

# Algoritmia y Programación

Oscar Bedoya

`oscar.bedoya@correounivalle.edu.co`

# Arreglos y Matrices

**Problema:** solicitar los nombres de cada estudiante del curso y mostrarlos todos al final

**Problema:** solicitar los nombres de cada estudiante del curso y mostrarlos todos al final

```
Digite su nombre: ANDRES ALVAREZ
```

```
Digite su nombre: JUAN CAICEDO
```

...

```
Digite su nombre: PEDRO ZAPATA
```

**Problema:** solicitar los nombres de cada estudiante del curso y mostrarlos todos al final

Digite su nombre: ANDRES ALVAREZ

Digite su nombre: JUAN CAICEDO

...

Digite su nombre: PEDRO ZAPATA

El estudiante 1 es ANDRES ALVAREZ

El estudiante 2 es JUAN CAICEDO

...

El estudiante 50 es PEDRO ZAPATA

**Problema:** solicitar los nombres de cada estudiante del curso y mostrarlos todos al final

```
nombre1=input("Digite su nombre")
```

```
nombre2=input("Digite su nombre")
```

```
...
```

```
nombre50=input("Digite su nombre")
```

**Problema:** solicitar los nombres de cada estudiante del curso y mostrarlos todos al final

```
nombre1=input("Digite su nombre")
```

```
nombre2=input("Digite su nombre")
```

```
...
```

```
nombre50=input("Digite su nombre")
```

- Se deben definir 50 variables
- Python tiene una forma corta de trabajar con muchas variables







Caben máximo 50 datos

**Problema:** solicitar los nombres y la nota del parcial de cada estudiante del curso y mostrarlos todos al final

Digite su nombre: ANDRES ALVAREZ

Digite la nota del parcial: 4.5

Digite su nombre: JUAN CAICEDO

Digite la nota del parcial: 3.7

...

...

Digite su nombre: PEDRO ZAPATA

Digite la nota del parcial: 1.6



# Arreglos

---

- ¿Qué es un arreglo?

Es un **conjunto de datos**, todos del mismo tipo, a los que se referencia bajo el mismo nombre

# Arreglos

---

Suponga que se tiene un arreglo donde se van a almacenar máximo 4 nombres

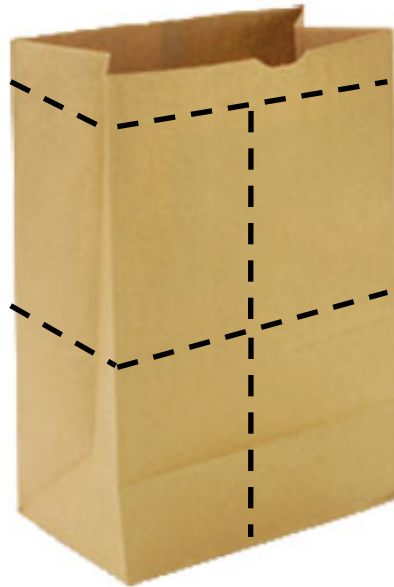


Nombres

# Arreglos

---

Suponga que se tiene un arreglo donde se van a almacenar máximo 4 nombres



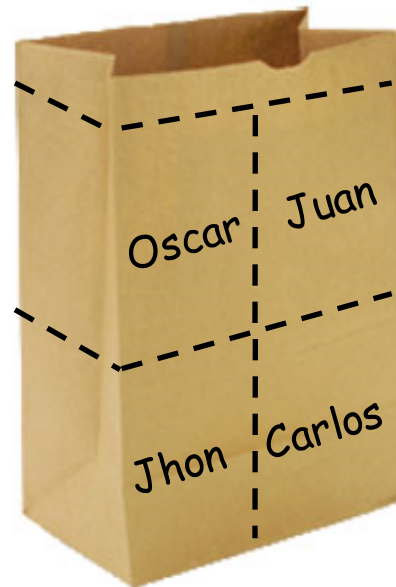
Nombres

# Arreglos

---

## Nombres

"Oscar"	"Juan"	"Jhon"	"Carlos"
---------	--------	--------	----------

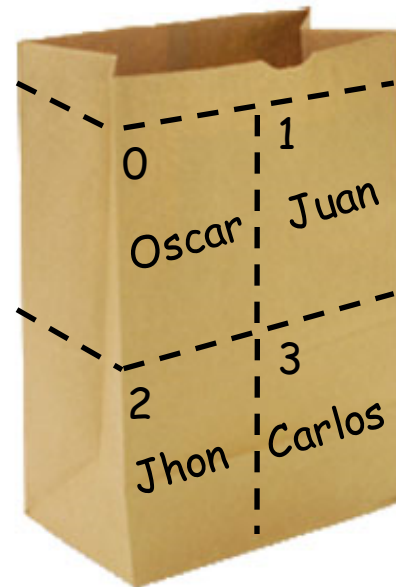


# Arreglos

---

## Nombres

0	1	2	3
"Oscar"	"Juan"	"Jhon"	"Carlos"

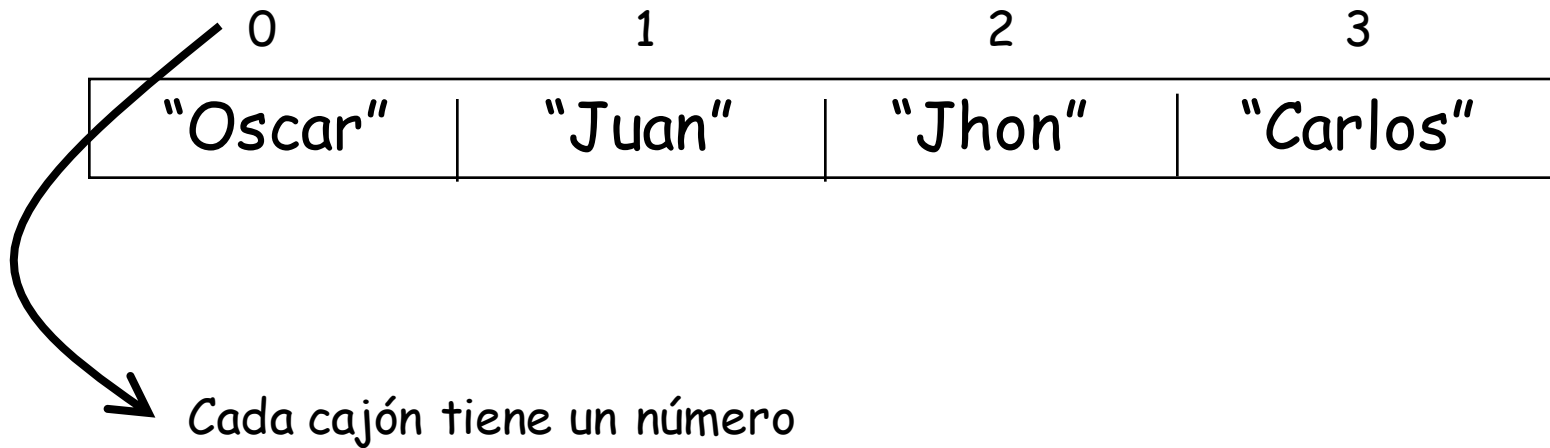




# Arreglos

---

## Nombres



# Arreglos

---

## Nombres

0	1	2	3
"Oscar"	"Juan"	"Jhon"	"Carlos"

- Los números indican las posiciones donde está cada elemento
- Las posiciones siempre son números enteros que inician en 0

