In [1]:

```
import numpy as np
```

In [2]:

```
b = np.array([1,2,3,4,5,6])
```

In [3]:

```
print(b)
```

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

In [8]:

```
c = np.array([[1,2,3,4,5,6],[7,8,9,3,2,1]])
print(c)
```

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

In [9]:

```
m = [[1,2,3,4,5,6],[7,8,9,3,2,1]]
print(m)
```

```
[[1, 2, 3, 4, 5, 6], [7, 8, 9, 3, 2, 1]]
```

In [10]:

```
print(type(b))
```

```
<class 'numpy.ndarray'>
```

In [11]:

```
print(type(m))
```

```
<class 'list'>
```

In [12]:

```
print(b.ndim)
```

```
2
```

In [13]:

```
print(b)
```

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

In [14]:

```
d = np.array([[[1,2,3],[4,5,6]],[[7,8,9],[6,5,4]]])
print(d)
```

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

 [[7 8 9]
  [6 5 4]]]
```

In [15]:

```
print(d.ndim)
```

```
3
```

In [16]:

```
# itemsize()  - calculate the byte size of each element

print(b)
```

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

In [17]:

```
print(b.itemsize)
```

```
4
```

In [18]:

```
# dtype - used to understand the data type of given element
```

In [19]:

```
print(b.dtype)
```

```
int32
```

In [20]:

```
# shape - returns a tuple consisting of array dimensions
print(b.shape)
```

```
(2, 6)
```

In [22]:

```
# reshape = used to reshape the array
print(b.reshape(3,4))
```

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

In [ ]: