NYU TANDON SCHOOL OF ENGINEERING

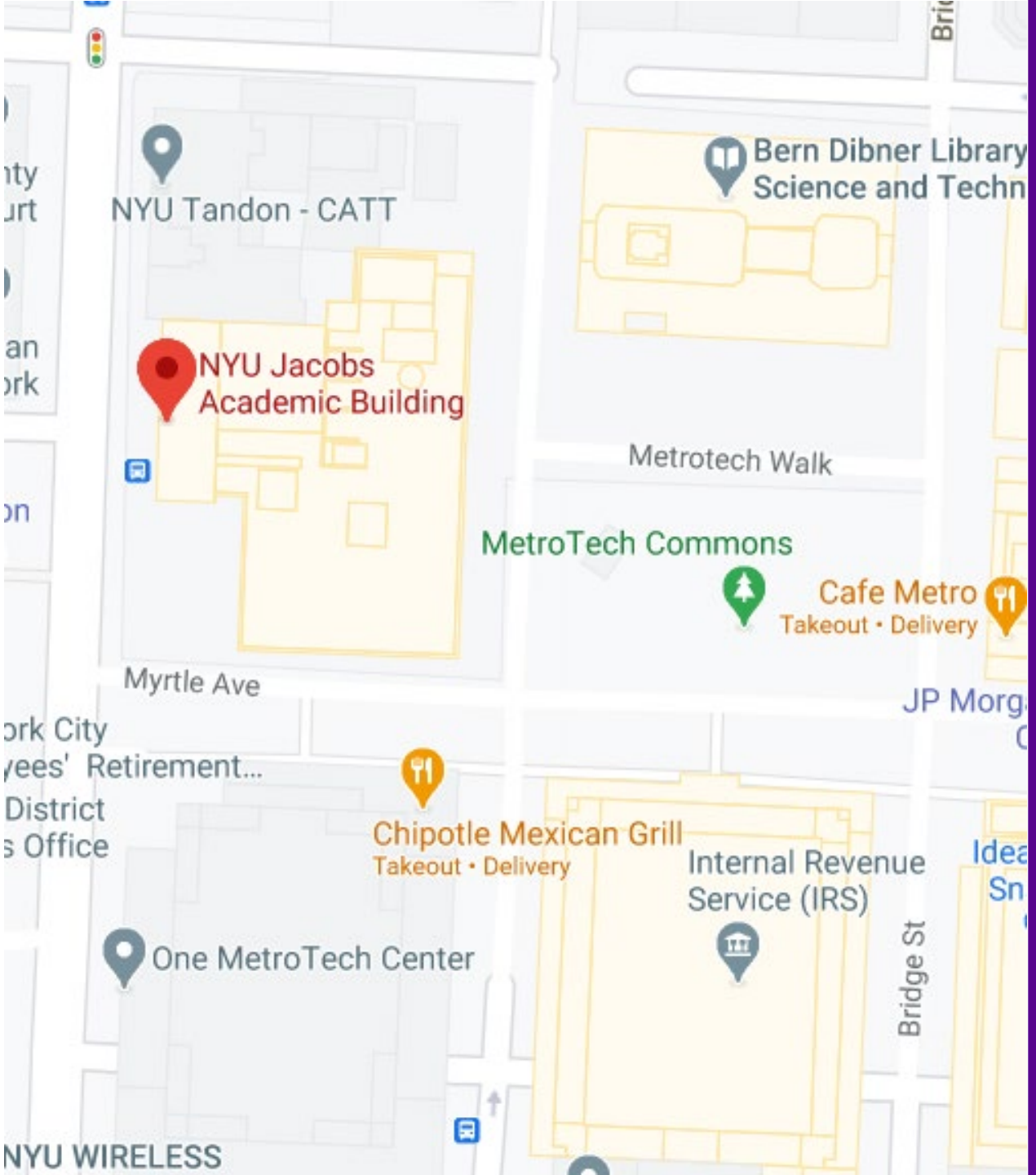
Lecture 3
Arrays

Programming for Business Analytics
MG-GY 8401

NYU TANDON SCHOOL OF ENGINEERING

Agenda

- Arrays
  - Accessing
  - Manipulating
  - Calculating



NYU TANDON SCHOOL OF ENGINEERING

Logistics

- Office Hours
  - Tuesday's 6-7PM ET
- Homework
  - Homework 3
  - Homework 2

TechRepublic.

