

# Anidación de Listas

Listas Paralelas y  
Listas Compuestas





# Listas Paralelas

- Podemos decir que dos listas son paralelas cuando hay una relación entre las componentes de igual subíndice (misma posición) de una lista y otra.

nombres	Juan	Ana	Marcos	Pablo	Laura
edades	12	21	27	14	21
	[0]	[1]	[2]	[3]	[4]



# Listas Paralelas

## Ejemplo:

- Desarrollar un programa que permita cargar 5 nombres de personas y sus edades respectivas.
- Luego de realizar la carga por teclado de todos los datos imprimir los nombres de las personas mayores de edad (mayores o iguales a 18 años)



# Listas Paralelas

```
1  nombres=[]
2  edades=[]
3  for x in range(5):
4      nom=input("Ingrese el nombre de la
5          persona:")
6      nombres.append(nom)
7      ed=int(input("Ingrese la edad de dicha
8          persona:"))
9      edades.append(ed)
10
11 print("Nombre de las personas mayores de
12     edad:")
13
14 for x in range(5):
15     if edades[x]>=18:
16         print(nombres[x])
```





# Listas Compuestas

Se han trabajado listas cuyos componentes son de tipo:

- Entero
- Flotante
- Cadena de caracteres
- Mezcla de los anteriores

```
notas=[8, 6, 8]  
alturas=[1.73, 1.55, 1.92]  
dias=["lunes", "martes", "miércoles"]
```



# Listas Compuestas

- Pero lo que hace tan flexible a esta estructura de datos, es que podemos almacenar componentes de tipo LISTA.
- Es equivalente al concepto de matrices o arreglos de arreglos.

```
notas=[ [4,5], [6,9], [7,3] ]
```



# Listas Compuestas

- Hemos definido una lista de tres elementos de tipo lista.
- El primer elemento de la lista es otra lista de dos elementos de tipo entero.
- De forma similar los otros dos elementos de la lista notas son listas de dos elementos de tipo entero.

```
notas=[[4,5], [6,9], [7,3]]
```



# Listas Compuestas

- Para definir y crear por asignación una lista de listas tenemos:

```
lista=[[1,2,3], [4,5,6], [7,8,9], [10,11,12]]
```





# Listas Compuestas

- El primer elemento de la variable **lista** es: [1,2,3]
- El segundo elemento de la variable **lista** es: [4,5,6] (y así sucesivamente)

```
lista=[[1,2,3], [4,5,6], [7,8,9], [10,11,12]]
```



# Listas Compuestas

- A La función **print**, si le pasamos como parámetro la **lista**, nos muestra la lista completa por pantalla:

```
print(lista)
```

- Salida por consola o terminal:  
[[1, 2, 3], [4, 5, 6], [7, 8, 9], [10, 11, 12]]



# Listas Compuestas

- Cuando pasamos a la función **print** el primer elemento de la lista:

```
print(lista[0])
```

- Nos muestra la lista contenida en la primera componente de la **lista** principal:

[1, 2, 3]



# Listas Compuestas

- Si queremos acceder al primer entero almacenado en la lista contenida en la primera componente de la lista principal:

```
print(lista[0][0])
```

- Imprimiendo:  
1



# Listas Compuestas

- Acceder a todos los elementos de la primera componente de una lista compuesta por listas:

```
for x in range(len(lista[0])):  
    print(lista[0][x])
```





# Listas Compuestas

- Acceder a cada elemento de cada componente tipo lista de la lista principal o lista de listas o matriz de elementos:

```
# imprimimos cada elemento entero de cada lista  
# contenida en la lista principal  
for k in range(len(lista)):  
    for x in range(len(lista[k])):  
        print(lista[k][x])
```



# Listas Compuestas

**Ejemplo 1:** Requerimiento aplicando los conceptos anteriores.

- Crear una lista por asignación.
- La lista tiene que tener cuatro elementos.
- Cada elemento debe ser una lista de 3 enteros.
- Imprimir sus elementos accediendo de diferentes modos.



