Usando Listas (Vectores, Matrices o Arreglos)

Las tablas en Python forman parte de lo que se conoce como estructuras de datos. Cabe resaltar que, si has programado en otros lenguajes, las listas no son más que lo que conociste como vectores, matrices, tablas o arreglos n-dimensionales y tienen muchas propiedades de las listas dinámicas.

Las listas o tablas: son estructuras de datos de vital importancia, pues son útiles para resolver problemas que no podríamos solucionar sin ellas. Las tablas son utilizadas para almacenar muchos valores (uno a continuación del otro) en una única variable y además, podemos acceder a cada uno de ellos en forma independiente de los otros. Y si estas pensando en listas de dos o mas dimensiones (matrices), debes saber que, simplemente se trata de una lista que contiene otras listas en su interior.

Sin importar si utilizas una lista simple (vector) o una lista con otras listas en su interior (matrices), en general, la lógica del manejo es la misma. Sin embargo, las tablas en Python van un poco más allá que simplemente guardar muchos datos en una sola variable, ya que también nos proveen de funciones (métodos) para manejar esos datos almacenados.

Ejemplo de tablas: Imaginemos que queremos crear un programa con el cual podamos, de algún modo, almacenar los nombres y números de documentos de diferentes personas. Esto lo podríamos resolver usando dos listas simples (vectores) o una lista que contenga dos listas en su interior (Matriz). Para cualquiera de las dos formas de escribir el programa, tendríamos una lista con los nombres y otra con los números de documento. Pero todo esto los analizaremos mas adelante.

Usando tablas (matrices): Las tablas o matrices han venido para ayudarnos en múltiples circunstancias, similares a las del ejemplo. Dado que una lista es capaz de almacenar múltiples listas en su interior, la podemos pensar como una tabla (con filas y columnas, igual que en el juego de la guerra naval) y asignamos valores en cada casilla según sea necesario, indicando la fila y la columna correspondiente.

Muy bien, ahora que mencionamos superficialmente la gran utilidad de las tablas, listas de listas o matrices, aprendamos más acerca de ellas, veamos cómo crearlas, cómo recorrerlas y algunos ejemplos de esto.

Ejemplos de Listas Simples (Vectores - Arreglos)

L - 1a - Vector - Definir y mostrar elementos

```
# -*- coding: utf-8 -*-
"""
Creado 24/04/2022
Definición de una lista simple (vector) con elementos y diferentes
métodos para  recorrerla mientras se muestran los elementos
@author: Horacio
"""
# Definición e impresión de una lista (vector) ya cargada
edades = [20, 41, 6, 18, 23]

print("-----------------------------------------------------------------")
print("Definir lista con edades y mostrarla con métodos diferentes")
print("-----------------------------------------------------------------")

print("\n---------( for i in range(len(edades)): )--------------------")
# Con for Recorriendo los índices
for i in range( len(edades) ):
    Cadena = "Elemento " + str(i) + ": "
    print(Cadena, edades[i])

print("\n--------( while indice < len(edades): ) ---------------------")
# Con while y recorriendo los índices
indice = 0
while indice < len(edades):
    Cadena = "Elemento " + str(indice) + ": "
    print(Cadena, edades[indice])
    indice += 1

print("\n---------( for h in edades: )-------------------------------")
# Saltando de elemento en elemento. La variable del for "h" va
# tomando los distintos valores de los elemento de la lista y en
# ordenadamente
for h in edades:
    Cadena = "Elemento " + str(h) + ": "
    print(Cadena, h)

# Termina Programa
print("\n-------------------------------------------------------")
print("Programa Terminado.")
```

L - 101 - Matriz - Definir y mostrar elementos

Debes entender que: En este ejemplo, la Lista principal (que llamaremos tabla_matriz) tiene en su interior dos listas (Una contendra nombre, y la otra edades). Entonces la lista usara dos pares de corchetes, los primeros corchetes indican la lista que usaremos (Si ponemos 0 indicaremos que usamos lista de nombres y si ponemos un 1, es que usaremos la lista de documentos), y los segundos corchetes, se usan para indicar que elemento queremos (una posición de esa lista).

```
# -*- coding: utf-8 -*-
"""
Creado 24/04/2022
Listas compuestas
@author: Horacio
"""
#Definición de Lista bidimensional ya cargada
tabla_matriz = [ ['Jose', 'Daniela'] , [18, 21] ]

# Muestra Jose (la primera posición es la 0, 0)
print(tabla_matriz[0][0])

# Muestra 21 (última fila, segunda columna)
print(tabla_matriz[-1][1])

# Muestra 18
print(tabla_matriz[1][0])

# Muestra Daniela
print(tabla_matriz[0][1])

# Muestra toda la lista con nombres
print(tabla_matriz[0])

# Termina Programa
print("\n-----------------------------------------------------------------")
print("Programa Terminado.")

# Este tipo de listas (compuestas) las analizaremos más adelante a
# partir del ejemplo L-101, por ahora solo este.

# Se Recomienda ampliar información referente a los comandos
# usados, puedes consultar  el el  documento "Listado de Comandos".
```

Ejemplos de Listas Simples (Vectores - Arreglos)

L - 1b - Definir, Cargar y mostrar elementos

```
# -*- coding: utf-8 -*-
"""
Creado 24/04/2022
Definir, cargar e imprimir elementos de una lista (una dimensión
vector)
@author: Horacio
"""
# Define lista vacía
Valores = []

# Titulo
print("-----------------------------------------------------------------")
print(" Definir lista vacía y cargarlos con elementos")
print("-----------------------------------------------------------------")

# incorporo 3 elementos (números enteros) a la lista
Valores.append(int(10))
Valores.append(int(9))
Valores.append(int(8))

# incorporo desde teclado 4 elementos (números enteros) a la lista
for i in range(4):
    elem = int(input("Ingrese Elemento: "))
    Valores.append(elem)

# incorporo 3 elementos (números enteros) a la lista
Valores.append(int(3))
Valores.append(int(2))
Valores.append(int(1))

print("\n-----------------------------------------------------------------")
# Muestro todos los elementos de la lista
for i in range( len(Valores) ):
    Cadena = "Elemento " + str(i+1) + ": "
    print(Cadena, Valores[i])

# Termina Programa
print("\n-------------------------------------------------------")
print("Programa Terminado.")
```

L - 1C - Definir, Cargar y Eliminar elementos

Nota Importante, para el próximo ejemplo "L-2b": Como ya habíamos visto antes, se puedes acceder a una posición usando el nombre de la lista y corchetes "[ ]" indicando en su interior el índice o posición del elemento que quieres obtener.
También se había mencionado que las posiciones de las listas arrancan enumeradas desde cero, así la primera posición se accede como [0] y la última sería el tamaño total menos 1.
Lo nuevo es que se pueden usar índices negativos, es decir, accediendo a los elementos de la lista contando desde el final hasta el primero. Lo que nos da un modo sencillo para obtener el último elemento, será entonces con el índice [-1]
Siempre debes recordar que si accedes directamente a un índice que no existe, el programa fallará.