# Arreglos

---

## Notas

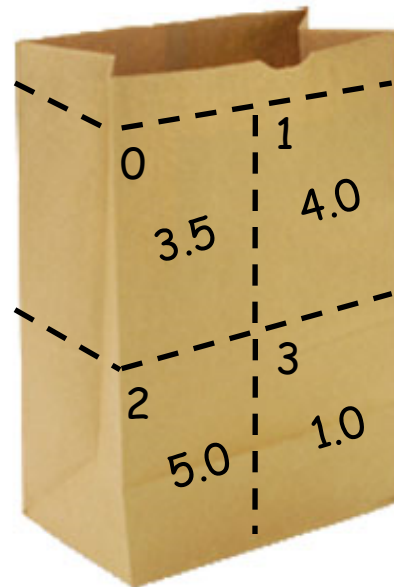
3.5	4.0	5.0	1.0
-----	-----	-----	-----

# Arreglos

---

## Notas

0	1	2	3
3.5	4.0	5.0	1.0



# Arreglos

---

- **Cómo definir un arreglo**

# Arreglos

---

- Cómo **definir** un arreglo



Se debe indicar:

- El nombre del arreglo
- La capacidad máxima

# Arreglos

---

- **Cómo definir un arreglo**

`nombre=[None]*n`

# Arreglos

---

- **Cómo definir un arreglo**

nombre=[None]\*n



nombre del  
arreglo



n es la capacidad  
máxima del arreglo



# Arreglos

---

- **Cómo definir un arreglo**

nombre=[None]\*n



nombre del  
arreglo



n es la capacidad  
máxima del arreglo

Nombres=[None]\*4

Notas=[None]\*4

# Arreglos

---

## Nombres

0	1	2	3
None	None	None	None

Nombres=[None]\*4

# Arreglos

---

## Notas

0	1	2	3
None	None	None	None

Notas=[None]\*4

# Arreglos

---

- ¿Cómo sería la definición del siguiente arreglo?

Edades

0	1	2
None	None	None

# Arreglos

---

- ¿Cómo sería la definición del siguiente arreglo?

Edades

0	1	2
None	None	None

Edades=[None]\*3

# Arreglos

---

- Cómo **ingresar datos** a un arreglo

0	1	2	3
None	None	None	None



¿Cómo guardar el dato  
"Juan" en el arreglo?

# Arreglos

---

- Cómo **ingresar datos** a un arreglo

Debe indicar en qué posición del arreglo va a insertar el valor

nombreDelArreglo[**posicion**]=**valor**



Cajón en el que  
se quiere guardar

# Arreglos

---

Nombres=[None]\*4

None	None	None	None
0	1	2	3



# Arreglos

---

Nombres=[None]\*4

Nombres[1]="Sarah"



*Guarde el texto  
"Sarah" en la  
posición 1 del  
arreglo*

None	None	None	None
0	1	2	3

# Arreglos

---

Nombres=[None]\*4

Nombres[1]="Sarah"



*Guarde el texto  
"Sarah" en la  
posición 1 del  
arreglo*

None	"Sarah"	None	None
0	1	2	3

# Arreglos

---

Nombres=[None]\*4

Nombres[1]="Sarah"

Nombres[2]="Juan"

¿Qué cambia en el arreglo?

None	"Sarah"	None	None
0	1	2	3

# Arreglos

---

Nombres=[None]\*4

Nombres[1]="Sarah"

Nombres[2]="Juan"

None	"Sarah"	"Juan"	None
0	1	2	3

# Arreglos

---

Nombres=[None]\*4

Nombres[1]="Sarah"

Nombres[2]="Juan"

Nombres[?]="Oscar"



¿Cuál debe ser el índice?

"Oscar"	"Sarah"	"Juan"	None
0	1	2	3

# Arreglos

---

Nombres=[None]\*4

Nombres[1]="Sarah"

Nombres[2]="Juan"

Nombres[0]="Oscar"

"Oscar"	"Sarah"	"Juan"	None
0	1	2	3

# Arreglos

---

Nombres=[None]\*4

Nombres[1]="Sarah"

Nombres[2]="Juan"

Nombres[0]="Oscar"

Note que el orden  
en el cual se guardan  
los valores NO  
importa

"Oscar"	"Sarah"	"Juan"	None
0	1	2	3

# Arreglos

---

Nombres=[None]\*4

Nombres[1]="Sarah"

Nombres[2]="Juan"

Nombres[0]="Oscar"

¿La instrucción  
**Nombres[4]="Jose";**  
genera algún error?

"Oscar"	"Sarah"	"Juan"	None
0	1	2	3



# Arreglos

---

Notas=[None]\*3

Notas[?]=?

Notas[?]=?

Notas[?]=?

*¿Cuáles son las  
instrucciones necesarias  
para obtener el arreglo que  
se muestra abajo?*

2.7	3.1	2.5
0	1	2

# Arreglos

---

Notas=[None]\*3

Notas[0]=2.7

Notas[1]=3.1

Notas[2]=2.5

*¿Cuáles son las instrucciones necesarias para obtener el arreglo que se muestra abajo?*

2.7	3.1	2.5
0	1	2

# Arreglos

---

- Cómo **recuperar el valor** contenido en un lugar del arreglo

0	1	2	3
"Oscar"	"Sarah"	"Juan"	None



¿Qué datos se han guardado en el arreglo?

# Arreglos

---

- Cómo **recuperar el valor** contenido en un lugar del arreglo

Debe indicar la posición del arreglo que quiere conocer

nombreDelArreglo[**posicion**]

↑  
Cajón que se  
quiere conocer

# Arreglos

---

## Nombres

0	1	2	3
"Oscar"	"Sarah"	"Juan"	None

# Arreglos

---

## Nombres

0	1	2	3
"Oscar"	"Sarah"	"Juan"	None

`Nombres[2]` indica que el valor en la posición 2 es "Juan"

# Arreglos

---

## Nombres

0	1	2	3
"Oscar"	"Sarah"	"Juan"	None

```
print("Uno de los nombres almacenados es",Nombres[2])
```

# Arreglos

---

## Nombres

0	1	2	3
"Oscar"	"Sarah"	"Juan"	None

```
print("Uno de los nombres almacenados es",Nombres[2])
```



Uno de los nombres almacenados es Juan



# Arreglos

---

```
Notas=[None]*3
```

```
Notas[0]=2.7
```

```
Notas[1]=3.1
```

```
Notas[2]=2.5
```

```
x = Notas[0] + Notas[1]
```

```
print("x es ",x)
```

# Arreglos

---

```
Notas=[None]*3
```

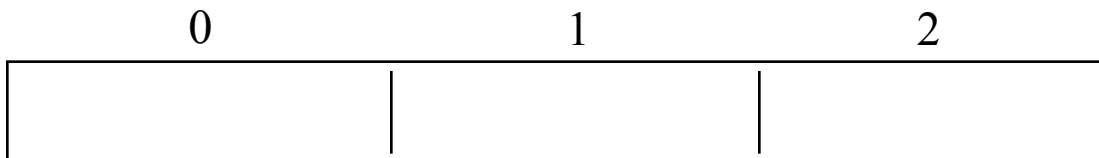
```
Notas[0]=2.7
```

```
Notas[1]=3.1
```

```
Notas[2]=2.5
```

```
x = Notas[0] + Notas[1]
```

```
print("x es ",x)
```



# Arreglos

---

```
Notas=[None]*3
```

```
Notas[0]=2.7
```

```
Notas[1]=3.1
```

```
Notas[2]=2.5
```

```
x = Notas[0] + Notas[1]
```

```
print("x es ",x)
```

0	1	2
2.7	3.1	2.5

# Arreglos

---

```
Notas=[None]*3
```

```
Notas[0]=2.7
```

```
Notas[1]=3.1
```

```
Notas[2]=2.5
```

```
x = Notas[0] + Notas[1]
```

```
print("x es ",x)
```

x es 5.8

0	1	2
2.7	3.1	2.5



```
valores=[None]*4
```

```
valores[1]=10
```

```
valores[0]=20
```

```
valores[3]=30
```

```
print(valores[3])
```

```
print(valores[2])
```

```
calculo1=(valores[0] + valores[1] + valores[3])/3.0
```

```
calculo2=(valores[0]*valores[0] + valores[1]*valores[1])
```

```
print(calculo1)
```

```
print(calculo2)
```

```
valores=[None]*4
```

```
valores[1]=10
```

```
valores[0]=20
```

```
valores[3]=30
```

```
print(valores[3])
```

```
print(valores[2])
```

```
calculo1=(valores[0] + valores[1] + valores[3])/3.0
```

```
calculo2=(valores[0]*valores[0] + valores[1]*valores[1])
```

```
print(calculo1)
```

```
print(calculo2)
```

