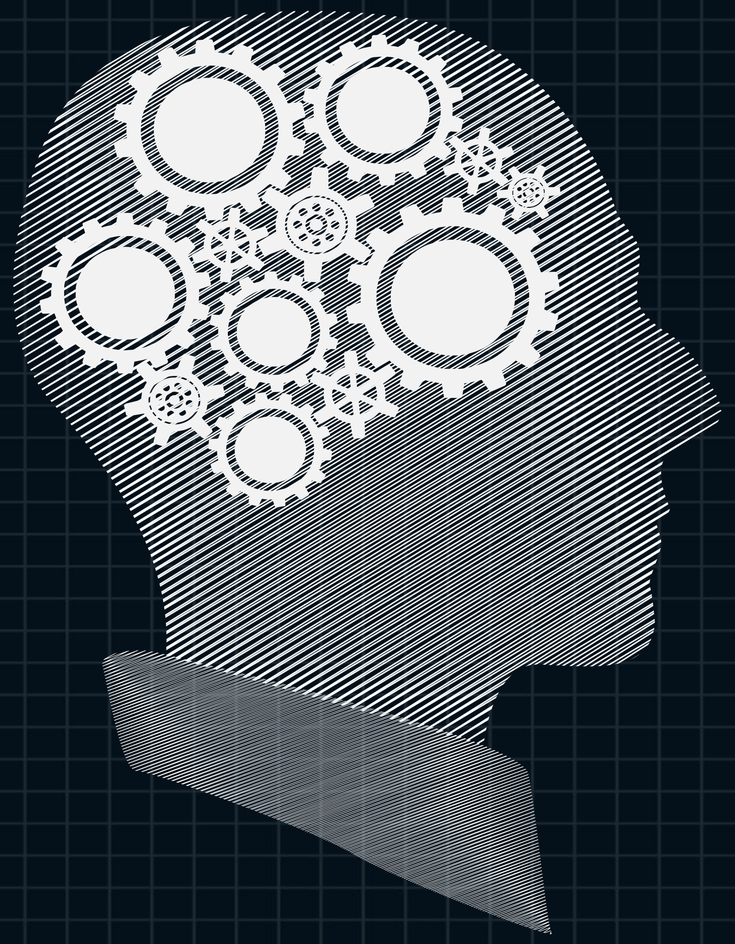


NumPy Master Class

Lecture.6
Element-wise Operations and Broadcasting

Lecture.6 Element-wise Operations and Broadcasting - Element-wise Operations of ndarrays

Vector-vector Case

```
import numpy as np

a = np.random.randint(-5, 5, (5, ))
b = np.random.randint(-5, 5, (5, ))

print("a: ", a)
print("b: ", b, '\n')

print("a + b: ", a + b)
```

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

a + b:  [ -4  -2  -5  -2 -10]
```

Lecture.6 Element-wise Operations and Broadcasting - Element-wise Operations of ndarrays

Vector-vector Case

```
import numpy as np

a = np.random.randint(1, 5, (5, ))
b = np.random.randint(1, 5, (5, ))

print("a: ", a)
print("b: ", b, '\n')


print("a + b: ", a + b)
print("a - b: ", a - b)
print("a * b: ", a * b)
print("a / b: ", a / b)
print("a // b: ", a // b)
print("a % b: ", a % b)
print("a ** b: ", a ** b)
```

```
a:  [2 3 1 2 4]
b:  [3 2 1 3 1]


a + b:  [5 5 2 5 5]
a - b:  [-1  1  0 -1  3]
a * b:  [6 6 1 6 4]
a / b:  [0.66666667 1.5        1.         0.66666667 4.        ]
a // b:  [0 1 1 0 4]
a % b:  [2 1 0 2 0]
a ** b:  [8 9 1 8 4]
```

Lecture.6 Element-wise Operations and Broadcasting - Element-wise Operations of ndarrays

Vector-vector Case

```
import numpy as np

a = np.random.randint(1, 5, (5, ))
b = np.random.randint(1, 5, (5, ))

print("a: ", a)
print("b: ", b, '\n')


print("a > b: ", a > b)
print("a >= b: ", a >= b)
print("a < b: ", a < b)
print("a <= b: ", a <= b)
print("a == b: ", a == b)
print("a != b: ", a != b)
```

```
a:  [1 1 1 4 3]
b:  [2 2 3 2 1]


a > b:  [False False False  True  True]
a >= b:  [False False False  True  True]
a < b:  [ True  True  True False False]
a <= b:  [ True  True  True False False]
a == b:  [False False False False False]
a != b:  [ True  True  True  True  True]
```

Lecture.6 Element-wise Operations and Broadcasting - Element-wise Operations of ndarrays

Vector-vector Case

```
import numpy as np

a = np.random.randint(1, 5, (5, ))
b = np.random.randint(1, 5, (5, ))

print("a: ", a)
print("b: ", b, '\n')

print("a + b: ", a.__add__(b))
print("a - b: ", a.__sub__(b))
print("a * b: ", a.__mul__(b))
print("a / b: ", a.__truediv__(b))
print("a // b: ", a.__floordiv__(b))
print("a % b: ", a.__mod__(b))
print("a ** b: ", a.__pow__(b))
```

```
a:  [3 2 4 2 2]
b:  [2 2 2 4 1]

a + b:  [5 4 6 6 3]
a - b:  [ 1  0  2 -2  1]
a * b:  [6 4 8 8 2]
a / b:  [1.5 1.  2.  0.5 2. ]
a // b:  [1 1 2 0 2]
a % b:  [1 0 0 2 0]
a ** b:  [ 9  4 16 16  2]
```

Lecture.6 Element-wise Operations and Broadcasting - Element-wise Operations of ndarrays

Matrix-matrix Case

```
import numpy as np

M = np.random.randint(1, 5, (2, 3))
N = np.random.randint(1, 5, (2, 3))

print("M: \n", M)
print("N: \n", N, '\n')


print("M + N: \n", M + N)
print("M - N: \n", M - N)

print("M > N: \n", M > N)
print("M >= N: \n", M >= N)
```

```
M:
 [[2 4 1]
 [4 2 1]]
N:
 [[2 3 4]
 [1 4 4]]



M + N:
 [[4 7 5]
 [5 6 5]]
M - N:
 [[ 0  1 -3]
 [ 3 -2 -3]]
M > N:
 [[False  True False]
 [ True False False]]
M >= N:
 [[ True  True False]
 [ True False False]]
```

Lecture.6 Element-wise Operations and Broadcasting - Element-wise Operations of ndarrays

Element-wise Multiplication and Masking

```
import numpy as np

a = np.arange(5)
mask = np.array([0, 1, 0, 1, 0])

print("input: ", a)
print("mask: ", mask)
print("output: ", a*b)
```

```
input:  [0 1 2 3 4]
mask:  [0 1 0 1 0]
output:  [ 0  1  0 3  0]
```

Lecture.6 Element-wise Operations and Broadcasting - Element-wise Operations of ndarrays

Element-wise Multiplication and Masking

