

In [24]:

```
import numpy as np

a = np.arange(9).reshape(3,3)
print('a =\n', a, '\n')

b = a*3
print('b =\n', b)

c = a*4
print('c =\n', c)
```

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

b =
[[ 0  3  6]
 [ 9 12 15]
 [18 21 24]]

c =
[[ 0  4  8]
 [12 16 20]
 [24 28 32]]
```

In [25]:

```
print('a =\n', a, '\n')
print('b =\n', b, '\n')
print('c =\n', c, '\n')

print('Apilamiento horizontal =\n', np.hstack((a,b,c)) )
```

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

b =
[[ 0  3  6]
 [ 9 12 15]
 [18 21 24]]

c =
[[ 0  4  8]
 [12 16 20]
 [24 28 32]]
```

```
Apilamiento horizontal =
[[ 0  1  2  0  3  6]
 [ 3  4  5  9 12 15]
 [ 6  7  8 18 21 24]]
```

In [26]:

```
print('a =\n', a, '\n')
print('b =\n', b, '\n')
print('c =\n', c, '\n')

print( 'Apilamiento horizontal con concatenate = \n',
np.concatenate((a,b,c), axis=1) )
```

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

```
b =
[[ 0  3  6]
 [ 9 12 15]
 [18 21 24]]
```

```
c =
[[ 0  4  8]
 [12 16 20]
 [24 28 32]]
```

```
Apilamiento horizontal con concatenate =
[[ 0  1  2  0  3  6]
 [ 3  4  5  9 12 15]
 [ 6  7  8 18 21 24]]
```

In [27]:

```
print('a =\n', a, '\n')
print('b =\n', b, '\n')
print('c =\n', c, '\n')

print( 'Apilamiento vertical =\n', np.vstack((a,b,c)) )
```

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

```
b =
[[ 0  3  6]
 [ 9 12 15]
 [18 21 24]]
```

```
c =
[[ 0  4  8]
 [12 16 20]
 [24 28 32]]
```

```
Apilamiento vertical =
[[ 0  1  2]
 [ 3  4  5]
 [ 6  7  8]
 [ 0  3  6]
 [ 9 12 15]
 [18 21 24]
 [ 0  4  8]
 [12 16 20]
 [24 28 32]]
```

In [28]:

```
print('a =\n', a, '\n')
print('b =\n', b, '\n')
print('c =\n', c, '\n')

print( 'Apilamiento vertical con concatenate =\n',
np.concatenate((a,b,c), axis=1) )
```

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

```
b =
[[ 0  3  6]
 [ 9 12 15]
 [18 21 24]]
```

```
c =
[[ 0  4  8]
 [12 16 20]
 [24 28 32]]
```

```
Apilamiento vertical con concatenate =
[[ 0  1  2  0  3  6  0  4  8]
 [ 3  4  5  9 12 15 12 16 20]
 [ 6  7  8 18 21 24 24 28 32]]
```

In [29]:

```
print('a =\n', a, '\n')
print('b =\n', b, '\n')
print('c =\n', c, '\n')

print( 'Apilamiento en profundidad =\n', np.dstack((a,b,c)) )
```

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

```
b =
[[ 0  3  6]
 [ 9 12 15]
 [18 21 24]]
```

```
c =
[[ 0  4  8]
 [12 16 20]
 [24 28 32]]
```

Apilamiento en profundidad =

```
[[[ 0  0  0]
 [ 1  3  4]
 [ 2  6  8]]
```

```
[[ 3  9 12]
 [ 4 12 16]
 [ 5 15 20]]
```

```
[[ 6 18 24]
 [ 7 21 28]
 [ 8 24 32]]]
```

In [30]:

```
print('a =\n', a, '\n')
print('b =\n', b, '\n')
print('c =\n', c, '\n')

print( 'Apilamiento por columnas =\n',
np.column_stack((a,b,c)) )
```

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

```
b =
[[ 0  3  6]
 [ 9 12 15]
 [18 21 24]]
```

```
c =
[[ 0  4  8]
 [12 16 20]
 [24 28 32]]
```

```
Apilamiento por columnas =
[[ 0  1  2  0  3  6  0  4  8]
 [ 3  4  5  9 12 15 12 16 20]
 [ 6  7  8 18 21 24 24 28 32]]
```

In [31]:

```
print('a =\n', a, '\n')
print('b =\n', b, '\n')
print('c =\n', c, '\n')

print( 'Apilamiento por filas =\n',
np.row_stack((a,b,c)) )
```

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

```
b =
[[ 0  3  6]
 [ 9 12 15]
 [18 21 24]]
```

```
c =
[[ 0  4  8]
 [12 16 20]
 [24 28 32]]
```

```
Apilamiento por filas =
[[ 0  1  2]
 [ 3  4  5]
 [ 6  7  8]
 [ 0  3  6]
 [ 9 12 15]
 [18 21 24]
 [ 0  4  8]
 [12 16 20]
 [24 28 32]]
```

In [32]:

```
print(a, '\n')
print(b, '\n')
print(c, '\n')

print('Array con división horizontal =\n', np.hsplit(a, 3), '\n')
print('Array con división horizontal =\n', np.hsplit(b, 3), '\n')
print('Array con división horizontal =\n', np.hsplit(c, 3), '\n')

print('Array con división horizontal, uso de split() =\n',
      np.split(a, 3, axis=1))
print('Array con división horizontal, uso de split() =\n',
      np.split(b, 3, axis=1))
print('Array con división horizontal, uso de split() =\n',
      np.split(c, 3, axis=1))
```