Ejemplos de Listas Simples (Vectores - Arreglos)

L - 2a - Carga de N Elementos, Sumarlos y mostrar resultado

```
# -*- coding: utf-8 -*-
"""
Creado 24/04/2022
Vectores, Carga y suma de elementos
@author: Horacio
"""
# Titulo del programa
print("-----------------------------------------------------------------")
print("Cargar un vector (lista) con N elementos, luego calcular")
print("y mostrar la suma de todos sus elementos por pantalla.")
print("-----------------------------------------------------------------\n")

Valor = []
Suma = 0.0
Cantidad = 5

print("Ingrese", Cantidad, " números.")
for i in range(Cantidad):
    texto = "Ingrese el " + str(i+1) + " Número: "
    Valor.append(int(input(texto)))

for i in range(Cantidad):
    Suma += Valor[i]

print("La suma de los números cargados es: ", end = '')
for i in range(Cantidad):
    print(Valor[i], end = '')
    if i<Cantidad - 1:
        print(" + ", end = '')

print(" = ", Suma )

# Termina Programa
print("\n-----------------------------------------------------------")
print("Programa Terminado.")



# Se Recomienda ampliar información referente a los comandos
# usados, puedes consultar el el documento "Listado de Comandos".
```

L - 2b - Recorriendo y Mostrando en Orden Inverso

```
# -*- coding: utf-8 -*-
"""
Creado 24/04/2022
Vectores: Definir lista (una dimensión) e imprimir en orden inverso
@author: Horacio
"""
# Define lista vacía
Valores = []

# Titulo del programa
print("-----------------------------------------------------------------")
print("Definir lista y cargarlos en el programa, luego mostrar todos")
print("los elementos en orden inverso")
print("-----------------------------------------------------------------")

# incorporo 6 elementos (números enteros) a la lista
Valores.append(int(10))
Valores.append(int(9))
Valores.append(int(8))
Valores.append(int(3))
Valores.append(int(2))
Valores.append(int(1))

print("\n-----------------------------------------------------------------")
# Informo cuantos elementos tiene la lista
Cantidad = len(Valores)
print("La Lista Valores Tiene ", Cantidad, " elementos" )

print("\n-----------------------------------------------------------------")
# Muestro todos loes elementos desde el ultimo al primero
for i in range( Cantidad-1, -1, -1 ):
        Cadena = "Elemento " + str(i+1) + ": "
        print(Cadena, Valores[i])

print("\n-----------------------------------------------------------------")
# Elementos desde el primero hacia atrás hasta primero otra vez
for i in range( 0, -Cantidad-1, -1 ):
        Cadena = "Elemento " + str(i) + ": "
        print(Cadena, Valores[i])

print("\n-----------------------------------------------------------------")
# Asi También podríamos haber logrado la cuenta Regresiva
for i in range(1, Cantidad+1):
      print(Cantidad - i)

# Termina Programa
print("\n-----------------------------------------------------------")
print("Programa Terminado.")
```

Ejemplos de Listas Simples (Vectores - Arreglos)

L - 2c - Cargar N elementos, Buscar y mostrar el Mayor

```
# -*- coding: utf-8 -*-
"""
Creado 24/04/2022
Vectores: Cargar N números enteros, buscar y mostrar el mayor de los elementos (N se define como constante del programa)
cargados
@author: Horacio
"""
# Titulo del programa
print("-----------------------------------------------------------------")
print("Cargar un vector (lista) con N elementos enteros, luego")
print("buscar y mostrar el mayor valor cargado.")
print(" IMPORTANTE: N se define como una constante.")
print("-----------------------------------------------------------------\n")

# Definimos lista Vacía
Valor = []

# Definimos cantidad de elementos a procesar
Cantidad = 5

print("Ingrese", Cantidad, " números.")
for i in range(Cantidad):
    texto = "Ingrese el " + str(i+1) + " Número: "
    Valor.append(int(input(texto)))

Mayor = 0
for i in range(Cantidad):
    if Mayor < Valor[i] or i == 0:
        Mayor = Valor[i]

print("\nEl mayor elemento del vector es: ", Mayor)

# Termina Programa
print("\n-----------------------------------------------------------")
print("Programa Terminado.")
```

L - 2d - Carga (con 0 termina) Calcula y mostrar Promedio

```
# -*- coding: utf-8 -*-
"""
Creado 24/04/2022
Vectores, Cargar números enteros, buscar y mostrar el promedio de los elementos cargados(La carga termina con una clave)
@author: Horacio
"""
# Titulo del programa
print("-----------------------------------------------------------------")
print("Cargar un vector (lista) con N elementos enteros, luego")
print("calcular y mostrar el promedio de todos los números ")
print("cargados.")
print(" IMPORTANTE: Ingresar cero para terminar Carga de datos.")
print("-----------------------------------------------------------------\n")

# Definimos lista Vacía
Valor = []

# Definimos cantidad de elementos a procesar
Cantidad = 0

# Acumulador en cero
Suma = 0

texto = "Ingrese " + str(Cantidad + 1) + " Elemento (Cero termina): "
Nro = int(input(texto))
while Nro != 0:
    Cantidad += 1
    Valor.append(Nro)
    texto = "Ingrese " + str(Cantidad + 1) + " Elemento (Cero termina):"
    Nro = int(input(texto))

# Sumo elementos
for i in range(Cantidad):
    Suma += Valor[i]

# Calculo y muestro Promedio
Promedio = float(Suma/Cantidad)
print("\nEl promedio de los ", Cantidad," elementos cargado es: ",
Promedio)

# Termina Programa
print("\n-----------------------------------------------------------")
print("Programa Terminado.")
```

