

NumPy Master Class

Lecture.7 Indexing and Slicing ndarrays

Lecture.7 Indexing and Slicing ndarrays

- Basic Indexing and Slicing

Indexing and Slicing Vector ndarrays

```
import numpy as np
```

```
a = np.arange(10)
print(f"ndarray: \n{a}\n")
```

```
ndarray:
[0 1 2 3 4 5 6 7 8 9]
```

```
print("a[0] : ", a[0])
print("a[2] : ", a[2])
```

```
a[0] : 0
a[2] : 2
```

```
print("a[-1] : ", a[-1]) # the last element
print("a[-2] : ", a[-2]) # the 2nd last element
```

```
a[-1] : 9
a[-2] : 8
```

Lecture.7 Indexing and Slicing ndarrays

- Basic Indexing and Slicing

Indexing and Slicing Vector ndarrays

```
import numpy as np
```

```
a = np.arange(5)
```

```
print(f"ndarray: \n{a}\n")
```

```
ndarray:
```

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

```
for data in a:
```

```
    print(data)
```

```
0
```

```
1
```

```
2
```

```
3
```

```
4
```


Lecture.7 Indexing and Slicing ndarrays

- Basic Indexing and Slicing

Indexing and Slicing Vector ndarrays

[start : end]

```
import numpy as np
```

```
a = np.arange(10)
print(f"ndarray: \n{a}\n")
```

```
ndarray:
[0 1 2 3 4 5 6 7 8 9]
```

```
print("a[0:3] : ", a[0:3])
```

```
a[0:3] :  [0 1 2]
```

```
print("a[3:7] : ", a[3:7])
```

```
a[3:7] :  [3 4 5 6]
```

```
print("a[5:-1] : ", a[5:-1])
```

```
a[5:-1] :  [5 6 7 8]
```

Lecture.7 Indexing and Slicing ndarrays

- Basic Indexing and Slicing

Indexing and Slicing Vector ndarrays

```
import numpy as np
```

```
a = np.arange(10)
print(f"ndarray: \n{a}\n")
```

```
ndarray:
[0 1 2 3 4 5 6 7 8 9]
```

```
print("a[2:] : ", a[2:])
```

```
a[2:] : [2 3 4 5 6 7 8 9]
```

```
print("a[-2:] : ", a[-2:]) # the last two elements
```

```
a[-2:] : [8 9]
```

```
print("a[:5] : ", a[:5])
```

```
a[:5] : [0 1 2 3 4]
```

```
print("a[:-3] : ", a[:-3])
```

```
a[:-3] : [0 1 2 3 4 5 6]
```

```
print("a[:] : ", a[:])
```

```
a[:] : [0 1 2 3 4 5 6 7 8 9]
```


Lecture.7 Indexing and Slicing ndarrays

- Basic Indexing and Slicing

Indexing and Slicing Vector ndarrays

[start : end : step]

```
import numpy as np
```

```
a = np.arange(10)
print(f"ndarray: \n{a}\n")
```

```
ndarray:
[0 1 2 3 4 5 6 7 8 9]
```

```
print("a[2:7:2] : ", a[2:7:2])
```

```
a[2:7:2] :  [2 4 6]
```

```
print("a[::3] : ", a[::3])
```

```
a[::3] :  [0 3 6 9]
```

Lecture.7 Indexing and Slicing ndarrays

- Basic Indexing and Slicing

Indexing and Slicing Vector ndarrays

```
import numpy as np
```

```
a = np.arange(10)  
print(f"ndarray: \n{a}\n")
```

```
ndarray:  
[0 1 2 3 4 5 6 7 8 9]
```

```
print("a[7:2:-1] : ", a[7:2:-1])
```

```
a[7:2:-1] :  [7 6 5 4 3]
```

```
print("a[::-1] : ", a[::-1])
```

```
a[::-1] :  [9 8 7 6 5 4 3 2 1 0]
```

```
print("a[8:3:-2] : ", a[8:3:-2])
```

```
a[8:3:-2] :  [8 6 4]
```

```
print("a[::-3] : ", a[::-3])
```

```
a[::-3] :  [9 6 3 0]
```


Lecture.7 Indexing and Slicing ndarrays

- Basic Indexing and Slicing

Indexing and Slicing Vector ndarrays

```
import numpy as np
```

```
a = np.arange(10)  
indices = np.array([0, 3, 6, -1])  
print(f"ndarray: \n{a}\n")
```

```
ndarray:  
[0 1 2 3 4 5 6 7 8 9]
```

```
print(a[[0, 3, 6, -1]])
```

```
[0 3 6 9]
```

```
print(a[indices])
```

```
[0 3 6 9]
```

```
print(a[a%3==0])
```

```
[0 3 6 9]
```


Lecture.7 Indexing and Slicing ndarrays

- Basic Indexing and Slicing

Element Assignment with Slicing

```
a = list(range(1, 11))  
print(a)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

```
for data_idx in range(5):  
    a[data_idx] = 0
```

```
print(a)
```

```
[0, 0, 0, 0, 0, 6, 7, 8, 9, 10]
```

```
a = np.arange(1, 11)  
print(a)
```

```
[ 1  2  3  4  5  6  7  8  9 10]
```

```
a[:5] = 0  
print(a)
```

```
[ 0  0  0  0  0  6  7  8  9 10]
```

Lecture.7 Indexing and Slicing ndarrays

- Basic Indexing and Slicing

Element Assignment with Slicing

```
a = np.arange(1, 11)  
print(a)
```

```
[ 1  2  3  4  5  6  7  8  9 10]
```

```
a[::2] = 200  
print(a)
```

```
[200   2 200   4 200   6 200   8 200  10]
```

```
a[5:-1:3] = 300  
print(a)
```

```
[200   2 200   4 200 300 200   8 300  10]
```


Lecture.7 Indexing and Slicing ndarrays

- Basic Indexing and Slicing

Indexing and Slicing Matrix ndarrays

```
a = [[0, 1, 2], [3, 4, 5], [6, 7, 8]]
```

```
print(a[0])    [0, 1, 2]  
print(a[1])    [3, 4, 5]  
print(a[2])    [6, 7, 8]
```

```
a = [[0, 1, 2], [3, 4, 5], [6, 7, 8]]
```

```
print(a[0][0], a[0][1], a[0][2])    0 1 2  
print(a[1][0], a[1][1], a[0][2])    3 4 2  
print(a[2][0], a[2][1], a[2][2])    6 7 8
```

Lecture.7 Indexing and Slicing ndarrays

- Basic Indexing and Slicing

Indexing and Slicing Matrix ndarrays

```
import numpy as np
```

```
a = np.arange(12).reshape((3, 4))  
print(f"ndarray: \n{a}\n")
```

```
print("a[0]: ", a[0])
```

```
print("a[1]: ", a[1])
```

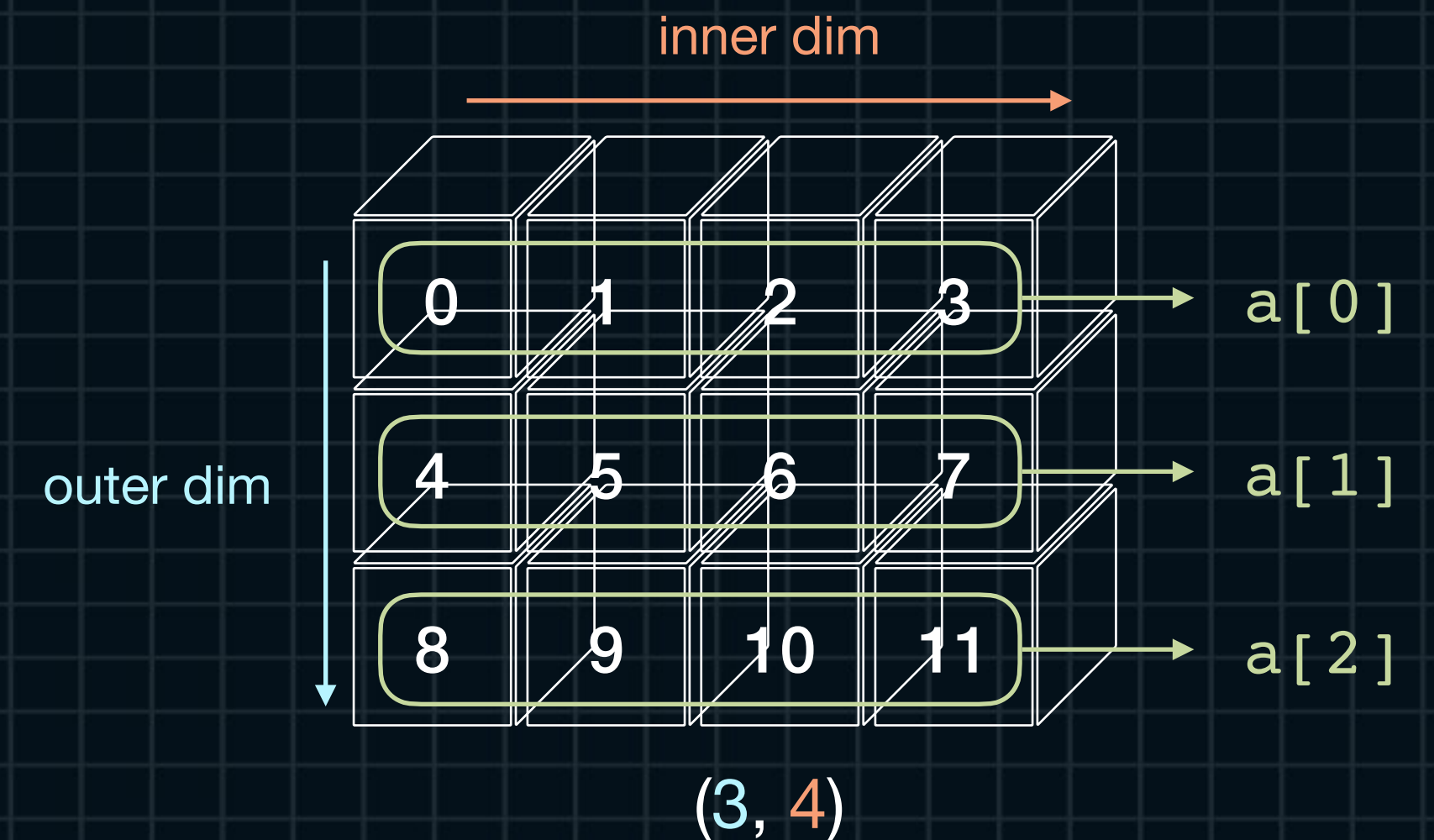
```
print("a[2]: ", a[2])
```

```
ndarray:  
[[ 0  1  2  3]  
 [ 4  5  6  7]  
 [ 8  9 10 11]]
```

```
a[0]: [0 1 2 3]
```

```
a[1]: [4 5 6 7]
```

```
a[2]: [ 8  9 10 11]
```



Lecture.7 Indexing and Slicing ndarrays

- Basic Indexing and Slicing

Indexing and Slicing Matrix ndarrays

```
import numpy as np
```

```
a = np.arange(12).reshape((3, 4))  
print(f"ndarray: \n{a}\n")
```

```
ndarray:  
[[ 0  1  2  3]  
 [ 4  5  6  7]  
 [ 8  9 10 11]]
```

```
for data in a:  
    print(data)
```

```
[0 1 2 3]  
[4 5 6 7]  
[ 8  9 10 11]
```

Lecture.7 Indexing and Slicing ndarrays

- Basic Indexing and Slicing

Indexing and Slicing Matrix ndarrays

```
import numpy as np

a = np.arange(9).reshape((3, 3))
print(f"ndarray: \n{a}\n")
```

```
ndarray:
[[0 1 2]
 [3 4 5]
 [6 7 8]]
```

```
print(a[0, 0], a[0, 1], a[0, 2])
print(a[1, 0], a[1, 1], a[1, 2])
print(a[2, 0], a[2, 1], a[2, 2])
```

```
0 1 2
3 4 5
6 7 8
```

```
import numpy as np

a = np.arange(9).reshape((3, 3))
print(f"ndarray: \n{a}\n")
```

```
ndarray:
[[0 1 2]
 [3 4 5]
 [6 7 8]]
```

```
print(a[0][0], a[0][1], a[0][2])
print(a[1][0], a[1][1], a[1][2])
print(a[2][0], a[2][1], a[2][2])
```

```
0 1 2
3 4 5
6 7 8
```


Lecture.7 Indexing and Slicing ndarrays

- Basic Indexing and Slicing

Indexing and Slicing Matrix ndarrays

```
import numpy as np
```

```
a = np.arange(12).reshape((3, 4))  
print(f"ndarray: \n{a}\n")
```

ndarray:

```
[[ 0  1  2  3]  
 [ 4  5  6  7]  
 [ 8  9 10 11]]
```

```
print("a[0, 1:] : ", a[0, 1:])
```

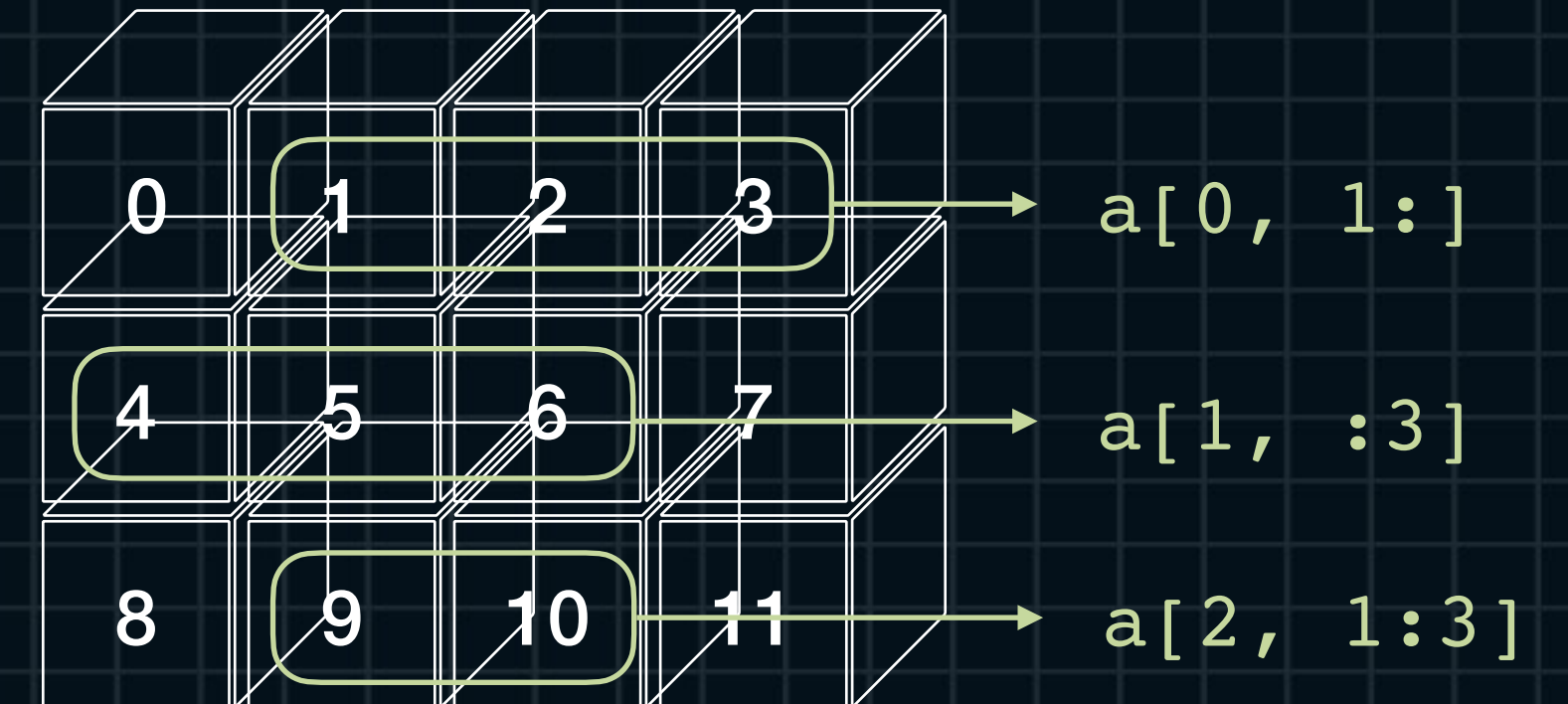
a[0, 1:] : [1 2 3]

```
print("a[1, :3] : ", a[1, :3])
```

a[1, :3] : [4 5 6]

```
print("a[2, 1:3] : ", a[2, 1:3])
```

a[2, 1:3] : [9 10]



Lecture.7 Indexing and Slicing ndarrays

- Basic Indexing and Slicing

Indexing and Slicing Matrix ndarrays

```
import numpy as np

a = np.arange(12).reshape((4, 3))
print(f"ndarray: \n{a}\n")
```

```
print("a[1:, 0] : ", a[1:, 0])
```

```
print("a[:3, 1] : ", a[:3, 1])
```

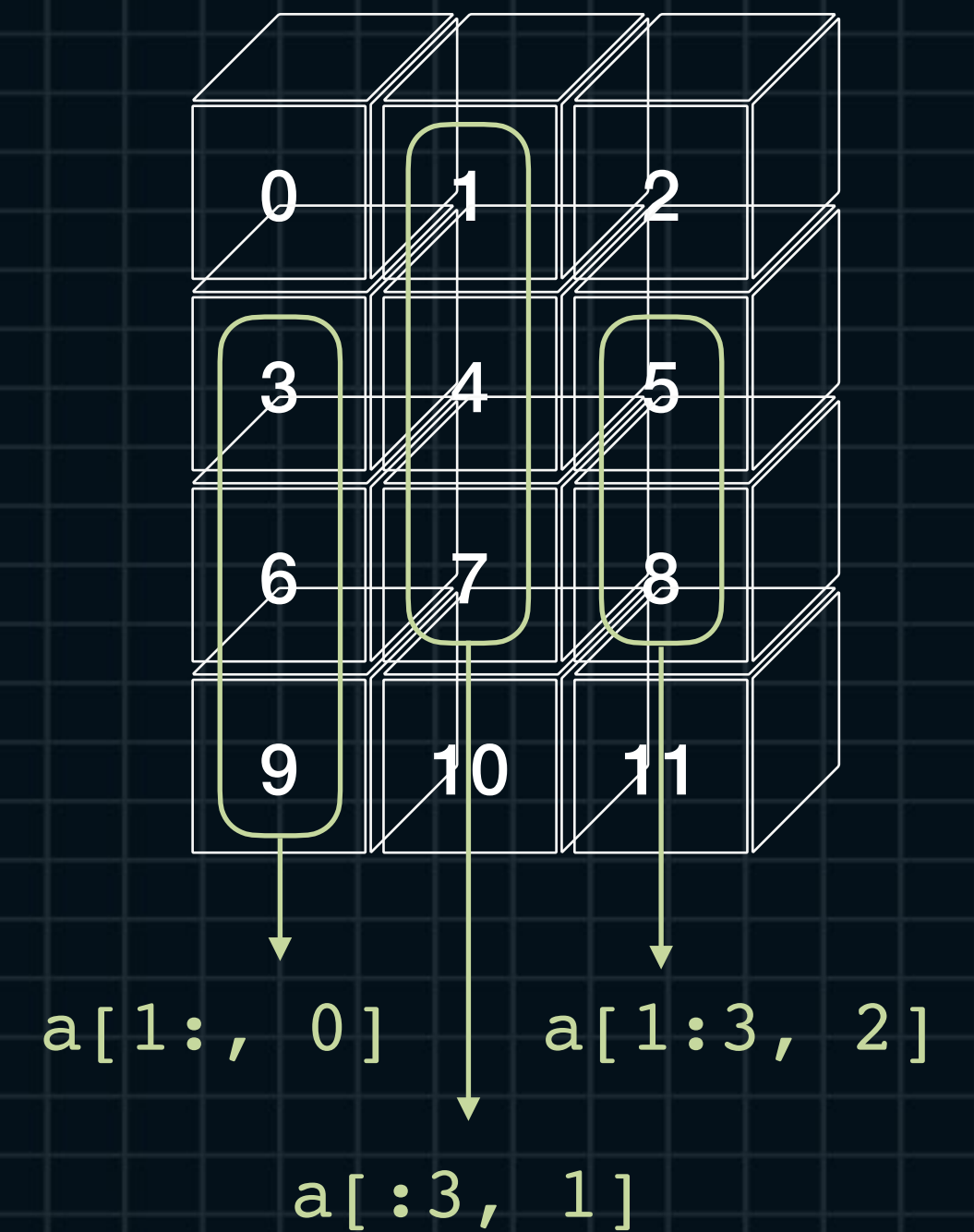
```
print("a[1:3, 2] : ", a[1:3, 2])
```

```
ndarray:
[[ 0  1  2]
 [ 3  4  5]
 [ 6  7  8]
 [ 9 10 11]]
```

```
a[1:, 0] : [3 6 9]
```

```
a[:3, 1] : [1 4 7]
```

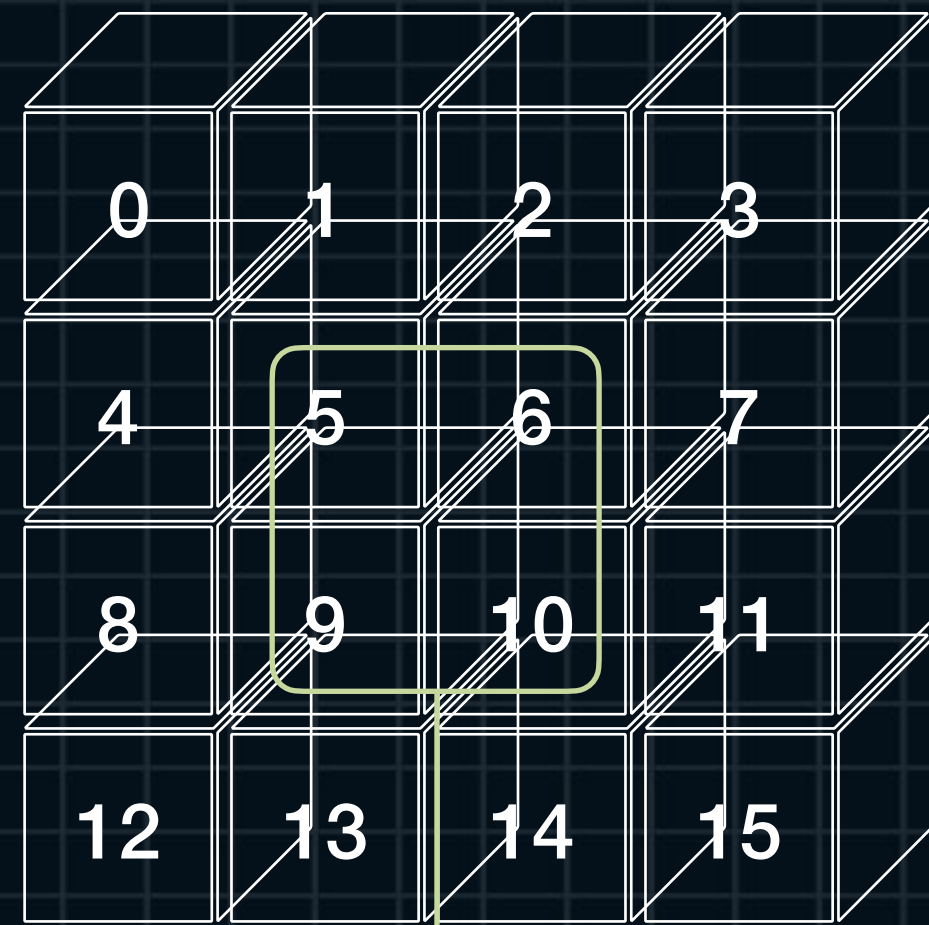
```
a[1:3, 2] : [5 8]
```



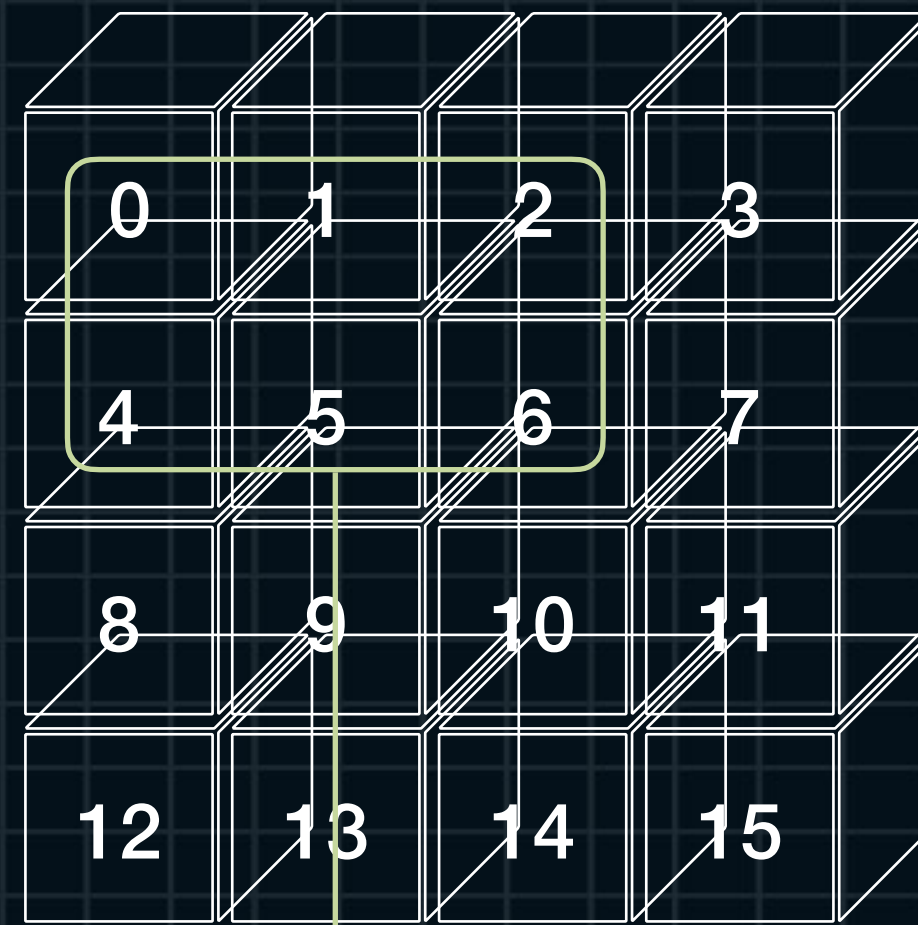
Lecture.7 Indexing and Slicing ndarrays

- Basic Indexing and Slicing

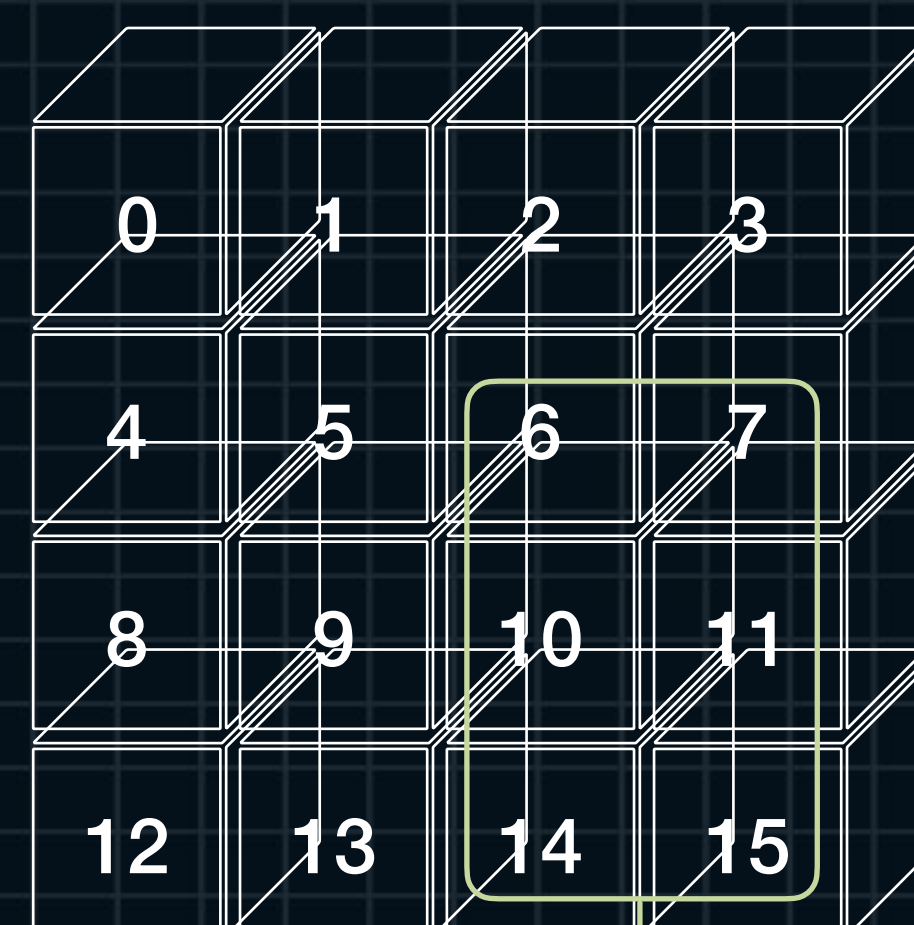
Indexing and Slicing Matrix ndarrays



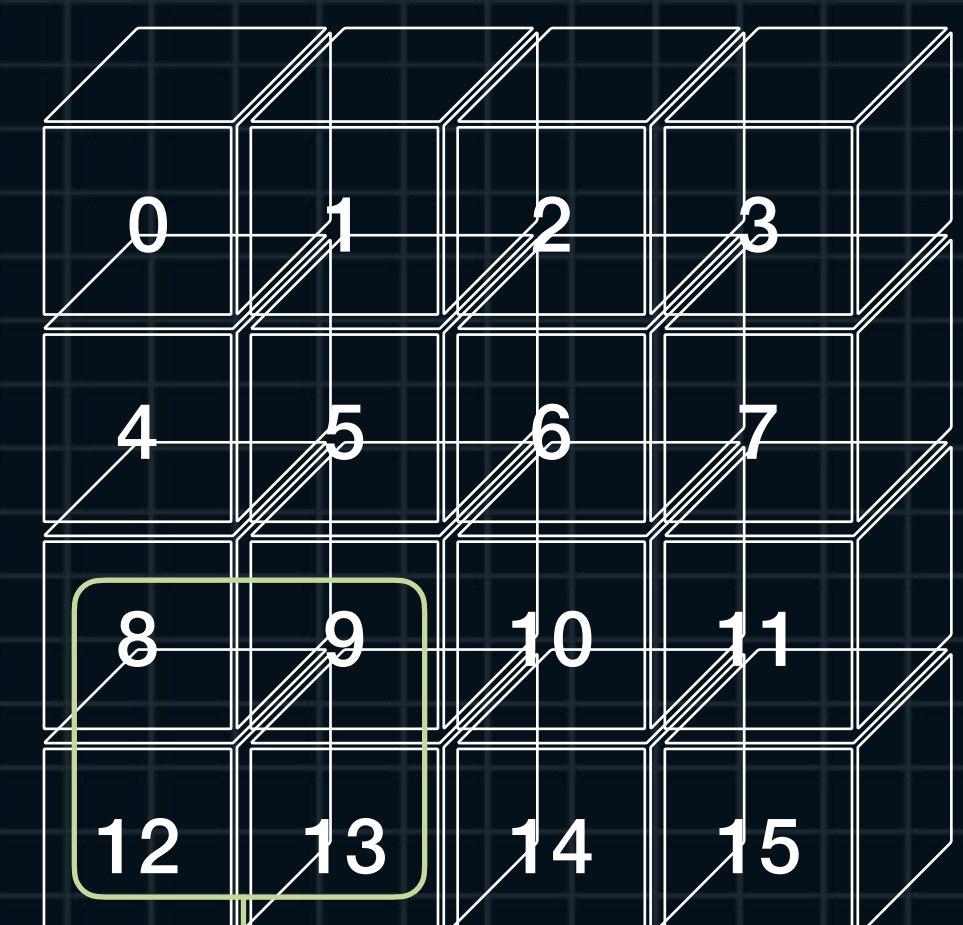
`a[1:3, 1:3]`



`a[:2, :3]`



`a[1:, 2:]`



`a[2:, :-2]`

Lecture.7 Indexing and Slicing ndarrays

- Basic Indexing and Slicing

Indexing and Slicing Matrix ndarrays