These old programming languages are still critical to big companies. But nobody wants to learn them

by **Owen Hughes** in **Developer**
on June 30, 2021, 8:38 AM PST

Large organizations still rely on ageing IT systems and programming languages to run their mainframes. But as traditional developers reach retirement age, new hires are reluctant to pick up old skills.

Classes

- Object oriented programming is a programming paradigm focused on classes
- Python is an object oriented programming language

Classes

- Classes ***encapsulate*** containers and functions
  - Store data
  - Operate on data
- Dot notation

Classes

- Objects instantiate classes
- Classes can ***inherit*** attributes and methods from another class

Poll

True or False: Does the following code raise an exception?

A. True
B. False

```
class ExampleClass(object):
    def __init__(self):
        pass

    def example_instance_method():
        return "Hello World"

example_object = ExampleClass()
example_object.example_instance_method()
```

True or False: Does the following code raise an exception?

A. True
B. False

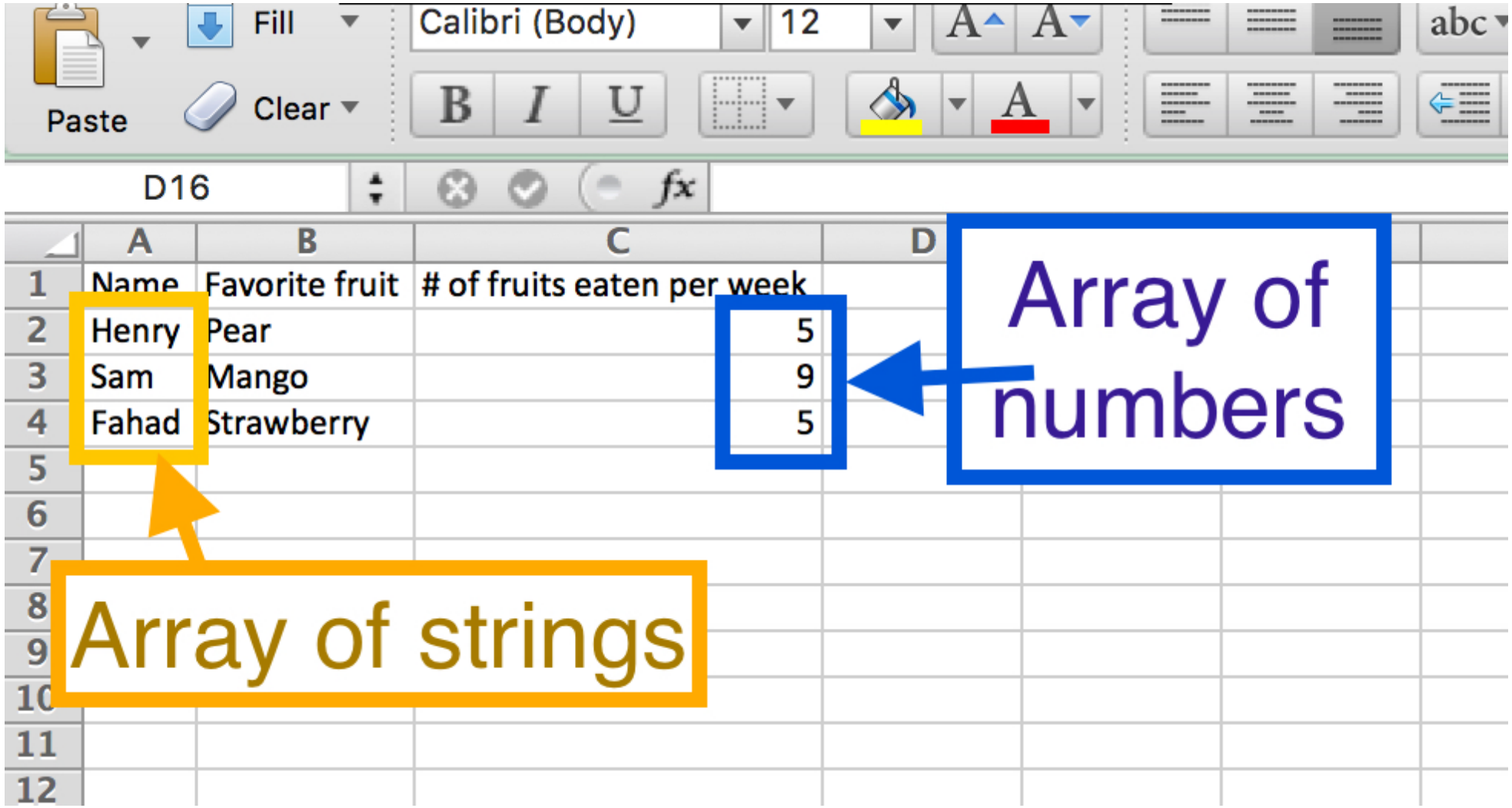
```
class ExampleClass(object):
    def __init__(self):
        pass

    def example_instance_method():
        return "Hello World"

example_object = ExampleClass()
example_object.example_instance_method()
```

