

TEL-101 Ayudantía 3
Vicente Llanos Á - Thomas Soto F
(vicente.llanos@usm.cl - thomas.soto@usm.cl)
5 de mayo de 2021

1. Listas

1. Introducción a las Listas

Una lista es una estructura y un tipo de dato en Python con características especiales. Lo bueno de las listas en Python es que nos permiten almacenar cualquier tipo de valor, ya sea enteros, cadenas y hasta otras funciones.

2. Funciones propias de las listas.

- `max(lista)` # Nos entrega el valor máximo dentro de una lista.
- `min(lista)` # Nos entrega el valor mínimo dentro de una lista.
- `sum(lista)` # Nos entrega la suma de los valores dentro de una lista.

3. Métodos para utilizar en las listas.

- `lista.count(valor)` # Nos retorna la cantidad de veces repetidas el dato **valor** dentro de la lista.
- `lista.append(valor)` # Nos agrega valor dentro de la lista.
- `lista.insert(posición, valor)` # Nos agrega valor en la posición indicada.
- `lista.remove(valor)` # Elimina el valor la primera posición en que se lo encuentre

```
1      lista = [1,2,3,4,5,6,1]
2      lista.remove(1)
3      print(lista)
4
5      [2,3,4,5,6,1]
```

- `lista.sort()` # Ordena de menor a mayor los elementos de la lista
- `lista.reverse()` # Invierte los valores de la lista

```
1      lista = [1,2,3,4,5,6,1]
2      lista.reverse()
3      print(lista)
4
5      [1,6,5,4,3,2,1]
6
7      #Otra forma de hacerlo
8
9      listaReversa = lista[::-1]
10     print(listaReversa)
11
12     [1,6,5,4,3,2,1]
```

2. Ciclo for.

1. Introducción al ciclo for.

Recorre secuencias finitas, por lo tanto, itera una cantidad fija de veces.
Funciona sobre tipos de datos iterables, vale decir, secuencias.

2. Sintaxis de un ciclo for

```
1      #Sintaxis de un ciclo for
2
3      for variable in secuencia:
4
5          # Variable viene siendo un iterador interno para poder recorrer listas, strings, tuplas, etc
6          # Se le puede colocar cualquier nombre
7          #Ejemplo:
8
9          notas = [100,80,93]
10
11         for nota in notas:
12             if nota >=55:
13                 print("Tu nota ha sido aprobatoria")
14             else:
15                 print("Esa nota fue menor a 55:c")
```

3. Recorriendo estructuras iterables con ciclo for.

```
1      #Dos formas de recorrer estructuras iterables con ciclo for
2
3      notas = [60,70,85,90]
4
5      for i in notas:
6          print(i)
7
8      60
9      70
10     85
11     90
12
13     #####
14
15     notas = [60,70,85,90]
16
17     for i in range(0,len(notas)):
18         print(i)
19         #Si hacemos print(i), lo que mostrará por pantalla es el valor de i, que en este caso es de 0 a largoNotas-1.
20         print(notas[i])
21         #En este caso nos imprimirá el valor dentro de la lista nota que ocupa la posición i.
```

3. Ejercicios.

1. Escriba un programa el cual pida la cantidad N de datos y luego los almacene en una lista. Luego de eso, todos los datos que no estén duplicados, deben ser agregados a otra lista y ordenarlos de mayor a menor. Debe mostrar el resultado final por pantalla.
2. Escriba un programa el cual le pida la cantidad de notas a un estudiante y luego las almacene en una lista. Luego de eso, debe calcular el promedio de las notas y mostrarlo por pantalla. Si el promedio es mayor o igual 55, el programa debe imprimir que ha aprobado la asignatura, de lo contrario, ha reprobado.

Tener cuidado con las restricciones de la nota

3. Escriba un programa el cual le pida la cantidad de notas a un estudiante y luego las almacene en una lista. Luego de eso, debe recorrer la lista en búsqueda de notas destacadas (notas sobre 90). Si es que se llega a encontrar una, se debe agregar a otra lista la cual se llame **notasDestacadas**, de lo contrario, las notas no destacadas se deben eliminar de la lista principal y almacenarse en otra lista llamada **notasNoDestacadas**. Para finalizar, el programa debe mostrar por pantalla la lista inicial luego de los cambios, la lista notasDestacadas y la lista notasNoDestacadas.
4. Se le solicita a los estudiantes destacados de telemática generación 2021, que escriban un programa computacional en Python con los siguientes requisitos:
 - Se debe pedir X datos y almacenarlos en una lista.
 - Se debe pedir N datos y almacenarlos en otra lista.
 - El programa debe ser capaz de almacenar en algún lugar los datos repetidos en ambas listas y almacenar en otro lugar los datos que no están en las dos listas de entrada.

Utilizar ciclos para la resolución del programa