```
import numpy as np
```

```
a = np.arange(16).reshape((4, 4))  
print(f"ndarray: \n{a}\n")
```

```
print("a[1:3, 1:3] : \n", a[1:3, 1:3])
```

```
print("a[:2, :3] : \n", a[:2, :3])
```

```
print("a[1:, 2:] : \n", a[1:, 2:])
```

```
print("a[2:, :-2] : \n", a[2:, :-2])
```

ndarray:

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

a[1:3, 1:3] :

```
[[ 5  6]  
 [ 9 10]]
```

a[:2, :3] :

```
[[0 1 2]  
 [4 5 6]]
```

a[1:, 2:] :

```
[[ 6  7]  
 [10 11]  
 [14 15]]
```

a[2:, :-2] :

```
[[ 8  9]  
 [12 13]]
```


Lecture.7 Indexing and Slicing ndarrays

- Basic Indexing and Slicing

Indexing and Slicing Matrix ndarrays

```
import numpy as np
```

```
image = np.arange(9).reshape((3, 3))  
print(f"ndarray: \n{image}\n")
```

```
ndarray:  
[[0 1 2]  
 [3 4 5]  
 [6 7 8]]
```

```
horizontal_flip = image[:, ::-1]  
print(f"horizontal_flip: \n{horizontal_flip}\n")
```

```
horizontal_flip:  
[[2 1 0]  
 [5 4 3]  
 [8 7 6]]
```

```
vertical_flip = image[::-1, :]  
print(f"vertical_flip: \n{vertical_flip}\n")
```

```
vertical_flip:  
[[6 7 8]  
 [3 4 5]  
 [0 1 2]]
```

```
rotation = image[::-1, ::-1]  
print(f"rotation: \n{rotation}\n")
```

```
rotation:  
[[8 7 6]  
 [5 4 3]  
 [2 1 0]]
```

Lecture.7 Indexing and Slicing ndarrays

- Basic Indexing and Slicing

Indexing and Slicing Matrix ndarrays

```
import numpy as np
```

```
a = np.arange(16).reshape((4, 4))  
print(f"ndarray: \n{a}\n")
```

ndarray:

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

```
print("a[0, :] : \n", a[0, :])
```

a[0, :] :
[0 1 2 3]

```
print("a[0, ...] : \n", a[0, ...])
```

a[0, ...] :
[0 1 2 3]

```
print("a[:, 1] : \n", a[:, 1])
```

a[:, 1] :
[1 5 9 13]

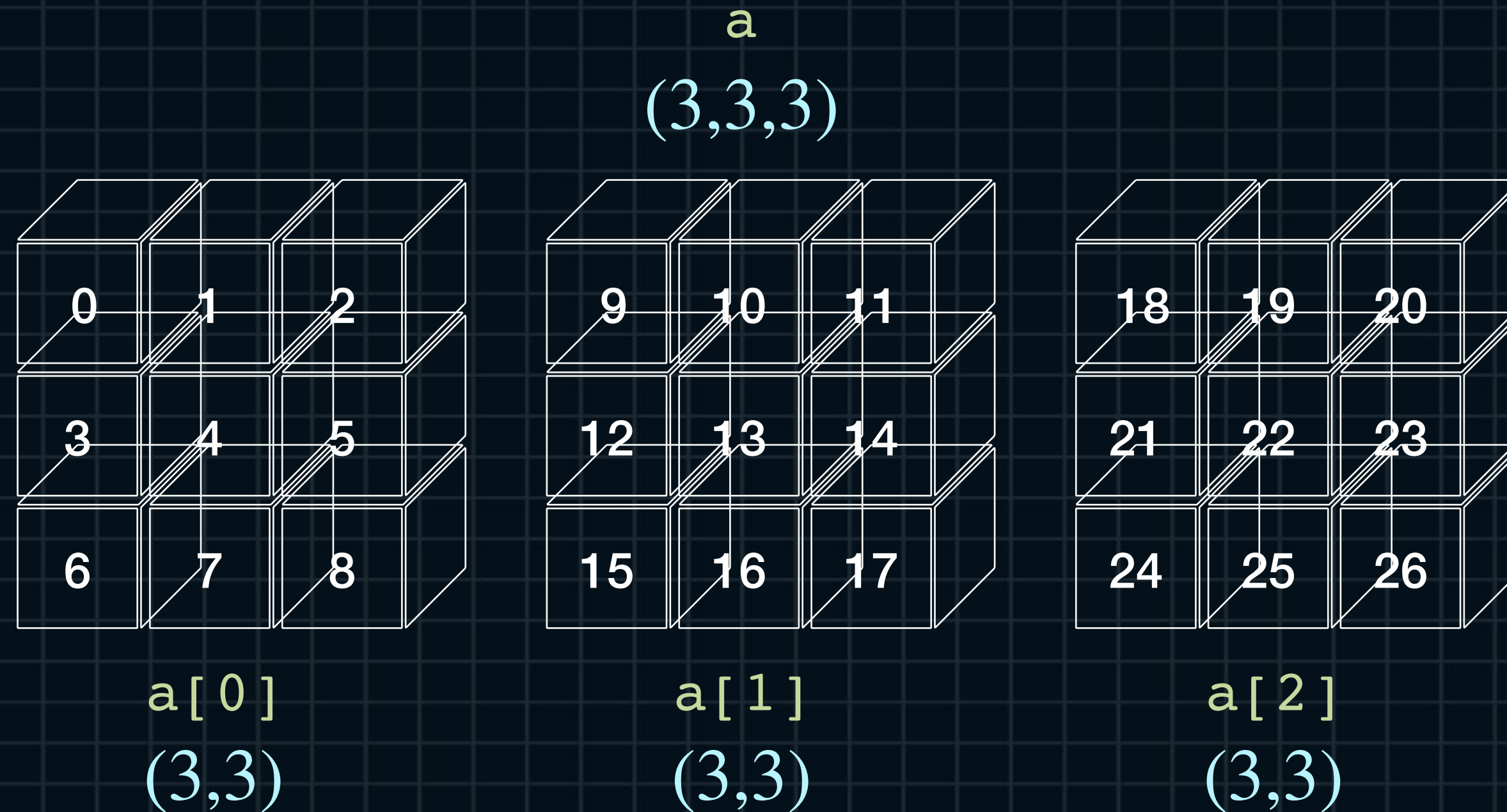
```
print("a[..., 1] : \n", a[..., 1])
```

a[..., 1] :
[1 5 9 13]

Lecture.7 Indexing and Slicing ndarrays

- Basic Indexing and Slicing

Indexing and Slicing 3rd Order Tensor ndarrays



Lecture.7 Indexing and Slicing ndarrays

- Basic Indexing and Slicing

Indexing and Slicing 3rd Order Tensor ndarrays