Class Method not Instance Method

Spreadsheets

"Microsoft Excel is probably the most successful data analytics platform of all times."

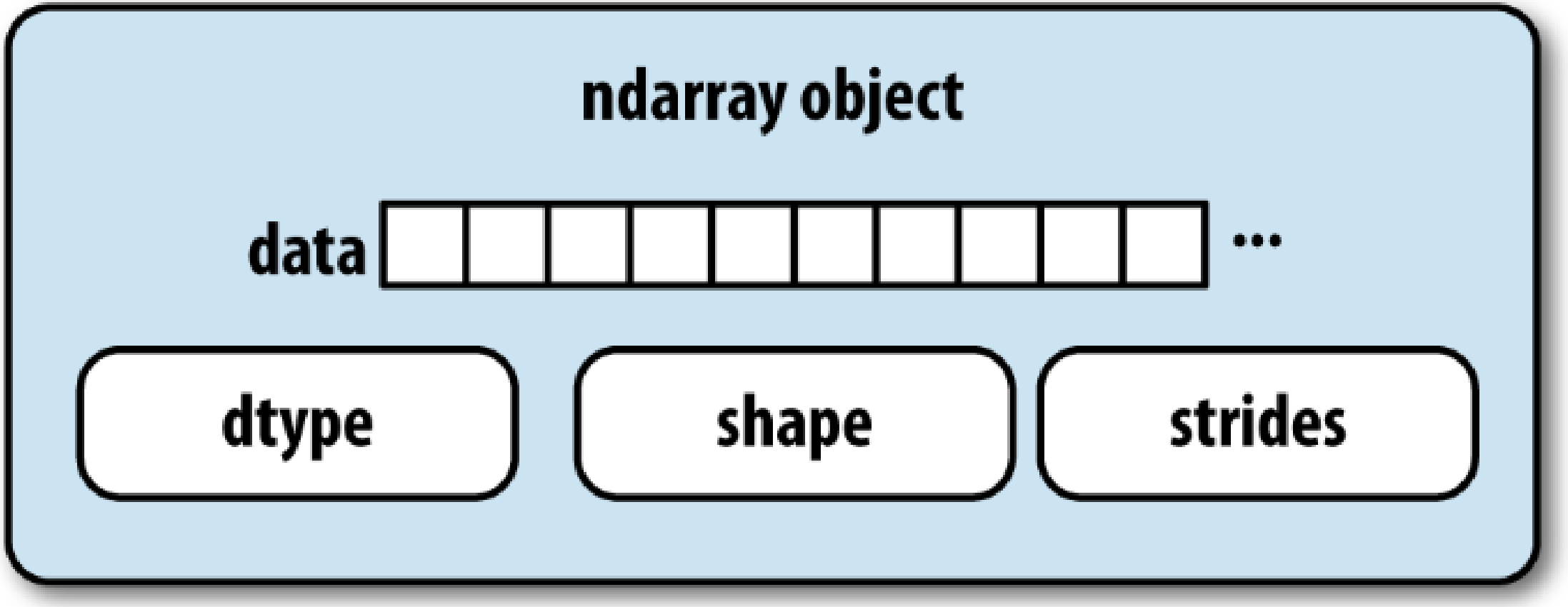
