]
lista.insert(2,9)
print(lista) #[3,4,9,5]
```

▼ Métodos de eliminación

```
'''El metodo pop(<pos>) eliminar y devuelve un elemento de la lista, identificado por su posicion. Si la posicion se omite se elimina el ultimo elemento de la lista'''
lista = [3,4,5]
elem = lista.pop(1)
print(elem) #4
print(lista) #[3,5]
#Da un error si la pos esta fuera de rango
```

```
'''El metodo remove(<elem>) elimina la primera aparicion de un elemento en la lista , identificado por su valor'''
lista = [3,4,5,4]
lista.remove(4) #queda [3,5,4]
```

```
#Provoca un error si no existe

'''ELIMINA TODAS LAS APARICIONES DE UN NRO EN UNA LISTA'''
myList = [2, 1, 3, 5, 1, 1, 1, 0]
while 1 in myList:
    myList.remove(1)
print(myList)
```

```
'''El metodo clear() elimina in situ* todos los elementos
de la lista.'''
lista = [3,4,5]
lista.clear()
print(lista) #[]
```

▼ Métodos de búsqueda

```
'''El metodo index(<elem>) busca un elemento y devuelve su
posicion'''
lista = [3,4,5]
print(lista.index(5)) #2
#Provoca un error si no lo encuentra.
```

```
'''El metodo index() tambien permite elegir la region de b
usqueda.'''
print(lista.index(5,2)) #Inicio
print(lista.index(5,2,4)) #Inicio y fin
#El valor final no esta incluido en el intervalo de busque
da
```

```
'''El metodo count(<elem>) devuelve la cantidad de repetic
iones de un elemento.'''
lista = [3,4,5,3]
```

```
print(lista.count(3)) #2  
#Devuelve 0 si no lo encuentra
```

▼ Métodos de ordenamiento

```
'''El metodo reverse() invierte in situ el orden de los elementos de la lista'''  
lista = [3,4,5]  
lista.reverse()  
print(lista) #[5,4,3]
```

```
'''El metodo sort() ordena in situ los elementos de la lista.'''  
lista = [5,1,7]  
lista.sort()  
print(lista) #[1,5,7]
```

```
"""-----Otro caso-----"""\n'''El parametro reverse = True hace que se ordene de mayor a menor.'''\nlista = [5,1,7]\nlista.sort(reverse = True)\nprint(lista) #[7,5,1]
```

```
"""-----Otro caso-----"""\n'''El parametro key = <clave> permite establecer el criterio de ordenamiento cuando este no sea el valor del item'''  
lista = [1,2,3,4,5,6]  
lista.sort(key = lambda x: x%2)  
print(lista) #[2,4,6,1,3,5]
```

```
"""-----Otro caso-----"""\n'''Si se utilizan funciones incorporadas no es necesario crear una funcion lambda'''
```

```

nombres = ["Andres","Ariel","Juan"]
nombres.sort(key=len) #Ordena por el largo
print(nombres)#["Juan","Ariel","Juan"]
numeros = [3,-2,4,-1]
numeros.sort(key=abs)#Ordena valor absoluto
print(numeros)[-1,-2,3,4]

"""-Otro caso-"""
'''Cuando se necesite ordenar una lista por mas de un atributo es necesario crear una lista con esos atributos, la que sera utilizada como clave'''
numeros = [9,6,7,8,1,5,4]
numeros.sort(key=lambda x:[x%2,x])
print(numeros) #[4,6,8,1,5,7,9]

'''Funcion Sorted'''
'''Ordena cualquier iterable'''
'''(rango,lista,string,tupla,conjunto)'''
numeros = [9,6,7,8,1,5,4]
ordenada = sorted(numeros)
'''Tambien admite parametros reverse y key'''
'''En todos los casos sorted() devuelve una lista con elementos ordenados'''

```

▼ Números al azar

```

'''Numeros generados o inventados por la computadora, son utilizados cuando se requiere un factor de azar por ejemplo videojuegos, criptografia o simulacion de eventos'''
'''Las funciones que se utilizan vienen del modulo random'''

        "import random"
"-----"
random.random():devuelve un numero real al azar dentro dell intervalo [0,1]

```

```

"-----"
random.randint(<min>,<max>):devuelve un
numero entero al azar dentro del intervalo
dados. El intervalo se considera cerrado
"-----"
random.choice(<secuencia>):devuelve un elemento elegido al
azar dentro de una
secuencia pasada como parametro.

import random
nombre = ["delfi", "teo", "lali", "nacho"]
opciones = ["Piedra","Papel","Tijera"]
for nom in range(len(nombre)):
    print(f"{nombre[nom]} --- {random.choice(opciones)}")

"-----"
situacion = random.choice(opciones)
'''La secuencia puede ser una lista, un string, una tupla,
un rango,etc.'''
"-----"
random.shuffle(<lista>): mezcla in situ los elementos de u
na lista, es decir que altera la posicion de los mismos.
lista = [3,4,5,6]
random.shuffle(lista)
print(lista) #ej [6,4,5,3]
"-----"

```

EJEMPLO 4

Desarrollar función que simule lanzamiento de cinco dados de un cubilete.
Escribir un programa para verificar su funcionamiento

```
import random

def lanzadados(cuantos):
    dados = []
    for i in range(cuantos):
        dados.append(random.randint(1,6))
    return dados

#Programa Principal
jugada = lanzadados(5) #Cinco dados
print(jugada) #[5,1,2,6,3]
```