```
import numpy as np
```

```
a = np.arange(3*3*3).reshape((3, 3, 3))
```

```
print(f"ndarray: \n{a}\n")
```

```
ndarray:
```

```
[[[ 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 25 26]]]
```

```
print("a[0] : \n", a[0])
```

```
a[0] :
[[0 1 2]
 [3 4 5]
 [6 7 8]]
```

```
print("a[1] : \n", a[1])
```

```
a[1] :
[[ 9 10 11]
 [12 13 14]
 [15 16 17]]
```

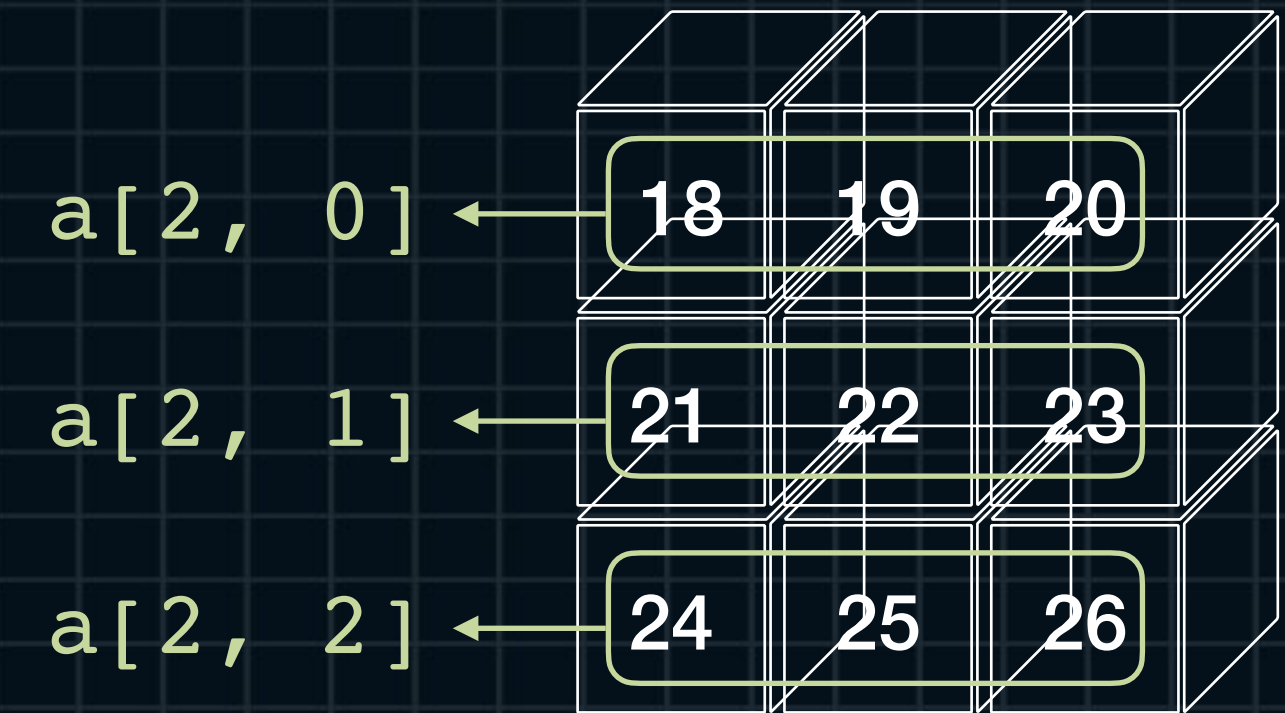
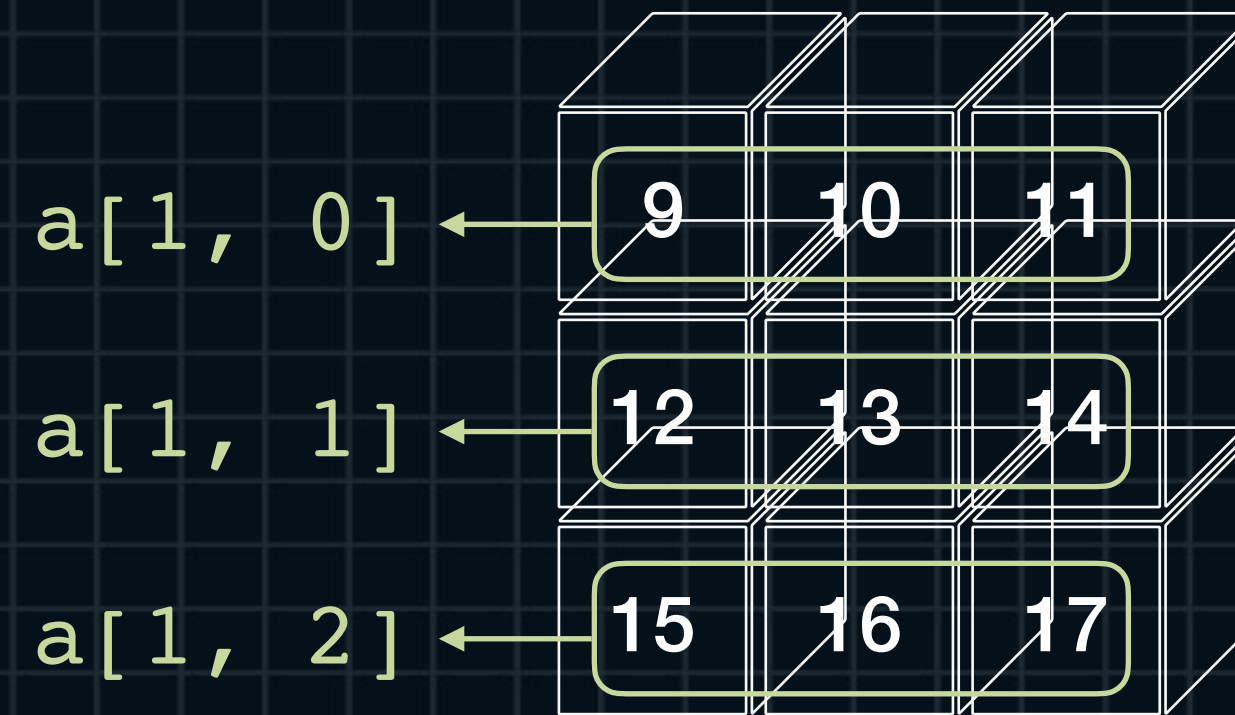
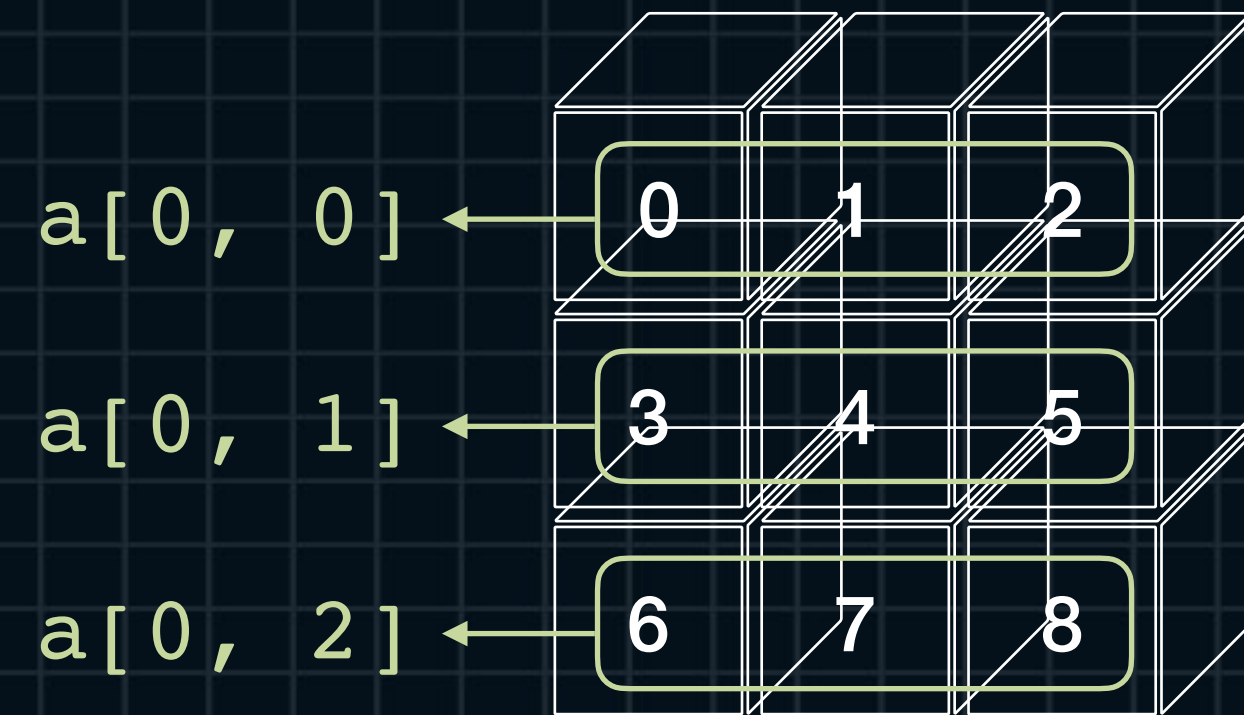
```
print("a[2] : \n", a[2])
```

```
a[2] :
[[18 19 20]
 [21 22 23]
 [24 25 26]]
```


Lecture.7 Indexing and Slicing ndarrays

- Basic Indexing and Slicing

Indexing and Slicing 3rd Order Tensor ndarrays



Lecture.7 Indexing and Slicing ndarrays

- Basic Indexing and Slicing

Indexing and Slicing 3rd Order Tensor ndarrays

```
import numpy as np
```

```
a = np.arange(3*3*3).reshape((3, 3, 3))  
print(f"ndarray: \n{a}\n")
```

```
ndarray:  
[[[ 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 25 26]]]
```

```
print("a[0, 0] : \n", a[0, 0])  
print("a[0, 1] : \n", a[0, 1])  
print("a[0, 2] : \n", a[0, 2], '\n')
```

```
a[0, 0] :  
[0 1 2]  
a[0, 1] :  
[3 4 5]  
a[0, 2] :  
[6 7 8]
```

```
print("a[1, 0] : \n", a[1, 0])  
print("a[1, 1] : \n", a[1, 1])  
print("a[1, 2] : \n", a[1, 2], '\n')
```

```
a[1, 0] :  
[ 9 10 11]  
a[1, 1] :  
[12 13 14]  
a[1, 2] :  
[15 16 17]
```

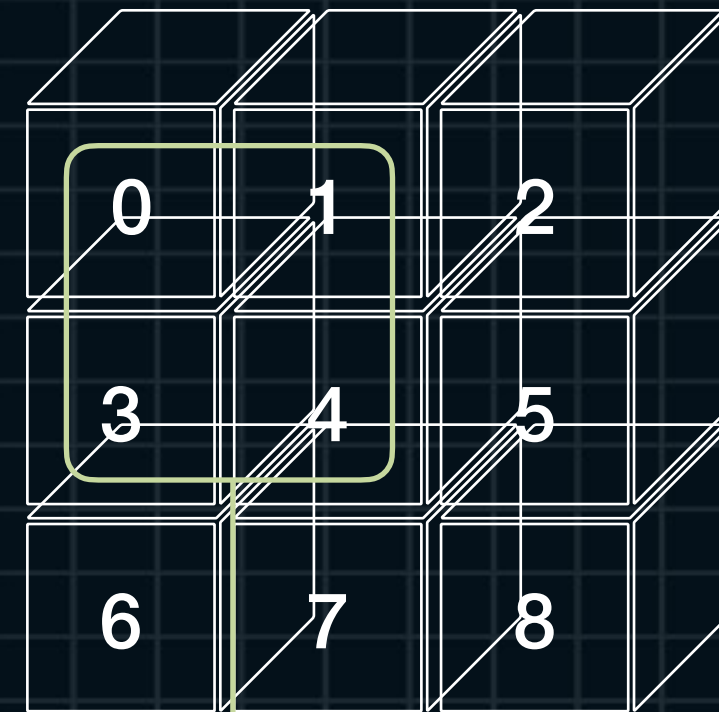
```
print("a[2, 0] : \n", a[2, 0])  
print("a[2, 1] : \n", a[2, 1])  
print("a[2, 2] : \n", a[2, 2])
```

```
a[2, 0] :  
[18 19 20]  
a[2, 1] :  
[21 22 23]  
a[2, 2] :  
[24 25 26]
```

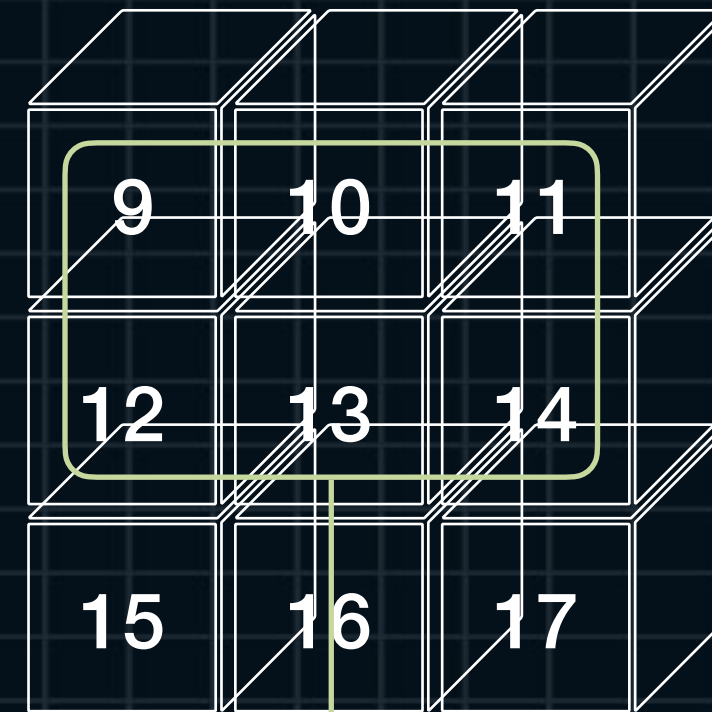

Lecture.7 Indexing and Slicing ndarrays

- Basic Indexing and Slicing

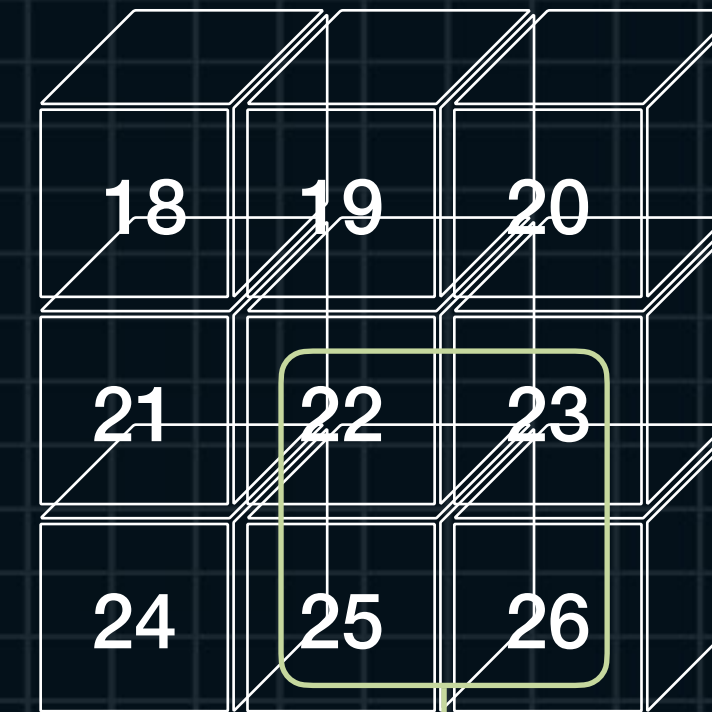
Indexing and Slicing 3rd Order Tensor ndarrays



`a[0, :2, :2]`



`a[1, :2, :]`



`a[2, -2:, -2:]`

Lecture.7 Indexing and Slicing ndarrays

- Basic Indexing and Slicing

Indexing and Slicing 3rd Order Tensor ndarrays

```
import numpy as np
```

```
a = np.arange(3*3*3).reshape((3, 3, 3))
```

```
print(f"ndarray: \n{a}\n")
```

```
ndarray:
```

```
[[[ 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 25 26]]]
```

```
print("a[0, :2, :2] : \n", a[0, :2, :2])
```

```
a[0, :2, :2] :
[[0 1]
 [3 4]]
```

```
print("a[1, :2, :] : \n", a[1, :2, :])
```

```
a[1, :2, :] :
[[ 9 10 11]
 [12 13 14]]
```

```
print("a[2, -2:, -2:] : \n", a[2, -2:, -2:])
```

```
a[2, -2:, -2:] :
[[22 23]
 [25 26]]
```


Lecture.7 Indexing and Slicing ndarrays

- Basic Indexing and Slicing

Indexing and Slicing 3rd Order Tensor ndarrays

```
import numpy as np
```

```
images = np.random.normal(size=(32, 100, 200))
```

```
print("image set: ", images.shape)
```

```
image set: (32, 100, 200)
```

```
image0 = images[0, :, :]
```

```
print(image0.shape) (100, 200)
```

```
image0 = images[0, ...]
```

```
print(image0.shape) (100, 200)
```

Lecture.7 Indexing and Slicing ndarrays

- Basic Indexing and Slicing

Indexing and Slicing 3rd Order Tensor ndarrays