Ejemplos de Listas Simples (Vectores - Arreglos)

L - 2e - Carga (con 0 termina) Mostrar Posición del Mayor y Menor elemento

```
# -*- coding: utf-8 -*-
"""
Creado 24/04/2022
Vectores, Cargar N números enteros, buscar y mostrar las
posiciones del mayor y menor de los elementos cargados
@author: Horacio
"""
# Titulo del programa
print("----------------------------------------------------------------")
print("Cargar un vector (lista) con elementos enteros, buscar y")
print("mostrar las posiciones del mayor y menor valor cargado.")
print(" IMPORTANTE: Ingresa cero para terminar Carga de datos.")
print("----------------------------------------------------------------\n")

# Definimos lista Vacía
Valor = []

# Definimos cantidad de elementos a procesar
Cantidad = 0

texto = "Ingrese " + str(Cantidad + 1) + " Elemento (Cero termina): "
Nro = int(input(texto))
while Nro != 0:
    Cantidad += 1
    Valor.append(Nro)
    texto = "Ingrese " + str(Cantidad + 1) + " Elemento (Cero termina):"
    Nro = int(input(texto))

Mayor = 0
Menor = 0
for i in range(Cantidad):
    if Mayor < Valor[i] or i == 0:
        Mayor = Valor[i]
        PosMay = i

    if Menor > Valor[i] or i == 0:
        Menor = Valor[i]
        PosMen = i

print("\nLa posicion del Mayor es: ", PosMay + 1)
print("\nLa posicion del Menor es: ", PosMen + 1)

# Termina Programa
print("\n----------------------------------------------------------")
print("Programa Terminado.")
```

L - 2f - Cargar 10 elem y Poner en cero Todos sus Elementos

```
# -*- coding: utf-8 -*-
"""
Creado 24/04/2022
Define Vector de 10 elementos y pone en cero cada Posición
@author: Horacio
"""
print("----------------------------------------------------------------")
print("Define Vector de 10 elementos y pone en cero cada Posición")
print("    Mostrar elementos del vector resultante antes de terminar")
print("----------------------------------------------------------------")

Valores = []

for i in range(10):
    Valores.append(int(0))

for i in range(10):
    texto = "En la Posición " + str(i) + " el Vector contiene: "
    print(texto, Valores[i])

# Termina Programa
print("\n----------------------------------------------------------")
print("Programa Terminado.")



# Se Recomienda ampliar información referente a los comandos
# usados, puedes consultar  el el  documento "Listado de Comandos".
```

Ejemplos de Listas Simples (Vectores - Arreglos)

L - 2g - Cargar y Mostrar Lista con números pares múltiplos de 3

```
# -*- coding: utf-8 -*-
"""
Creado 24/04/2022
Guarda en una lista todos los números pares que a la vez sean
Multiplos de 3 contenidos en el intervalo Amin y Bmax
@author: Horacio
"""
print("----------------------------------------------------------------")
print("Guarda en una lista todos los números pares que a la vez sean")
print("múltiplos de 3 contenidos en el intervalo Amin y Bmax")
print("----------------------------------------------------------------\n")

# Limites del intervalo
Amin = 1
Bmax = 50

# Defino la lista vacía
divisible = []

for i in range(Amin, Bmax, 1):
    if i%2 == 0:
        if i%3 == 0:
            divisible.append(i)
print("Forma 1: ", divisible)

# También podríamos haber escrito así:
divisible = [i for i in range(Amin, Bmax) if i%2==0 if i%3==0]
print("\nForma 2: ", divisible)

# Termina Programa
print("\n----------------------------------------------------------")
print("Programa Terminado.")
```

Ejemplos de Listas Simples (Vectores - Arreglos)

L - 3 - Unión de Listas y Mostrar elementos

```
# -*- coding: utf-8 -*-
"""
Creado 24/04/2022
Carga y unión de diferentes listas
@author: Horacio
"""
# Definición de Listas
Lista_01 = []
Lista_02 = ['Casa', 'árbol']

# Titulo
print("----------------------------------------------------------------")
print("     Carga y Unión de listas - Programa Ejemplo")
print("----------------------------------------------------------------")

# incorporo 2 elementos (Cadenas) a la lista_01
Lista_01.append("Elem_01")
Lista_01.append("Elem_02")

# incorporo desde teclado 2 elementos a la lista_01
for i in range(2):
        elem = input("Ingrese Elemento: ")
        Lista_01.append(elem)

# Se crea la Lista_03 al momento de usar y...
# Union de Listas - Diferentes formas de Unir
Lista_03 = Lista_01 + Lista_02
Lista_03 = Lista_03 + [10, 5]

print("\n----------------------------------------------------------------")
# Informo cuantos elementos tiene la lista
Cantidad = len(Lista_03)
print("La Lista_03 Tiene ", Cantidad, " elementos" )

print("----------------------------------------------------------------")
# Muestro todos los elementos de la lista
for i in range( len(Lista_03) ):
     Cadena = "Elemento " + str(i+1) + ": "
      print(Cadena, Lista_03[i])

# Termina Programa
print("\n----------------------------------------------------------")
print("Programa Terminado.")
```

 # Se Recomienda ampliar información referente a los comandos
 # usados, puedes consultar el el documento "Listado de Comandos".