0	1	2	3

```
valores=[None]*4
```

```
valores[1]=10
```

```
valores[0]=20
```

```
valores[3]=30
```

```
print(valores[3])
```

```
print(valores[2])
```

```
calculo1=(valores[0] + valores[1] + valores[3])/3.0
```

```
calculo2=(valores[0]*valores[0] + valores[1]*valores[1])
```

```
print(calculo1)
```

```
print(calculo2)
```

0	1	2	3
20	10	None	30



```
valores=[None]*4
```

```
valores[1]=10
```

```
valores[0]=20
```

```
valores[3]=30
```

```
print(valores[3])
```

```
print(valores[2])
```

```
calculo1=(valores[0] + valores[1] + valores[3])/3.0
```

```
calculo2=(valores[0]*valores[0] + valores[1]*valores[1])
```

```
print(calculo1)
```

```
print(calculo2)
```

0	1	2	3
20	10	None	30

30
None
20.0
500

```
numeros=[None]*4
```

```
numeros[2]=3.7
```

```
numeros[0]=1.0
```

```
numeros[3]=1.3
```

```
numeros[1]=2.0
```

```
calculo1 = (numeros[2] + numeros[3])/2.0
```

```
calculo2=(numeros[0]*2.0) + (numeros[1]*3)
```

```
calculo3=(numeros[1]/numeros[0]) - 4.0
```

```
print(calculo1)
```

```
print(calculo2)
```

```
print(calculo3)
```

```
numeros=[None]*4
```

```
numeros[2]=3.7
```

```
numeros[0]=1.0
```

```
numeros[3]=1.3
```

```
numeros[1]=2.0
```

```
calculo1 = (numeros[2] + numeros[3])/2.0
```

```
calculo2=(numeros[0]*2.0) + (numeros[1]*3)
```

```
calculo3=(numeros[1]/numeros[0]) - 4.0
```

```
print(calculo1)
```

```
print(calculo2)
```

```
print(calculo3)
```

0	1	2	3

```
numeros=[None]*4
```

```
numeros[2]=3.7
```

```
numeros[0]=1.0
```

```
numeros[3]=1.3
```

```
numeros[1]=2.0
```

```
calculo1 = (numeros[2] + numeros[3])/2.0
```

```
calculo2=(numeros[0]*2.0) + (numeros[1]*3)
```

```
calculo3=(numeros[1]/numeros[0]) - 4.0
```

```
print(calculo1)
```

```
print(calculo2)
```

```
print(calculo3)
```

0	1	2	3
1.0	2.0	3.7	1.3

```
numeros=[None]*4
```

```
numeros[2]=3.7
```

```
numeros[0]=1.0
```

```
numeros[3]=1.3
```

```
numeros[1]=2.0
```

```
calculo1 = (numeros[2] + numeros[3])/2.0
```

```
calculo2=(numeros[0]*2.0) + (numeros[1]*3)
```

```
calculo3=(numeros[1]/numeros[0]) - 4.0
```

```
print(calculo1)
```

```
print(calculo2)
```

```
print(calculo3)
```

0	1	2	3
1.0	2.0	3.7	1.3

2.5
8.0
-2.0

# Arreglos

---

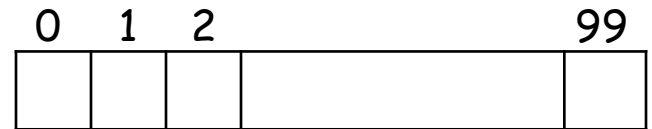
- Presente el conjunto de instrucciones en Python para crear un arreglo de tamaño 100 llamado **numeros**

# Arreglos

---

- Presente el conjunto de instrucciones en Python para crear un arreglo de tamaño 100 llamado **numeros**

```
numeros=[None]*100
```



# Arreglos

---

- Presente el conjunto de instrucciones en Python para crear un arreglo de tamaño 100
- Adicione las instrucciones que necesite para solicitar al usuario cada uno de los 100 números



# Arreglos

---

- Presente el conjunto de instrucciones en Python para crear un arreglo de tamaño 100
- Adicione las instrucciones que necesite para solicitar al usuario cada uno de los 100 números

```
numeros=[None]*100
```

0	1	2		99

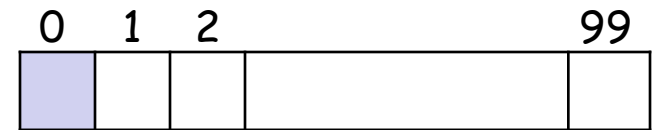
# Arreglos

---

- Presente el conjunto de instrucciones en Python para crear un arreglo de tamaño 100
- Adicione las instrucciones que necesite para solicitar al usuario cada uno de los 100 números

```
numeros=[None]*100
```

```
numeros[0]=int(input("Ingrese un valor"))
```

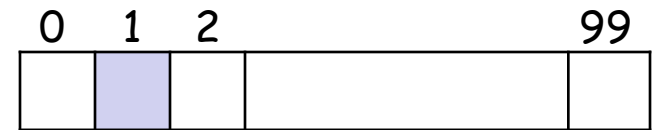


# Arreglos

---

- Presente el conjunto de instrucciones en Python para crear un arreglo de tamaño 100
- Adicione las instrucciones que necesite para solicitar al usuario cada uno de los 100 números

```
numeros=[None]*100
```



```
numeros[0]=int(input("Ingrese un valor"))
```

```
numeros[1]=int(input("Ingrese un valor"))
```

# Arreglos

---

```
numeros=[None]*100
```

```
numeros[0]=int(input("Ingrese un valor"))
```

```
numeros[1]=int(input("Ingrese un valor"))
```

```
...
```

```
numeros[99]=int(input("Ingrese un valor"))
```

# Arreglos

---

```
numeros=[None]*100
```

```
for i in range(0,100,1):
```

```
    numeros[i]=int(input("Ingrese un valor"))
```

# Arreglos

---

- Presente el conjunto de instrucciones en Python para crear un arreglo de tamaño 100
- Adicione las instrucciones que necesite para solicitar al usuario cada uno de los 100 números
- Ahora, **muestre** todos los números

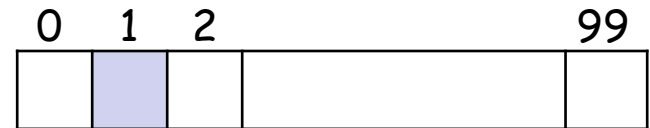
```
print(numeros[0])
```



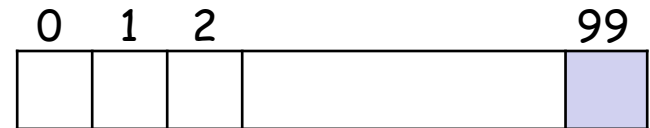
# Arreglos

```
print(numeros[0])
```

```
print(numeros[1])
```







# Arreglos

---

```
for i in range(0,100,1):  
    print(numeros[i])
```

# Arreglos

---

```
numeros=[None]*100
```

```
for i in range(0,100,1):
```

```
    numeros[i]=int(input("Ingrese un valor"))
```

```
for i in range(0,100,1):
```

```
    print(numeros[i])
```

# Arreglos

---

- Presente el conjunto de instrucciones en Python para crear un arreglo de tamaño 100
- Adicione las instrucciones que necesite para solicitar al usuario cada uno de los 100 números
- Ahora, muestre todos los números
- Luego, **muestre** solo los **números pares**

# Arreglos

---

```
for i in range(0,100,1):  
    print(numeros[i])
```

} Se muestran  
todos los  
números

# Arreglos

---

```
for i in range(0,100,1):  
    print(numeros[i])
```

} Se muestran  
todos los  
números

} Se muestran  
solo los  
números pares

# Arreglos

---

```
for i in range(0,100,1):  
    print(numeros[i])
```

} Se muestran  
todos los  
números

0	1	2	3	4		99
4	7	10	36	13		20

} Se muestran  
solo los  
números pares

# Arreglos

---

```
for i in range(0,100,1):  
    print(numeros[i])
```

} Se muestran  
todos los  
números

} Se muestran  
solo los  
números pares

0	1	2	3	4		99
4	7	10	36	13		20

↑  
¿es par?



# Arreglos

---

```
for i in range(0,100,1):  
    print(numeros[i])
```

} Se muestran  
todos los  
números

} Se muestran  
solo los  
números pares

0	1	2	3	4		99
4	7	10	36	13		20

↑  
¿es par?

# Arreglos

---

```
for i in range(0,100,1):  
    print(numeros[i])
```

} Se muestran  
todos los  
números

} Se muestran  
solo los  
números pares

0	1	2	3	4		99
4	7	10	36	13		20

↑  
¿es par?

# Arreglos

---

```
for i in range(0,100,1):  
    print(numeros[i])
```

Se muestran  
todos los  
números

0	1	2	3	4		99
4	7	10	36	13		20

Se muestran  
solo los  
números pares

↑  
¿es par?

# Arreglos

---

```
for i in range(0,100,1):  
    print(numeros[i])
```

} Se muestran  
todos los  
números

```
for i in range(0,100,1):  
    if ( ??? ):  
        print(numeros[i])
```

} Se muestran  
solo los  
números pares

# Arreglos

---

```
for i in range(0,100,1):  
    print(numeros[i])
```

} Se muestran  
todos los  
números

```
for i in range(0,100,1):  
    if (numeros[i]%2==0):  
        print(numeros[i])
```

} Se muestran  
solo los  
números pares

# Arreglos

---

- Presente el conjunto de instrucciones en Python para crear un arreglo de tamaño 100
- Adicione las instrucciones que necesite para solicitar al usuario cada uno de los 100 números
- Ahora, muestre todos los números
- Luego, muestre solo los números pares
- Muestre la **suma** de los números contenidos en el arreglo

# Arreglos

---

```
for i in range(0,100,1):  
    print(numeros[i])
```

} Se muestran  
todos los  
números

# Arreglos

---

```
for i in range(0,100,1):  
    print(numeros[i])
```

} Se muestran  
todos los  
números

```
s=0
```

```
for i in range(1,6,1):
```

```
    s = s + (i*i)
```

```
print(s)
```

➡  $1^2+2^2+3^2+4^2+5^2$



# Arreglos

---

```
for i in range(0,100,1):  
    print(numeros[i])
```

} Se muestran  
todos los  
números

```
suma=0  
  
for i in range(0,100,1):  
    suma = suma + ???  
  
print(suma)
```

} Se muestra la suma  
de todos los  
números en el  
arreglo

# Arreglos

---

```
for i in range(0,100,1):  
    print(numeros[i])
```

} Se muestran  
todos los  
números

```
suma=0  
  
for i in range(0,100,1):  
    suma = suma + numeros[i]  
  
print(suma)
```

} Se muestra la suma  
de todos los  
números en el  
arreglo

Suponga que se ha definido el siguiente arreglo:

```
datos=[None]*6
datos[5]=3
datos[2]=9
datos[3]=7
datos[1]=1
datos[0]=6
datos[4]=0
```

- Al ejecutar las siguientes instrucciones se muestra el mensaje \_\_\_\_\_

```
calculo1=(2*datos[2])+(4*datos[4])-(5*datos[0])
print("Cálculo1 es:",calculo1)
```

- Al ejecutar las siguientes instrucciones se muestra el mensaje \_\_\_\_\_

```
calculo2=0
for i in range(0,6,1):
    if (datos[i]%3==0):
        calculo2=calculo2+datos[i]
print("Cálculo2 es: ",calculo2)
```

Suponga que se ha definido el siguiente arreglo:

```
datos=[None]*6  
datos[5]=3  
datos[2]=9  
datos[3]=7  
datos[1]=1  
datos[0]=6  
datos[4]=0
```

0	1	2	3	4	5

- Al ejecutar las siguientes instrucciones se muestra el mensaje \_\_\_\_\_

```
calculo1=(2*datos[2])+(4*datos[4])-(5*datos[0])  
print("Cálculo1 es:",calculo1)
```

- Al ejecutar las siguientes instrucciones se muestra el mensaje \_\_\_\_\_

```
calculo2=0  
for i in range(0,6,1):  
    if (datos[i]%3==0):  
        calculo2=calculo2+datos[i]  
print("Cálculo2 es: ",calculo2)
```

Suponga que se ha definido el siguiente arreglo:

```
datos=[None]*6
datos[5]=3
datos[2]=9
datos[3]=7
datos[1]=1
datos[0]=6
datos[4]=0
```

0	1	2	3	4	5
6	1	9	7	0	3

- Al ejecutar las siguientes instrucciones se muestra el mensaje Calculo1 es -12

```
calculo1=(2*datos[2])+(4*datos[4])-(5*datos[0])
print("Cálculo1 es:",calculo1)
```

- Al ejecutar las siguientes instrucciones se muestra el mensaje Calculo2 es 18

```
calculo2=0
for i in range(0,6,1):
    if (datos[i]%3==0):
        calculo2=calculo2+datos[i]
print("Cálculo2 es: ",calculo2)
```

Suponga que se ha definido el siguiente arreglo:

```
datos=[None]*7
datos[4]=3
datos[6]=8
datos[0]=5
datos[2]=10
datos[1]=4
datos[3]=8
datos[5]=7
```

- Al ejecutar las siguientes instrucciones se muestra el mensaje \_\_\_\_\_

```
s=0
for i in range(0,7,1):
    if ( i>4 ):
        s=s+datos[i]
print("Cálculo1 es: ",s)
```

- Al ejecutar las siguientes instrucciones se muestra el mensaje \_\_\_\_\_

```
s=0
for i in range(0,7,1):
    if (datos[i]%2==0 or datos[i]<4):
        s=s+datos[i]
print("Cálculo2 es: ",s)
```

Suponga que se ha definido el siguiente arreglo:

```
datos=[None]*7
```

```
datos[4]=3
```

```
datos[6]=8
```

```
datos[0]=5
```

```
datos[2]=10
```

```
datos[1]=4
```

```
datos[3]=8
```

```
datos[5]=7
```

0	1	2	3	4	5	6

- Al ejecutar las siguientes instrucciones se muestra el mensaje \_\_\_\_\_

```
s=0
```

```
for i in range(0,7,1):
```

```
    if ( i>4 ):
```

```
        s=s+datos[i]
```

```
print("Cálculo1 es: ",s)
```

- Al ejecutar las siguientes instrucciones se muestra el mensaje \_\_\_\_\_

```
s=0
```

```
for i in range(0,7,1):
```

```
    if (datos[i]%2==0 or datos[i]<4):
```

```
        s=s+datos[i]
```

```
print("Cálculo2 es: ",s)
```

Suponga que se ha definido el siguiente arreglo:

```
datos=[None]*7
```

```
datos[4]=3
```

```
datos[6]=8
```

```
datos[0]=5
```

```
datos[2]=10
```

```
datos[1]=4
```

```
datos[3]=8
```

```
datos[5]=7
```

0	1	2	3	4	5	6
5	4	10	8	3	7	8

- Al ejecutar las siguientes instrucciones se muestra el mensaje Cálculo 1 es: 15

```
s=0
```

```
for i in range(0,7,1):
```

```
    if ( i>4 ):
```

```
        s=s+datos[i]
```

```
print("Cálculo1 es: ",s)
```

- Al ejecutar las siguientes instrucciones se muestra el mensaje Cálculo 2 es: 33

```
s=0
```

```
for i in range(0,7,1):
```

```
    if (datos[i]%2==0 or datos[i]<4):
```

```
        s=s+datos[i]
```

```
print("Cálculo2 es: ",s)
```





# Matrices

# Matrices

---

**Notas:**

3.5	4.0	5.0	5.0
5.0	5.0	3.0	2.5
4.5	4.5	4.0	5.0

Matriz de 3 filas y 4 columnas

# Matrices

---

**Notas:**

	Parcial1	Parcial2	Labs	Proyecto
Estudiante1	3.5	4.0	5.0	5.0
Estudiante2	5.0	5.0	3.0	2.5
Estudiante3	4.5	4.5	4.0	5.0

Matriz de 3 filas y 4 columnas

# Matrices

---

## Notas:

	Parcial1	Parcial2	Labs	Proyecto
Estudiante1				
Estudiante2				
Estudiante3				
	...	...	...	...
Estudiante50				

Matriz de 50 filas y 4 columnas

# Matrices

---

**Notas:**

	Parcial1	Parcial2	Labs	Proyecto
Estudiante1	3.5	4.0	5.0	5.0
Estudiante2	5.0	5.0	3.0	2.5
Estudiante3	4.5	4.5	4.0	5.0

Matriz de 3 filas y 4 columnas

# Matrices

---

**Notas:**

3.5	4.0	5.0	5.0
5.0	5.0	3.0	2.5
4.5	4.5	4.0	5.0

Se debe conocer la cantidad **máxima** de filas y columnas

# Matrices

---

## Notas:

	columna0	columna1	columna2	columna3
fila 0	3.5	4.0	5.0	5.0
fila 1	5.0	5.0	3.0	2.5
fila 2	4.5	4.5	4.0	5.0



# Matrices

---

**Notas:**

	0	1	2	3
0	3.5	4.0	5.0	5.0
1	5.0	5.0	3.0	2.5
2	4.5	4.5	4.0	5.0

# Matrices

---

**Notas:**

	0	1	2	3
0	3.5	4.0	5.0	5.0
1	5.0	5.0	3.0	2.5
2	4.5	4.5	4.0	5.0



fila 2, column 1

# Matrices

---

Notas:

	0	1	2	3
0	3.5	4.0	5.0	5.0
1	5.0	5.0	3.0	2.5
2	4.5	4.5	4.0	5.0

↓  
fila 1, column 3

# Matrices

---

Notas:

	0	1	2	3	← posición columna
0	3.5	4.0	5.0	5.0	
1	5.0	5.0	3.0	2.5	
2	4.5	4.5	4.0	5.0	

↑ posición fila

# Matrices

---

- **Cómo definir una matriz**

Para definir una matriz se debe conocer:

- El nombre de la matriz
- La cantidad máxima de filas y columnas

# Matrices

---

- Cómo **definir** una matriz

nombre=numpy.zeros((m,n))

↑  
nombre de la matriz

↑  
donde **m** es la cantidad de filas  
y **n** es la cantidad de columnas

# Matrices

---

```
Notas=numpy.zeros((50,4))
```

Matriz llamada Notas con 50 filas y 4 columnas

Notas=numpy.zeros((50,4))

	0	1	2	3
0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0
	...	...	...	...
49	0.0	0.0	0.0	0.0



# Matrices

---

Cómo **definir una matriz** de enteros, con 4 filas y 3 columnas, llamada Numeros

# Matrices

---

Cómo **definir una matriz** de enteros, con 4 filas y 3 columnas, llamada Numeros

```
Numeros=numpy.zeros((4,3))
```

	0	1	2
0	0.0	0.0	0.0
1	0.0	0.0	0.0
2	0.0	0.0	0.0
3	0.0	0.0	0.0

# Matrices

---

¿Cómo **insertar** datos en una matriz?

	0	1	2	3
0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0



¿Cómo guardar 4.5  
en la matriz?

# Matrices

---

¿Cómo **insertar** datos en una matriz?

Debe indicar la posición de la **fila** y de la **columna** donde va a almacenar el dato

`nombreMatriz[posicionFila][posicionColumna]=valor`

# Matrices

---

¿Cómo *insertar* datos en una matriz?

Notas[0][0]=4.5

Notas[0][1]=3.2

	0	1	2	3
0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0

# Matrices

---

¿Cómo *insertar* datos en una matriz?

Notas[0][0]=4.5

Notas[0][1]=3.2

	0	1	2	3
0	4.5	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0

# Matrices

---

¿Cómo *insertar* datos en una matriz?

Notas[0][0]=4.5

Notas[0][1]=3.2

	0	1	2	3
0	4.5	3.2	0.0	0.0
1	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0

# Matrices

---

¿Cómo **insertar** datos en una matriz?

Notas[0][0]=4.5

Notas[0][1]=3.2

Notas[?][?]=**5.0**

Notas[?][?]=**4.1**

	0	1	2	3
0	4.5	3.2	0.0	0.0
1	0.0	0.0	0.0	<b>5.0</b>
2	<b>4.1</b>	0.0	0.0	0.0



# Matrices

---

¿Cómo **insertar** datos en una matriz?

Notas[0][0]=4.5

Notas[0][1]=3.2

Notas[**1**][**3**]=**5.0**

Notas[**?**][**?**]=**4.1**

	0	1	2	3
0	4.5	3.2	0.0	0.0
1	0.0	0.0	0.0	<b>5.0</b>
2	<b>4.1</b>	0.0	0.0	0.0

# Matrices

---

¿Cómo **insertar** datos en una matriz?

Notas[0][0]=4.5

Notas[0][1]=3.2

Notas[1][3]=5.0

Notas[2][0]=4.1

	0	1	2	3
0	4.5	3.2	0.0	0.0
1	0.0	0.0	0.0	5.0
2	4.1	0.0	0.0	0.0

# Matrices

---

¿Cómo recuperar datos de una matriz?

	0	1	2	3
0	4.5	3.2	1.5	3.4
1	4.2	4.7	2.1	4.0
2	2.1	1.0	1.0	2.5



¿Qué valores se han guardado en la matriz?

# Matrices

---

¿Cómo recuperar datos de una matriz?

Debe indicar la posición de la fila y de la columna

`nombreMatriz[posicionFila][posicionColumna]`

# Matrices

---

¿Cómo recuperar datos de una matriz?

Notas[0][0]

Notas[0][3]

	0	1	2	3
0	4.5	3.2	1.5	3.4
1	4.2	4.7	2.1	4.0
2	2.1	1.0	1.0	2.5

# Matrices

---

¿Cómo recuperar datos de una matriz?

Notas[0][0]

Notas[0][3]

¿Cómo obtener el 4.0?

Notas[?][?]

	0	1	2	3
0	4.5	3.2	1.5	3.4
1	4.2	4.7	2.1	4.0
2	2.1	1.0	1.0	2.5

# Matrices

---

¿Cómo recuperar datos de una matriz?

Notas[0][0]

Notas[0][3]

¿Cómo obtener el 4.0?