```
import numpy as np
```

```
images = np.random.normal(size=(32, 100, 200))
```

```
print("image set: ", images.shape)
```

```
image set: (32, 100, 200)
```

```
col0 = images[:, :, 0]
```

```
print(col0.shape)
```

```
(32, 100)
```

```
col0 = images[:, :, 0]
```

```
print(col0.shape, '\n')
```

```
(32, 100)
```

```
image0_col0 = images[0, :, 0]
```

```
print(image0_col0.shape)
```

```
(100,)
```

```
image0_col0 = images[0, :, 0]
```

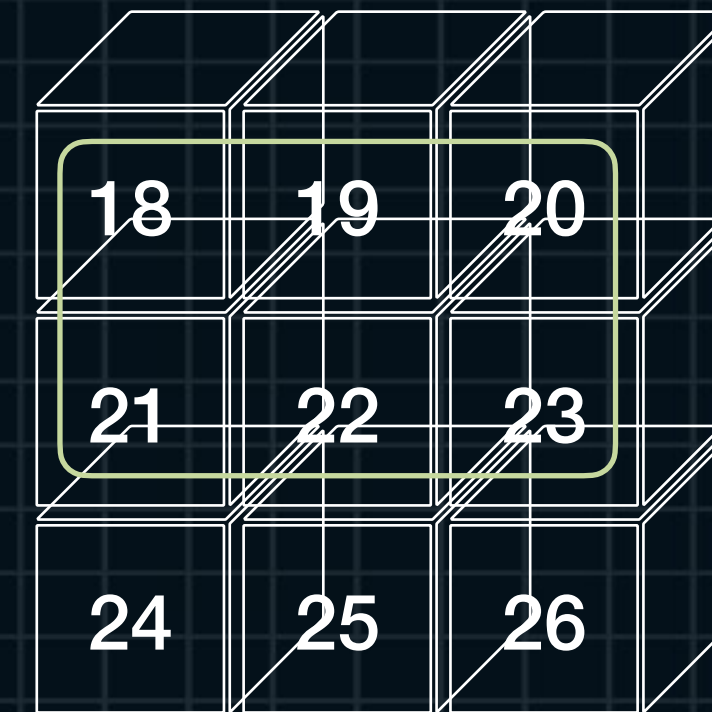
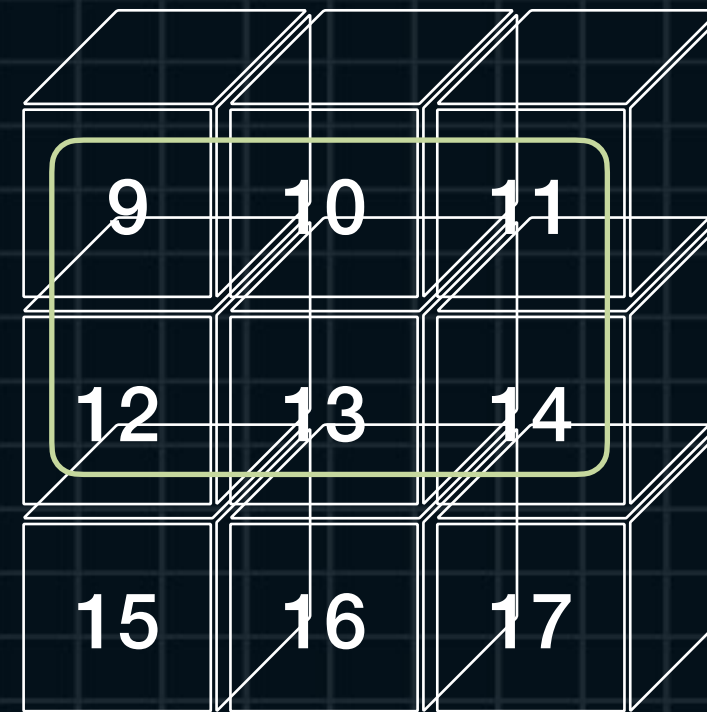
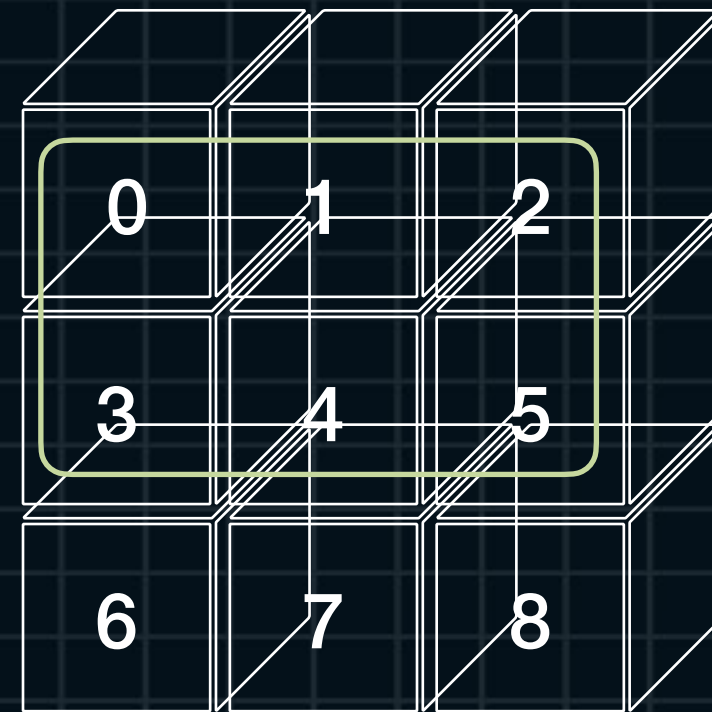
```
print(image0_col0.shape)
```

```
(100,)
```


Lecture.7 Indexing and Slicing ndarrays

- Basic Indexing and Slicing

Exercises



```
import numpy as np
```

```
a = np.arange(3*3*3).reshape((3, 3, 3))  
print(f"ndarray: \n{a}\n")
```

```
ndarray:  
[[[ 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 25 26]]]
```

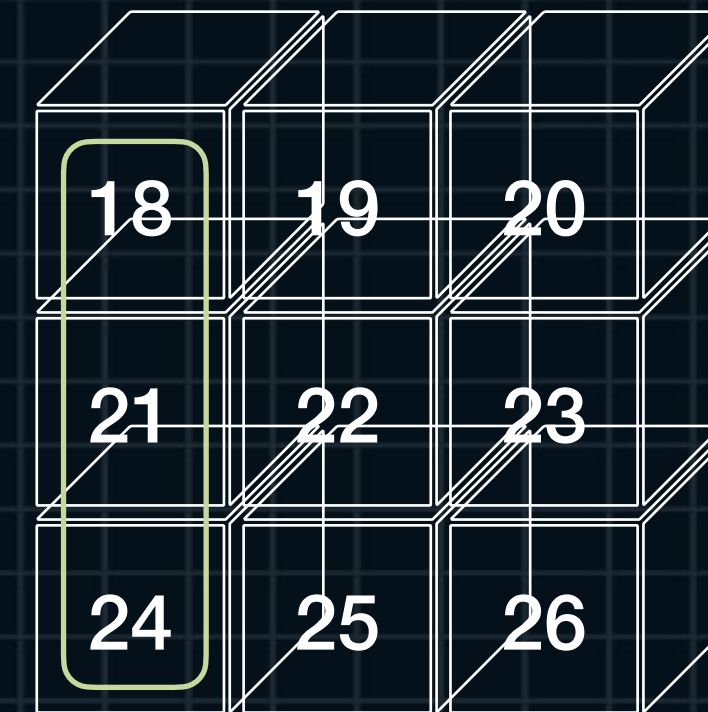
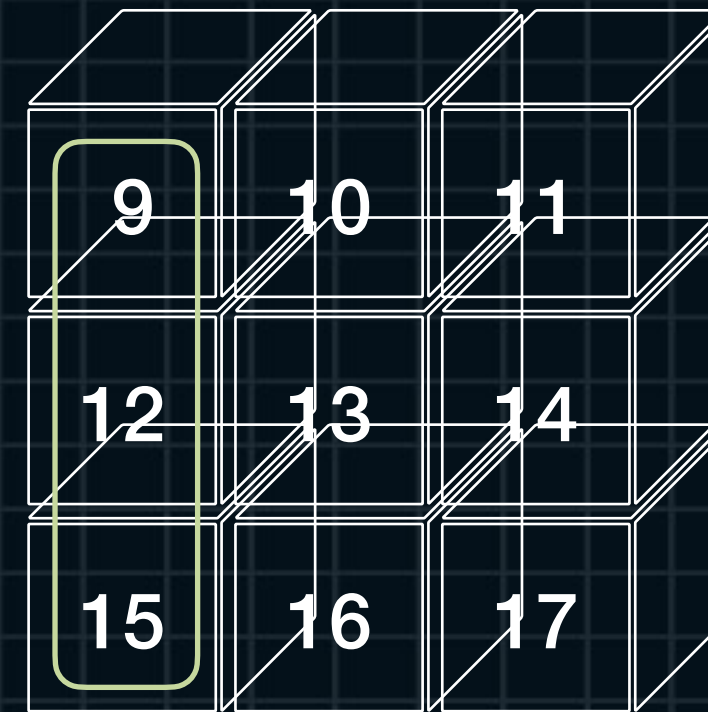
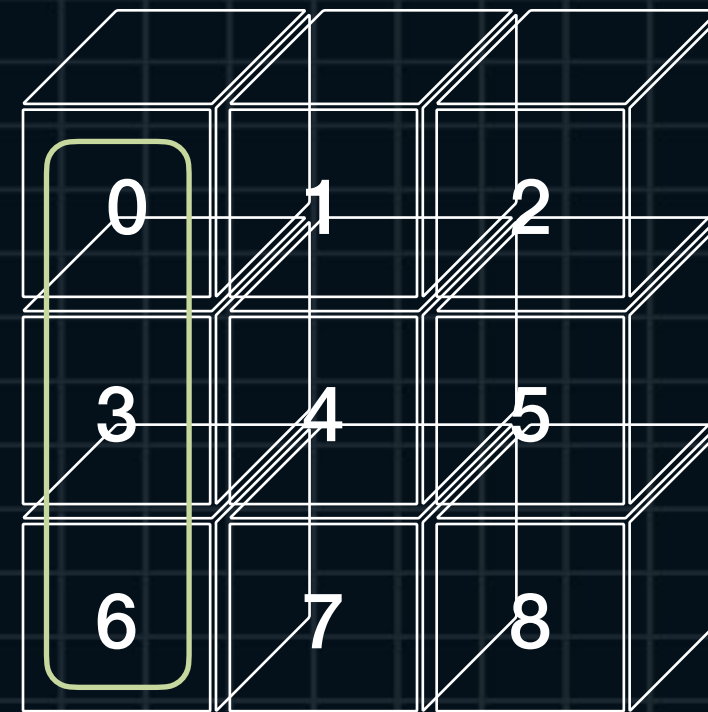
```
print("a[:, :2, :] : \n", a[:, :2, :])
```

```
a[:, :2, :] :  
[[[ 0  1  2]  
  [ 3  4  5]]  
  
 [[ 9 10 11]  
  [12 13 14]]  
  
 [[18 19 20]  
  [21 22 23]]]
```

Lecture.7 Indexing and Slicing ndarrays

- Basic Indexing and Slicing

Exercises



```
import numpy as np
```

```
a = np.arange(3*3*3).reshape((3, 3, 3))
```

```
print(f"ndarray: \n{a}\n")
```

```
ndarray:
[[[ 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 25 26]]]
```

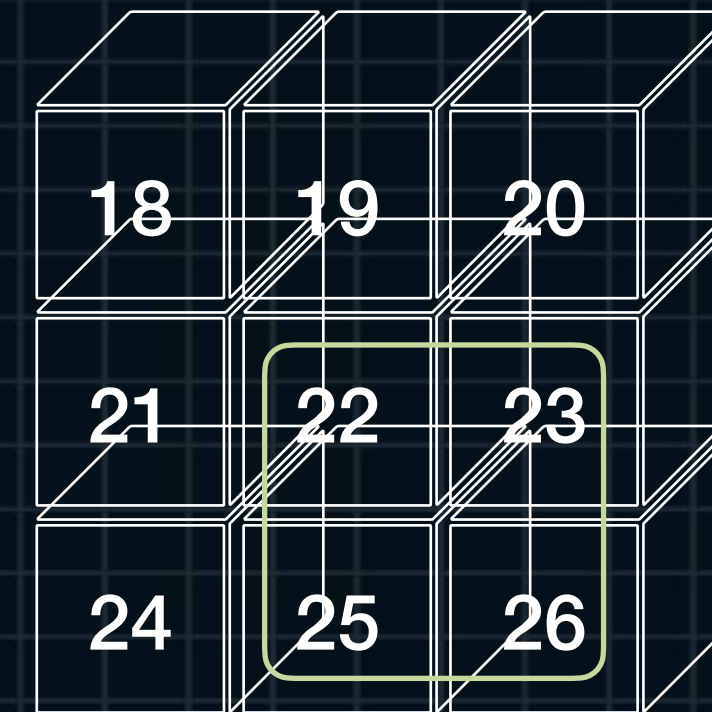
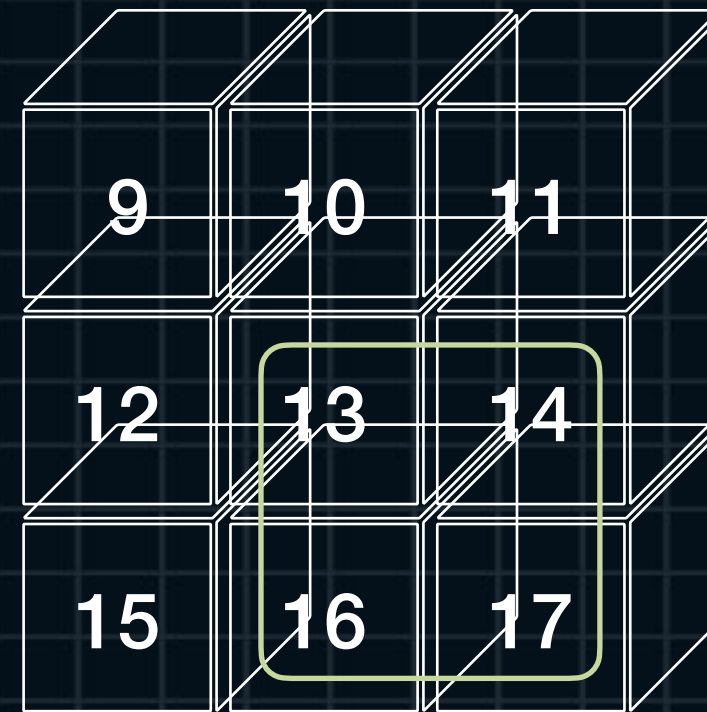
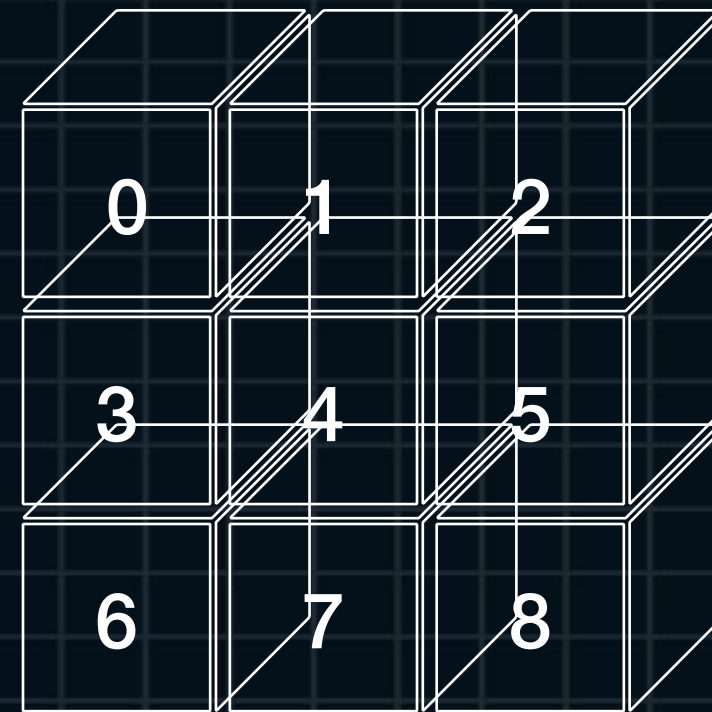
```
print("a[:, :, 0] : \n", a[:, :, 0])
```