L - 4 - Insertar y eliminar elemento de la lista

```
# -*- coding: utf-8 -*-
"""
Creado 24/04/2022
Inserción y eliminación dinámica de elementos en una lista
@author: Horacio
"""
import time

# Declaro una lista con sus elementos
Lista_01 = ['casa', 'árbol', 'cielo', 'montaña', 'hormiga']

# Titulo
print("----------------------------------------------------------------")
print("           Inserción y eliminación dinámica")
print("              de los elementos de una lista")
print("----------------------------------------------------------------\n")

# Informo cuantos elementos tiene la lista
Cantidad = len(Lista_01)
print("La Lista_01 Tiene ", Cantidad, " elementos" )
print("Elementos de la Lista", Lista_01)
print("----------------------------------------------------------------\n")
time.sleep(2)

print("Elimino los 2 primeros elementos de la Lista")
# Elimino el primer elemento de la lista
Lista_01.pop(1)
# Elimino el nuevo primer elemento de la lista
Lista_01.pop(1)
# Informo cuantos elementos tiene la lista
Cantidad = len(Lista_01)
print("La Lista_01 Tiene ", Cantidad, " elementos" )
print("Elementos de la Lista", Lista_01)
print("----------------------------------------------------------------\n")
time.sleep(2)

# incorporo 2 elementos a la lista
print("Incorporo 2 nuevos elementos a la Lista")
Lista_01.append("Elem_01")
Lista_01.append("Elem_02")
# Informo cuantos elementos tiene la lista
Cantidad = len(Lista_01)
print("La Lista_01 Tiene ", Cantidad, " elementos" )
print("Elementos de la Lista", Lista_01)
print("----------------------------------------------------------------\n")
time.sleep(2)

print("Elimino un elemento 'hormiga' de la Lista (este donde este)")
# Elimino el elemento 'hormiga' de la lista
Lista_01.remove('hormiga')
# Informo cuantos elementos tiene la lista
Cantidad = len(Lista_01)
print("La Lista_01 Tiene ", Cantidad, " elementos" )
print("Elementos de la Lista", Lista_01)
print("----------------------------------------------------------------\n")
time.sleep(2)

print("----------------------------------------------------------------")
# Muestro todos los elementos que persisten en la Lista
for i in range( len(Lista_01) ):
      Cadena = "Elemento " + str(i+1) + ": "
      print(Cadena, Lista_01[i])

# Termina Programa
print("\n----------------------------------------------------------")
print("Programa Terminado.")
```

Ejemplos de Listas Simples (Vectores - Arreglos)

L - 5 - Insertar y Mostrar Elementos

```
# -*- coding: utf-8 -*-
"""
Creado 24/04/2022
Usar Listas para almacenar números y resultado del un Cálculo
@author: Horacio
"""
# Titulo del programa
print("-----------------------------------------------------------------")
print("        Carga e impresión de elementos en listas")
print("Crear un programa que al ingresar un número calcule")
print("el factorial. Deberá cada uno en una lista, el número")
print("ingresado en una y el factorial en otra. Al finalizar")
print("deberá mostrar las dos listas. ")
print("-----------------------------------------------------------------\n")

# Declaracion de Listas
Numeros=[]
Factorial=[]

Num=int(input("Cantidad de números que calculará factorial: "))
for i in range(Num):
    Val = int(input("Ingrese un numero: ")) # Numero para calculo
    Numeros.append(Val) # Guardo Numero en lista de Números
    Calculo = Val
    while Val > 1:
        Val = Val - 1
        Calculo = Calculo * Val # Calculo Factorial
        Factorial.append(Calculo) # Guardo Factorial en lista

# Muestro Valores guardados en la listas
for i in range(Num):
    print("Número: ", Numeros[i], " - Factorial: ", Factorial[i])

# Termina Programa
print("\n-----------------------------------------------------------")
print("Programa Terminado.")


# Se Recomienda ampliar información referente a los comandos
# usados, puedes consultar  el el  documento "Listado de Comandos".
```

L - 6 - Cargar una serie de nombres

```
# -*- coding: utf-8 -*-
"""
Creado 24/04/2022
Carga de una lista con cadenas de Caracteres.
@author: Horacio
"""
# Titulo del programa
print("-----------------------------------------------------------------")
print("Cargar una lista con los nombres de alumnos de un curso.")
print("Terminar la carga con la Palabra FIN. o letra f.")
print("Mostrar todos los elementos de la lista al terminar la carga")
print("de datos.")
print("-----------------------------------------------------------------")

import time

# Definimos la lista vacía al comienzo
Alumnos = []

# Lista de palabras con las que puedo finalizar carga
Finaliza = ['F', 'FIN']

# Inicializo el contador
i=0
# Lectura del Primer nombre (Fin termina la carga)
texto = "Ingrese Nombre alumno "+str(i+1)+" (Fin para terminar): "
Nombre = input(texto)
while Nombre.upper() not in Finaliza:
    # Cuento y guardo el nombre en la lista    i = i + 1
    Alumnos.append(Nombre)

# Lectura del l resto de los nombres (Fin termina la carga)
    texto="Ingrese Nombre alumno "+str(i+1)+" (Fin para terminar): "
    Nombre = input(texto)

print("\n-----------------------------------------------------------------")
# Ahora se muestran los elementos de la tabla
if i > 0:   #verifico que se haya cargado algún nombre
   tamaño = len(Alumnos) #Veo cuantos nombres hay
   for i in range(tamaño):
       print("Nombre: ", i+1," : ", Alumnos [i])
else:
    print("No se ingresaron Datos")

# Termina Programa
print("\n-----------------------------------------------------------------")
print("Programa Terminado.")
```

Ejemplos de Listas Simples (Vectores - Arreglos)

L - 7 - Ordenar y Mostrar Elementos

```
# -*- coding: utf-8 -*-
"""
Creado 24/04/2022
Ordenar Listas por Dos métodos:
1)-  Generando copia ordenada de la lista
2)- Ordenar la lista sobre si misma
@author: Horacio
"""
print("----------------------------------------------------------------")
print("     Ordenamiento de listas Alfabética y Numéricas")
print("              Ordenar Listas por Dos métodos: ")
print("1)-  Generando copia ordenada de la lista. ")
print("2)- Ordenar la lista sobre si misma. ")
print("Enunciado: Crear una lista cargada con nombres y otra")
print("con números enteros, ordenarlas de Mayor a Menor y")
print("de Menor a Mayor. Antes de finalizar el programa, ")
print("mostrar las listas resultantes. ")
print("----------------------------------------------------------------\n")

# Declaro una lista con sus elementos
Lista_Str = ['hormiga', 'papel', 'casa', 'árbol', 'tijera', 'cielo', 'montaña',
'estrella', 'piedra']
Lista_Int = [ 6, 1, 9, 5, 2, 8, 4, 3, 7]

#Copia de Lista Ordenada de Menor a Mayor
ordenados = sorted(Lista_Str)
print("Copia la Lista y ordena: ", ordenados)

#Copia de Lista Ordenada, en este caso  Menor a Mayor
ordenados.sort(reverse=True)
print("Lista Propia de Men a May: ", ordenados)

print("\n----------------------------------------------------------------\n")
#Lista ordenada (se ordena ella misma) de Menor a Mayor
Lista_Str.sort()
print("Lista Propia de May a Men: ", Lista_Str)

#Lista ordenada (se ordena ella misma) de Mayor a Menor
Lista_Str.sort(reverse=True)
print("Lista Propia de Men a May: ", Lista_Str)

print("\n----------------------------------------------------------------")
print("              Ordenamiento de una lista Numérica")
print("----------------------------------------------------------------\n")

ordenados = sorted(Lista_Int)  #Copia ordenada de Men a May
print("Copia la Lista y ordena: ", ordenados)

ordenados.sort(reverse=True) #Copia ordenada de May a Men
print("Copia de la lista ordenada de Men a May: ", ordenados)

print("\n----------------------------------------------------------------\n")
Lista_Int.sort()  #Ordeno Lista sobre si misma de Menor a Mayor
print("Lista Ordenada de May a Men: ", Lista_Int)

Lista_Int.sort(reverse=True) #Ordeno sobre si misma de May a Men
print("Lista Ordenada de Men a May: ", Lista_Int)

# Termina Programa
print("\n----------------------------------------------------------------")
print("Programa Terminado.")
```

TE PODES IMAGINAR ESTO?

```
Lista_Lineal       = []  # 1 x N elementos
Lista_Rectangulo_01  = [[], []]  # 2 x N elementos
Lista_Rectangulo_02      = [[], [], []]  # 3 x N elementos

Lista_Rectangulo_02        = [[], [], [], []]  # 4 x N elementos
```

 # Se Recomienda ampliar información referente a los comandos
 # usados, puedes consultar el el documento "Listado de Comandos".

Listas Compuestas (Matrices - Arreglos)

Debes entender que: En este tipo de listas, la Lista principal (a la que le ponemos el nombre) podrá tener en cada posición (como su elemento) una lista y cada lista tendrá su propio grupo de elementos. Es por este motivo que la lista que declaramos, para ser recorrida usará tantos pares de corchetes (indices), como listas (grupos de elementos) tenga en su interior.

Ejemplos de Listas Compuestas (Matrices - Arreglos)

L - 101 - Matriz - Definir y mostrar elementos

```
# -*- coding: utf-8 -*-
"""
Creado 24/04/2022
Listas compuestas
@author: Horacio
"""
# Titulo del programa
print("-----------------------------------------------------------------")
print("                    Primera Lista Doble.")
print("También llamada Tabla, Matriz o Arreglo Bidimensional.")
print("      Definir lista con datos y mostrar los elementos.")
print("-----------------------------------------------------------------\n")

#Definición de Lista bidimensional ya cargada
tabla_matriz = [ ['Jose', 'Daniela'] ,  #Fila 1 - Los nombres
                 [18, 21] ]             #Fila  2 - Edades

# Muestra Jose (la primera posición es la 0, 0)
print(tabla_matriz[0][0])

# Muestra 21 (última fila, segunda columna)
print(tabla_matriz[-1][1])

# Muestra 18
print(tabla_matriz[1][0])

# Muestra Daniela
print(tabla_matriz[0][1])

# Muestra toda la lista con nombres
print(tabla_matriz[0])

# Termina Programa
print("\n-----------------------------------------------------------------")
print("Programa Terminado.")


# Se Recomienda ampliar información referente a los comandos
# usados, puedes consultar  el  documento "Listado de Comandos".
```

Como puedes ver en esta y **siguiente programa**, el ciclo for puede acceder directamente a los elementos y recorrerlos uno por uno. **Sin embargo**, a veces puedes necesitar los índices y también puedes acceder a cada elemento por medio de ese índice ya sea usando un ciclo for o incluso un while con un contador.

==========>

L - 102 - Matriz - Recorrido y mostrar elementos

```
# -*- coding: utf-8 -*-
"""
Creado 24/04/2022
Recorrer y mostrar los elementos de una lista doble.
También llamada como tabla, matriz o arreglo bidimensional
@author: Horacio
"""
# Titulo del programa
print("-----------------------------------------------------------------")
print("  Definir lista de 2x2 con datos y mostrar los elementos.")
print("   Recorrer y mostrar los elementos de una lista doble.")
print("-----------------------------------------------------------------\n")

import time

#Definición de Lista bidimensional ya cargada
tabla_matriz = [ ['Jose', 'Daniela'] ,  #Fila 1 - Los nombres
                 [18, 21] ]             #Fila  2 - Edades

print("-----------------------------------------------------------------")
print("RECORRIDO 01 - Accede cada columna dentro de la fila\n")
# Accedemos a cada fila (que es una lista)
for fila in Matriz:
    # Accedemos a cada columna dentro de la fila
    for columna in fila:
        print(columna, end = ' ') #Ver parámetros Función print()
        time.sleep(2)  # Espera dos segundos
    print("\n") # Bajará dos renglones

print("-----------------------------------------------------------------")
print("RECORRIDO 02 - Índices, Se posiciona en fila y recorre columnas\n")
# Recorriendo los índices - i representa las filas
for i in range(len(Matriz)):
    for j in range(len(Matriz[i])):
        print(Matriz[i][j], end = ' ') #Ver parámetros Función print()
        time.sleep(2)  # Espera dos segundos
    print("\n") # Bajará dos renglones

print("-----------------------------------------------------------------")
print("RECORRIDO 03 - Usando while y los índices\n")
# Con while y sus dos índices
fila = 0
while fila < len(Matriz):
    columna = 0
    while columna < len(Matriz[fila]):
        print(Matriz[fila][columna], end = ' ')
        columna += 1
        time.sleep(2)  # Espera dos segundos
    fila += 1
    print("\n") # Bajará dos renglones

# Termina Programa
print("\n-----------------------------------------------------------------")
print("Programa Terminado.")
```