# Listas Compuestas

```
1 lista=[[1,2,3], [4,5,6], [7,8,9], [10,11,12]]
2 # imprimimos la lista completa
3 print(lista)
4 print("-----")
5 # imprimimos la primera componente
6 print(lista[0])
7 print("-----")
8 # imprimimos la primera componente de la lista contenida
9 # en la primera componente de la lista principal
10 print(lista[0][0])
11 print("-----")
12 # imprimimos con un for la lista contenida en la primera componente
13 for x in range(len(lista[0])):
14     print(lista[0][x])
15 print("-----")
16 # imprimimos cada elemento entero de cada lista contenida en la lista
17 for k in range(len(lista)):
18     for x in range(len(lista[k])):
19         print(lista[k][x])
```



# Listas Compuestas

## Ejemplo 2:

- Crear una lista por asignación.
- La lista debe tener 2 elementos.
- Cada elemento debe ser una lista de 5 enteros.
- Calcular y mostrar la suma de cada lista contenida en la lista principal.





# Listas Compuestas

## Ejemplo 2:

```
lista=[[1,3,5,7,9], [2,4,6,8,10]]  
for k in range(len(lista)):  
    suma=0  
    for x in range(len(lista[k])):  
        suma=suma+lista[k][x]  
    print(suma)
```





# Listas Compuestas

## Ejemplo 3:

- Crear una lista por asignación. La lista debe que tener 5 elementos.
- Cada elemento debe ser una lista, la primera lista tiene que tener un elemento, la segunda dos elementos, la tercera tres elementos y así sucesivamente (componentes tipo lista de tamaño variable)
- Sumar todos los valores de las listas.



# Listas Compuestas

## Ejemplo 3:

```
lista=[[1], [1,2], [1,2,3], [1,2,3,4], [1,2,3,4,5]]
suma=0
for k in range(len(lista)):
    for x in range(len(lista[k])):
        suma=suma+lista[k][x]
print(suma)
```



# Listas Compuestas

## Carga dinámica de listas:

- En algunas situaciones debemos crear una nueva lista ingresando los datos por teclado o por operaciones del mismo programa.
- Se puede requerir la anidación de los elementos dependiendo de lo que se quiera representar.



# Listas Compuestas

## Ejemplo 4:

- Definir dos listas de 3 elementos.
- En la primera lista cada elemento es una sublista con el nombre del padre y la madre de una familia.
- La segunda lista está constituida por listas con los nombres de los hijos de cada familia. (Puede haber familias sin hijos)
- Imprimir los nombres del padre, la madre y sus hijos. También imprimir solo el nombre del padre y la cantidad de hijos que tiene dicho padre.



# Listas Compuestas

- Listas posibles según el enunciado:

```
padres=[["juan","ana"], ["carlos","maria"], ["pedro","laura"]]  
hijos=[["marcos","alberto","silvia"], [], ["oscar"]]
```





# Listas Compuestas

## Ejemplo 4:

```
padres=[]
hijos=[]
for k in range(3):
    pa=input("Ingrese el nombre del padre:")
    ma=input("Ingrese el nombre de la madre:")
    padres.append([pa, ma])
    cant=int(input("Cuantos hijos tienen esta familia:"))
    hijos.append([])
    for x in range(cant):
        nom=input("Ingrese el nombre del hijo:")
        hijos[k].append(nom)

print("Listado del padre, madre y sus hijos")
for k in range(3):
    print("Padre:",padres[k][0])
    print("Madre:",padres[k][1])
    for x in range(len(hijos[k])):
        print("Hijo:",hijos[k][x])

print("Listado del padre y cantidad de hijos que tiene")
for x in range(3):
    print("padre:",padres[x][0])
    print("Cantidad de hijos:",len(hijos[x]))
```