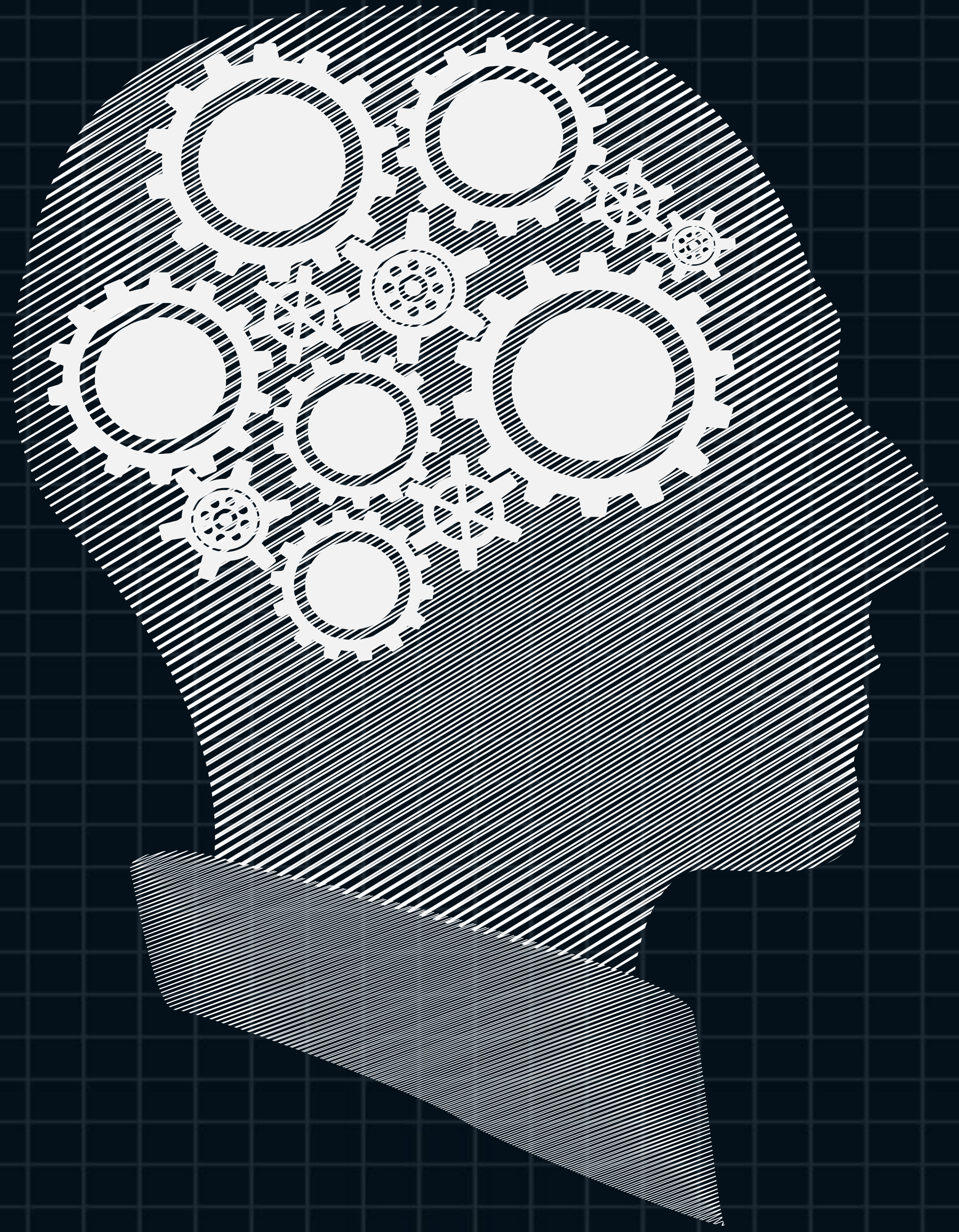


NumPy Master Class

Lecture.15 Repeating ndarrays

Lecture.15
Repeating ndarrays

- Element-wise Repetition

np.repeat

`numpy.repeat(a, repeats, axis=None)`

```python
import numpy as np

x = 3

rep = np.repeat(x, 2)

print(f"x: {x}")
print(f"np.repeat(x, 2): \n{rep}\n")
```

```
x: 3
np.repeat(x, 2):
[3 3]
```

```python
import numpy as np

x = np.array([1, 2, 3])

rep = np.repeat(x, 3)

print(f"x: {x}")
print(f"np.repeat(x, 3): \n{rep}\n")
```

```
x: [1 2 3]
np.repeat(x, 3):
[1 1 1 2 2 2 3 3 3]
```

Lecture.15
Repeating ndarrays

- Element-wise Repetition

np.repeat with axis

```python
import numpy as np

x = np.arange(4).reshape((2, 2))


rep = np.repeat(x, 3)

print(f"x: \n{x}")
print(f"np.repeat(x, 3): \n{rep}\n")
```

```
x:
[[0 1]
 [2 3]]
np.repeat(x, 3):
[0 0 0 1 1 1 2 2 2 3 3 3]
```

Lecture.15
Repeating ndarrays

- Element-wise Repetition

np.repeat with axis

```
x = np.arange(4).reshape((2, 2))
print(f"x: {x.shape}\n{x}")
```

```
x: (2, 2)
[[0 1]
 [2 3]]
```

```
rep = np.repeat(x, repeats=3, axis=0)
print(f"np.repeat(x, 3, 0): {rep.shape}\n{rep}\n")
```

```
np.repeat(x, 3, 0): (6, 2)
[[0 1]
 [0 1]
 [0 1]
 [2 3]
 [2 3]
 [2 3]]
```

```
rep = np.repeat(x, repeats=3, axis=1)
print(f"np.repeat(x, 3, 1): {rep.shape}\n{rep}\n")
```

```
np.repeat(x, 3, 1): (2, 6)
[[0 0 0 1 1 1]
 [2 2 2 3 3 3]]
```

Lecture.15 Repeating ndarrays

- Element-wise Repetition

np.repeat with axis

```
x = np.arange(4).reshape((2, 2))
print("x: \n", x, '\n')
```

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

```
rep = np.repeat(x, repeats=[2, 1], axis=0)
print(f"repeats=[2, 1]: {rep.shape}\n{rep}\n”)
```

```
repeats=[2, 1]: (3, 2)
[[0 1]
 [0 1]
 [2 3]]
```

```
rep = np.repeat(x, repeats=[1, 2], axis=0)
print(f"repeats=[1, 2]: {rep.shape}\n{rep}\n”)
```

```
repeats=[1, 2]: (3, 2)
[[0 1]
 [2 3]
 [2 3]]
```

```
rep = np.repeat(x, repeats=[2, 2], axis=0)
print(f"repeats=[2, 2]: {rep.shape}\n{rep}\n")
```

```
repeats=[2, 2]: (4, 2)
[[0 1]
 [0 1]
 [2 3]
 [2 3]]
```

Lecture.15
Repeating ndarrays

- Element-wise Repetition

np.repeat with axis

```python
import numpy as np

x = np.arange(6).reshape((2, 3))
print("x: \n", x, '\n')
```

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

```python
rep = np.repeat(x, repeats=[2, 1, 2], axis=1)
print(f"repeats=[2, 1, 2]: {rep.shape}\n{rep}\n”)
```

```
repeats=[2, 1, 2]: (2, 5)
[[0 0 1 2 2]
 [3 3 4 5 5]]
```

```python
rep = np.repeat(x, repeats=[1, 2, 2], axis=1)
print(f”repeats=[1, 2, 2]: {rep.shape}\n{rep}\n")
```

```
repeats=[1, 2, 2]: (2, 5)
[[0 1 1 2 2]
 [3 4 4 5 5]]
```