Ejemplos de Listas Compuestas (Matrices - Arreglos)

L - Matriz 103 - Cargar y Mostrar elementos

```
# -*- coding: utf-8 -*-
"""
Creado 24/04/2022
Carga de una lista bidimensional (matriz de 3xN) que contenga
Nombre, DNI y la antigüedad de cada empleado de un comercio.
@author: Horacio
"""
# Titulo del programa
print("----------------------------------------------------------------")
print("CARGA LISTA DE 3xN ELEMENTOS: Cargar una lista")
print("con el Nombre, DNI y Antigüedad de los empleados de un")
print("comercio. Ingresar la Palabra FIN o letra f para terminar la")
print("carga. Mostrar la lista al terminar la carga de datos.")
print("----------------------------------------------------------------\n")

import time
# Lista de palabras con las que puedo finalizar carga
Finaliza = ['F', 'FIN', 'DESPEDIDO']
# Creamos la tabla con listas vacías al comienzo
Empleados = [[],  #Fila de todos los nombres
             [],  #Fila de todos los DNI
             []] #Fila de todos las Antigüedad

# Inicializo el contador
i=0
# Lectura del Primer nombre (Fin termina la carga
print("Ingrese los datos de la persona 1")
Nombre = input("Nombre (Finaliza con 'fin'): ")
while Nombre.upper() not in Finaliza:
    # Cuento y completo la carga de datos del empleado
    i = i + 1
    identificación_Leido = input("DNI: ")

    Antiguedad_Leido = input("Antiguedad: ")

    # Guardo datos en fila correspondiente
    # La primera lista (Primera fila) es para los Nombres
    Empleados[0].append(Nombre)

    # La segunda lista (Segunda fila) es para los Documentos
    Empleados[1].append(identificación_Leido)

    # La tercera lista (Tercera fila) es para la Antiguedad
    Empleados[2].append(Antiguedad_Leido)

    print("\n================================")
    print("========= Empleado: ", i + 1 ," =========\n")

    Nombre = input("Nombre (Finaliza con 'fin'): ")

# Ahora mostremos los valores en la tabla
if i > 0:
   tamaño = len(Empleados[0]) #Cantos empleados (columnas)
   for i in range(tamaño):
       print("Mostrando datos de Empleados", i + 1)
       print("Nombre: ", Empleados[0][i])
       print("DNI: ", Empleados[1][i])
       print("Antigüedad: ", Empleados[2][i])
else:
    print("No se ingresaron Datos")

# Termina Programa
print("\n----------------------------------------------------------------")
print("Programa Terminado.")
```

L - Matrices 104_Suma_x_Filas

```
# -*- coding: utf-8 -*-
"""
Creado 24/04/2022
Carga de una lista bidimensional (matriz de 4x4) y a continuación
sumar cada fila y mostrar resultados
@author: Horacio
"""
# Titulo del programa
print("----------------------------------------------------------------")
print("CARGA LISTA de 4 filas con 4 columnas cada fila (Matriz)")
print("de 4x4). Enunciado: Cargar una lista de 4x4 con Números")
print("enteros y a continuación Calcular la suma de cada fila y")
print("mostrar por pantalla los resultados.")
print("----------------------------------------------------------------\n")

import time

Cantidad_Fil = 4
Cantidad_Col = 4

# Creamos la tabla con listas vacías al comienzo
Matriz = [[],  # Primera Fila     # También podría haber
          [],  # Segunda Fila    # pensado la Matriz como
          [],  # Tercera Fila    # que tiene una sola fila y 4
          []]  # Cuarta Fila     # columnas.

# Comienza Carga de elementos por filas
for i in range(len(Matriz)):
 # Podría poner el limite del for con la variable Cantidad_Fil que
 # tiene el valor 4, pero puedo poner len(Matriz) porque se
 # definieron 4 filas y len(Matriz) retorna 4
  for j in range(Cantidad_Col):
     Cadena = "Ingresar elemento Fila: "+str(i+1)+" Columna
"+str(j+1)+": "
     Matriz[i].append( input(Cadena) )

for i in range(len(Matriz)):
  Sum_Fil = 0
  for j in range(Cantidad_Col):
     Cadena = "Fila: "+str(i+1)+" Columna: "+str(j+1)
     print(Cadena)
     Sum_Fil = Sum_Fil + int(Matriz[i][j])
  print("La suma de los elementos de la fila ", i, " es: ", Sum_Fil )
  time.sleep(1)  # Espera un segundo

# Termina Programa
print("\n----------------------------------------------------------------")
print("Programa Terminado.")



# Se Recomienda ampliar información referente a los comandos
# usados, puedes consultar  el el  documento "Listado de Comandos".
```

Ejemplos de Listas Compuestas (Matrices - Arreglos)