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

```
[[ 0  3  6]
 [ 9 12 15]
 [18 21 24]]
```

```
[[ 0  4  8]
 [12 16 20]
 [24 28 32]]
```

```
Array con división horizontal =
[array([[0],
        [3],
        [6]]), array([[1],
        [4],
        [7]]), array([[2],
        [5],
        [8]])]
```

```
Array con división horizontal =
[array([[ 0],
        [ 9],
        [18]]), array([[ 3],
        [12],
        [21]]), array([[ 6],
        [15],
        [24]])]
```

```
Array con división horizontal =
[array([[ 0],
        [12],
        [24]]), array([[ 4],
        [16],
        [28]]), array([[ 8],
        [20],
        [32]])]
```

```
Array con división horizontal, uso de split() =
[array([[0],
        [3],
        [6]]), array([[1],
        [4],
        [7]]), array([[2],
        [5],
        [8]])]
```

```
Array con división horizontal, uso de split() =
[array([[ 0],
        [ 9],
        [18]]), array([[ 3],
        [12],
        [21]]), array([[ 6],
        [15],
        [24]])]
```

```
Array con división horizontal, uso de split() =
[array([[ 0],
```

```
[12],
[24]]), array([[ 4],
[16],
[28]]), array([[ 8],
[20],
[32]])]
```

In [33]:

```
print(a, '\n')
print(b, '\n')
print(c, '\n')

print('División Vertical = \n', np.vsplit(a, 3), '\n')
print('División Vertical = \n', np.vsplit(b, 3), '\n')
print('División Vertical = \n', np.vsplit(c, 3), '\n')

print('Array con división vertical, uso de split() =\n',
np.split(a, 3, axis=0))
print('Array con división vertical, uso de split() =\n',
np.split(b, 3, axis=0))
print('Array con división vertical, uso de split() =\n',
np.split(c, 3, axis=0))
```

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

```
[[ 0  3  6]
 [ 9 12 15]
 [18 21 24]]
```

```
[[ 0  4  8]
 [12 16 20]
 [24 28 32]]
```

```
División Vertical =
[array([[0, 1, 2]]), array([[3, 4, 5]]), array([[6, 7, 8]])]
```

```
División Vertical =
[array([[0, 3, 6]]), array([[ 9, 12, 15]]), array([[18, 21, 24]])]
```

```
División Vertical =
[array([[0, 4, 8]]), array([[12, 16, 20]]), array([[24, 28, 32]])]
```

```
Array con división vertical, uso de split() =
[array([[0, 1, 2]]), array([[3, 4, 5]]), array([[6, 7, 8]])]
Array con división vertical, uso de split() =
[array([[0, 3, 6]]), array([[ 9, 12, 15]]), array([[18, 21, 24]])]
Array con división vertical, uso de split() =
[array([[0, 4, 8]]), array([[12, 16, 20]]), array([[24, 28, 32]])]
```

In [41]:

```
c = np.arange(27).reshape(3, 3, 3)

print(c, '\n')

print('División en profundidad =\n', np.dsplit(c,3), '\n')
```

```
[[[ 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]]]
```

```
División en profundidad =
[array([[ 0],
        [ 3],
        [ 6]],

       [[ 9],
        [12],
        [15]],

       [[18],
        [21],
        [24]]], array([[ 1],
        [ 4],
        [ 7]],

       [[10],
        [13],
        [16]],

       [[19],
        [22],
        [25]]], array([[ 2],
        [ 5],
        [ 8]],

       [[11],
        [14],
        [17]],

       [[20],
        [23],
        [26]]])]
```

In [43]:

```
print(b, '\n')
print('ndim: ', b.ndim)
```

```
[[ 0  3  6]
 [ 9 12 15]
 [18 21 24]]
```

ndim: 2

In [44]:

```
print(a, '\n')
print('size: ', b.size)
```

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

size: 9

In [45]:

```
print('itemsize: ', b.itemsize)
```

itemsize: 8

In [46]:

```
print(c, '\n')
print('nbytes: ', b.nbytes, '\n')

print('nbytes equivalente: ', c.size * c.itemsize)
```

```
[[[ 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]]]
```

nbytes: 72

nbytes equivalente: 216

In [48]:

```
b.resize(6,4)
print(b, '\n')
print('Transpuesta: ', b.T)
```

```
[[ 0  3  6  9]
 [12 15 18 21]
 [24  0  0  0]
 [ 0  0  0  0]
 [ 0  0  0  0]
 [ 0  0  0  0]]
```

```
Transpuesta: [[ 0 12 24  0  0  0]
 [ 3 15  0  0  0  0]
 [ 6 18  0  0  0  0]
 [ 9 21  0  0  0  0]]
```

In [49]:

```
c = np.array([1.j + 1, 2.j + 3])
print('Complejo: \n', c)
```

```
Complejo:
[1.+1.j 3.+2.j]
```

In [52]:

```
print('real: ', c.real, '\n')
print('imaginario: ', c.imag)
```

```
real: [1. 3.]
```

```
imaginario: [1. 2.]
```

In [53]:

```
print(a.dtype)
```

```
int64
```

In [54]:

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

for item in f: print (item)

print('\n')
print('Elemento 2: ', b.flat[2])

b.flat = 7
print(b, '\n')

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

```
[[0 1]
 [2 3]]
```

<numpy.flatiter object at 0x558e235fd7e0>

```
0
1
2
3
```

```
Elemento 2:  2
[[7 7]
 [7 7]]
```

```
[[7 1]
 [7 1]]
```

In []:

In []: