In [1]:
```python
import numpy as np
```

In [2]:
```python
#create a 0 dimension array
a = np.array(0)
```

In [3]:
```python
#create a 1 dimension array
b = np.array([1, 2, 3, 4, 5])
```

In [4]:
```python
#create a 2 dimension array
c = np.array([[1, 3, 5, 7], [2, 4, 6, 8]])
```

Why NumPy not lists

In [5]:
```python
# Get Dimension
a.ndim
```

Out[5]:
```
0
```

In [6]:
```python
# Get Shape
b.shape
```

Out[6]:
```
(5,)
```

In [7]:
```python
# Get Type
a.dtype
```

Out[7]:
```
dtype('int32')
```

In [8]:
```python
# Get Size
a.itemsize
```

Out[8]:
```
4
```

In [9]:
```python
# Get total size
a.nbytes
```

Out[9]:
```
4
```

In [10]:
```python
# Get number of elements
a.size
```

Out[10]:
```
1
```

Accessing/Changing specific elements, rows, columns, etc

In [11]:
```
a = np.array([[1,2,3,4,5,6,7],[8,9,10,11,12,13,14]])
print(a)
```

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

- Numpy Array Exercises

In [12]:
```
# Create an array with variable name a and the following contents (shape (3, 4)):
a = np.array([[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12]])
```

In [13]:
```
#What is the array shape?
#Array shape is the representation of its row and width ie: (row, column)
a.shape
```

Out[13]:
```
(3, 4)
```

In [14]:
```
#What is the array ndim?
# Ndim is the number of dimensions in an array
a.ndim
```

Out[14]:
```
2
```

In [15]:
```
#How about the len of the array?
# len is the length or number of elements in an array (outermost elements or rows)
len(a)
```

Out[15]:
```
3
```

Creating arrays using functions

In [16]:
```
#Create a 1D array from 2 through 5 inclusive.
np.linspace(2, 5)
```

Out[16]:
```
array([2.        , 2.06122449, 2.12244898, 2.18367347, 2.24489796,
       2.30612245, 2.36734694, 2.42857143, 2.48979592, 2.55102041,
       2.6122449 , 2.67346939, 2.73469388, 2.79591837, 2.85714286,
       2.91836735, 2.97959184, 3.04081633, 3.10204082, 3.16326531,
       3.2244898 , 3.28571429, 3.34693878, 3.40816327, 3.46938776,
       3.53061224, 3.59183673, 3.65306122, 3.71428571, 3.7755102 ,
       3.83673469, 3.89795918, 3.95918367, 4.02040816, 4.08163265,
       4.14285714, 4.20408163, 4.26530612, 4.32653061, 4.3877551 ,
       4.44897959, 4.51020408, 4.57142857, 4.63265306, 4.69387755,
       4.75510204, 4.81632653, 4.87755102, 4.93877551, 5.        ])
```

In [17]:
```
#Make an array with 10 equally spaced elements between 2 and 5 inclusive.
np.linspace(2, 5, 10)
```

Out[17]:
```
array([2.        , 2.33333333, 2.66666667, 3.        , 3.33333333,
       3.66666667, 4.        , 4.33333333, 4.66666667, 5.        ])
```

In [18]:
```
#Make an all-ones array shape (4, 4).
np.ones([4,4])
```

Out[18]:
```
array([[1., 1., 1., 1.],
       [1., 1., 1., 1.],
       [1., 1., 1., 1.],
       [1., 1., 1., 1.]])
```

In [19]:
```
#Make an identity array shape (6, 6).
np.identity(6)
```

Out[19]:
```
array([[1., 0., 0., 0., 0., 0.],
       [0., 1., 0., 0., 0., 0.],
       [0., 0., 1., 0., 0., 0.],
       [0., 0., 0., 1., 0., 0.],
       [0., 0., 0., 0., 1., 0.],
       [0., 0., 0., 0., 0., 1.]])
```

In [20]:
```
#Make this array with a single Python / numpy command:
# 1  0  0
# 0  2  0
# 0  0  3
np.array([[1, 0, 0], [0, 2, 0], [0, 0, 3]])
```

Out[20]:
```
array([[1, 0, 0],
       [0, 2, 0],
       [0, 0, 3]])
```

In [21]:
```
#Make a Array of random numbers shape 3, 5
np.random.rand(3,5)
```

Out[21]:
```
array([[0.12847827, 0.36892995, 0.80495675, 0.69669976, 0.09042565],
       [0.14321002, 0.27691425, 0.99529548, 0.08099657, 0.45228937],
       [0.70651963, 0.22588775, 0.00701879, 0.29196232, 0.81509621]])
```

Indexing and slicing, array creation

In [22]:
```
#Create the following array, call this a:
# 2  7 12  0
# 3  9  3  4
# 4  0  1  3

a = np.array([
    [2, 7, 12, 0],
    [3, 9, 3, 4],
    [4, 0, 1, 3]
])

#Get the 2nd row of a
print ("2nd row of a is: ", a[1])

#Get the 3rd column of a
print ("3rd column of a is: ", a[:,2])

#Create the following arrays (with correct data types):
# [[1, 1, 1, 1],
#  [1, 1, 1, 1],
#  [1, 1, 1, 2],
#  [1, 6, 1, 1]]
a = np.array([
     [1, 1, 1, 1],
     [1, 1, 1, 1],
     [1, 1, 1, 2],
     [1, 6, 1, 1]
])

# [[0., 0., 0., 0., 0.],
#  [2., 0., 0., 0., 0.],
#  [0., 3., 0., 0., 0.],
#  [0., 0., 4., 0., 0.],
#  [0., 0., 0., 5., 0.],
#  [0., 0., 0., 0., 6.]]
b = np.array(
    [[0., 0., 0., 0., 0.],
     [2., 0., 0., 0., 0.],
     [0., 3., 0., 0., 0.],
     [0., 0., 4., 0., 0.],
     [0., 0., 0., 5., 0.],
     [0., 0., 0., 0., 6.]]
)
print ("Datatype of a is:", a.dtype)
print ("Datatype of b is:", b.dtype)
```

```
2nd row of a is:  [3 9 3 4]
3rd column of a is:  [12  3  1]
Datatype of a is: int32
Datatype of b is: float64
```