L - Matrices 105_Suma_x_Filas_Guardo_Vector

```
# -*- coding: utf-8 -*-
"""
Creado 24/04/2022
Carga de una lista bidimensional (matriz de 4x4) y a continuación sumar cada fila, guardarla la suma en el vector. Para finalizar mostrar los elementos cargados en el vector
@author: Horacio
"""
# Titulo del programa
print("----------------------------------------------------------------")
print("CARGA LISTA de 4 filas con 4 columnas cada fila (Matriz")
print("de 4x4). Enunciado: Cargar una lista de 4x4 con Números")
print("enteros y a continuación Calcular la suma de cada fila y")
print("Guardarla en un Vector y al finalizar mostrar por pantalla")
print("el Vector.")
print("----------------------------------------------------------------\n")

import time

Cantidad_Fil = 4
Cantidad_Col = 4

# Creamos la tabla con listas vacías al comienzo
Matriz = [[],  # Primera Fila      # También podría haber
          [],  # Segunda Fila     # pensado la Matriz como
          [],  # Tercera Fila     # que tiene una sola fila y 4
          []]  # Cuarta Fila      # columnas.

Vector = []

# Comienza Carga de elementos por filas
for i in range(len(Matriz)):
 # Podría poner el limite del for con la variable Cantidad_Fil que
 # tiene el valor 4, pero también puedo poner len(Matriz) porque se definieron 4 filas y len(Matriz) retorna 4
  for j in range(Cantidad_Col):
    Cadena = "Ingresar elemento Fila: "+str(i+1)+" Columna "+str(j+1)+": "
    Matriz[i].append( input(Cadena) )

for i in range(len(Matriz)):
  Sum_Fil = 0
  for j in range(Cantidad_Col):
     Cadena = "Fila: "+str(i+1)+" Columna: "+str(j+1)
     print(Cadena)
     Sum_Fil = Sum_Fil + int(Matriz[i][j])
  print("La suma de los elementos de la fila ", i, " es: ", Sum_Fil )
  Vector.append( Sum_Fil )  #Cargo en vector la suma de cada fila
  time.sleep(1)  # Espera un segundo

print("---------- Muestro Sumas Guardadas en Vector ----------\n")
for i in range(len(Vector)):
  print("La suma de la fila ", i+1, " es: ", Vector[i] )
  time.sleep(1)  # Espera un segundo

# Termina Programa
print("\n----------------------------------------------------------------")
print("Programa Terminado.")
```

```
# -*- coding: utf-8 -*-
"""
Creado 24/04/2022
Carga de una lista bidimensional (matriz de 4x4) por Filas y a continuación sumar cada columna y mostrar por pantalla la suma.
@author: Horacio
"""
# Titulo del programa
print("----------------------------------------------------------------")
print("CARGAR por filas una LISTA de 4 filas y 4 columnas cada")
print("fila (Matriz de 4x4). Enunciado: Cargar una Matriz de 4x4")
print("con Números enteros y a continuación Calcular la suma de ")
print("cada columna, mostrando cada suma.")
print("----------------------------------------------------------------\n")

import time

Cantidad_Fil = 4
Cantidad_Col = 4

# Creamos la tabla con listas vacías al comienzo
Matriz = [[],  # Primera Fila      # También podría haber
          [],  # Segunda Fila     # pensado la Matriz como
          [],  # Tercera Fila     # que tiene una sola fila y 4
          []]  # Cuarta Fila      # columnas.

# Comienza Carga de elementos por filas
for i in range(len(Matriz)):
 # Podría poner el limite del for con la variable Cantidad_Fil que
 # tiene el valor 4, pero puedo poner len(Matriz) porque se
 # definieron 4 filas y len(Matriz) retorna 4
  for j in range(Cantidad_Col):
    Cadena = "Ingresar elemento Fila: "+str(i+1)+" Columna "+str(j+1)+": "
    Matriz[i].append( input(Cadena) )

# Comienza recorrido por columnas
for j in range(Cantidad_Col):
  Sum_Col = 0
  for i in range(len(Matriz)):
     Cadena = "Fila: "+str(i+1)+" Columna: "+str(j+1)
     print(Cadena)
     Sum_Col = Sum_Col + int(Matriz[i][j])
  print("La suma de elementos de Columna ", j+1, " es: ", Sum_Col )
  time.sleep(1)  # Espera un segundo

# Termina Programa
print("\n----------------------------------------------------------------")
print("Programa Terminado.")


# Se Recomienda ampliar información referente a los comandos
# usados, puedes consultar  el el  documento "Listado de Comandos".
```

Ejemplos de Listas Compuestas (Matrices - Arreglos)

Listas_07_Listas_Matrices_107_Suma_Col_Guardo_Vector

```
# -*- coding: utf-8 -*-
"""
Creado 24/04/2022
Carga de una lista bidimensional (matriz de 4x4) por Filas y a
continuación sumar cada columna, guardarla la suma en un
vector. Para finalizar mostrar los elementos cargados en el vector.
@author: Horacio
"""
# Titulo del programa
print("-----------------------------------------------------------------")
print("CARGAR por filas una LISTA de 4 filas con 4 columnas")
print("cada fila (Matriz de 4x4). Enunciado: Cargar una Matriz")
print("de 4x4 con Números enteros y a continuación Calcular la")
print("suma de cada columna, guardando en un Vector la suma")
print("de cada una. Antes de finalizar el programa, mostrar por")
print("pantalla el Vector.")
print("-----------------------------------------------------------------\n")

import time

Cantidad_Fil = 4
Cantidad_Col = 4

# Creamos la tabla con listas vacías al comienzo
Matriz = [[],  # Primera Fila      # También podría haber
          [],  # Segunda Fila     # pensado la Matriz como
          [],  # Tercera Fila     # que tiene una sola fila y 4
          []]  # Cuarta Fila      # columnas.

Vector = []

# Comienza Carga de elementos por filas?
for i in range(len(Matriz)):
 # Podría poner el limite del for con la variable Cantidad_Fil que
 # tiene el valor 4, pero puedo poner len(Matriz) porque se
 # definieron 4 filas y len(Matriz) retorna 4
  for j in range(Cantidad_Col):
    Cadena = "Ingresar elemento Fila: "+str(i+1)+" Columna
"+str(j+1)+": "
    Matriz[i].append( input(Cadena) )

# Comienza recorrido por columnas
for j in range(Cantidad_Col):
  Sum_Col = 0
  for i in range(len(Matriz)):
     Cadena = "Fila: "+str(i+1)+" Columna: "+str(j+1)
     print(Cadena)
     Sum_Col = Sum_Col + int(Matriz[i][j])
  print("La suma de elementos de Columna ", j+1, " es: ", Sum_Col )
  Vector.append( Sum_Col )  #Cargo en vector suma de Columna
  time.sleep(1)  # Espera un segundo

print("---------- Muestro Sumas Guardadas en Vector ----------\n")
for j in range(len(Vector)):
  print("La suma de la Columna ", j+1, " es: ", Vector[j] )
  time.sleep(1)  # Espera un segundo

# Termina Programa
print("\n-----------------------------------------------------------------")
print("Programa Terminado.")
```

Listas Dinámicas con Más de Una Dimensión (Matrices y Cubos)

Veamos el siguiente ejemplo. La Carga de una lista doble o matriz de 2x3 (2 filas por 3 columnas). El código está justo abajo de la explicacion.