Notas[1][3]

	0	1	2	3
0	4.5	3.2	1.5	3.4
1	4.2	4.7	2.1	4.0
2	2.1	1.0	1.0	2.5

# Matrices

---

```
print("La nota almacenada en la fila 1 columna 3 es ",Notas[1][3])
```

```
La nota almacenada en la fila 1 columna 3 es 4.0
```

	0	1	2	3
0	4.5	3.2	1.5	3.4
1	4.2	4.7	2.1	4.0
2	2.1	1.0	1.0	2.5



```
Numeros=numpy.zeros((3,3))
```

```
Numeros[0][0]=3
```

```
Numeros[1][0]=7
```

```
Numeros[1][1]=4
```

```
Numeros[2][0]=8
```

```
Numeros[2][1]=9
```

```
Numeros[2][2]=5
```

```
print("Valor",Numeros[2][2])
```

```
print("Valor",Numeros[1][0])
```

```
print("Valor",Numeros[0][2])
```

```
calculo1=Numeros[0][0] + Numeros[2][1]
```

```
print("Valor",calculo1)
```

```
calculo2=(Numeros[0][0] + Numeros[1][1] + Numeros[2][2])/3
```

```
print("Valor",calculo2)
```

```
Numeros=numpy.zeros((3,3))
```

```
Numeros[0][0]=3
```

```
Numeros[1][0]=7
```

```
Numeros[1][1]=4
```

```
Numeros[2][0]=8
```

```
Numeros[2][1]=9
```

```
Numeros[2][2]=5
```

	0	1	2
0			
1			
2			

```
print("Valor",Numeros[2][2])
```

```
print("Valor",Numeros[1][0])
```

```
print("Valor",Numeros[0][2])
```

```
calculo1=Numeros[0][0] + Numeros[2][1]
```

```
print("Valor",calculo1)
```

```
calculo2=(Numeros[0][0] + Numeros[1][1] + Numeros[2][2])/3
```

```
print("Valor",calculo2)
```

```
Numeros=numpy.zeros((3,3))
```

```
Numeros[0][0]=3
```

```
Numeros[1][0]=7
```

```
Numeros[1][1]=4
```

```
Numeros[2][0]=8
```

```
Numeros[2][1]=9
```

```
Numeros[2][2]=5
```

	0	1	2
0	3	0	0
1	7	4	0
2	8	9	5

```
print("Valor",Numeros[2][2])
```

```
print("Valor",Numeros[1][0])
```

```
print("Valor",Numeros[0][2])
```

```
calculo1=Numeros[0][0] + Numeros[2][1]
```

```
print("Valor",calculo1)
```

```
calculo2=(Numeros[0][0] + Numeros[1][1] + Numeros[2][2])/3
```

```
print("Valor",calculo2)
```

```
Numeros=numpy.zeros((3,3))
```

```
Numeros[0][0]=3
```

```
Numeros[1][0]=7
```

```
Numeros[1][1]=4
```

```
Numeros[2][0]=8
```

```
Numeros[2][1]=9
```

```
Numeros[2][2]=5
```

```
print("Valor",Numeros[2][2])
```

```
print("Valor",Numeros[1][0])
```

```
print("Valor",Numeros[0][2])
```

```
calculo1=Numeros[0][0] + Numeros[2][1]
```

```
print("Valor",calculo1)
```

```
calculo2=(Numeros[0][0] + Numeros[1][1] + Numeros[2][2])/3
```

```
print("Valor",calculo2)
```

	0	1	2
0	3	0	0
1	7	4	0
2	8	9	5

Valor 5

Valor 7

Valor 0

```
Numeros=numpy.zeros((3,3))
```

```
Numeros[0][0]=3
```

```
Numeros[1][0]=7
```

```
Numeros[1][1]=4
```

```
Numeros[2][0]=8
```

```
Numeros[2][1]=9
```

```
Numeros[2][2]=5
```

```
print("Valor",Numeros[2][2])
```

```
print("Valor",Numeros[1][0])
```

```
print("Valor",Numeros[0][2])
```

```
calculo1=Numeros[0][0] + Numeros[2][1]
```

```
print("Valor",calculo1)
```

```
calculo2=(Numeros[0][0] + Numeros[1][1] + Numeros[2][2])/3
```

```
print("Valor",calculo2)
```

	0	1	2
0	3	0	0
1	7	4	0
2	8	9	5

Valor 5

Valor 7

Valor 0

Valor 12

Valor 4

```
Numeros=numpy.zeros((2,3))
```

```
Numeros[1][1]=2
```

```
Numeros[1][0]=4
```

```
Numeros[1][2]=6
```

```
Numeros[0][1]=7
```

```
Numeros[0][2]=1
```

```
calculo1=(Numeros[0][1]*Numeros[0][2]) – Numeros[1][2]
```

```
calculo2=Numeros[1][1] + (Numeros[1][0]/2) – (Numeros[0][0]*4)
```

```
calculo3=(4*Numeros[1][2]) – (10*Numeros[1][0])
```

```
print("Valor1",calculo1)
```

```
print("Valor2",calculo2)
```

```
print("Valor3",calculo3)
```

```
Numeros=numpy.zeros((2,3))
```

```
Numeros[1][1]=2
```

```
Numeros[1][0]=4
```

```
Numeros[1][2]=6
```

```
Numeros[0][1]=7
```

```
Numeros[0][2]=1
```

	0	1	2
0			
1			

```
calculo1=(Numeros[0][1]*Numeros[0][2]) – Numeros[1][2]
```

```
calculo2=Numeros[1][1] + (Numeros[1][0]/2) – (Numeros[0][0]*4)
```

```
calculo3=(4*Numeros[1][2]) – (10*Numeros[1][0])
```

```
print("Valor1",calculo1)
```

```
print("Valor2",calculo2)
```

```
print("Valor3",calculo3)
```

```
Numeros=numpy.zeros((2,3))
Numeros[1][1]=2
Numeros[1][0]=4
Numeros[1][2]=6
Numeros[0][1]=7
Numeros[0][2]=1
```

	0	1	2
0	0	7	1
1	4	2	6

```
calculo1 = 1
calculo2 = 4
calculo3 = -16
```

```
calculo1=(Numeros[0][1]*Numeros[0][2]) – Numeros[1][2]
calculo2=Numeros[1][1] + (Numeros[1][0]/2) – (Numeros[0][0]*4)
calculo3=(4*Numeros[1][2]) – (10*Numeros[1][0])
```

```
print("Valor1",calculo1)
print("Valor2",calculo2)
print("Valor3",calculo3)
```



# Matrices

---

- Presente el conjunto de instrucciones en Python para crear una matriz de 50x4

# Matrices

---

- Presente el conjunto de instrucciones en Python para crear una matriz de 50x4

```
Numeros=numpy.zeros((50,4))
```

	0	1	2	3
0				
1				
2				
	...	...	...	...
49				

# Matrices

---

- Presente el conjunto de instrucciones en Python para crear una matriz de 50x4
- Adicione las instrucciones necesarias para **solicitar** al usuario cada uno de los números

	0	1	2	3
0				
1				
2				
	...	...	...	...
49				

	0	1	2	3
0				
1				
2				
	...	...	...	...
49				

	0	1	2	3
0				
1				
2				
	...	...	...	...
49				

	0	1	2	3
0				
1				
2				
	...	...	...	...
49				



```
for i in range(0,50,1):
```

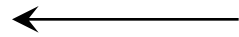
```
    for j in range(0,4,1):
```

```
        numeros[i][j]=float(input("Digite un número"))
```

```
for i in range(0,50,1):  
    for j in range(0,4,1):  
        numeros[i][j]=float(input("Digite un número"))
```

	0	1	2	3
0				
1				
2				
	...	...	...	...
49				

```
for i in range(0,50,1):
```



La variable i maneja las filas.  
Comienza en 0, hasta 49

```
for j in range(0,4,1):
```

```
    numeros[i][j]=float(input("Digite un número"))
```

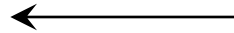
	0	1	2	3
0				
1				
2				
	...	...	...	...
49				

```
for i in range(0,50,1):
```

```
    for j in range(0,4,1):
```

```
        numeros[i][j]=float(input("Digite un número"))
```

La variable **j** maneja las columnas.  
Comienza en 0, hasta 3



	0	1	2	3
0				
1				
2				
	...	...	...	...
49				

## Matriz de 20x10

	0	1	...	9
0			...	
1				
2				
...	...	...	...	...
19				

```
for i in range(?,?,1):
```

```
    for j in range(?,?,1):
```

```
        numeros[i][j]=float(input("Digite un número"))
```

## Matriz de 20x10

	0	1	...	9
0			...	
1				
2				
...	...	...	...	...
19				

```
for i in range(0,20,1):
```

```
    for j in range(0,10,1):
```

```
        numeros[i][j]=float(input("Digite un número"))
```

## Matriz de 3x3

	0	1	2
0			
1			
2			

```
for i in range(?,?,1):
```

```
    for j in range(?,?,1):
```

```
        numeros[i][j]=float(input("Digite un número"))
```

## Matriz de 3x3

	0	1	2
0			
1			
2			

```
for i in range(0,3,1):
```

```
    for j in range(0,3,1):
```

```
        numeros[i][j]=float(input("Digite un número"))
```



# Matrices

---

- Presente el conjunto de instrucciones en Python para crear una matriz de 50x4
- Adicione las instrucciones necesarias para solicitar al usuario cada uno de los números
- Ahora, **muestre todos** los números

```
for i in range(0,50,1):
```

```
    for j in range(0,4,1):
```

```
        numeros[i][j]=float(input("Digite un número"))
```

```
for i in range(0,50,1):  
    for j in range(0,4,1):  
        numeros[i][j]=float(input("Digite un número"))
```

```
for i in range(0,50,1):  
    for j in range(0,4,1):  
        print(numeros[i][j])
```

# Matrices

---

- Presente el conjunto de instrucciones en Python para crear una matriz de 50x4
- Adicione las instrucciones necesarias para solicitar al usuario cada uno de los números
- Ahora, muestre todos los números
- Muestre todos los números de la primera fila

# Matrices

---

	0	1	2	3
0	2	5	2	4
1	3	5	12	5
2	6	12	43	4
3	21	32	31	5

- Cómo mostrar solo los elementos de la primera fila

```
for i in range(0,50,1):  
    for j in range(0,4,1):  
        print(numeros[i][j])
```

```
for i in range(0,50,1):  
    for j in range(0,4,1):  
        if ( ??? ):   
            print(numeros[i][j])
```

```
for i in range(0,50,1):  
    for j in range(0,4,1):  
        if ( i==0 ):   
            print(numeros[i][j])
```

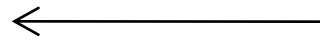


```
for i in range(0,50,1):
```

```
    for j in range(0,4,1):
```

```
        if ( i==0 ):
```

```
            print(numeros[i][j])
```



De todas las posiciones,  
solo muestra el número si  
la fila es cero

# Matrices

---

	0	1	2	3
0	2	5	2	4
1	3	5	12	5
2	6	12	43	4
3	21	32	31	5

- Cómo mostrar solo los elementos de la primera columna

```
for i in range(0,50,1):  
    for j in range(0,4,1):  
        if ( ??? ):   
            print(numeros[i][j])
```

```
for i in range(0,50,1):  
    for j in range(0,4,1):  
        if ( j==0 ):   
            print(numeros[i][j])
```

# Matrices

---

	0	1	2	3
0	2	5	2	4
1	3	5	12	5
2	6	12	43	4
3	21	32	31	5

- Cómo mostrar los elementos de la diagonal \

```
for i in range(0,50,1):  
    for j in range(0,4,1):  
        if ( ??? ):   
            print(numeros[i][j])
```

```
for i in range(0,50,1):  
    for j in range(0,4,1):  
        if ( i==j ):   
            print(numeros[i][j])
```

# Matrices

---

- Muestre la **suma** de todos los números en la matriz



```
for i in range(0,50,1):  
    for j in range(0,4,1):  
        print(numeros[i][j])
```

```
suma=0.0
```

```
for i in range(0,50,1):
```

```
    for j in range(0,4,1):
```

```
        suma = suma + ???
```

```
print("La suma es",suma)
```

```
suma=0.0
```

```
for i in range(0,50,1):
```

```
    for j in range(0,4,1):
```

```
        suma = suma + numeros[i][j]
```

```
print("La suma es",suma)
```



```
datos=numpy.zeros((3,3))
datos[0][0]=7
datos[0][1]=3
datos[0][2]=9
datos[1][0]=12
datos[1][1]=4
datos[1][2]=5
datos[2][0]=2
datos[2][1]=10
datos[2][2]=1
```

- Al ejecutar las siguientes instrucciones se muestran los valores \_\_\_\_\_

```
for i in range(0,3,1):
    for j in range(0,3,1):
        if (i+j==1):
            print(datos[i][j])
```

```
datos=numpy.zeros((3,3))
datos[0][0]=7
datos[0][1]=3
datos[0][2]=9
datos[1][0]=12
datos[1][1]=4
datos[1][2]=5
datos[2][0]=2
datos[2][1]=10
datos[2][2]=1
```

	0	1	2
0			
1			
2			

- Al ejecutar las siguientes instrucciones se muestran los valores \_\_\_\_\_

```
for i in range(0,3,1):
    for j in range(0,3,1):
        if (i+j==1):
            print(datos[i][j])
```

```
datos=numpy.zeros((3,3))
datos[0][0]=7
datos[0][1]=3
datos[0][2]=9
datos[1][0]=12
datos[1][1]=4
datos[1][2]=5
datos[2][0]=2
datos[2][1]=10
datos[2][2]=1
```

	0	1	2
0	7	3	9
1	12	4	5
2	2	10	1

- Al ejecutar las siguientes instrucciones se muestran los valores \_\_\_\_\_

```
for i in range(0,3,1):
    for j in range(0,3,1):
        if (i+j==1):
            print(datos[i][j])
```

```
datos=numpy.zeros((3,3))
datos[0][0]=7
datos[0][1]=3
datos[0][2]=9
datos[1][0]=12
datos[1][1]=4
datos[1][2]=5
datos[2][0]=2
datos[2][1]=10
datos[2][2]=1
```

	0	1	2
0	7	3	9
1	12	4	5
2	2	10	1

- Al ejecutar las siguientes instrucciones se muestran los valores 3,12

```
for i in range(0,3,1):
    for j in range(0,3,1):
        if (i+j==1):
            print(datos[i][j])
```



```
datos=numpy.zeros((3,3))
datos[0][0]=7
datos[0][1]=3
datos[0][2]=9
datos[1][0]=12
datos[1][1]=4
datos[1][2]=5
datos[2][0]=2
datos[2][1]=10
datos[2][2]=1
```

	0	1	2
0	7	3	9
1	12	4	5
2	2	10	1

• Al ejecutar las siguientes instrucciones se muestra el mensaje \_\_\_\_\_

```
suma=0
for i in range(0,3,1):
    for j in range(0,3,1):
        if (datos[i][j]%5==0):
            suma = suma + datos[i][j]
print(suma)
```

```
datos=numpy.zeros((3,3))
```

```
datos[0][0]=7
```

```
datos[0][1]=3
```

```
datos[0][2]=9
```

```
datos[1][0]=12
```

```
datos[1][1]=4
```

```
datos[1][2]=5
```

```
datos[2][0]=2
```

```
datos[2][1]=10
```

```
datos[2][2]=1
```

	0	1	2
0	7	3	9
1	12	4	5
2	2	10	1

• Al ejecutar las siguientes instrucciones se muestra el mensaje 15

```
suma=0
```

```
for i in range(0,3,1):
```

```
    for j in range(0,3,1):
```

```
        if (datos[i][j]%5==0):
```

```
            suma = suma + datos[i][j]
```

```
print(suma)
```



```
datos=numpy.zeros((3,3))
datos[0][0]=7
datos[0][1]=5
datos[0][2]=9
datos[1][0]=12
datos[1][1]=4
datos[1][2]=2
datos[2][0]=2
datos[2][1]=10
datos[2][2]=5
```

- Al ejecutar las siguientes instrucciones se muestra el mensaje \_\_\_\_\_

```
for i in range(0,3,1):
    for j in range(0,3,1):
        if (j==2):
            print(datos[i][j])
```

```
datos=numpy.zeros((3,3))
datos[0][0]=7
datos[0][1]=5
datos[0][2]=9
datos[1][0]=12
datos[1][1]=4
datos[1][2]=2
datos[2][0]=2
datos[2][1]=10
datos[2][2]=5
```

	0	1	2
0			
1			
2			

- Al ejecutar las siguientes instrucciones se muestra el mensaje \_\_\_\_\_

```
for i in range(0,3,1):
    for j in range(0,3,1):
        if (j==2):
            print(datos[i][j])
```

```
datos=numpy.zeros((3,3))
datos[0][0]=7
datos[0][1]=5
datos[0][2]=9
datos[1][0]=12
datos[1][1]=4
datos[1][2]=2
datos[2][0]=2
datos[2][1]=10
datos[2][2]=5
```

	0	1	2
0	7	5	9
1	12	4	2
2	2	10	5

- Al ejecutar las siguientes instrucciones se muestra el mensaje \_\_\_\_\_

```
for i in range(0,3,1):
    for j in range(0,3,1):
        if (j==2):
            print(datos[i][j])
```

```
datos=numpy.zeros((3,3))
datos[0][0]=7
datos[0][1]=5
datos[0][2]=9
datos[1][0]=12
datos[1][1]=4
datos[1][2]=2
datos[2][0]=2
datos[2][1]=10
datos[2][2]=5
```

	0	1	2
0	7	5	9
1	12	4	2
2	2	10	5

- Al ejecutar las siguientes instrucciones se muestra el mensaje 9,2,5

```
for i in range(0,3,1):
    for j in range(0,3,1):
        if (j==2):
            print(datos[i][j])
```

```
datos=numpy.zeros((3,3))
```

```
datos[0][0]=7
```

```
datos[0][1]=5
```

```
datos[0][2]=9
```

```
datos[1][0]=12
```

```
datos[1][1]=4
```

```
datos[1][2]=2
```

```
datos[2][0]=2
```

```
datos[2][1]=10
```

```
datos[2][2]=5
```

	0	1	2
0	7	5	9
1	12	4	2
2	2	10	5

• Al ejecutar las siguientes instrucciones se muestra el mensaje \_\_\_\_\_

```
suma=0
```

```
for i in range(0,3,1):
```

```
    for j in range(0,3,1):
```

```
        if ( datos[i][j]%2==0 ):
```

```
            suma = suma + datos[i][j]
```

```
print(suma)
```



```
datos=numpy.zeros((3,3))
```

```
datos[0][0]=7
```

```
datos[0][1]=5
```

```
datos[0][2]=9
```

```
datos[1][0]=12
```

```
datos[1][1]=4
```

```
datos[1][2]=2
```

```
datos[2][0]=2
```

```
datos[2][1]=10
```

```
datos[2][2]=5
```

	0	1	2
0	7	5	9
1	12	4	2
2	2	10	5

• Al ejecutar las siguientes instrucciones se muestra el mensaje 30

```
suma=0
```

```
for i in range(0,3,1):
```

```
    for j in range(0,3,1):
```

```
        if ( datos[i][j]%2==0 ):
```

```
            suma = suma + datos[i][j]
```

```
print(suma)
```

```
datos=numpy.zeros((3,3))
datos[0][0]=7
datos[0][1]=5
datos[0][2]=9
datos[1][0]=12
datos[1][1]=4
datos[1][2]=2
datos[2][0]=2
datos[2][1]=10
datos[2][2]=5
```

	0	1	2
0	7	5	9
1	12	4	2
2	2	10	5

• Al ejecutar las siguientes instrucciones se muestra el mensaje \_\_\_\_\_

```
suma=0
for i in range(0,3,1):
    for j in range(0,3,1):
        if ( datos[i][j]%3==1 ):
            suma = suma + datos[i][j]
print(suma)
```

```
datos=numpy.zeros((3,3))
```

```
datos[0][0]=7
```

```
datos[0][1]=5
```

```
datos[0][2]=9
```

```
datos[1][0]=12
```

```
datos[1][1]=4
```

```
datos[1][2]=2
```

```
datos[2][0]=2
```

```
datos[2][1]=10
```

```
datos[2][2]=5
```

	0	1	2
0	7	5	9
1	12	4	2
2	2	10	5

• Al ejecutar las siguientes instrucciones se muestra el mensaje 21

```
suma=0
```

```
for i in range(0,3,1):
```

```
    for j in range(0,3,1):
```

```
        if ( datos[i][j]%3==1 ):
```

```
            suma = suma + datos[i][j]
```

```
print(suma)
```

```
datos=numpy.zeros((3,3))
datos[0][0]=7
datos[0][1]=5
datos[0][2]=9
datos[1][0]=12
datos[1][1]=4
datos[1][2]=2
datos[2][0]=2
datos[2][1]=10
datos[2][2]=5
```

	0	1	2
0	7	5	9
1	12	4	2
2	2	10	5

• Al ejecutar las siguientes instrucciones se muestra el mensaje \_\_\_\_\_

```
suma=0
for i in range(0,3,1):
    for j in range(0,3,1):
        if ( i==1 or j==0):
            suma = suma + datos[i][j]
print(suma)
```

```
datos=numpy.zeros((3,3))
```

```
datos[0][0]=7
```

```
datos[0][1]=5
```

```
datos[0][2]=9
```

```
datos[1][0]=12
```

```
datos[1][1]=4
```

```
datos[1][2]=2
```

```
datos[2][0]=2
```

```
datos[2][1]=10
```

```
datos[2][2]=5
```

	0	1	2
0	7	5	9
1	12	4	2
2	2	10	5

• Al ejecutar las siguientes instrucciones se muestra el mensaje 27

```
suma=0
```

```
for i in range(0,3,1):
```

```
    for j in range(0,3,1):
```

```
        if ( i==1 or j==0):
```

```
            suma = suma + datos[i][j]
```

```
print(suma)
```

```
datos=numpy.zeros((3,3))
```

```
datos[0][0]=7
```

```
datos[0][1]=5
```

```
datos[0][2]=9
```

```
datos[1][0]=12
```

```
datos[1][1]=4
```

```
datos[1][2]=2
```

```
datos[2][0]=2
```

```
datos[2][1]=10
```

```
datos[2][2]=5
```

	0	1	2
0	7	5	9
1	12	4	2
2	2	10	5

• Al ejecutar las siguientes instrucciones se muestra el mensaje \_\_\_\_\_

```
suma=0
```

```
for i in range(0,3,1):
```

```
    for j in range(0,3,1):
```

```
        if ( i==1 or datos[i][j]>8):
```

```
            suma = suma + datos[i][j]
```

```
print(suma)
```

```
datos=numpy.zeros((3,3))
```

```
datos[0][0]=7
```

```
datos[0][1]=5
```

```
datos[0][2]=9
```

```
datos[1][0]=12
```

```
datos[1][1]=4
```

```
datos[1][2]=2
```

```
datos[2][0]=2
```

```
datos[2][1]=10
```

```
datos[2][2]=5
```

	0	1	2
0	7	5	9
1	12	4	2
2	2	10	5

• Al ejecutar las siguientes instrucciones se muestra el mensaje 37

```
suma=0
```

```
for i in range(0,3,1):
```

```
    for j in range(0,3,1):
```

```
        if ( i==1 or datos[i][j]>8):
```

```
            suma = suma + datos[i][j]
```

```
print(suma)
```