

Sesión 30

Integra las listas en varias dimensiones, para el uso de grandes cantidades de datos y de diverso tipo de datos.

1. Introducción:

En este subindicador los alumnos aprenderán a manipular los datos desde las listas/arreglos bidimensionales, que permiten un almacenamiento temporal de estos en diversas cantidades y tipos.

MÓDULO 3: Valores booleanos, Instrucciones if-elif-else, Bucles while y for, Control de flujo, Operaciones lógicas y bit a bit, Listas y arreglos.

2. Colecciones de datos(b).

a. Listas dentro de otras listas (listas o arreglos bidimensionales).

Cuando creamos una lista, esta puede tener diversos datos y por consiguiente diversos tipos de datos; cuando los componentes de una lista son otras listas se generan las listas o arreglos bidimensionales, los cuales permiten operar con mayor cantidad de datos, su almacenamiento simula a un tablero de ajedrez si se viera de manera visual.

Para poder acceder a los elementos de una lista de este tipo, debemos trabajar con dos sentencias for, por ejemplo:

Lbid	0	1	2	3	4
0					
1					
2					
3					
4					
5					
6					

La lista bidimensional **Lbid**, está formada por 7 filas y 5 columnas; vamos a proceder a cargar dicha matriz:

```
import random
Lbid=[]
for f in range(7):
    Lst=[]
    for c in range(5):
        Lst.append(random.randint(10,99))
    Lbid.extend([Lst])
for f in range(7):
    for c in range(5):
        print(Lbid[f][c],end=' ')
    print("\n")
```

El primer for es quien controla el número de filas, mientras que el segundo el número de columnas; al momento de imprimir, para que se observe la matriz por cada fila impresa se realiza un salto y de ese modo tenemos la matriz; al ejecutar se observa de este modo **Lbid**:

```
77 54 22 74 77
64 57 52 93 13
42 28 83 46 63
98 38 87 58 72
36 30 89 74 78
38 11 66 98 11
58 99 34 21 63
```

Otra forma de poder cargar los datos es de la siguiente manera:

```
Lbid=[]
for f in range(7):
    Lbid.append([])
    for c in range(5):
        Lbid[f].append(random.randint(10,99))
for f in range(7):
    for c in range(5):
        print(Lbid[f][c],end=' ')
    print("\n")
```

Al ejecutar el código se muestra:

```
56 66 12 42 69
94 84 77 79 63
93 18 76 71 14
15 92 26 79 47
47 87 68 38 45
26 81 70 76 93
53 57 67 80 21
```

b. Operaciones dentro de las listas bidimensionales

En cuanto a las operaciones que se pueden realizar dentro de estas listas, son las mismas que las observadas anteriormente. Sin embargo, existe un tipo de lista bidimensional denominada cuadrada, que tiene una tratativa diferencial al momento de realizar ejercicios. (se muestra una matriz de 5x5)

Lbid	0	1	2	3	4
0					
1					
2					
3					
4					

```
import random
Lbid=[]
for f in range(5):
    Lbid.append([])
    for c in range(5):
        Lbid[f].append(random.randint(10,99))
for f in range(5):
    for c in range(5):
        print(Lbid[f][c],end=' ')
    print("\n")
```

28 11 60 26 19
63 52 79 80 26
88 25 70 89 51
60 51 23 31 83
48 50 48 82 33

Triangular superior

Lbid	0	1	2	3	4
0					
1					
2					
3					
4					

Triangular inferior

Lbid	0	1	2	3	4
0					
1					
2					
3					
4					

Diagonal Principal

Lbid	0	1	2	3	4
0					
1					
2					
3					
4					

Diagonal Secundaria

Lbid	0	1	2	3	4
0					
1					
2					
3					
4					

c. Listas o arreglos Tridimensionales

No se limita la profundidad de seguir incluyendo listas dentro de otras listas, es por ello que Python puede permitir la creación de más dimensiones donde almacenar datos.

3. Desarrollo de ejercicios tipo

Los ejercicios que veremos a continuación operan mediante el uso de listas, están nivelados en cuanto a la dificultad para que se comprenda lo mejor posible.

1. Problema Prg1

En una matriz bidimensional de 4x3, se ingresan números de 2 cifras, mediante un algoritmo mostrar la suma de los elementos de cada fila.

Solución:

En este ejercicio se van autogenerar los números y luego cargarlos en la matriz, finalmente por cada fila se deberán acumular y mostrar.

Programa:

```
import random
Lbid=[]
for f in range(4):
    Lbid.append([])
    for c in range(3):
        Lbid[f].append(random.randint(10,99))
for f in range(4):
    for c in range(3):
        print(Lbid[f][c],end=' ')
    print("\n")

for f in range(4):
    Ac=0
    for c in range(3):
        print(Lbid[f][c],end=' ')
        Ac=Ac+Lbid[f][c]
    print(":",Ac)
    print("\n")
```

Al ejecutar el programa, se muestra lo siguiente:

```
21 14 16
56 23 50
88 77 87
12 50 60
21 14 16 : 51
56 23 50 : 129
88 77 87 : 252
12 50 60 : 122
```

2. Problema Prg2

En una matriz cuadrada de 10x10, ingrese números de 2 cifras y luego muestre la suma de los elementos de la diagonal principal.

Solución:

Cuando hablamos de la diagonal principal debemos considerar que en esta coinciden los índices, ya que se trata de una matriz cuadrada y por ende no existe a necesidad de recorrer toda la matriz.

Programa:

```
import random
Lbid=[]
for f in range(10):
    Lbid.append([])
    for c in range(10):
        Lbid[f].append(random.randint(10,99))
for f in range(10):
    for c in range(10):
        print(Lbid[f][c],end=' ')
    print("\n")

Ac=0
for f in range(10):
    Ac=Ac+Lbid[f][f]
print("La suma de elementos es:",Ac)
```

Al ejecutar el programa, se muestra lo siguiente:

```
11 28 94 67 29 56 66 92 82 51
72 10 60 88 34 64 10 20 30 71
84 93 59 79 95 49 72 51 32 29
87 57 66 93 36 16 85 73 66 70
86 44 67 90 28 43 63 15 70 14
12 88 46 77 71 68 22 57 33 17
45 35 49 53 12 21 48 31 31 72
60 90 25 66 51 50 64 65 88 88
61 43 79 66 50 42 70 35 40 73
48 15 16 88 15 13 66 56 84 72

La suma de elementos es: 494
```

3. Problema Prg3

Vamos a simular una tabla de alumnos, donde se tienen los siguientes datos: código, nombre, nota1 y nota2; el algoritmo deberá permitir el ingreso de 10 registros, para luego indicar cuantos aprobaron y desaprobaron el curso, validar las notas al momento de ingresarlas.

Solución:

La lista bidimensional que se pretende crear simula a una tabla de alumnos, debemos usar una de las formas de carga de datos y luego proceder a calcular el promedio para finalmente tener el número de aprobados y desaprobados.

```
ARALU= []
for f in range(3):
    LST= []
    LST.append(input("Ingrese el código:"))
    LST.append(input("Ingrese el nombre:"))
    op=True
    while(op):
        N1=int(input("Ingrese la nota1:"))
        if (N1>=0) and (N1<=20):
            LST.append(N1)
            break
        else:
            print("Error en NOTA...!!!")
```



```

op=True
while (op) :
    N2=int(input("Ingrese la nota2:"))
    if (N2>=0) and (N2<=20) :
        LST.append(N2)
        break
    else:
        print("Error en NOTA...!!!")
ARALU.extend([LST])
print("Reporte de Alumnos")
for f in range(3):
    for c in range(4):
        print(ARALU[f][c],end=' ')
    print("\n")
print("Reporte de Promedios")
for f in range(3):
    for c in range(4):
        print(ARALU[f][c],end=' ')
    print((int(ARALU[f][2])+int(ARALU[f][3]))/2)
print("\n")
print("Nro de Aprobados y Desaprobados")
Aca=0
Acd=0
for f in range(3):
    PR=(int(ARALU[f][2])+int(ARALU[f][3]))/2
    if (PR>=12.5) :
        Aca=Aca+1
    else:
        Acd=Acd+1
print("Nro de aprobados:",Aca)
print("Nro de desaprobados:",Acd)

```

Al ejecutar el programa, se muestra lo siguiente:

```

Ingrese el codigo:AL01
Ingrese el nombre:JHON
Ingrese la nota1:12
Ingrese la nota2:17
Ingrese el codigo:AL02
Ingrese el nombre:ROSA
Ingrese la nota1:11
Ingrese la nota2:18
Ingrese el codigo:AL03
Ingrese el nombre:MILY
Ingrese la nota1:8
Ingrese la nota2:7

```

Reporte de Alumnos

AL01 JHON 12 17

AL02 ROSA 11 18

AL03 MILY 8 7

Reporte de Promedios

AL01 JHON 12 17 14.5

AL02 ROSA 11 18 14.5

AL03 MILY 8 7 7.5

Nro de Aprobados y Desaprobados

Nro de aprobados: 2

Nro de desaprobados: 1

4. Actividad**A) Ejercicios en general**

Desarrolle los siguientes ejercicios usando todas las sentencias e instrucciones desarrolladas a lo largo del curso.

- Se tiene una matriz cuadrada de 20x20, desarrolle un algoritmo que permita mostrar la suma de los elementos de la diagonal secundaria.
- Se tienen dos matrices de 3x2 y 2x3; desarrolle un algoritmo que permita multiplicarlas, tener en consideración:

$$\begin{pmatrix} 1 & 2 \\ 4 & 5 \\ 7 & 8 \end{pmatrix} \cdot \begin{pmatrix} 1 & 2 & 3 \\ 0 & 5 & 2 \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{pmatrix}$$

3 x 2 2 x 3 3 x 3

Si se pueden multiplicar

Relación de columnas y filas en la multiplicación de matrices

- Desarrolle un algoritmo que permita ingresar números a una matriz cuadrada de 10x10, luego muestre la suma de todos los elementos de la diagonal superior.
- Se pretende simular la tabla clientes de una base de datos, para ello se debe considerar los siguientes campos que se deben ingresar en la lista bidimensional que debe contener 10 registros:

Código
Nombre
Edad
Nhijos
SBasico

Deberá calcular el Sueldo Neto a pagar por cada trabajador, para ello considerar que si tiene más de 3 hijos se le debe considerar un 20% adicional sobre el Sbasico en otro caso no aplica; por otro lado, si la edad del trabajador es mayor a 50 años del mismo modo deberá considerarse un 10% adicional sobre el Sbasico.

Finalmente, deberá generar el código de los clientes: C001, C002, C003...; así mismo validar que la edad debe ser mayor a 18 años y el sueldo básico es como mínimo 1100 soles.

5. Fuentes consultadas:

- A) <https://edube.org/learn/programming-essentials-in-python-part-1-spanish>
- B) Edison Zavaleta C. (2005). *Fundamentos de Programación*. Perú, Editorial Abaco-Lima.