slicing

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

In [2]:
```
# We create a 4 x 5 ndarray that contains integers from 0 to 19
X = np.arange(20).reshape(4, 5)
print(X)
```

```
[[ 0  1  2  3  4]
 [ 5  6  7  8  9]
 [10 11 12 13 14]
 [15 16 17 18 19]]
```

In [3]:
```
# We select all the elements that are in the 2nd through 4th rows and i
n the 3rd to 5th columns
Z = X[1:4,2:5]
print(Z)
Z[0,1] = 20
print(Z)
```

```
[[ 7  8  9]
 [12 13 14]
 [17 18 19]]
[[ 7 20  9]
 [12 13 14]
 [17 18 19]]
```

x value will change as z[0,1] = 20 so we use #copy

In [ ]:
```
print(X)
```

In [ ]:

```
# We can select the same elements as above using method 2
W = X[1:,2:5]
print(W)
```

In [ ]:

```
# We select all the elements that are in the 1st through 3rd rows and i
n the 3rd to 4th columns
Y = X[:3,2:5]
print(Y)
```

In [ ]:

```
# We select all the elements in the 3rd row
v = X[2,:]
print(v)
```

In [ ]:

```
# We select all the elements in the 3rd column
q = X[:,2]
print(q)
```

In [ ]:

```
# We select all the elements in the 3rd column but return a rank 2 ndar
ray
R = X[:,2:3]
print(R)
```

copy

In [ ]:

```
X = np.arange(20).reshape(4, 5)
print(X)
```

In [ ]:

```
# create a copy of the slice using the np.copy() function
Z = np.copy(X[1:4,2:5])
```

In [ ]:

```
#  create a copy of the slice using the copy as a method
W = X[1:4,2:5].copy()
```

In [ ]:
```
Z[0,1] = 20
```

In [ ]:
```
print(X)
print(Z)
```

indices , slice using rank 1 in rank 2

In [ ]:
```
X = np.arange(20).reshape(4, 5)
print(X)
```

In [ ]:
```
# We create a rank 1 ndarray that will serve as indices to select eleme
nts from X
indices = np.array([1,3])
```

In [ ]:
```
# We use the indices ndarray to select the 2nd and 4th row of X
Y = X[indices,:]
print(Y)
```

In [ ]:
```
# We use the indices ndarray to select the 2nd and 4th column of X
Z = X[:, indices]
print(Z)
```

np.diag(ndarray, k=N) default is k=0, k > 0 are used to select elements in diagonals above the main diagonal, k < 0 are used to select elements in diagonals below the main diagonal

In [4]:
```
# We create a 4 x 5 ndarray that contains integers from 0 to 19
X = np.arange(25).reshape(5, 5)
print(X)
```

```
[[ 0  1  2  3  4]
```

```
 [ 5  6  7  8  9]
 [10 11 12 13 14]
 [15 16 17 18 19]
 [20 21 22 23 24]]
```

```
In [5]: # We print the elements in the main diagonal of X
        print('z =', np.diag(X))
```

```
z = [ 0  6 12 18 24]
```

```
In [6]: # We print the elements above the main diagonal of X
        print('y =', np.diag(X, k=1))
```

```
y = [ 1  7 13 19]
```

```
In [7]: # We print the elements below the main diagonal of X
        print('w = ', np.diag(X, k=-1))
```

```
w =  [ 5 11 17 23]
```