```
import numpy as np
a = np.arange(0, 10) #one-dimensional array
```

```
a.reshape(2,5) #two-diensional array defining row num and col num, respectively
creates a different array than the original one
```

```
b = np.arange(10,20)
```

```
listA = list(range(0,10))
listB = list(range(10,20))
listA + listB
```

```
np.where(a>5)
np.nonzero(a)
reshape = makes a COPY of the array
```

```
d = np.array([np.arange(0,5), np.arange(0,5)]) #two dimensional array
```

```
e = np.array(np.arange(0,20)).reshape(2,2,5) #three-dimensional array; 2 rows,
5 columns
e
```

```
#get num 2,3,7,8 select all rows -> do start num to end num of cols
#num of rows and cols start at 0; last num exclusive
b = a.reshape(2,5)
b
b[::, 2:4:]
#row, col intvl
```

```
a[5:8] = 12 #replaces value for 5th index in slice of array
#says that value index range 5 - 8 will be 12

a[5:8] with no equal will create a slice array consisting of index 5 - 8
```

```
c_slice = a.copy()[7:9]
creates another array based on a copy of a, which means original a still exists
change value of slice will change original array
```

```
1. slicing 2-D array
In: a[3:7, :]
In: a[:, 3:7]

Does a[0:1, ] produce the result as a[0,]?
No bc first one produces two rows w/ all columns
second one produce one row w/ all columns

Write the code to select a single row or column?
a[0, ::] single row
```

a[::, 0] single column

Write the code to select the rows in odd numbers?
a[::2, ::]

Use negative number to output a row from backward?
a[0, -1::]

2. slicing a 3-D array
ex, m = np.array([np.array([np.arange(4), np.arange(4), np.arange(4)])
In: a[0:2, 1, 2]

What will show running this code?
row 0 and 1 from 0 axis, then 3 columns across

How can you slice out the highlighted area? (in ppt)