Lecture.15 Repeating ndarrays

- Element-wise Repetition

np.repeat with Vectors

```python
import numpy as np

row_vec = np.arange(4).reshape((1, -1))
print(f"ndarray: {row_vec.shape}\n{row_vec}\n")


rep = np.repeat(row_vec, repeats=3, axis=0)
print(f"repeats=3, axis=0: {rep.shape}\n{rep}\n")
```

```
ndarray: (1, 4)
[[0 1 2 3]]
```

```
repeats=3, axis=0: (3, 4)
[[0 1 2 3]
 [0 1 2 3]
 [0 1 2 3]]
```

Lecture.15
Repeating ndarrays

- Element-wise Repetition

np.repeat with Vectors

```python
import numpy as np

col_vec = np.arange(4).reshape((-1, 1))
print(f"ndarray: {col_vec.shape}\n{col_vec}\n")


rep = np.repeat(col_vec, repeats=3, axis=1)
print(f"repeats=3, axis=1: {rep.shape}\n{rep}")
```

```
ndarray: (4, 1)
[[0]
 [1]
 [2]
 [3]]


repeats=3, axis=1: (4, 3)
[[0 0 0]
 [1 1 1]
 [2 2 2]
 [3 3 3]]
```

Lecture.15
Repeating ndarrays

- Overall Repetition

np.tile

numpy.tile(*A*, *reps*)

```python
import numpy as np

a = np.arange(4)
print(f"ndarray: {a.shape}\n{a}\n")


tile = np.tile(a, reps=3)
print(f"reps=3: {tile.shape}\n{tile}\n")
```

```
ndarray: (4,)
[0 1 2 3]


reps=3: (12,)
[0 1 2 3 0 1 2 3 0 1 2 3]
```

Lecture.15
Repeating ndarrays

- Overall Repetition

np.tile with reps

```python
import numpy as np

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

```
ndarray: (3,)
[0 1 2]
```

```python
tile = np.tile(a, reps=[1, 2])
print(f"reps=[1, 2]: {tile.shape}\n{tile}\n")
```

```
reps=[1, 2]: (1, 6)
[[0 1 2 0 1 2]]
```

```python
tile = np.tile(a, reps=[2, 1])
print(f"reps=[2, 1]: {tile.shape}\n{tile}\n”)
```

```
reps=[2, 1]: (2, 3)
[[0 1 2]
 [0 1 2]]
```

```python
tile = np.tile(a, reps=[2, 2])
print(f"reps=[2, 2]: {tile.shape}\n{tile}\n")
```

```
reps=[2, 2]: (2, 6)
[[0 1 2 0 1 2]
 [0 1 2 0 1 2]]
```

Lecture.15 Repeating ndarrays

- Overall Repetition

np.tile with reps

```python
import numpy as np

a = np.arange(6).reshape((2, 3))
print(f"ndarray: {a.shape}\n{a}\n")
```

```
ndarray: (2, 3)
[[0 1 2]
 [3 4 5]]
```

```python
tile = np.tile(a, reps=[1, 2])
print(f"reps=[1, 2]: {tile.shape}\n{tile}\n”)
```

```
reps=[1, 2]: (2, 6)
[[0 1 2 0 1 2]
 [3 4 5 3 4 5]]
```

```python
tile = np.tile(a, reps=[2, 1])
print(f"reps=[2, 1]: {tile.shape}\n{tile}\n")
```

```
reps=[2, 1]: (4, 3)
[[0 1 2]
 [3 4 5]
 [0 1 2]
 [3 4 5]]
```

```python
tile = np.tile(a, reps=[2, 2])
print(f"reps=[2, 2]: {tile.shape}\n{tile}\n")
```

```
reps=[2, 2]: (4, 6)
[[0 1 2 0 1 2]
 [3 4 5 3 4 5]
 [0 1 2 0 1 2]
 [3 4 5 3 4 5]]
```

Lecture.15
Repeating ndarrays

- Overall Repetition

np.tile with reps

```python
import numpy as np

a = np.arange(6).reshape((2, 3))
print(f"ndarray: {a.shape}\n")
```

```
ndarray: (2, 3)
```

```python
reps = np.array([3, 5])
tile = np.tile(a, reps=reps)
print(f"shapes: {tile.shape} - {reps*a.shape}")
```

```
shapes: (6, 15) - [ 6 15]
```

```python
reps = np.array([10, 8])
tile = np.tile(a, reps=reps)
print(f"shapes: {tile.shape} - {reps*a.shape}")
```

```
shapes: (20, 24) - [20 24]
```

Lecture.15
Repeating ndarrays

- Overall Repetition

np.tile with Vectors

```python
import numpy as np

a = np.arange(4).reshape((1, -1))

tile = np.tile(a, reps=[5, 1])

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

```
ndarray: (1, 4)
[[0 1 2 3]]
tile: (5, 4)
[[0 1 2 3]
 [0 1 2 3]
 [0 1 2 3]
 [0 1 2 3]
 [0 1 2 3]]
```

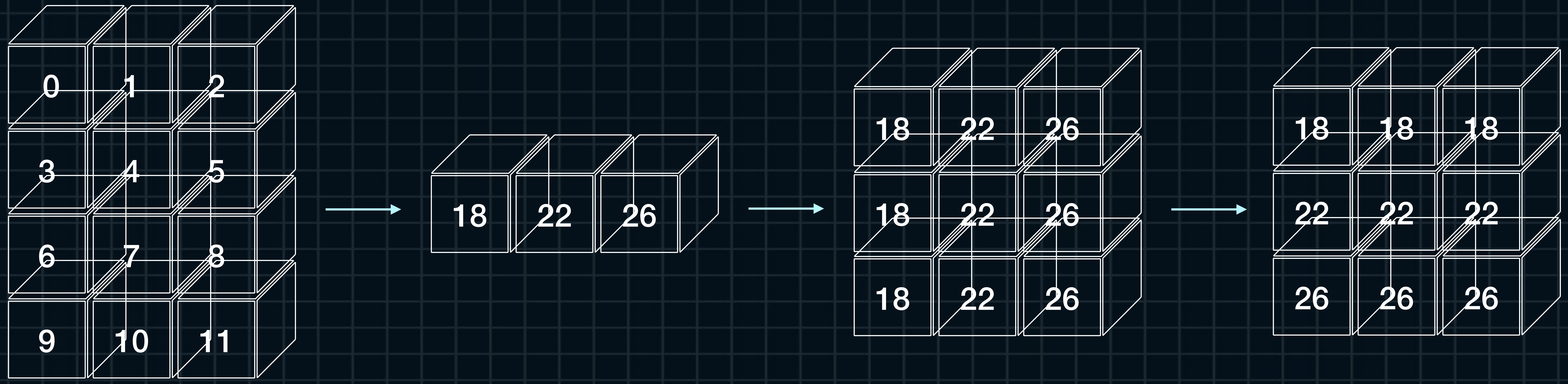
```python
a = np.arange(4).reshape((-1, 1))

tile = np.tile(a, reps=[1, 5])

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

```
ndarray: (4, 1)
[[0]
 [1]
 [2]
 [3]]
tile: (4, 5)
[[0 0 0 0 0]
 [1 1 1 1 1]
 [2 2 2 2 2]
 [3 3 3 3 3]]
```

Lecture.15
Repeating ndarrays

- Application of Repetition

Row-wise Case



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

18	22	26

18	22	26
18	22	26
18	22	26

18	18	18
22	22	22
26	26	26

Lecture.15
Repeating ndarrays

- Application of Repetition

Row-wise Case

```python
import numpy as np

x = np.arange(4*3).reshape((4, 3))
print(f"x: {x.shape}\n{x}\n")

x = x.sum(axis=0, keepdims=True)
print(f"x.sum: {x.shape}\n{x}\n")

x = x.repeat(repeats=3, axis=0)
print(f"x.repeat: {x.shape}\n{x}\n")

x = x.T
print(f"x.T: {x.shape}\n{x}\n")
```

```
x: (4, 3)
[[ 0  1  2]
 [ 3  4  5]
 [ 6  7  8]
 [ 9 10 11]]

x.sum: (1, 3)
[[18 22 26]]

x.repeat: (3, 3)
[[18 22 26]
 [18 22 26]
 [18 22 26]]

x.T: (3, 3)
[[18 18 18]
 [22 22 22]
 [26 26 26]]
```

Lecture.15
Repeating ndarrays

- Application of Repetition

Row-wise Case

