

Recordando conceptos útiles.

 Recuerdas cómo usabas el for anidado para generar pirámides...

```
pirámides con asteriscos
"""

n=int(input("De cuántos renglones quieres la pirámide?"))
for i in range(1,n+1):
    print("*"*i)

h=int(input("De cuántos renglones quieres la pirámide?"))
for i in range(1,n+1):
    print(" "*i,end=" ")
    for j in range (i, n+1):
        print("*", end="")
    print()
Shell ×
```

>>> %Run piramides.py

De cuántos renglones quieres la pirámide?5

De cuántos renglones quieres la pirámide?6



print()

pos-=1

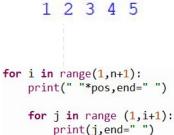


for j in range (1,i+1):

print()

print(j,end=" ")

```
for j in range (pos,0,-1):
    print(j,end=" ")
```

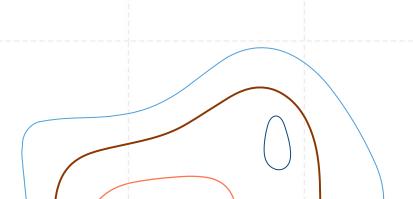


print()

pos-=1

matrices

Así como una lista puede tener cualquier tipo de dato almacenado, también puede tener una lista.



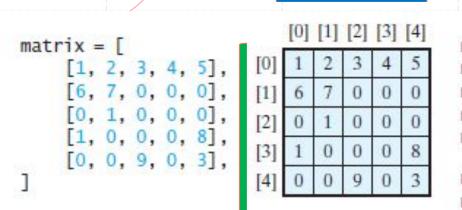
"Introduction to programming using Python", D. Liang

Matrices en python

- Podemos pensar que las matrices son listas donde cada elemento es una fila de la matriz
- Cada fila es, a su vez, una lista.
- Podemos utilizar el operador índice [] para acceder a las **renglón** y a los elementos de éstas.
- Los elementos de un renglón se acceden a través de un índice de columnas

Lista de listas bidimensional = matriz

Columnas



Renglón

```
matrix[0] is [1, 2, 3, 4, 5]
matrix[1] is [6, 7, 0, 0, 0]
matrix[2] is [0, 1, 0, 0, 0]
matrix[3] is [1, 0, 0, 0, 8]
matrix[4] is [0, 0, 9, 0, 3]

matrix[0][0] is 1
matrix[4][4] is 3
```

Las matrices se pueden procesar de la misma forma que las listas unidimensionales, pero ahora se requieren *ciclos anidados.*

Aún si capturas la matriz con una declaración como esta, requieres for anidados para

```
[6, 7, 0, 0, 0],
[0, 1, 0, 0, 0],
[1, 0, 0, 0, 8],
[0, 0, 9, 0, 3],

El primer for te permite cambiar de renglón

El segundo for te permite moverte en cada lista, pero visto como una matriz, es quien te permite recorrer las columnas.
```

RECORRERLA

matrix = [

[1, 2, 3, 4, 5],

```
ren=int(input("Dame le valor de renglones: "))
                                        col=int(input("Dame le valor de columna: "))
                                        matriz=[]
                                        #Llenado de Matriz
                                        for renglon in range(ren):
                                            fila=[]
                                            for columna in range(col):
                                                dato=int(input(f"Dame le valor {[renglon]} {[columna]}: "))
                                                fila.append(dato)
                                            matriz.append(fila)
                                        #Opción de impresión #1
                                        print(matriz)
                                                                                  [[1, 2], [3, 4]]
                                     16
                                        #Opción de impresión #2
                                        for renglon in matriz:
                                                for valor in renglon:
                                     19
                                                    print(valor,end= " ")
                                     20
Captura
                                                print()
                                        #Opción de impresión #3
                                        print(*matriz, sep="\n")
                                                                                  [1, 2]
                                                                                  [3, 4]
```

• Creación de matriz de longitud dada por el usuario

ren=int(input("Dame le valor de renglones: "))

```
col=int(input("Dame le valor de columna: "))
matriz=[]

#Llenado de Matriz
for renglon in range(ren):
    fila=[]
    for columna in range(col):
        dato=int(input(f"Dame le valor {[renglon]} {[columna]}: "))
        fila.append(dato)

matriz.append(fila)
```

Diversas formas de imprimir: 14 #Opción de impresión #1 15 print(matriz) 16 | 17 #Opción de impresión #2

print(valor,end= " ")

for valor in renglon:

for renglon in matriz:

print()

#Opción de impresión #3
print(*matriz, sep="\n")

18

19

20

22

*matriz -> desempaqueta la lista, es decir, es equivalente a pasarle a la función cada fila de la matriz por separado "\n"

[1, 2] [3, 4]

Impresión de listas

```
Funciones para impresión de matrices
```

Sumando todos los elementos matrix = [[1, 2, 3], [4, 5, 6], [7, 8, 9]] # Assume a list is given total = 0 for row in matrix:

print("Total is", total) # Print the total

for value in row:

total += value

Inicialización con números al azar

```
import random
matrix = [] # Create an empty list
numberOfRows = eval(input("Enter the number of rows: "))
numberOfColumns = eval(input("Enter the number of columns: "))
for row in range(numberOfRows):
    matrix.append([]) # Add an empty new row
    for column in range(numberOfColumns):
        matrix[row].append(random.randint(0, 99))
print(matrix)
```

Sumando

```
matrix = [[1, 2, 3], [4, 5, 6], [7, 8, 9]] # Assume a list is given

for column in range(len(matrix[0])):
   total = 0
   for row in range(len(matrix)):
```

total += matrix[row][column]

print("Sum for column", column, "is", total)

Las matrices se comportan como listas cuando son utilizadas en funciones Se pasa la referencia a la matriz Si la matriz se modifica en la función, automáticamente queda modificada.

```
Función para llenar matriz de números por teclado
def llena matriz usuario(ren, col):
                                                 Crea una lista vacía y va
    matriz=[]
                                                 agregando elemento por
    for renglon in range(ren):
                                                 elemento de cada fila de la matriz
        fila=[]
         for columna in range(col):
```

fila.append(dato)

matriz.append(fila)
return matriz

Llamando a la función:

anidada1=llena_matriz_usuario(3,3)

print(anidada1)

dato=int(input(f"Ingresa el dato {columna +1} de la fila {renglon+1}: "))

Función imprime_matriz

```
def imprime_matriz(matriz):
    for renglon in matriz:
        for columna in renglon:
            print(f"{columna:3d}", end=" ")
            print()
```

Función que suma todos los números de una matriz.

def acumulado(matriz):
 acum=0
 for renglon in matriz:
 acum+=sum(renglon)
 return acum

Recibe una matriz como parámetro y calcula la suma de todos los elementos
Nota: la matriz puede ser de cualquier tamaño

Ejemplo

```
#Programa principal
anidada1=llena_matriz_usuario(3,3)
print()
imprime_matriz(anidada1)
suma=acumulado(anidada1)
```

print(f"La suma de los números dentro de la lista anidada es: {suma}")

Prueba del código

- Primero se genera una matriz usando la función llena matriz usuario.
- Se imprime la matriz utilizando la función imprime matriz
- Luego se despliega la suma de todos los elementos usando la función acumulado

```
Ingresa el dato 1 de la fila 1: 1
Ingresa el dato 2 de la fila 1: 2
Ingresa el dato 3 de la fila 1: 3
Ingresa el dato 1 de la fila 2: 4
Ingresa el dato 2 de la fila 2: 5
Ingresa el dato 3 de la fila 2: 6
Ingresa el dato 1 de la fila 3: 7
Ingresa el dato 1 de la fila 3: 8
Ingresa el dato 2 de la fila 3: 9

1 2 3
4 5 6
7 8 9
La suma de los números dentro de la lista anidada es: 45
```





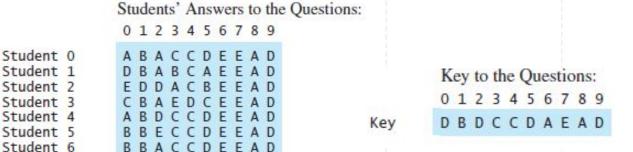
RETO: Examen opción múltiple

- Tenemos un examen de opción múltiple.
- En una tabla tenemos las respuestas de cada estudiante
- En una lista las respuestas correctas

EBECCDEEAD

Student 3 Student 4

Student 7



Reto (continuación)

- Para obtener la calificación de cada estudiante, hay que comparar
 cada respuesta con la llave y, si son iguales, sumar un acierto
- Al final se despliega el número de aciertos de cada estudiante

Antes de ver una posible solución, *inténtalo*.

```
def main():
    # Students' answers to the questions
   answers = [
        ['A', 'B', 'A', 'C', 'C', 'D', 'E', 'E', 'A', 'D'],
        ['D', 'B', 'A', 'B', 'C', 'A', 'E', 'E', 'A', 'D'],
       ['E', 'D', 'D', 'A', 'C', 'B', 'E', 'E', 'A', 'D'],
        ['C', 'B', 'A', 'E', 'D', 'C', 'E', 'E', 'A', 'D'],
       ['A', 'B', 'D', 'C', 'C', 'D', 'E', 'E', 'A', 'D'],
       ['B', 'B', 'E', 'C', 'C', 'D', 'E', 'E', 'A', 'D'],
        ['B', 'B', 'A', 'C', 'C', 'D', 'E', 'E', 'A', 'D'],
        ['E', 'B', 'E', 'C', 'C', 'D', 'E', 'E', 'A', 'D']]
   # Key to the questions
   keys = ['D', 'B', 'D', 'C', 'C', 'D', 'A', 'E', 'A', 'D']
    # Grade all answers
    for i in range (len (answers)):
       # Grade one student
       correctCount = 0
        for j in range(len(answers[i])):
            if answers[i][j] == keys[j]:
               correctCount += 1
       print("Student", i, "'s correct count is", correctCount)
main() # Call the main function
```





Llenado de la matriz

La matriz se va llenando renglón por renglón.
Cada renglón es una lista, una vez que tengo la lista llena, la "añado" a la matriz principal con append.



Para recorrer las matrices se necesitan dos *fors*. El primero es para cambiar de renglón y el segundo es para caminar dentro de cada lista.



matrices

Es una lista que tiene listas adentro. Las listas dentro son todas del mismo tamaño.

Recursos recomendados

 Videos sobre listas de listas desarrollado por Yolanda Martínez.

- https://youtu.be/\Jy3vPXzjs0
- https://youtu.be/fTG9S5MzxMQ
- https://youtu.be/tEAIR5UgcmA