Paso 1: Comenzamos definiendo una lista vacía (a la que desde ahorra llamaremos ***Matriz***). Esta lista recién definida se encuentra sin elementos.

Paso 2: Definimos e inicializamos una variable del tipo cadena de caracteres (a la que llamaremos Fila) guardándole la cadena vacía.

Paso 3: Leemos el numero 7 (siete), recordar que la sentencia input() entrega una cadena, aunque hayamos leído un numero, y concatenamos lo que pudiera tener Fila con la cadena leída más un espacio en blanco.

Paso 4: Repetimos el paso 3, es decir: *Leemos un numero esta vez el 23 y lo concatenamos lo que pudiera tener Fila (ya estaba el 7 y un espacio en blanco) con la cadena leída más otro espacio en blanco.*

Paso 5: Nuevamente repetimos el paso 3, es decir: *Leemos un numero ahora leemos el 13 y lo concatenamos lo que pudiera tener Fila (ya tenia guardado el 7+Espacio en blanco+23+espacio en blanco) con la cadena leída más un espacio en blanco.*

Paso 6: Aplicamos a la cadena **Fila** el método split y el resultado lo guardo nuevamente en **Fila**, que ahora quedo convertida en una **lista.**

Paso 7: Agrego a la Matriz un elemento (la Lista recién creada que llamamos Fila).

Paso 8: Ahora que ya tenemos cargada en la Matriz la primera fila, debemos cargar la segunda fila. Para esto, repetimos los pasos desde el 2 (dos) al paso 7 (siete), lo único que cambiara, es que los números que leemos son el 17, 6 y el 9. Y listo, ya queda cargada la segunda fila.

Veamos el código de este primer ejemplo que se explico gráficamente.

1) La Lista Principal (que llamaremos Matriz) inicialmente estará vacía

Matriz = [ ]

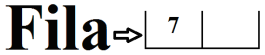
2) Vacío cadena **Fila** Asignándole NADA

Fila = ""

Fila

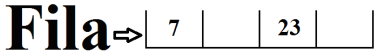
3) Leo una cadena (número) y guardo en Fila, lo que había en Fila más, la cadena leída mas, un espacio en blanco

Fila = Fila + input(Mensaje) + " "



4) Leo una cadena (número) y guardo en Fila, lo que había en Fila más, la cadena leída mas, un espacio en blanco

Fila = Fila + input(Mensaje) + " "

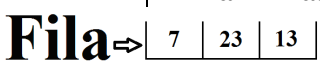


5) Leo una cadena (número) y guardo en Fila, lo que había en Fila más, la cadena leída mas, un espacio en blanco

Fila = Fila + input(Mensaje) + " "

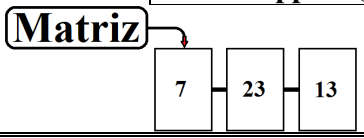
6) Aplico a la cadena Fila, el método split (repasa el funcionamiento consultando **el** "Listado de Comandos"

Fila = Fila.split()

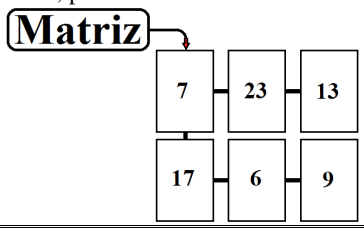


7) Agrego a Matriz, como primer elemento la lista Fila (antes de aplicar método slip era cadena de caracteres)

Matriz.append(Fila)



8) Repito pasos desde el 2 (dos) al 7 (siete) para la segunda fila, pero esta vez leo las cadenas (números) 17, 6 y 9:



Cuando lo pruebes en la PC, te sugiero que ingreses un 2 para las filas y un 3 para las columnas, así podrás seguir los pasos uno por uno.

L - 201 - Matriz Dinámica - Carga, muestra elementos y suma filas

```
# -*- coding: utf-8 -*-
"""
Creado 24/04/2022
Definición, carga y uso de Matrices Dinámicas - El usuario decide al momento de la
ejecución cuantas filas y columnas quiere usar.
@author: Horacio
"""
import time

# Definición de la Lista base para trabajar
```

```
Matriz = []    #Paso 1 de la explicación Gráfica

Fil_Texto = "Cantidad de Filas?: " #Para probar ingrese 2 filas
M = int(input(Fil_Texto))

Col_Tex = "Cantidad Columnas?: "  #Para probar ingrese 3 columnas
N = int(input(Col_Tex))
for i in range(M):
    Fila = ""     #Paso 2 de la explicación Gráfica
    for j in range(N):
        Mensaje = "Elemto Fila: " + str(i+1) + " Columna: " + str(j+1) + " : "
        Fila = Fila + input(Mensaje) + " " #Pasos  3, 4 y 5  de la explicación Gráfica

    Fila = Fila.split()   #Paso 6 de la explicación Gráfica

    print(Fila)  #Muestra la fila ingresada
#    for j in range(len(Fila)): #Se podría transformar todos los elementos
#        Fila[j] = int(Fila[j])    #En números en esta parte
    Matriz.append(Fila)  #Paso 7 de la explicación Gráfica (que repetiré en el Paso 8)

print("\n")

# Se muestran todos los elementos Cargados
for i in range(len(Matriz)):
    for j in range(len(Matriz[i])):
        print(Matriz[i][j], end = ' ')
    print("")
print("")

# Se calcula y muestra la suma de los elementos de cada fila
for i in range(len(Matriz)):
    Sum_Fil = 0  # Se comenzará a recorrer una fila, pongo en cero el acumulador
    for j in range(len(Matriz[i])):
        Cadena = "Fila: "+str(i+1)+" Columna: "+str(j+1)
        print(Cadena)   #Informo que parte de la matriz (fila - Columna)estoy recorriendo
        Sum_Fil = Sum_Fil + int(Matriz[i][j])

    print("La suma de los elementos de la fila ", i, " es: ", Sum_Fil )
    time.sleep(1)  # Espera un segundo

# Termina Programa
print("\n------------------------------------------------------------------")
print("Programa Terminado.")
```

Ahora Comencemos por algunos ejemplos simples, los debemos ejecutar e ir comparando el código con lo que visualizamos en la pantalla