Exercise Set 2

Hina Arora

- numpy arrays are more compact (and therefore more memory efficient) than python lists; unlike python lists, numpy is constrained to arrays that all contain the same type.

Copy

Create a numpy array as below:

```
np.random.seed(0)
arr = np.array(np.random.normal(size=(3,5)))
arr
```

```
array([[ 1.76405235,  0.40015721,  0.97873798,  2.2408932 ,  1.86755799],
       [-0.97727788,  0.95008842, -0.15135721, -0.10321885,  0.4105985 ],
       [ 0.14404357,  1.45427351,  0.76103773,  0.12167502,  0.44386323]])
```

Question

(1) print column 1
(2) print row 0
(3) print element at row 0 and column 1
(4) print columns 1 and 3
(5) print rows 0 and 2
(6) print rows 0 and 2 & columns 1 and 3

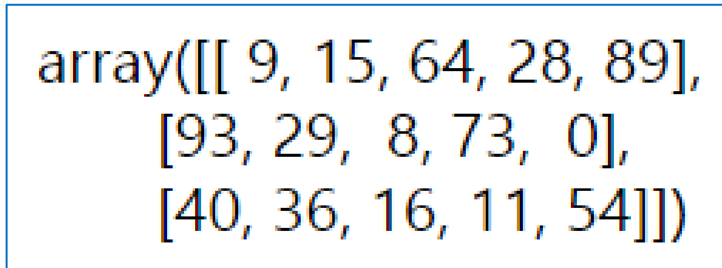
Answer

```
arr[:,1]

arr[0,:]
# arr[0]

arr[0,1] # multi-dimensional indexing
# arr[0][1] # chained indexing

arr[:,[1,3]]

arr[[0,2],:]
# arr[[0,2]]

arr[np.ix_([0,2],[1,3])]
```

Copy

Define a 3 x 5 array as follows:

```
np.random.seed(10)
nr, nc = 3, 5
arr = np.random.randint(0, 100, 15).reshape(nr, nc)
arr
```

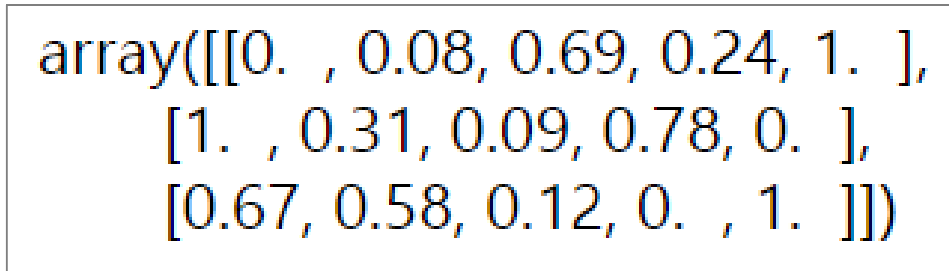


```
array([[ 9, 15, 64, 28, 89],
       [93, 29,  8, 73,  0],
       [40, 36, 16, 11, 54]])
```

Question

Min-Max scale each row:

- for each row, find the row minimum (rmin) and the row maximum (rmax)
- then for each element (e) in each row, transform as follows: (e – rmin)/(rmax – rmin)
- this will scale each row to the range [0, 1]



```
array([[0.  , 0.08, 0.69, 0.24, 1.  ],
       [1.  , 0.31, 0.09, 0.78, 0.  ],
       [0.67, 0.58, 0.12, 0.  , 1.  ]])
```

Answer

```
rmin = np.min(arr, axis=1).reshape(nr,1)  # row mins
rmax = np.max(arr, axis=1).reshape(nr,1) # row maxs
mmsarr = (arr - rmin) / (rmax - rmin)
np.round(mmsarr, 2)
```