e-7

March 11, 2025

```
[1]: import numpy as np
     from numpy import ma
     from PIL import Image
     from IPython.display import display
[2]: a = ma.masked_array([1, 2, 3])
[3]: b = np.array(a) # subok has default value (fasle)
     print(type(b))
     print(type(a))
    <class 'numpy.ndarray'>
    <class 'numpy.ma.MaskedArray'>
[4]: c = np.array(a, subok=True) # if true, so used from base class
     print(type(c))
     print(type(a))
    <class 'numpy.ma.MaskedArray'>
    <class 'numpy.ma.MaskedArray'>
[5]: c = np.array([1, 2, 3])
     d = np.array(c) # copy have default value (True)
     e = np.array(c, copy=False) # so anychange wass occured in e, c changed too
     d[0] = 15
     print(c)
     print('----')
     print(d)
     print('*****')
     e[0] = 17
     print(c)
```

```
print('----')
print(e)

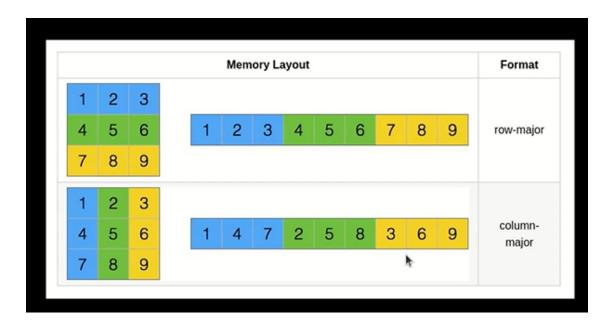
[1 2 3]
----
[15 2 3]
*****
[17 2 3]
----
[17 2 3]
[6]: display(Image.open('array_order.png'))
```

order: {'K', 'A', 'C', 'F'}, optional

Specify the memory layout of the array. If object is not an array, the newly created array will be in C order (row major) unless 'F' is specified, in which case it will be in Fortran order (column major). If object is an array the following holds.

order	no copy	copy=True
'K'	unchanged	F & C order preserved, otherwise most similar order
'A'	unchanged	F order if input is F and not C, otherwise C order
'C'	C order	C order
'F'	F order	F order

```
[7]: display(Image.open('array_order_2.png'))
```



[]: