In [1]:

```
#MANUEL FELIPE VALENCIA CEBALLOS
#1004768150
import numpy as np

#Se crea con 6 elementos
a=np.arange(10)

print('Arreglo a=', a,'\n')
#Tipo
print('Tipo de a=', a.dtype,'\n')
#Dimension
print('Dimension de a=', a,'\n')

print('Número de elementos de a =', a.shape)
```

```
Arreglo a= [0 1 2 3 4 5 6 7 8 9]

Tipo de a= int32

Dimension de a= [0 1 2 3 4 5 6 7 8 9]

Número de elementos de a = (10,)
```

In [2]:

```
m = np.array([np.arange(3), np.arange(3)])
print(m)
```

```
[[0 1 2]
 [0 1 2]]
```

In [3]:

```
a = np.array([[1,2,3], [4,5,6]])
print('a =\n', a, '\n')

print('a[0,0] =', a[0,0], '\n')
print('a[0,1] =', a[0,1], '\n')
print('a[0,2] =', a[0,2], '\n')
print('a[1,0] =', a[1,0], '\n')
print('a[1,1] =', a[1,1], '\n')
print('a[0,2] =', a[1,2])
```

```
a =
 [[1 2 3]
 [4 5 6]]

a[0,0] = 1

a[0,1] = 2

a[0,2] = 3

a[1,0] = 4

a[1,1] = 5

a[0,2] = 6
```

In [4]:

```
a = np.arange(10)
print('a =', a, '\n')

print('a[0:9] = ', a[0:10], '\n')

print('a[3,7] =', a[3:5])
```

```
a = [0 1 2 3 4 5 6 7 8 9]

a[0:9] =  [0 1 2 3 4 5 6 7 8 9]

a[3,7] = [3 4]
```

In [5]:

```
print('a[0:9:1] =', a[0:9:1], '\n')

print('a[:9:1] =', a[:9:1], '\n')

print('a[0:9:2] =', a[0:9:2], '\n')

print('a[0:9:3] =', a[0:9:3])
```

```
a[0:9:1] = [0 1 2 3 4 5 6 7 8]

a[:9:1] = [0 1 2 3 4 5 6 7 8]

a[0:9:2] = [0 2 4 6 8]

a[0:9:3] = [0 3 6]
```

In [6]:

```
print('a[9:0:-1] =', a[9:0:-1], '\n')

print('a[::-1] =', a[::-1])
```

```
a[9:0:-1] = [8 7 6 5 4 3 2 1]

a[::-1] = [8 7 6 5 4 3 2 1 0]
```

In [7]:

```
b = np.arange(24).reshape(2,3,4)
print('b =\n', b)
```

```
b =
 [[[ 0  1  2  3]
  [ 4  5  6  7]
  [ 8  9 10 11]]

 [[12 13 14 15]
  [16 17 18 19]
  [20 21 22 23]]]
```

In [8]:

```
print('b[1,2,3] =', b[1,2,3], '\n')

print('b[0,2,2] =', b[0,2,2], '\n')

print('b[0,1,1] =', b[0,1,1])
```

```
b[1,2,3] = 23

b[0,2,2] = 10

b[0,1,1] = 5
```

In [9]:

```
print('b[0,0,0] =', b[0,0,0], '\n')

print('b[1,0,0] =', b[1,0,0], '\n')

print('b[:,0,0] =', b[:,0,0])
```

```
b[0,0,0] = 0

b[1,0,0] = 12

b[:,0,0] = [ 0 12]
```

In [10]:

```
print('b[0] =\n', b[0])
```

```
b[0] =
 [[ 0  1  2  3]
 [ 4  5  6  7]
 [ 8  9 10 11]]
```

In [11]:

```
print('b[0,:,:] =\n', b[0,:,:])
```

```
b[0,:,:] =
 [[ 0  1  2  3]
 [ 4  5  6  7]
 [ 8  9 10 11]]
```

In [12]:

```
print('b[0, ...] =\n', b[0, ...])
```

```
b[0, ...] =
 [[ 0  1  2  3]
 [ 4  5  6  7]
 [ 8  9 10 11]]
```

In [13]:

```
print('b[0,1] =', b[0,1])
```

```
b[0,1] = [4 5 6 7]
```

In [14]:

```
z = b[0,1]
print('z =', z, '\n')

print('z[::2] =', z[::2])
```

```
z = [4 5 6 7] 

z[::2] = [4 6]
```

In [15]:

```
print('b[0,1,::2] =', b[0,1,::2])
```

```
b[0,1,::2] = [4 6]
```

In [16]:

```
print(b, '\n')
print('b[:,:,1] =\n', b[:,:,1], '\n')

print('b[...,1] =\n', b[...,1])
```

```
[[[ 0  1  2  3]
  [ 4  5  6  7]
  [ 8  9 10 11]]

 [[12 13 14 15]
  [16 17 18 19]
  [20 21 22 23]]] 

b[:,:,1] =
 [[ 1  5  9]
 [13 17 21]] 

b[...,1] =
 [[ 1  5  9]
 [13 17 21]]
```

In [17]:

```
print(b, '\n')
print('b[:,1] =', b[:,1])
```

```
[[[ 0  1  2  3]
  [ 4  5  6  7]
  [ 8  9 10 11]]

 [[12 13 14 15]
  [16 17 18 19]
  [20 21 22 23]]] 

b[:,1] = [[ 4  5  6  7]
 [16 17 18 19]]
```

In [18]:

```
print(b, '\n')
print('b[0,:,1] =', b[0,:,1])
```

```
[[[ 0  1  2  3]
  [ 4  5  6  7]
  [ 8  9 10 11]]

 [[12 13 14 15]
  [16 17 18 19]
  [20 21 22 23]]]

b[0,:,1] = [1 5 9]
```

In [8]:

```
b = np.arange(30).reshape(2,3,5)
print('b=', b,'\n')
print('b[0,:,-1] =', b[0,:,-1])

print('b[0, ::-1, -1] =', b[0, ::-1, -1])

print('b[0, ::2, -1] =', b[0, ::2, -1])
```

```
b= [[[ 0  1  2  3  4]
  [ 5  6  7  8  9]
  [10 11 12 13 14]]

 [[15 16 17 18 19]
  [20 21 22 23 24]
  [25 26 27 28 29]]]

b[0,:,-1] = [ 4  9 14]
b[0, ::-1, -1] = [14  9  4]
b[0, ::2, -1] = [ 4 14]
```

In [10]:

```
print(b,'\n------------------\n')
#invertir
print(b[::-1])
```

```
[[[ 0  1  2  3  4]
  [ 5  6  7  8  9]
  [10 11 12 13 14]]

 [[15 16 17 18 19]
  [20 21 22 23 24]
  [25 26 27 28 29]]]
------------------

[[[15 16 17 18 19]
  [20 21 22 23 24]
  [25 26 27 28 29]]

 [[ 0  1  2  3  4]
  [ 5  6  7  8  9]
  [10 11 12 13 14]]]
```

In [11]:

```
#con ravel se genera un vetor con la matriz
print('Vector b = \n', b.ravel())
```

```
Vector b =
 [ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
 24 25 26 27 28 29]
```

In [12]:

```
#flatten es similar, pero este genera un nuevo espacio me memoria
print('Vector b con flatten =\n', b.flatten())
```

```
Vector b con flatten =
 [ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
 24 25 26 27 28 29]
```

In [15]:

```
#shape cambia la estructura de la matriz
b.shape=(3,10)
print('b(5x5) = \n',b)
```

```
b(5x5) =
 [[ 0  1  2  3  4  5  6  7  8  9]
 [10 11 12 13 14 15 16 17 18 19]
 [20 21 22 23 24 25 26 27 28 29]]
```

In [16]:

```
#transpuesta

print('b=\n',b,'\n-------------\n')
#transpuesta
print('Transpuesta de b=\n', b.transpose(),'\n------------\n')
```

```
b=
 [[ 0  1  2  3  4  5  6  7  8  9]
 [10 11 12 13 14 15 16 17 18 19]
 [20 21 22 23 24 25 26 27 28 29]]
-------------

Transpuesta de b=
 [[ 0 10 20]
 [ 1 11 21]
 [ 2 12 22]
 [ 3 13 23]
 [ 4 14 24]
 [ 5 15 25]
 [ 6 16 26]
 [ 7 17 27]
 [ 8 18 28]
 [ 9 19 29]]
------------
```

In [17]:

```
#resize
b.resize([5,6])
print('b=\n',b)
```

```
b=
 [[ 0  1  2  3  4  5]
 [ 6  7  8  9 10 11]
 [12 13 14 15 16 17]
 [18 19 20 21 22 23]
 [24 25 26 27 28 29]]
```

In [ ]:

```

```