```python
import numpy as np

x = np.arange(4*3).reshape((4, 3))
print(f"x: {x.shape}\n{x}\n")
```

```
x: (4, 3)
[[ 0  1  2]
 [ 3  4  5]
 [ 6  7  8]
 [ 9 10 11]]
```

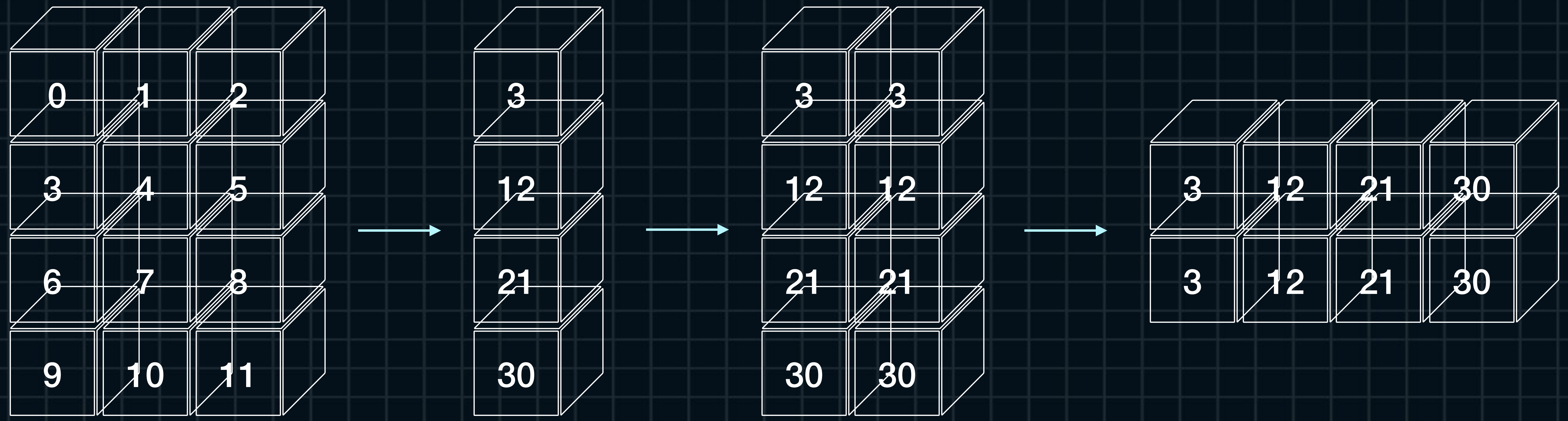
```python
y = x.sum(0, keepdims=True).repeat(3, 0).T
print(f"y: {y.shape}\n{y}\n")
```

```
y: (3, 3)
[[18 18 18]
 [22 22 22]
 [26 26 26]]
```

Lecture.15
Repeating ndarrays

- Application of Repetition

Column-wise Case



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

→

3
12
21
30

→

3	3
12	12
21	21
30	30

→

3	12	21	30
3	12	21	30

Lecture.15
Repeating ndarrays

- Application of Repetition

Column-wise Case

```python
import numpy as np

x = np.arange(4*3).reshape((4, 3))
print(f"x: {x.shape}\n{x}\n")


x = x.sum(axis=1, keepdims=True)
print(f"x.sum: {x.shape}\n{x}\n")




x = x.repeat(repeats=2, axis=1)
print(f"x.repeat: {x.shape}\n{x}\n")



x = x.T
print(f"x.T: {x.shape}\n{x}\n")
```

```
x: (4, 3)
[[ 0  1  2]
 [ 3  4  5]
 [ 6  7  8]
 [ 9 10 11]]

x.sum: (4, 1)
[[ 3]
 [12]
 [21]
 [30]]

x.repeat: (4, 2)
[[ 3  3]
 [12 12]
 [21 21]
 [30 30]]

x.T: (2, 4)
[[ 3 12 21 30]
 [ 3 12 21 30]]
```

Lecture.15
Repeating ndarrays

- Application of Repetition

Column-wise Case

```python
import numpy as np

x = np.arange(4*3).reshape((4, 3))
print(f"x: {x.shape}\n{x}\n")
```

```
x: (4, 3)
[[ 0  1  2]
 [ 3  4  5]
 [ 6  7  8]
 [ 9 10 11]]
```

```python
y = x.sum(1, keepdims=True).repeat(2, 1).T
print(f"y: {y.shape}\n{y}\n")
```

```
y: (2, 4)
[[ 3 12 21 30]
 [ 3 12 21 30]]
```

Lecture.15
Repeating ndarrays

- Making Coordinates

Coordinates from 1D Vectors

$$y = x^2 - x$$

Lecture.15
Repeating ndarrays

- Making Coordinates

Coordinates from 1D Vectors

$$z = x^2 + y^2$$

$$\begin{array}{ccccc}
(-2,-2), & (-1,-2), & (0,-2), & (1,-2), & (2,-2) \\
(-2,-1), & (-1,-1), & (0,-1), & (1,-1), & (2,-1) \\
(-2,\ \ 0), & (-1,\ \ 0), & (0,\ \ 0), & (1,\ \ 0), & (2,\ \ 0) \\
(-2,\ \ 1), & (-1,\ \ 1), & (0,\ \ 1), & (1,\ \ 1), & (2,\ \ 1) \\
(-2,\ \ 2), & (-1,\ \ 2), & (0,\ \ 2), & (1,\ \ 2), & (2,\ \ 2)
\end{array}$$

$$X = \begin{pmatrix} -2 & -1 & 0 & 1 & 2 \\ -2 & -1 & 0 & 1 & 2 \\ -2 & -1 & 0 & 1 & 2 \\ -2 & -1 & 0 & 1 & 2 \\ -2 & -1 & 0 & 1 & 2 \end{pmatrix}, \quad Y = \begin{pmatrix} -2 & -2 & -2 & -2 & -2 \\ -1 & -1 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 \end{pmatrix}$$

Lecture.15
Repeating ndarrays

- Making Coordinates

Coordinates from 1D Vectors

```python
import numpy as np

x = np.arange(-2, 3)
y = np.arange(-2, 3)

print(f"x: {x}")
print(f"y: {y}")
```

```
x: [-2 -1  0  1  2]
y: [-2 -1  0  1  2]
```

```python
X = x.reshape((1, -1)).repeat(y.shape[0], axis=0)
print(f"X: \n{X}")
```

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

```python
Y = y.reshape((-1, 1)).repeat(x.shape[0], axis=1)
print(f"Y: \n{Y}")
```

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

```python
Z = np.square(X) + np.square(Y)
print(f"Z: \n{Z}")
```

```
Z:
[[8 5 4 5 8]
 [5 2 1 2 5]
 [4 1 0 1 4]
 [5 2 1 2 5]
 [8 5 4 5 8]]
```

Lecture.15
Repeating ndarrays

- Making Coordinates

np.meshgrid

`numpy.meshgrid(*xi)`

```
import numpy as np

x = np.arange(-2, 3)
y = np.arange(-2, 3)

X, Y = np.meshgrid(x, y)
print(f"X: \n{X}")
print(f"Y: \n{Y}")
```

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

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

```
Z = np.square(X) + np.square(Y)
print(f"Z: \n{Z}")
```

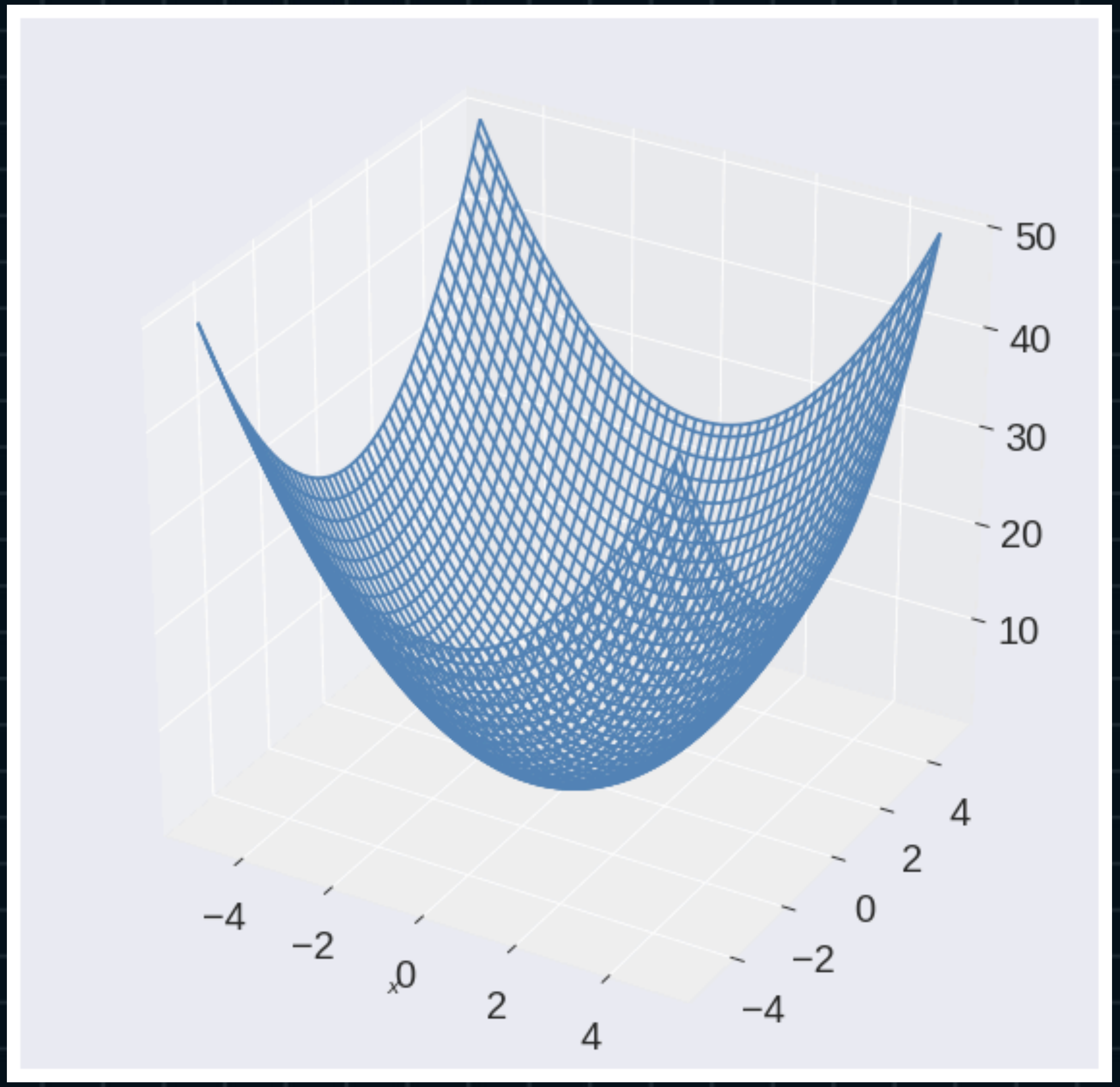
```
Z:
[[8 5 4 5 8]
 [5 2 1 2 5]
 [4 1 0 1 4]
 [5 2 1 2 5]
 [8 5 4 5 8]]
```

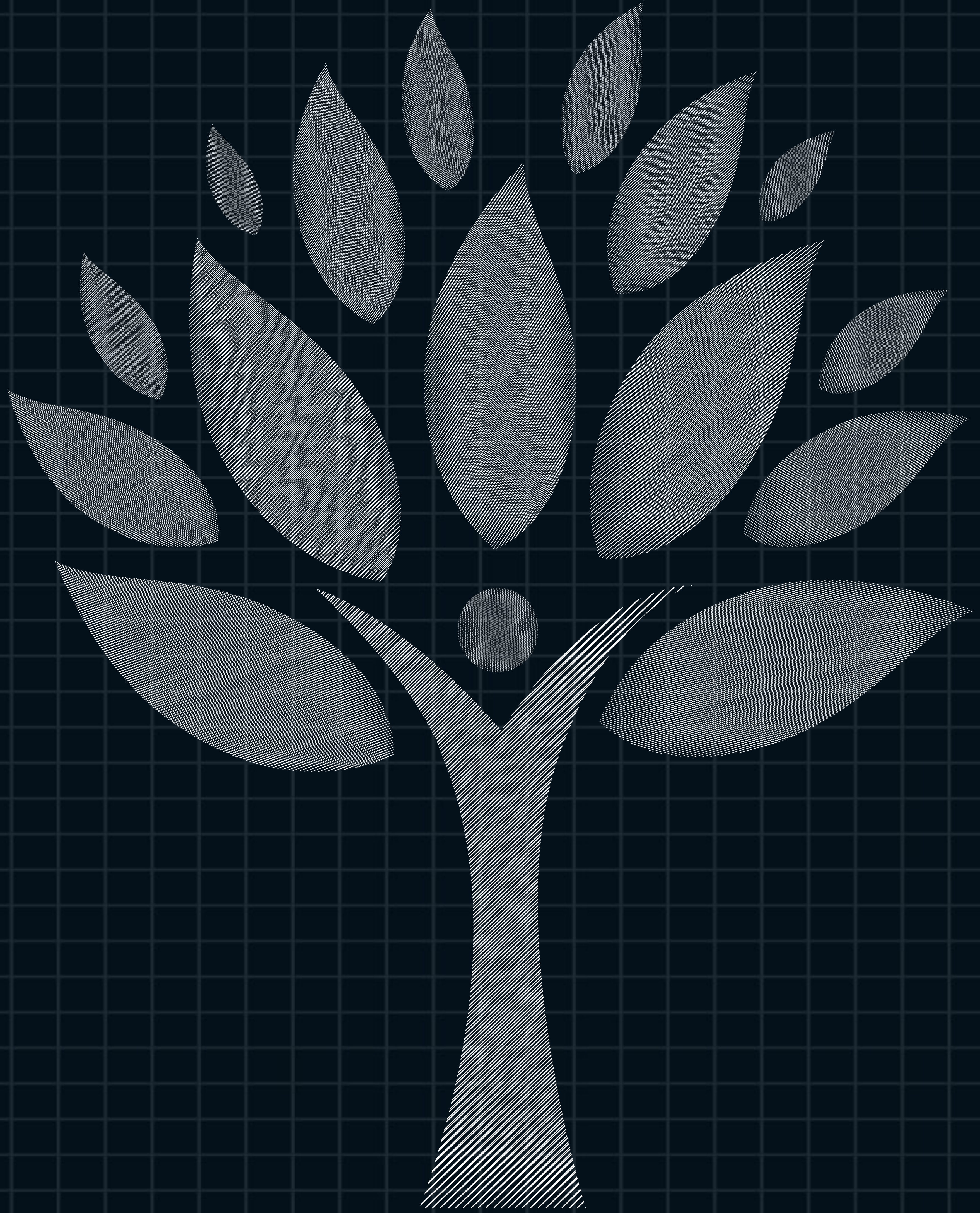
Lecture.15 Repeating ndarrays

- Making Coordinates

np.meshgrid

```
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(-5, 5, 100)
y = np.linspace(-5, 5, 100)

X, Y = np.meshgrid(x, y)
Z = np.square(X) + np.square(Y)

fig = plt.figure(figsize=(10, 10))
ax = fig.add_subplot(projection='3d')

ax.plot_wireframe(X, Y, Z)
ax.tick_params(labelsize=20)
```





NumPy Master Class

Lecture.15
Repeating ndarrays