```
import numpy as np

a = np.arange(1, 5).reshape((2, 2))
mask = np.array([[0, 0], [1, 0]])

print("input: \n", a)
print("mask: \n", mask)
print("output: \n", a*mask)
```

```
input:
 [[1 2]
 [3 4]]
mask:
 [[0 0]
 [1 0]]
output:
 [[0 0]
 [3 0]]
```

Lecture.6 Element-wise Operations and Broadcasting - Broadcasting in NumPy

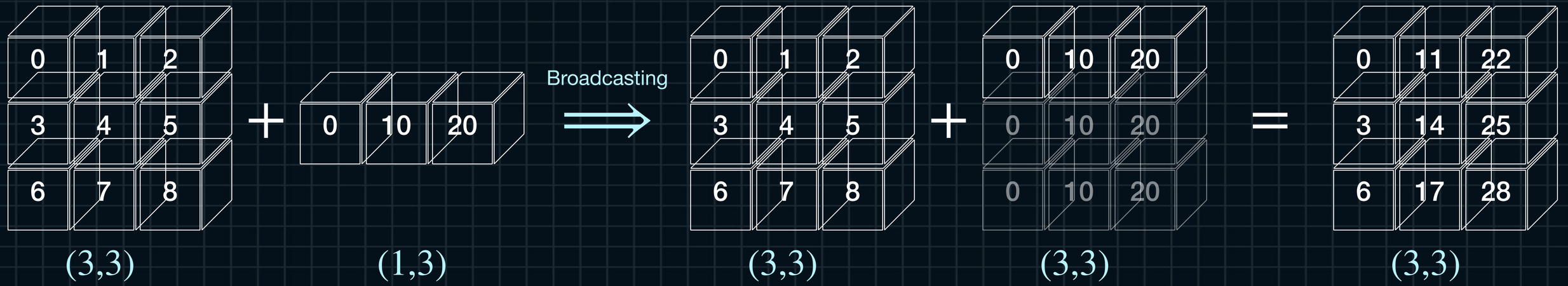
Broadcasting Cases

a.ndim = b.ndim

a.ndim ≠ b.ndim

Lecture.6 Element-wise Operations and Broadcasting - Broadcasting in NumPy

When ndims Are Equal(Matrices)



Lecture.6 Element-wise Operations and Broadcasting - Broadcasting in NumPy

When ndims Are Equal(Matrices)

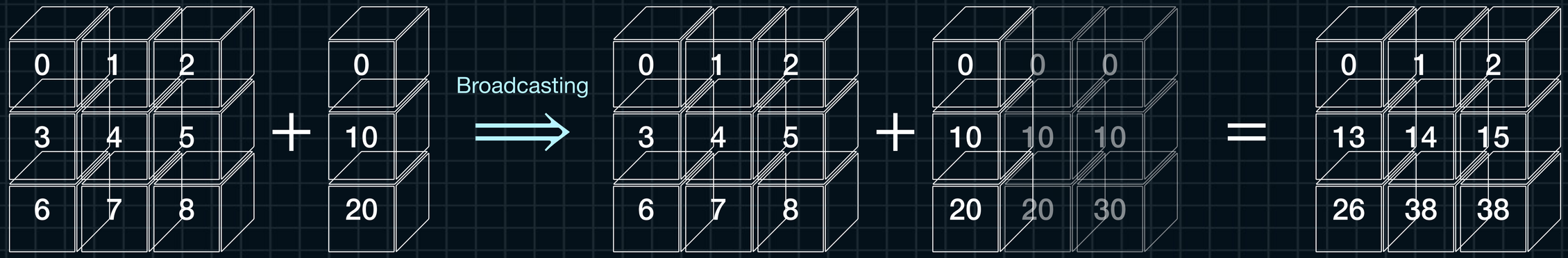
```
import numpy as np

A = np.arange(9).reshape(3, 3)
B = 10*np.arange(3).reshape((-1, 3))
C = A + B

print("A: {}/{}\n{}".format(A.ndim, A.shape, A))
print("B: {}/{}\n{}\n".format(A.ndim, B.shape, B))

print("A + B: {}/{}\n{}".format(A.ndim, C.shape, C))
```

```
A: 2/(3, 3)
[[0 1 2]
 [3 4 5]
 [6 7 8]]
B: 2/(1, 3)
[[ 0 10 20]]
```

```
A + B: 2/(3, 3)
[[ 0 11 22]
 [ 3 14 25]
 [ 6 17 28]]
```

Lecture.6 Element-wise Operations and Broadcasting - Broadcasting in NumPy

When ndims Are Equal(Matrices)



Lecture.6 Element-wise Operations and Broadcasting - Broadcasting in NumPy

When ndims Are Equal(Matrices)

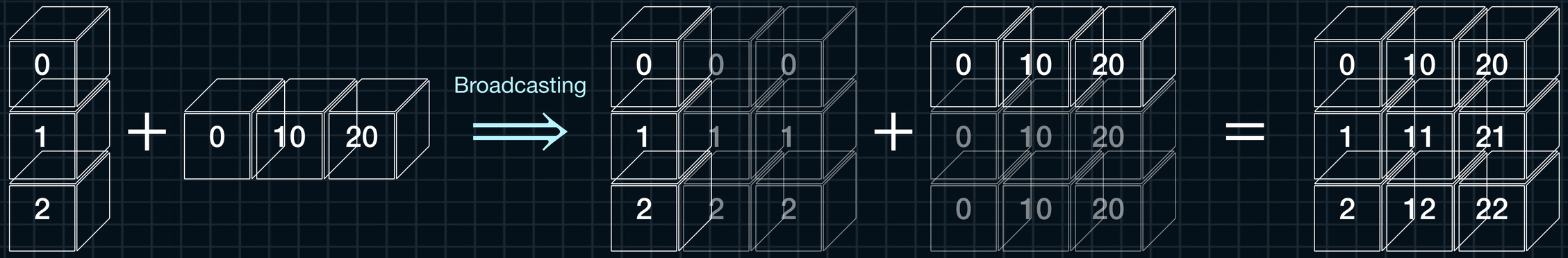
```
import numpy as np

A = np.arange(9).reshape(3, 3)
B = 10*np.arange(3).reshape((3, -1))
C = A + B

print("A: {}/{}\n{}".format(A.ndim, A.shape, A))
print("B: {}/{}\n{}\n".format(A.ndim, B.shape, B))

print("A + B: {}/{}\n{}".format(A.ndim, C.shape, C))
```

```
A: 2/(3, 3)
[[0 1 2]
 [3 4 5]
 [6 7 8]]
B: 2/(3, 1)
[[ 0]
 [10]
 [20]]
```

```
A + B: 2/(3, 3)
[[ 0  1  2]
 [13 14 15]
 [26 27 28]]
```

Lecture.6 Element-wise Operations and Broadcasting - Broadcasting in NumPy

When ndims Are Equal(Matrices)



(3,1) (1,3) (3,3) (3,3) (3,3)

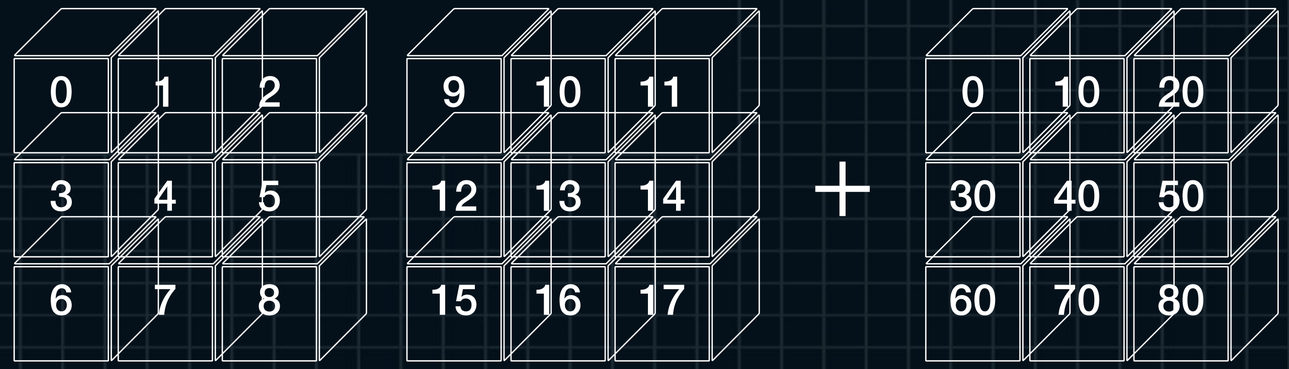
Lecture.6 Element-wise Operations and Broadcasting - Broadcasting in NumPy

When ndims Are Equal(Matrices)

```
import numpy as np

A = np.arange(3).reshape((3, -1))
B = 10*np.arange(3).reshape((-1, 3))
C = A + B

print("A: {}/{}\n{}".format(A.ndim, A.shape, A))
print("B: {}/{}\n{}\n".format(A.ndim, B.shape, B))

print("A + B: {}/{}\n{}".format(A.ndim, C.shape, C))
```

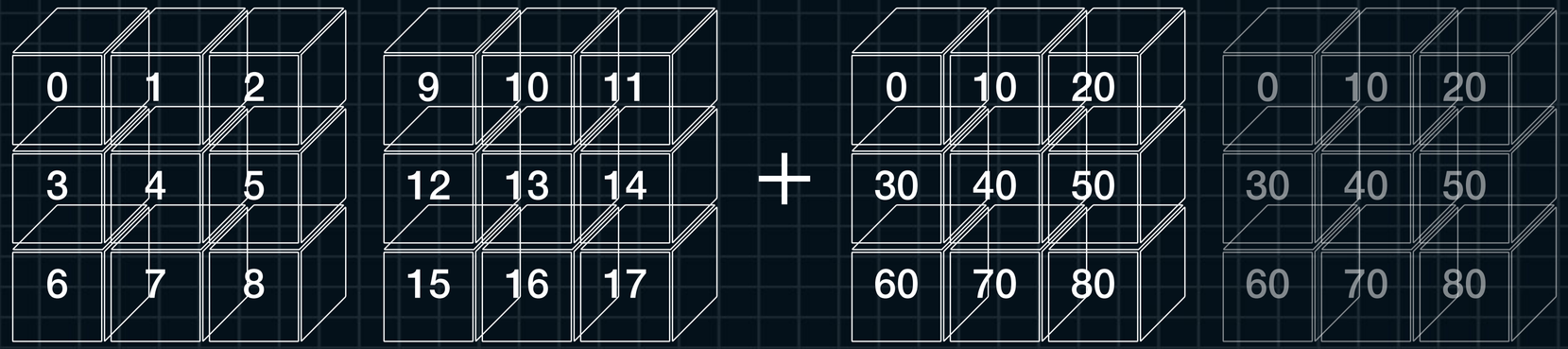
```
A: 2/(3, 1)
[[0]
 [1]
 [2]]
B: 2/(1, 3)
[[ 0 10 20]]
```

```
A + B: 2/(3, 3)
[[ 0 10 20]
 [ 1 11 21]
 [ 2 12 22]]
```



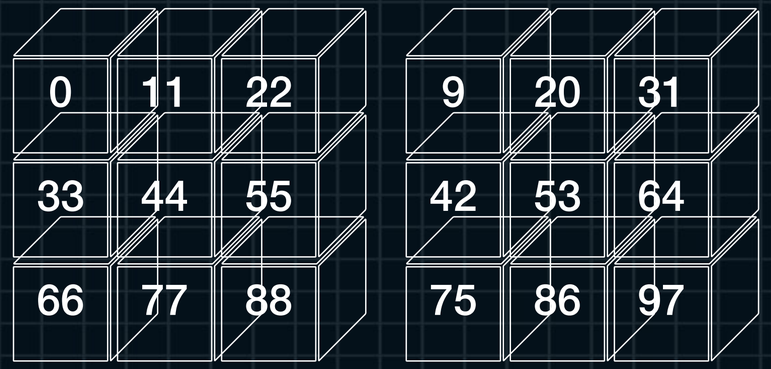
(2,3,3) (1,3,3)

Broadcasting



(2,3,3) (2,3,3)

=



(2,3,3)

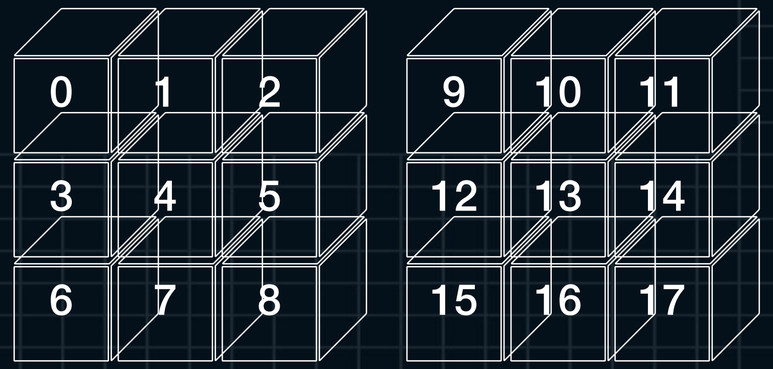
Lecture.6 Element-wise Operations and Broadcasting - Broadcasting in NumPy

When ndims Are Equal(3rd Order Tensors)

```
import numpy as np

A = np.arange(18).reshape((2, 3, 3))
B = 10*np.arange(9).reshape((1, 3, 3))
C = A + B

print("A: {}/{}\n{}".format(A.ndim, A.shape, A))
print("B: {}/{}\n{}\n".format(A.ndim, B.shape, B))

print("A + B: {}/{}\n{}".format(A.ndim, C.shape, C))
```

```
A + B: 3/(2, 3, 3)
[[[ 0 11 22]
  [33 44 55]
  [66 77 88]]

 [[ 9 20 31]
  [42 53 64]
  [75 86 97]]]
```

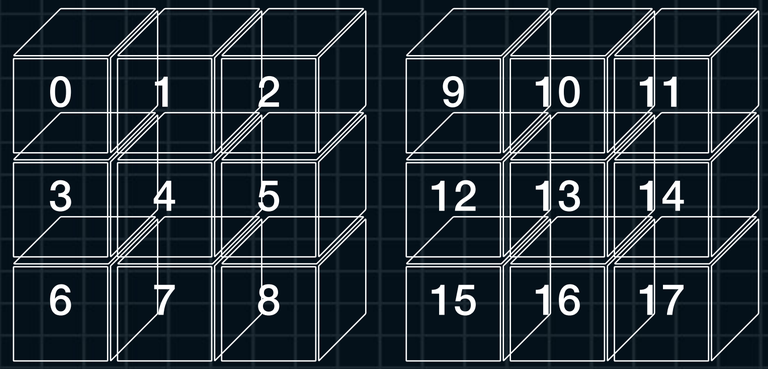
```
A: 3/(2, 3, 3)
[[[ 0  1  2]
  [ 3  4  5]
  [ 6  7  8]]