```
a[:, :, 0] :
[[ 0  3  6]
 [ 9 12 15]
 [18 21 24]]
```


Lecture.7 Indexing and Slicing ndarrays

- Basic Indexing and Slicing

Exercises



```
import numpy as np
```

```
a = np.arange(3*3*3).reshape((3, 3, 3))  
print(f"ndarray: \n{a}\n")
```

```
ndarray:  
[[[ 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 25 26]]]
```

```
print("a[1:, -2:, -2:] : \n", a[1:, -2:, -2:])
```

```
a[1:, -2:, -2:] :  
[[[13 14]  
  [16 17]]  
  
 [[22 23]  
  [25 26]]]
```

Lecture.7 Indexing and Slicing ndarrays

- Basic Indexing and Slicing

Exercises

```
import numpy as np

image = np.random.normal(size=(3, 500, 300))

image_r = image[0]
image_g = image[1]
image_b = image[2]

print(image_r.shape, image_g.shape, image_b.shape)
(500, 300) (500, 300) (500, 300)
```

```
import numpy as np

image = np.random.normal(size=(3, 500, 300))

image_r, image_g, image_b = image
print(image_r.shape, image_g.shape, image_b.shape)
(500, 300) (500, 300) (500, 300)
```


Lecture.7 Indexing and Slicing ndarrays

- Basic Indexing and Slicing

Exercises

```
import numpy as np
```

```
image = np.random.normal(size=(3, 500, 300))
```

```
top_left = image[:, :100, :100]
```

```
top_right = image[:, :100, -100:]
```

```
bottom_left = image[:, -100:, :100]
```

```
bottom_right = image[:, -100:, -100:]
```

```
print(top_left.shape, top_right.shape)
```

```
(3, 100, 100) (3, 100, 100)
```

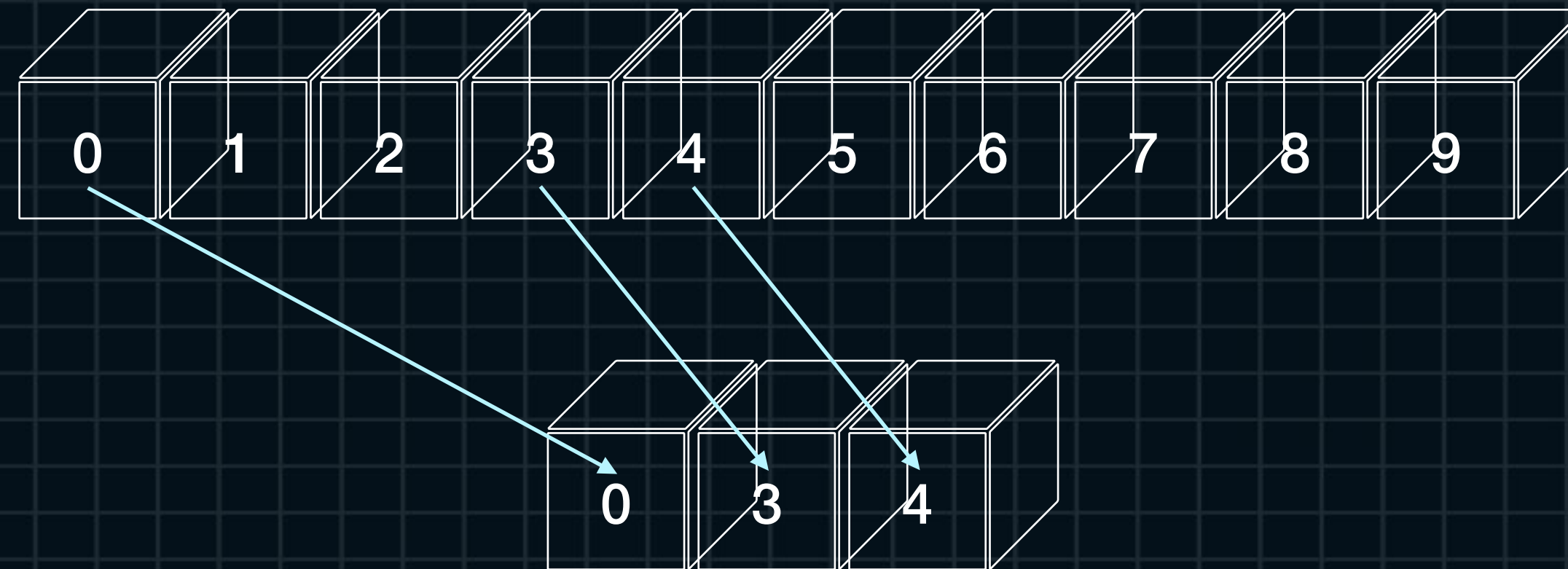
```
print(bottom_left.shape, bottom_right.shape)
```

```
(3, 100, 100) (3, 100, 100)
```

Lecture.7 Indexing and Slicing ndarrays

- Indexing with int ndarrays

Indexing and Slicing Vector ndarrays



```
import numpy as np
```

```
a = np.arange(10)  
print(f"ndarray: \n{a}\n")
```

```
ndarray:  
[0 1 2 3 4 5 6 7 8 9]
```

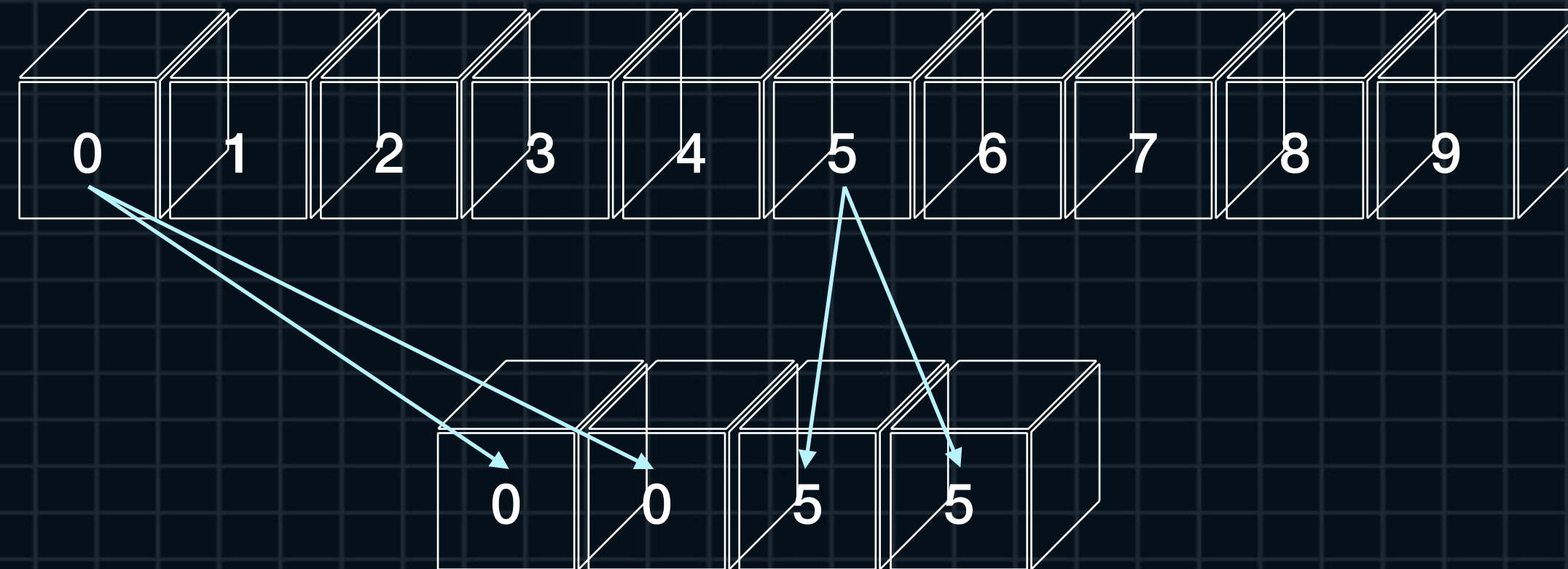
```
indices = np.array([0, 3, 4])  
print(a[indices])
```

```
[0 3 4]
```


Lecture.7 Indexing and Slicing ndarrays

- Indexing with int ndarrays

Indexing and Slicing Vector ndarrays



```
import numpy as np
```

```
a = np.arange(10)  
print(f"ndarray: \n{a}\n")
```

```
ndarray:  
[0 1 2 3 4 5 6 7 8 9]
```

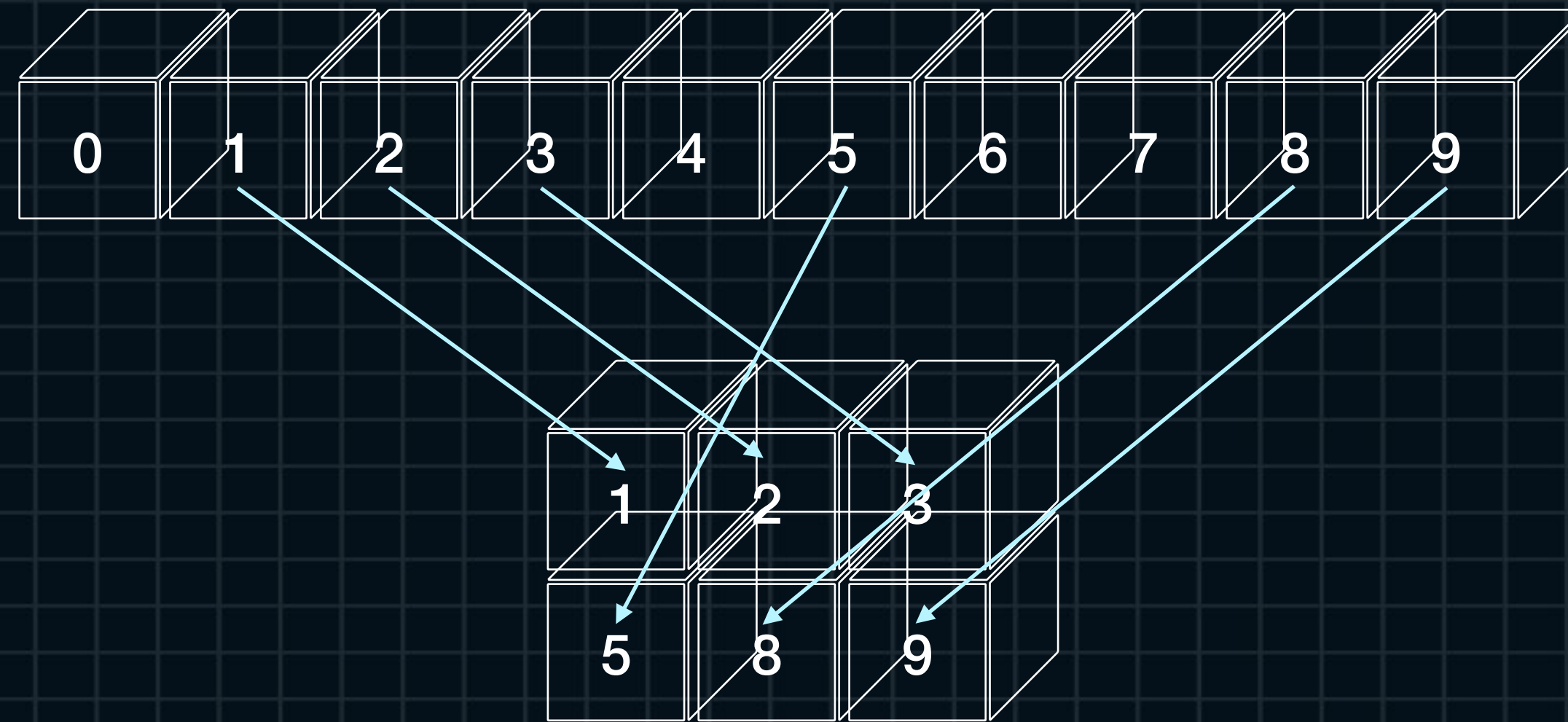
```
indices = np.array([0, 0, 5, 5])  
print(a[indices])
```

```
[0 0 5 5]
```

Lecture.7 Indexing and Slicing ndarrays

- Indexing with int ndarrays

Indexing and Slicing Vector ndarrays



```
import numpy as np
```

```
a = np.arange(10)
print(f"ndarray: \n{a}\n")
```

```
ndarray:
[0 1 2 3 4 5 6 7 8 9]
```

```
print(f"a[indices]: \n{a[indices]}")
```

```
indices = np.array([[1, 2, 3], [5, 8, 9]])
print(f"indices: \n{indices}")
```

```
indices:
[[1 2 3]
 [5 8 9]]
```

```
a[indices]:
[[1 2 3]
 [5 8 9]]
```


Lecture.7 Indexing and Slicing ndarrays

- Indexing with int ndarrays

Indexing and Slicing Vector ndarrays

```
import numpy as np
```

```
a = np.random.randint(0, 20, (10, ))  
print(f"ndarray: \n{a}\n")
```

```
ndarray:  
[14 12  4  6  0 19  9 14 17 15]
```

```
indices = np.random.randint(0, 10, size=(2, 3, 4))
```

```
print(f"indices: \n{indices}") indices:  
[[[2 5 7 0]  
  [2 3 6 6]  
  [1 8 1 5]]  
  
 [[7 9 4 0]  
  [0 7 5 8]  
  [4 3 5 1]]]
```

```
print(f"a[indices]: \n{a[indices]}") a[indices]:  
[[[ 4 19 14 14]  
  [ 4  6  9  9]  
  [12 17 12 19]]  
  
 [[14 15  0 14]  
  [14 14 19 17]  
  [ 0  6 19 12]]]
```

Lecture.7 Indexing and Slicing ndarrays

- Indexing with int ndarrays

Indexing and Slicing Matrix ndarrays

```
import numpy as np
```

```
a = np.arange(12).reshape((3, 4))
```

```
print(f"ndarray: \n{a}\n")
```

```
print("a[0]: ", a[0])
```

```
print("a[1]: ", a[1])
```

```
print("a[2]: ", a[2])
```

```
ndarray:
```

```
[[ 0  1  2  3]
 [ 4  5  6  7]
 [ 8  9 10 11]]
```

```
a[0]: [0 1 2 3]
```

```
a[1]: [4 5 6 7]
```

```
a[2]: [ 8  9 10 11]
```


Lecture.7 Indexing and Slicing ndarrays

- Indexing with int ndarrays

Indexing and Slicing Matrix ndarrays

```
import numpy as np

a = np.arange(12).reshape((3, 4))
print(f"ndarray: \n{a}\n")

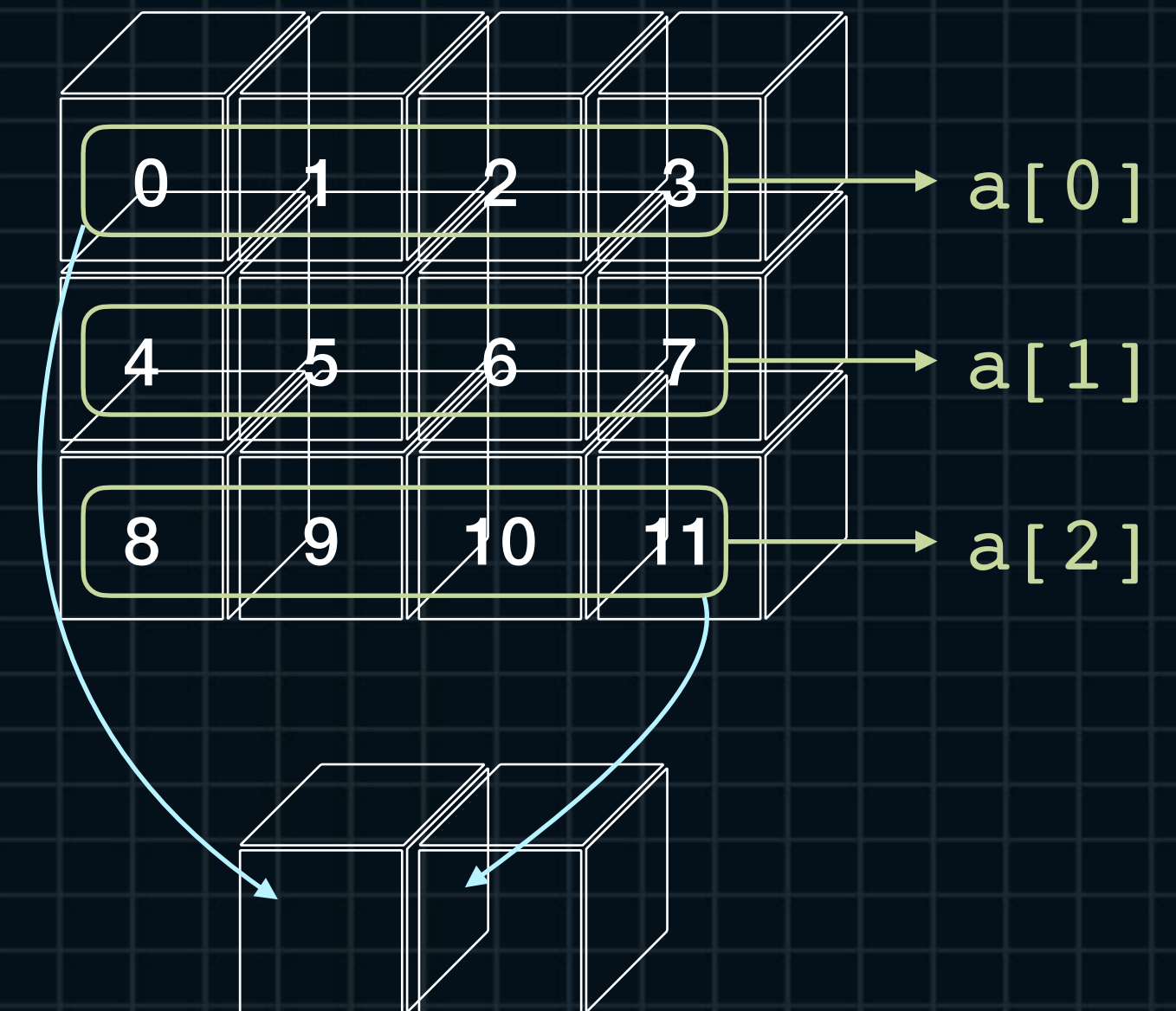
indices = np.array([0, 2])
print(f"indices: \n{indices}")

print(f"a[indices]: \n{a[indices]}")
```

```
ndarray:
[[ 0  1  2  3]
 [ 4  5  6  7]
 [ 8  9 10 11]]
```

```
indices:
[0 2]
```

```
a[indices]:
[[ 0  1  2  3]
 [ 8  9 10 11]]
```



Lecture.7 Indexing and Slicing ndarrays

- Indexing with int ndarrays

Indexing and Slicing Matrix ndarrays

```
import numpy as np
```

```
a = np.arange(12).reshape((3, 4))  
print(f"ndarray: \n{a}\n")
```

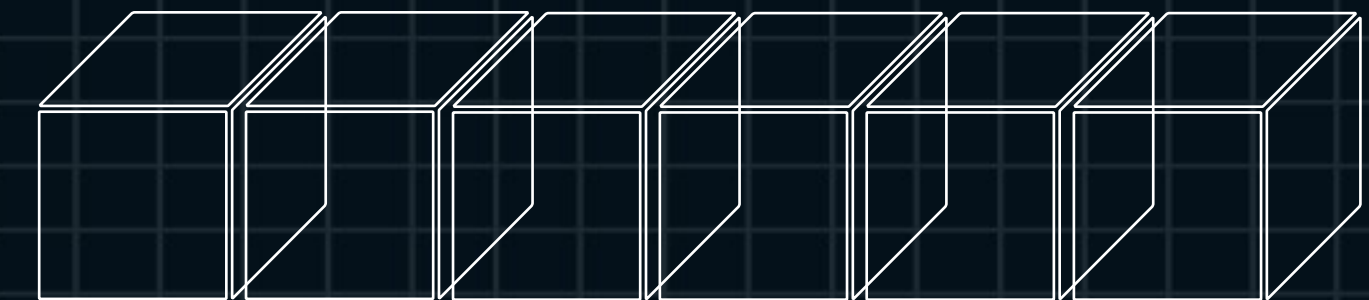
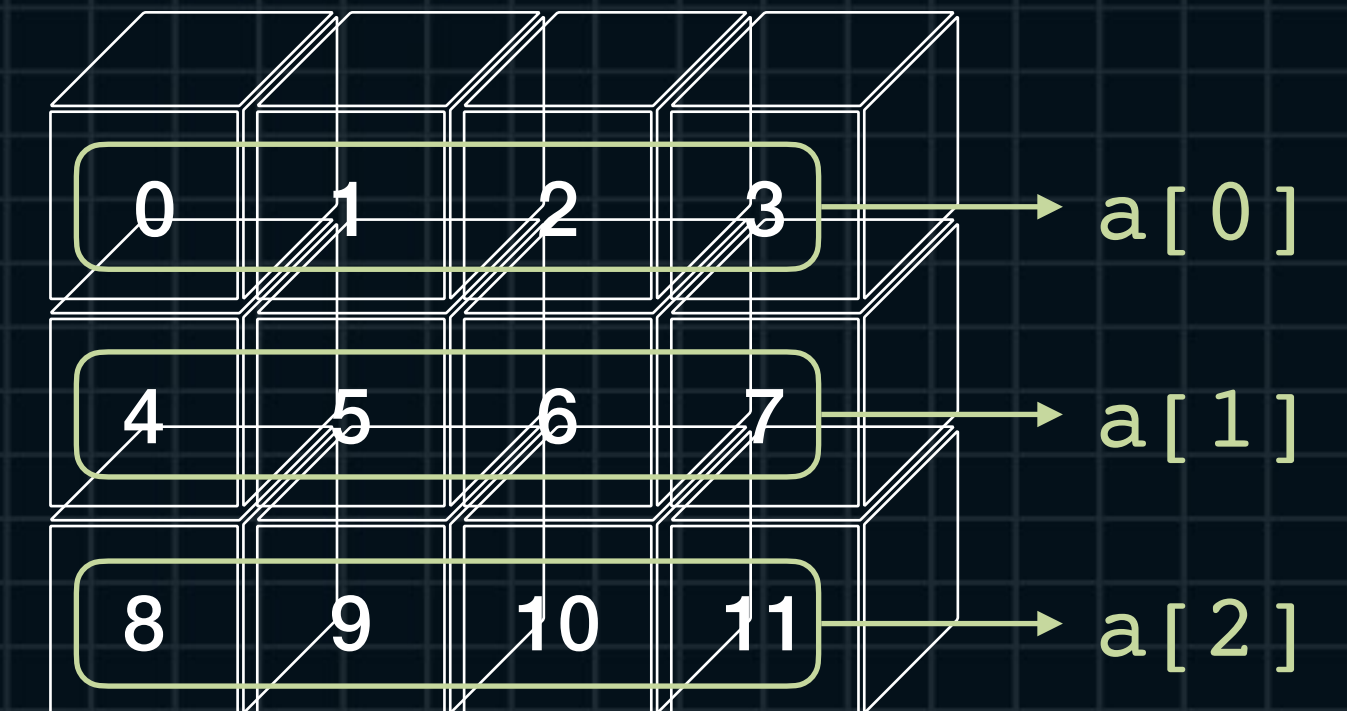
```
indices = np.array([0, 0, 1, 1, 2, 2])  
print(f"indices: \n{indices}")
```

```
print(f"a[indices]: \n{a[indices]}")
```

```
ndarray:  
[[ 0  1  2  3]  
 [ 4  5  6  7]  
 [ 8  9 10 11]]
```

```
indices:  
[0 0 1 1 2 2]
```

```
a[indices]:  
[[ 0  1  2  3]  
 [ 0  1  2  3]  
 [ 4  5  6  7]  
 [ 4  5  6  7]  
 [ 8  9 10 11]  
 [ 8  9 10 11]]
```



Lecture.7 Indexing and Slicing ndarrays

- Indexing with int ndarrays

Indexing and Slicing Matrix ndarrays

```
import numpy as np
```

```
a = np.arange(12).reshape((3, 4))  
print(f"ndarray: \n{a}\n")
```

```
indices = np.array([[0, 1, 2], [-3, -2, -1]])  
print(f"indices: \n{indices}")
```

```
print(f"a[indices]: \n{a[indices]}")
```

ndarray:

```
[[ 0  1  2  3]  
 [ 4  5  6  7]  
 [ 8  9 10 11]]
```

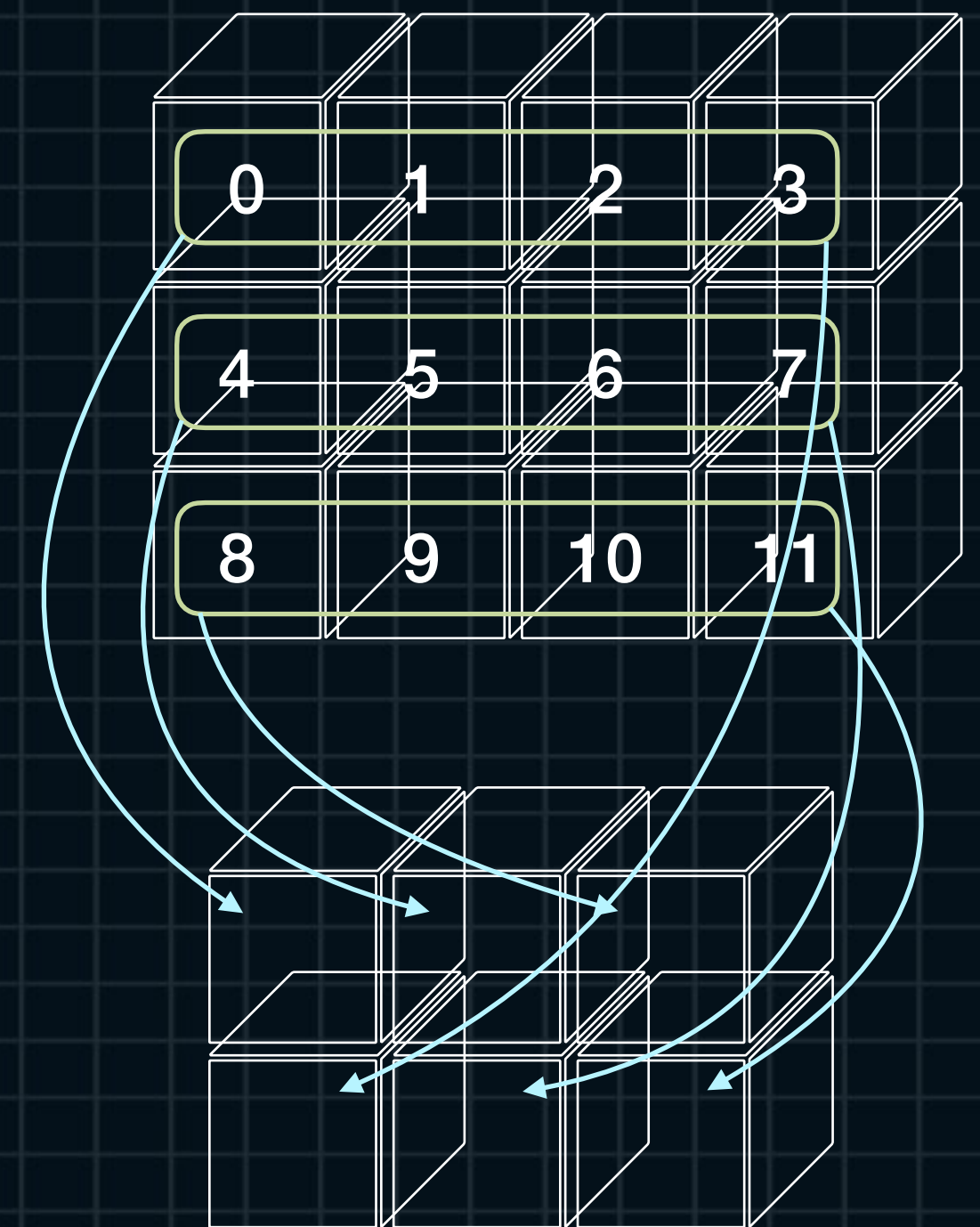
indices:

