In [22]:
```
import numpy as np
```

Method 1 ¶

In [23]:
```
Arr = np.arange(12)
Arr
```

Out[23]:
```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11])
```

In [24]:
```
import numpy as np
Arr = np.arange(12)
for val in Arr:
    print(val, end=' ')
```

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

Method 2

In [25]:
```
# we can also iterate the numpay array just like other
# programming languages like C++, C, etc.
```

In [26]:
```
import numpy as np
Arr = np.arange(12)
i = 0
while Arr[i] < Arr.size:
    print(Arr[i])
    i = i+1
    if(i==Arr.size):
        break
```

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

Method 3 : Iterating a Two-dimensional Array

In [27]:
```
Arr = np.arange(12).reshape(4,3)
for row in Arr:
    print(row)
```

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

Method 4: Iterate over each element on 2D array

In [44]:
```
Arr = np.arange(12).reshape(4,3)
for row in Arr:
    for cell in row:
        print(cell, end='\t')
    print("\n")
```

```
0	1	2	

3	4	5	

6	7	8	

9	10	11	
```

Method 5: Using Flatten method

In [55]:
```
Arr = np.arange(12).reshape(4,3)
for cell in Arr.flatten():
        print(cell, end=' ')
```

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

Method 6: Nditer Object

In [56]:
```
Arr
```

Out[56]:
```
array([[ 0,  1,  2],
       [ 3,  4,  5],
       [ 6,  7,  8],
       [ 9, 10, 11]])
```

In [57]:
```
Arr = np.arange(12).reshape(4,3)
for cell in np.nditer(Arr):
    print(cell, end=' ')
```

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

In [58]:
```
# C order iteration
Arr = np.arange(12).reshape(4,3)
for cell in np.nditer(Arr, order='C'):
    print(cell, end=' ')
```

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

In [59]:
```
# F order iteration
Arr = np.arange(12).reshape(4,3)
for cell in np.nditer(Arr, order='F'):
    print(cell, end=' ')
```

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

Modification to the Numpy array using nditer

In [60]:
```
for cell in np.nditer(Arr):
    cell[...] =cell*2
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-60-8d1b4acca918> in <module>
      1 for cell in np.nditer(Arr):
----> 2     cell[...] =cell*2

ValueError: assignment destination is read-only
```

In [61]:
```
for cell in np.nditer(Arr, op_flags=['readwrite']):
    cell[...] =cell-3
Arr
```

Out[61]:
```
array([[-3, -2, -1],
       [ 0,  1,  2],
       [ 3,  4,  5],
       [ 6,  7,  8]])
```

In [ ]:
```

```