 [[ 9 10 11]
  [12 13 14]
  [15 16 17]]]
B: 3/(1, 3, 3)
[[[ 0 10 20]
  [30 40 50]
  [60 70 80]]]
```



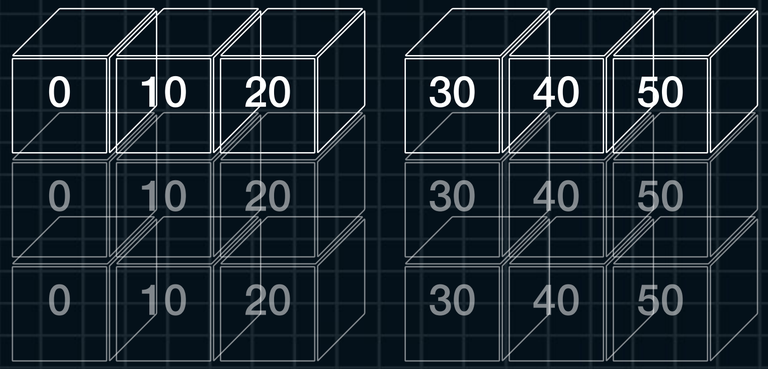
(2,3,3)

(2,1,3)

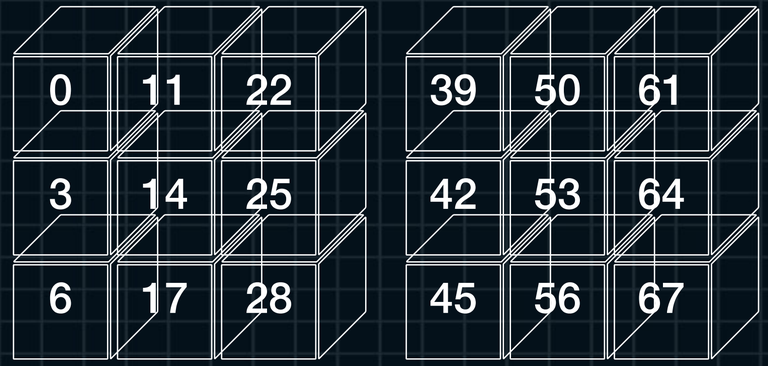
Broadcasting



(2,3,3)



(2,3,3)



(2,3,3)

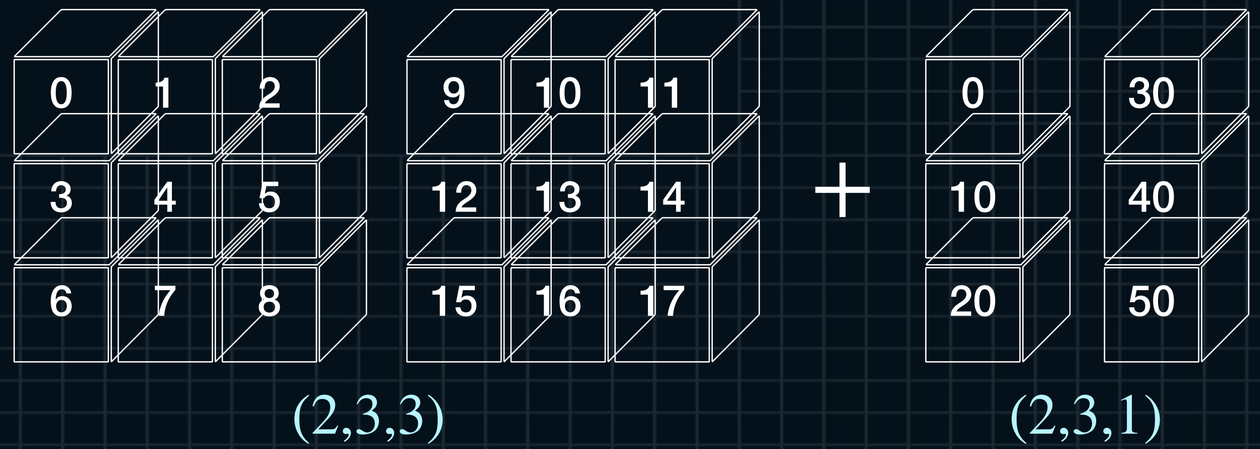
Lecture.6 Element-wise Operations and Broadcasting - Broadcasting in NumPy

When ndims Are Equal(3rd Order Tensors)

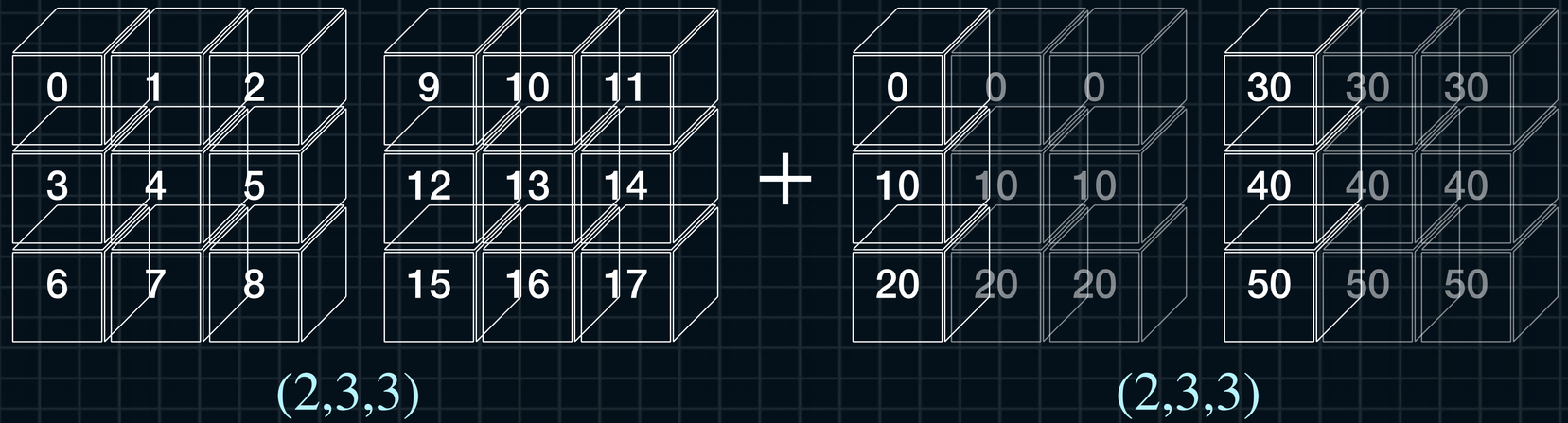
```
import numpy as np

A = np.arange(18).reshape((2, 3, 3))
B = 10*np.arange(6).reshape((2, 1, 3))
C = A + B

print("A: {}/{}\n{}".format(A.ndim, A.shape, A))
print("B: {}/{}\n{}\n".format(A.ndim, B.shape, B))

print("A + B: {}/{}\n{}".format(A.ndim, C.shape, C))
```

```
A: 3/(2, 3, 3)
[[[ 0  1  2]
  [ 3  4  5]
  [ 6  7  8]]

 [[ 9 10 11]
  [12 13 14]
  [15 16 17]]]
B: 3/(2, 1, 3)
[[[ 0 10 20]]

 [[30 40 50]]]
```

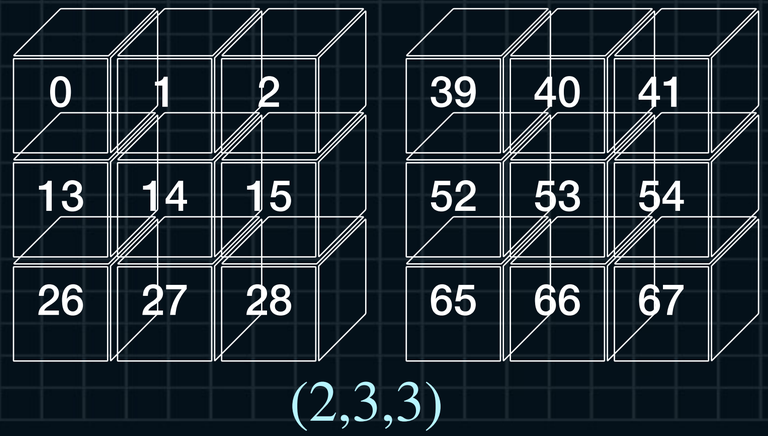
```
A + B: 3/(2, 3, 3)
[[[ 0 11 22]
  [ 3 14 25]
  [ 6 17 28]]