```
[[ 0  1  2]  
 [-3 -2 -1]]
```

a[indices]:

```
[[[ 0  1  2  3]  
 [ 4  5  6  7]  
 [ 8  9 10 11]]
```

```
[[ 0  1  2  3]  
 [ 4  5  6  7]  
 [ 8  9 10 11]]]
```



Lecture.7 Indexing and Slicing ndarrays

- Indexing with int ndarrays

Indexing and Slicing Matrix ndarrays

```
import numpy as np
```

```
a = np.arange(12).reshape((3, 4))  
print(f"ndarray: \n{a}\n")
```

ndarray:

```
[[ 0  1  2  3]  
 [ 4  5  6  7]  
 [ 8  9 10 11]]
```

```
indices0, indices1 = np.array([0]), np.array([0])  
print(a[indices0, indices1])
```

[0]

```
indices0, indices1 = np.array([1]), np.array([2])  
print(a[indices0, indices1])
```

[6]

```
indices0, indices1 = np.array([-1]), np.array([0])  
print(a[indices0, indices1])
```

[8]

```
indices0, indices1 = np.array([-1]), np.array([-1])  
print(a[indices0, indices1])
```

[11]

Lecture.7 Indexing and Slicing ndarrays

- Indexing with int ndarrays

Indexing and Slicing Matrix ndarrays

```
import numpy as np
```

```
a = np.arange(12).reshape((3, 4))  
print(f"ndarray: \n{a}\n")
```

```
ndarray:  
[[ 0  1  2  3]  
 [ 4  5  6  7]  
 [ 8  9 10 11]]
```

```
indices0, indices1 = np.array([0, 1, 2]), np.array([1, 2, 3])
```

```
print("Paired indices")  
for idx0, idx1 in zip(indices0, indices1):  
    print(f"({idx0}, {idx1})")
```

```
Paired indices  
(0, 1)  
(1, 2)  
(2, 3)
```

```
print(f"a[indices0, indices1]: \n{a[indices0, indices1]}")
```

```
a[indices0, indices1]:  
[ 1  6 11]
```

Lecture.7 Indexing and Slicing ndarrays

- Indexing with int ndarrays

Indexing and Slicing Matrix ndarrays

```
import numpy as np

a = np.arange(12).reshape((3, 4))
print(f"ndarray: \n{a}\n")

indices0 = np.array([[0, 1, 2], [0, 1, 2]])
indices1 = np.array([[0, 1, 2], [1, 2, 3]])

print("Paired indices")
for row_indices0, row_indices1 in zip(indices0, indices1):
    for idx0, idx1 in zip(row_indices0, row_indices1):
        print(f"({idx0}, {idx1})", end=' ')
    print()

print(f"a[indices0, indices1]: \n{a[indices0, indices1]}")
```

```
ndarray:
[[ 0  1  2  3]
 [ 4  5  6  7]
 [ 8  9 10 11]]
```

```
Paired indices
(0, 0) (1, 1) (2, 2)
(0, 1) (1, 2) (2, 3)
```

```
a[indices0, indices1]:
[[ 0  5 10]
 [ 1  6 11]]
```


Lecture.7 Indexing and Slicing ndarrays

- Indexing with bool ndarrays

Indexing and Slicing Vector ndarrays

```
import numpy as np
```

```
a = np.arange(5)  
print(f"ndarray: \n{a}")
```

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

```
b_indices = np.array([True, False, True, False, True])  
print(f"b_indices: \n{b_indices}\n")
```

```
b_indices:  
[ True False  True False  True]
```

```
print(f"a[b_indices]: \n{a[b_indices]}")
```

```
a[b_indices]:  
[0 2 4]
```

Lecture.7 Indexing and Slicing ndarrays

- Indexing with bool ndarrays

Indexing and Slicing Vector ndarrays

```
import numpy as np
```

```
a = np.random.randint(0, 20, (10, ))  
print(f"ndarray: \n{a}")
```

```
ndarray:  
[ 7  9 12 12  2  7 15  6 12  3]
```

```
b_indices = (a % 2 == 0)
```

```
print(f"b_indices: \n{b_indices}\n")
```

```
b_indices:  
[False False  True  True  True False False  True  True False]
```

```
print(f"a[b_indices]: \n{a[b_indices]}")
```

```
a[b_indices]:  
[12 12  2  6 12]
```


Lecture.7 Indexing and Slicing ndarrays

- Indexing with bool ndarrays

Indexing and Slicing Matrix ndarrays

```
import numpy as np
```

```
a = np.random.randint(0, 20, (2, 2))  
print(f"ndarray: \n{a}")
```

```
b_indices = np.array([[True, False],  
                     [False, True]])
```

```
print(f"b_indices: \n{b_indices}\n")
```

```
print(f"a[b_indices]: \n{a[b_indices]}")
```

```
ndarray:  
[[10  4]  
 [ 0 19]]
```

```
b_indices:  
[[ True False]  
 [False  True]]
```

```
a[b_indices]:  
[10 19]
```

Lecture.7 Indexing and Slicing ndarrays

- Indexing with bool ndarrays

Indexing and Slicing Matrix ndarrays

```
import numpy as np
```

```
a = np.random.randint(0, 20, (3, 4))  
print(f"ndarray: \n{a}")
```

```
ndarray:  
[[ 9 13  9  2]  
 [15 19 12  0]  
 [ 9 10 13 13]]
```

```
b_indices = (a > 10)  
print(f"b_indices: \n{b_indices}\n")
```

```
b_indices:  
[[False  True False False]  
 [ True  True  True False]  
 [False False  True  True]]
```

```
print(f"a[b_indices]: \n{a[b_indices]}")
```

```
a[b_indices]:  
[13 15 19 12 13 13]
```


Lecture.7 Indexing and Slicing ndarrays

- Extracting Indices

```
import numpy as np
```

```
a = np.array([True, False, True, False])
```

```
nonzero = np.nonzero(a)
```

```
where = np.where(a)
```

```
print(f"a: \n{a}\n")
```

```
print(f"nonzero: \n{nonzero}")
```

```
print(f"where: \n{where}")
```

```
a:
```

```
[ True False  True False]
```

```
nonzero:
```

```
(array([0, 2]),)
```

```
where:
```

```
(array([0, 2]),)
```

```
import numpy as np
```

```
a = np.array([[True, False],  
              [True, False]])
```

```
nonzero = np.nonzero(a)
```

```
where = np.where(a)
```

```
print(f"a: \n{a}\n")
```

```
print(f"nonzero: \n{nonzero}")
```

```
print(f"where: \n{where}")
```

```
a:
```

```
[[ True False]
```

```
 [ True False]]
```

```
nonzero:
```

```
(array([0, 1]), array([0, 0]))
```

```
where:
```

```
(array([0, 1]), array([0, 0]))
```

Lecture.7 Indexing and Slicing ndarrays

- Extracting Indices

```
import numpy as np
```

```
a = np.array([[True, False, True],  
              [True, False, False],  
  
              [False, True, False],  
              [True, False, True]])
```

```
nonzero = np.nonzero(a)  
where = np.where(a)
```

```
print(f"a: \n{a}\n")  
print(f"nonzero: \n{nonzero}")  
print(f"where: \n{where}")
```

```
a:  
[[ True False  True]  
 [ True False False]
```

```
[[False  True False]  
 [ True False  True]]]
```

```
nonzero:  
(array([0, 0, 0, 1, 1, 1]),  
 array([0, 0, 1, 0, 1, 1]),  
 array([0, 2, 0, 1, 0, 2]))  
where:  
(array([0, 0, 0, 1, 1, 1]),  
 array([0, 0, 1, 0, 1, 1]),  
 array([0, 2, 0, 1, 0, 2]))
```


Lecture.7 Indexing and Slicing ndarrays

- Extracting Indices

```
import numpy as np

a = np.random.randint(-2, 3, size=(3, 3))

u_nonzero = a[np.nonzero(a)]
u_where = a[np.where(a)]
u_bool = a[a != 0]

print(f"a: \n{a}\n")
print(f"using nonzero: \n{u_nonzero}")
print(f"using where: \n{u_where}")
print(f"using bool ndarray: \n{u_bool}")
```

```
a:
[[ 1  2  0]
 [ 1  0  0]
 [ 2 -2  1]]

using nonzero:
[ 1  2  1  2 -2  1]
using where:
[ 1  2  1  2 -2  1]
using bool ndarray:
[ 1  2  1  2 -2  1]
```

Lecture.7 Indexing and Slicing ndarrays

- Extracting Indices

```
import numpy as np
```

```
a = np.random.randint(-2, 3, size=(3, 3))
```

```
u_nonzero = a[np.nonzero(a > 0)]
```

```
u_where = a[np.where(a > 0)]
```

```
u_bool = a[a > 0]
```

```
print(f"a: \n{a}\n")
```

```
print(f"using nonzero: \n{u_nonzero}")
```

```
print(f"using where: \n{u_where}")
```

```
print(f"using bool ndarray: \n{u_bool}")
```

a:

```
[[ 2  0 -1]
 [ 0  0  2]
 [ 1  1  0]]
```

using nonzero:

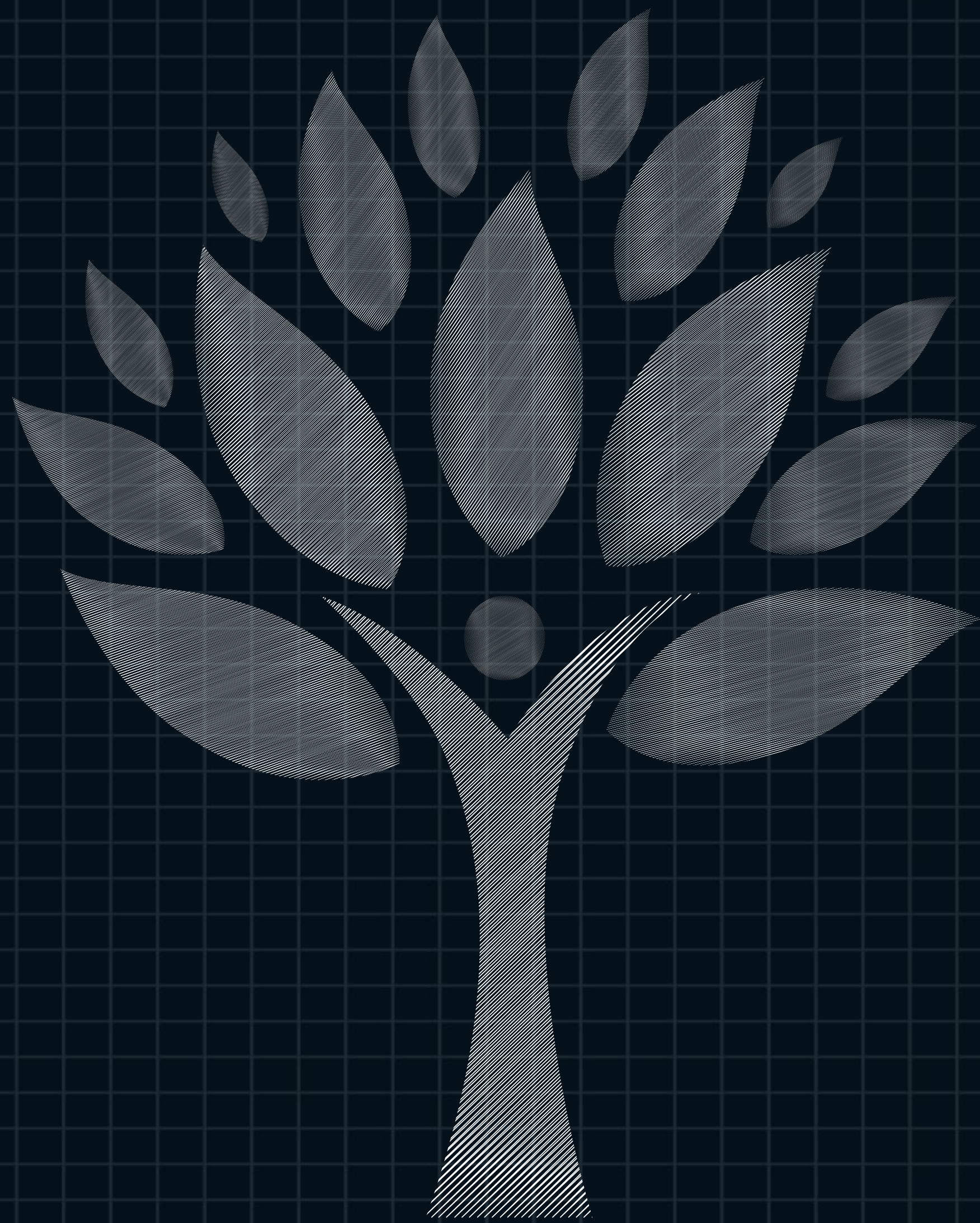
```
[2 2 1 1]
```

using where:

```
[2 2 1 1]
```

using bool ndarray:

```
[2 2 1 1]
```

NumPy Master Class

Lecture.7 Indexing and Slicing ndarrays