 [[39 50 61]
  [42 53 64]
  [45 56 67]]]
```



Broadcasting



=



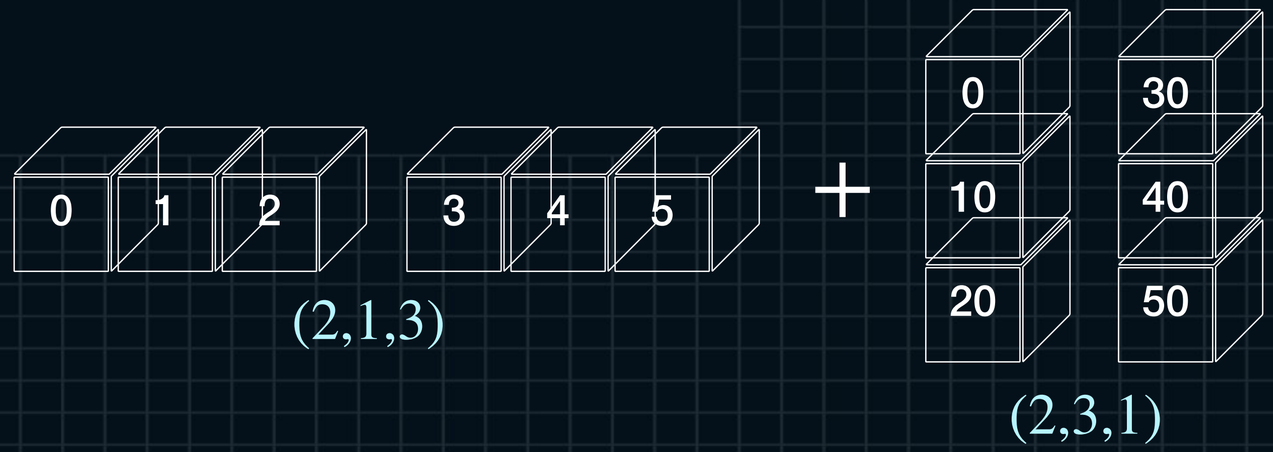
Lecture.6 Element-wise Operations and Broadcasting - Broadcasting in NumPy

When ndims Are Equal(3rd Order Tensors)

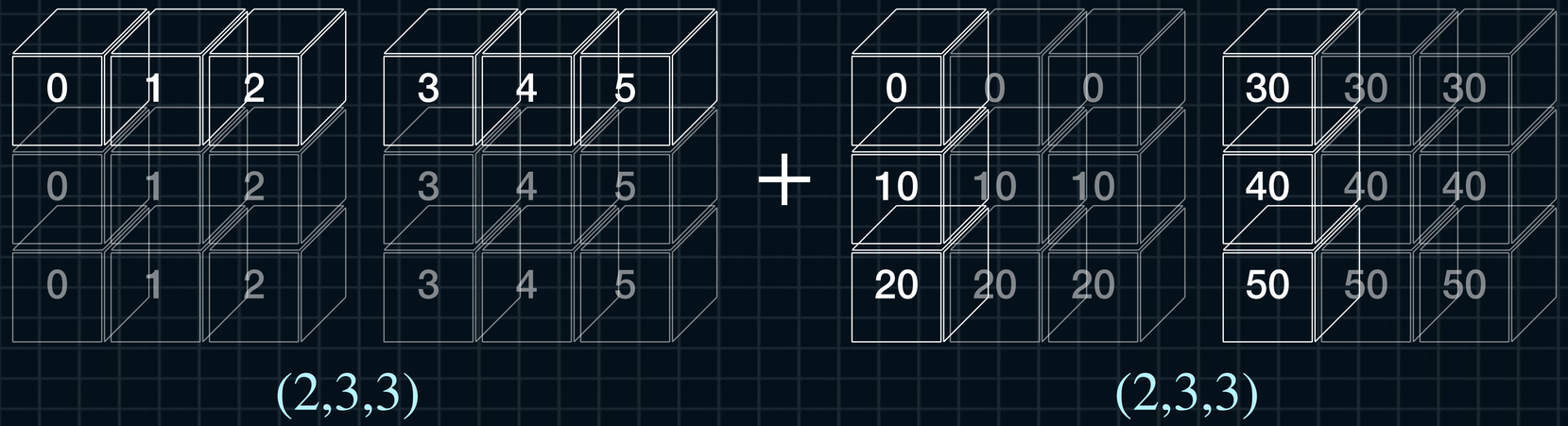
```
import numpy as np

A = np.arange(18).reshape((2, 3, 3))
B = 10*np.arange(6).reshape((2, 3, 1))
C = A + B

print("A: {}/{}\n{}".format(A.ndim, A.shape, A))
print("B: {}/{}\n{}\n".format(A.ndim, B.shape, B))

print("A + B: {}/{}\n{}".format(A.ndim, C.shape, C))
```

```
A: 3/(2, 3, 3)
[[[ 0  1  2]
  [ 3  4  5]
  [ 6  7  8]]

 [[ 9 10 11]
  [12 13 14]
  [15 16 17]]]
B: 3/(2, 3, 1)
[[[ 0]
  [10]
  [20]]

 [[30]
  [40]
  [50]]]
```

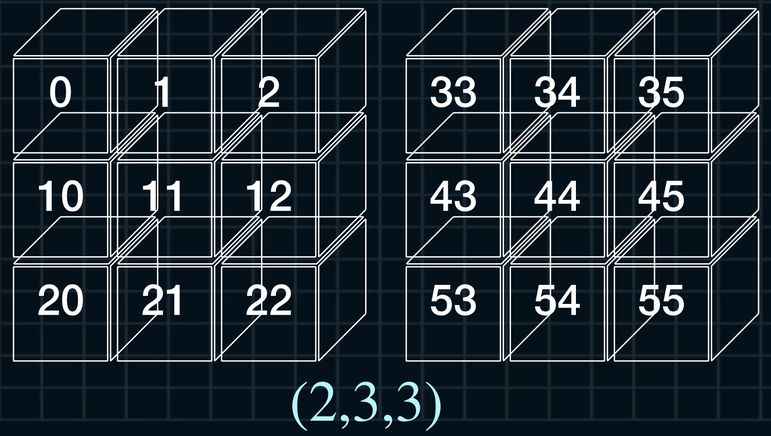
```
A + B: 3/(2, 3, 3)
[[[ 0  1  2]
  [13 14 15]
  [26 27 28]]

 [[39 40 41]
  [52 53 54]
  [65 66 67]]]
```



Broadcasting



=



Lecture.6 Element-wise Operations and Broadcasting - Broadcasting in NumPy

When ndims Are Equal(3rd Order Tensors)

```
import numpy as np

A = np.arange(6).reshape((2, 1, 3))
B = 10*np.arange(6).reshape((2, 3, 1))
C = A + B

print("A: {}/{}\n{}".format(A.ndim, A.shape, A))
print("B: {}/{}\n{}\n".format(A.ndim, B.shape, B))

print("A + B: {}/{}\n{}".format(A.ndim, C.shape, C))
```

```
A: 3/(2, 1, 3)
[[[0 1 2]]

 [[3 4 5]]]
B: 3/(2, 3, 1)
[[[ 0]
  [10]
  [20]]

 [[30]
  [40]
  [50]]]
```

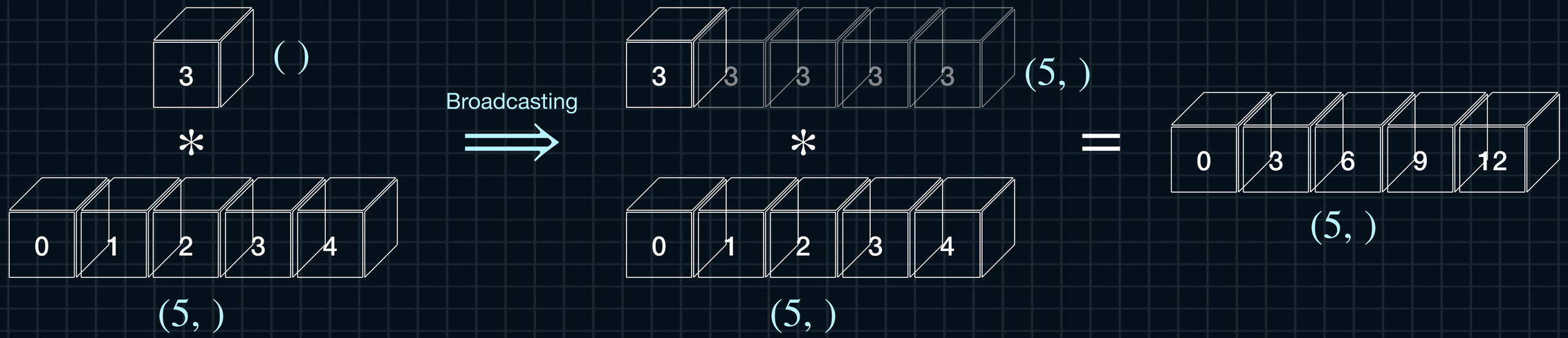
```
A + B: 3/(2, 3, 3)
[[[ 0  1  2]
  [10 11 12]
  [20 21 22]]

 [[33 34 35]
  [43 44 45]
  [53 54 55]]]
```

Lecture.6 Element-wise Operations and Broadcasting - Broadcasting in NumPy

When ndims Are Not Equal

```
import numpy as np

a = np.array(3)
u = np.arange(5)

print("shapes: {}/{}".format(a.shape, u.shape))
print("a: ", a)
print("u: ", u, '\n')

print("a*u: ", a*u)
```

```
shapes: ()/(5,)
a:  3
u:  [0 1 2 3 4]

a*u:  [ 0  3  6  9 12]
```



Lecture.6 Element-wise Operations and Broadcasting - Broadcasting in NumPy

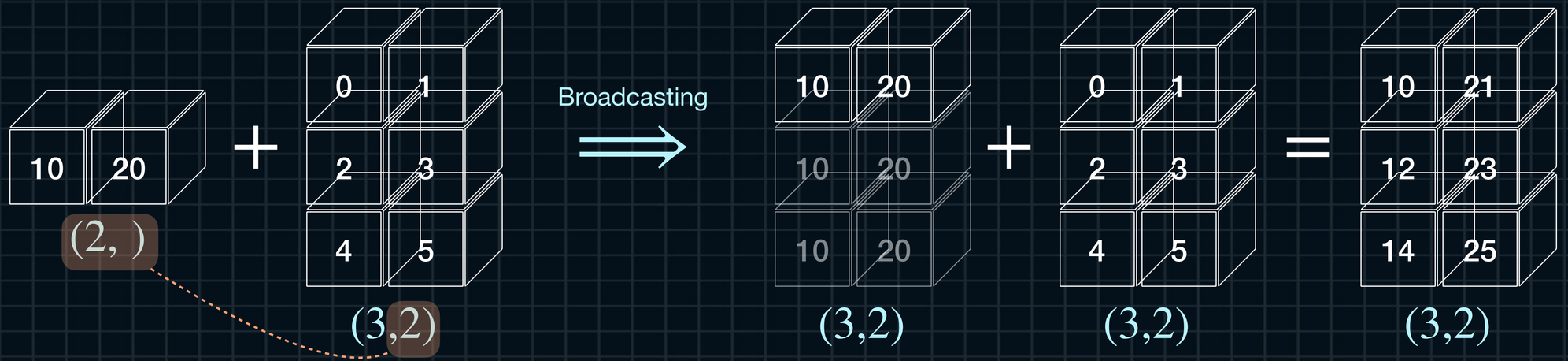
When ndims Are Not Equal

```
import numpy as np

a = np.array(3)
u = np.arange(1, 5)

shapes = "shapes: {}/{}"
print(shapes.format(a.shape,
                    u.shape))
print("a: ", a)
print("u: ", u, '\n')

print("a + u: ", a + u)
print("a - u: ", a - u)
print("a * u: ", a * u)
print("a / u: ", a / u)
print("a // u: ", a // u)
print("a % u: ", a % u)
print("a ** u: ", a ** u,
      '\n')

print("a > u: ", a > u)
print("a >= u: ", a >= u)
print("a < u: ", a < u)
print("a <= u: ", a <= u)
print("a == u: ", a == u)
print("a != u: ", a != u)
```

```
shapes: ()/(4,)
a:  3
u:  [1 2 3 4]

a + u:  [4 5 6 7]
a - u:  [ 2  1  0 -1]
a * u:  [ 3  6  9 12]
a / u:  [3.   1.5  1.   0.75]
a // u:  [3 1 1 0]
a % u:  [0 1 0 3]
a ** u:  [ 3  9 27 81]

a > u:  [ True  True False False]
a >= u:  [ True  True  True False]
a < u:  [False False False  True]
a <= u:  [False False  True  True]
a == u:  [False False  True False]
a != u:  [ True  True False  True]
```

Lecture.6 Element-wise Operations and Broadcasting - Broadcasting in NumPy

When ndims Are Not Equal

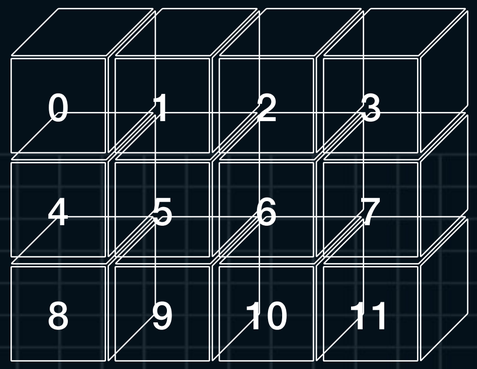


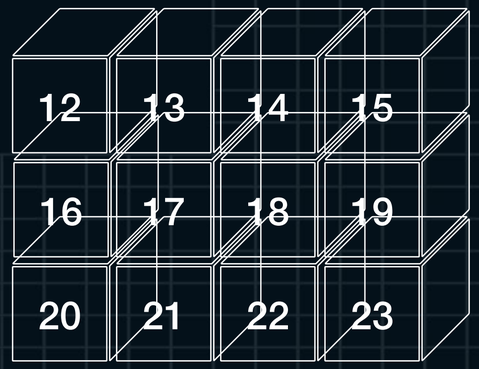
Lecture.6 Element-wise Operations and Broadcasting - Broadcasting in NumPy

When ndims Are Not Equal

```
import numpy as np

A = np.array([10, 20])
B = np.arange(6).reshape((3, 2))
C = A + B

print("A: {}/{}\n{}".format(A.ndim, A.shape, A))
print("B: {}/{}\n{}\n".format(A.ndim, B.shape, B))

print("A + B: {}/{}\n{}".format(A.ndim, C.shape, C))
```

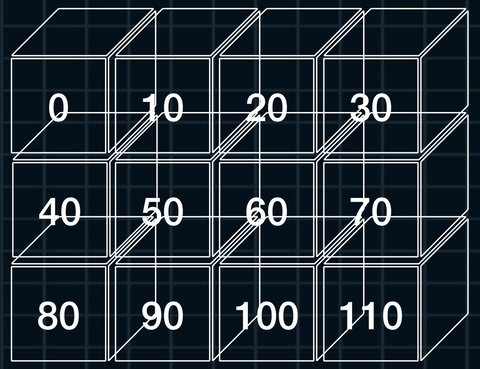
```
A: 1/(2,)
[10 20]
B: 2/(3, 2)
[[0 1]
 [2 3]
 [4 5]]

A + B: 2/(3, 2)
[[10 21]
 [12 23]
 [14 25]]
```





(2,3,4)

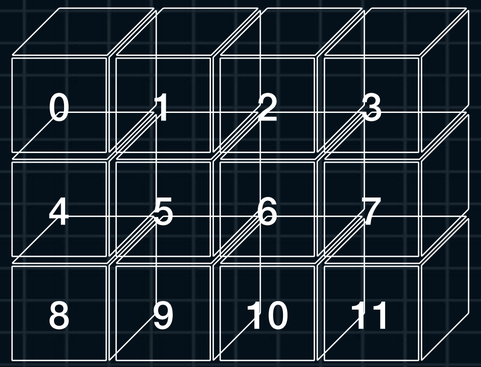
+

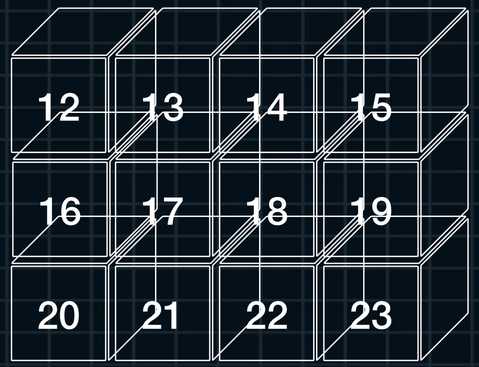


(3,4)

Broadcasting

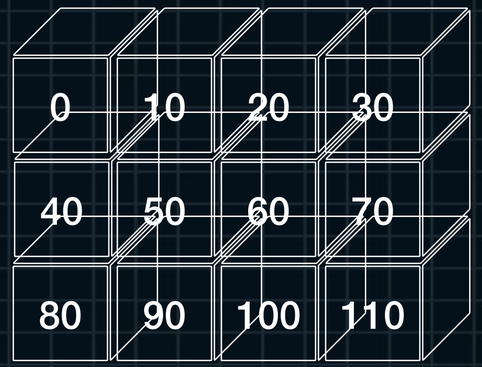
⟹

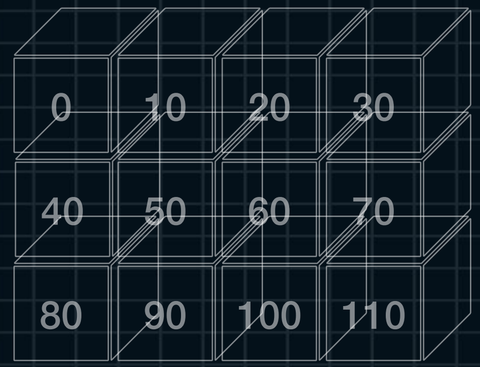




(2,3,4)

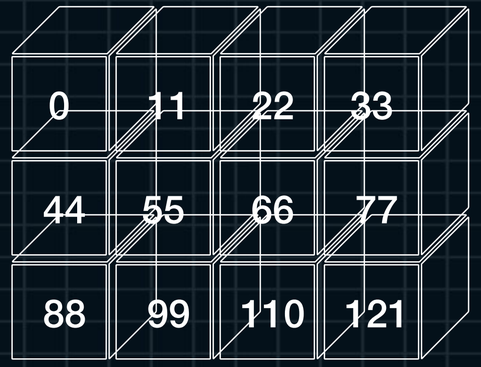
+

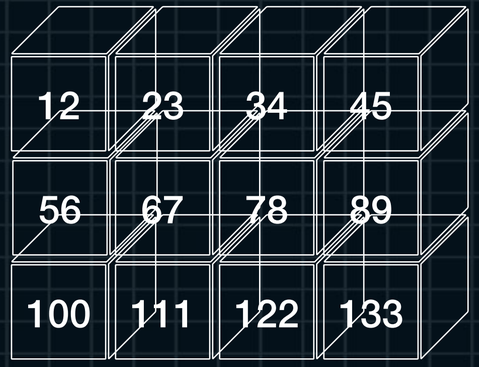




(2,3,4)

=





(2,3,4)

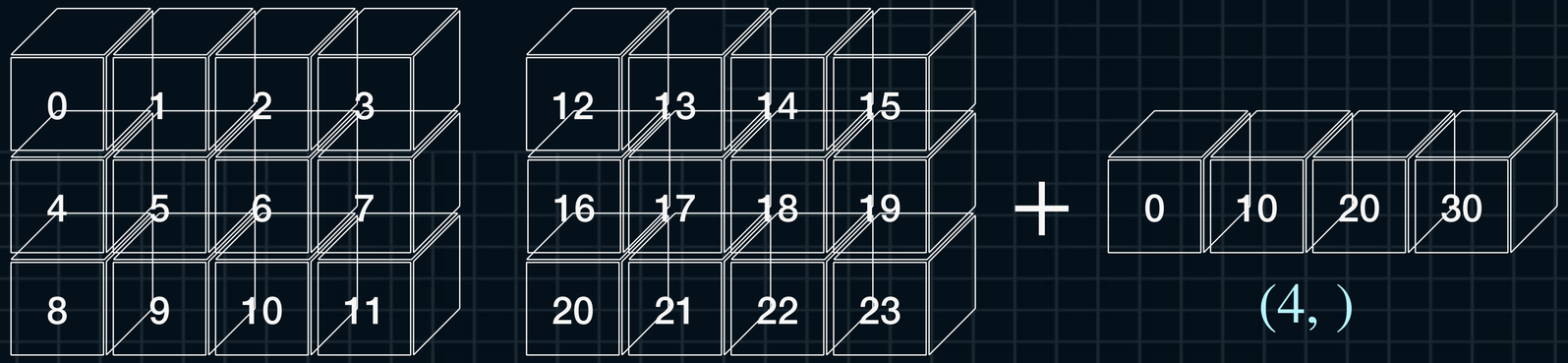
Lecture.6 Element-wise Operations and Broadcasting - Broadcasting in NumPy

When ndims Are Not Equal

```
import numpy as np

A = np.arange(2*3*4).reshape((2, 3, 4))
B = 10*np.arange(3*4).reshape((3, 4))
C = A + B

print("A: {}/{}\n{}".format(A.ndim, A.shape, A))
print("B: {}/{}\n{}\n".format(A.ndim, B.shape, B))

print("A + B: {}/{}\n{}".format(A.ndim, C.shape, C))
```

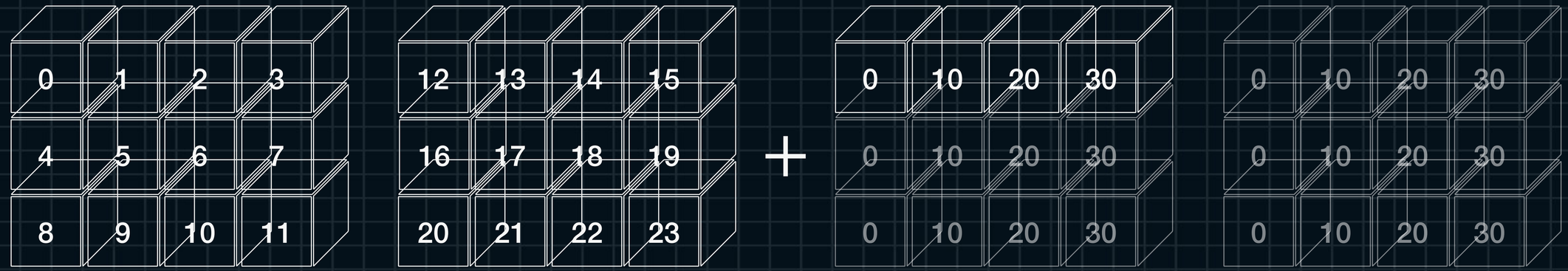
```
A: 3/(2, 3, 4)
[[[ 0  1  2  3]
  [ 4  5  6  7]
  [ 8  9 10 11]]

 [[12 13 14 15]
  [16 17 18 19]
  [20 21 22 23]]]
B: 2/(3, 4)
[[  0  10  20  30]
 [ 40  50  60  70]
 [ 80  90 100 110]]
```

```
A + B: 3/(2, 3, 4)
[[[  0  11  22  33]
  [ 44  55  66  77]
  [ 88  99 110 121]]

 [[ 12  23  34  45]
  [ 56  67  78  89]
  [100 111 122 133]]]
```



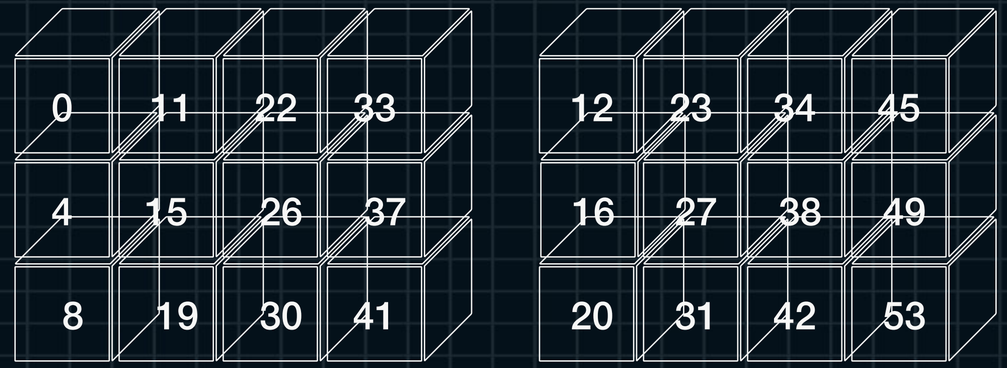
(2,3,4) (4,)

Broadcasting



(2,3,4) (2,3,4)

=



(2,3,4)

Lecture.6 Element-wise Operations and Broadcasting - Broadcasting in NumPy

When ndims Are Not Equal

```
import numpy as np

A = np.arange(2*3*4).reshape((2, 3, 4))
B = 10*np.arange(4)
C = A + B

print("A: {}/{}\n{}".format(A.ndim, A.shape, A))
print("B: {}/{}\n{}\n".format(A.ndim, B.shape, B))

print("A + B: {}/{}\n{}".format(A.ndim, C.shape, C))
```

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

 [[12 13 14 15]
  [16 17 18 19]
  [20 21 22 23]]]
B: 1/(4,)
[ 0 10 20 30]
```

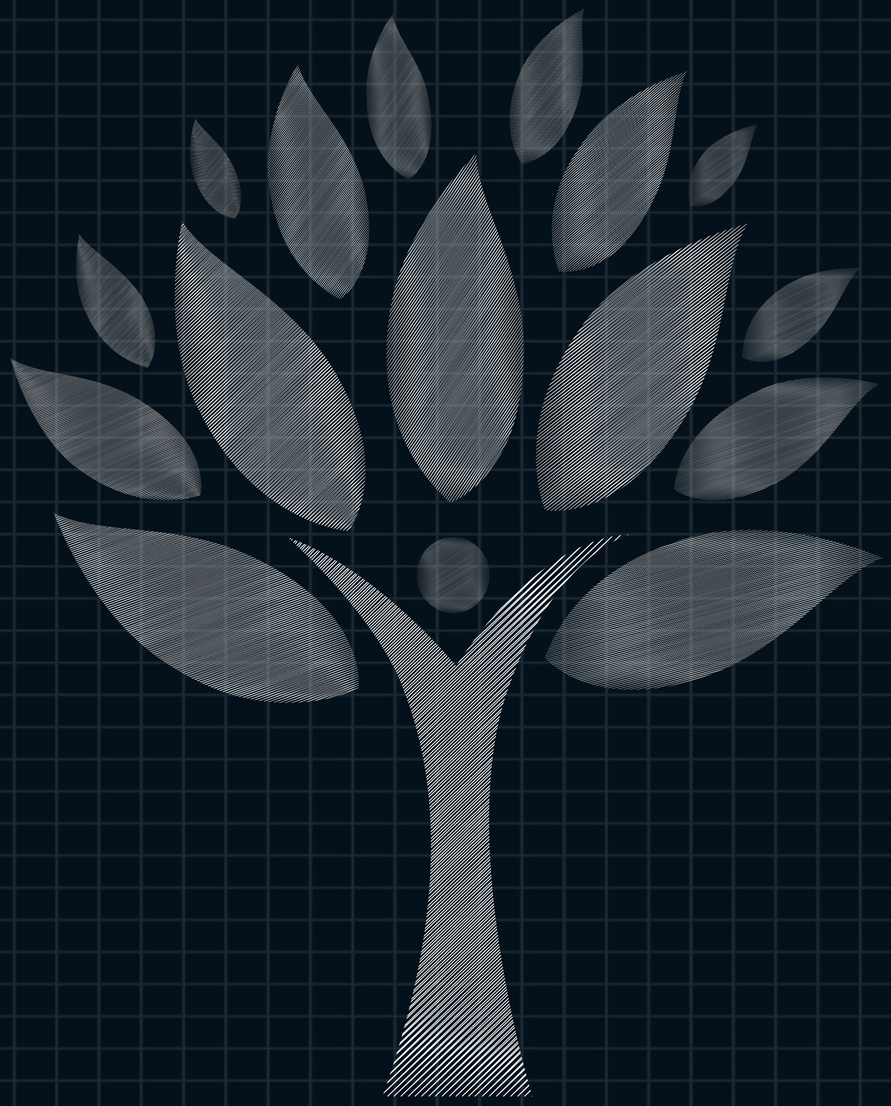
```
A + B: 3/(2, 3, 4)
[[[ 0 11 22 33]
  [ 4 15 26 37]
  [ 8 19 30 41]]

 [[12 23 34 45]
  [16 27 38 49]
  [20 31 42 53]]]
```

Lecture.6 Element-wise Operations and Broadcasting - Broadcasting in NumPy

When ndims Are Not Equal

A: (2, 3, 4)
B: (3, 4)
C: (4)

A: (a, b, c, d, e)
B: (b, c, d, e)
C: (c, d, e)
D: (d, e)
E: (e)



NumPy Master Class

Lecture.6
Element-wise Operations and Broadcasting