Spreadsheets



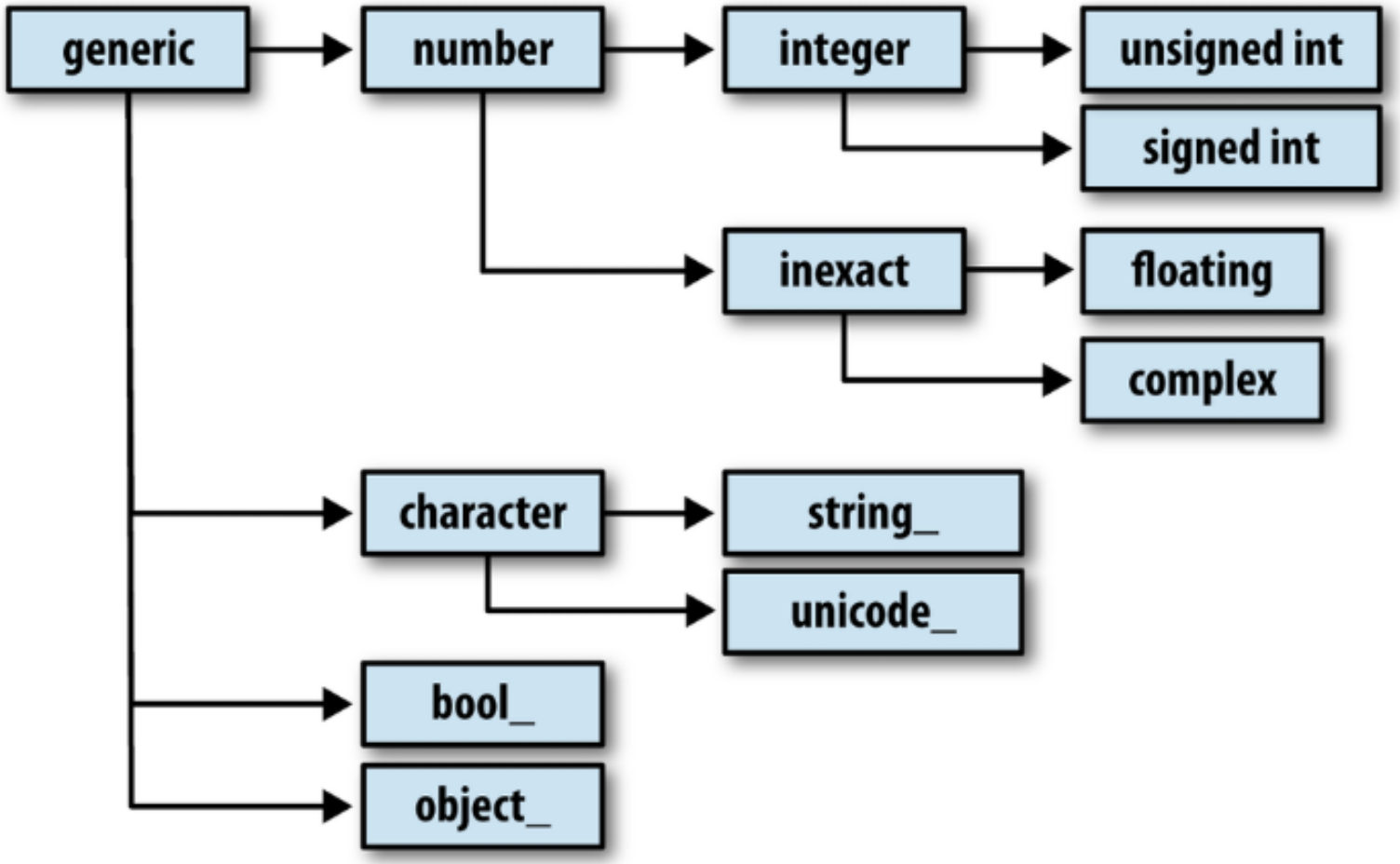
Package

- import numpy
- import numpy as np
- from numpy import *

Properties of Arrays



Data Type

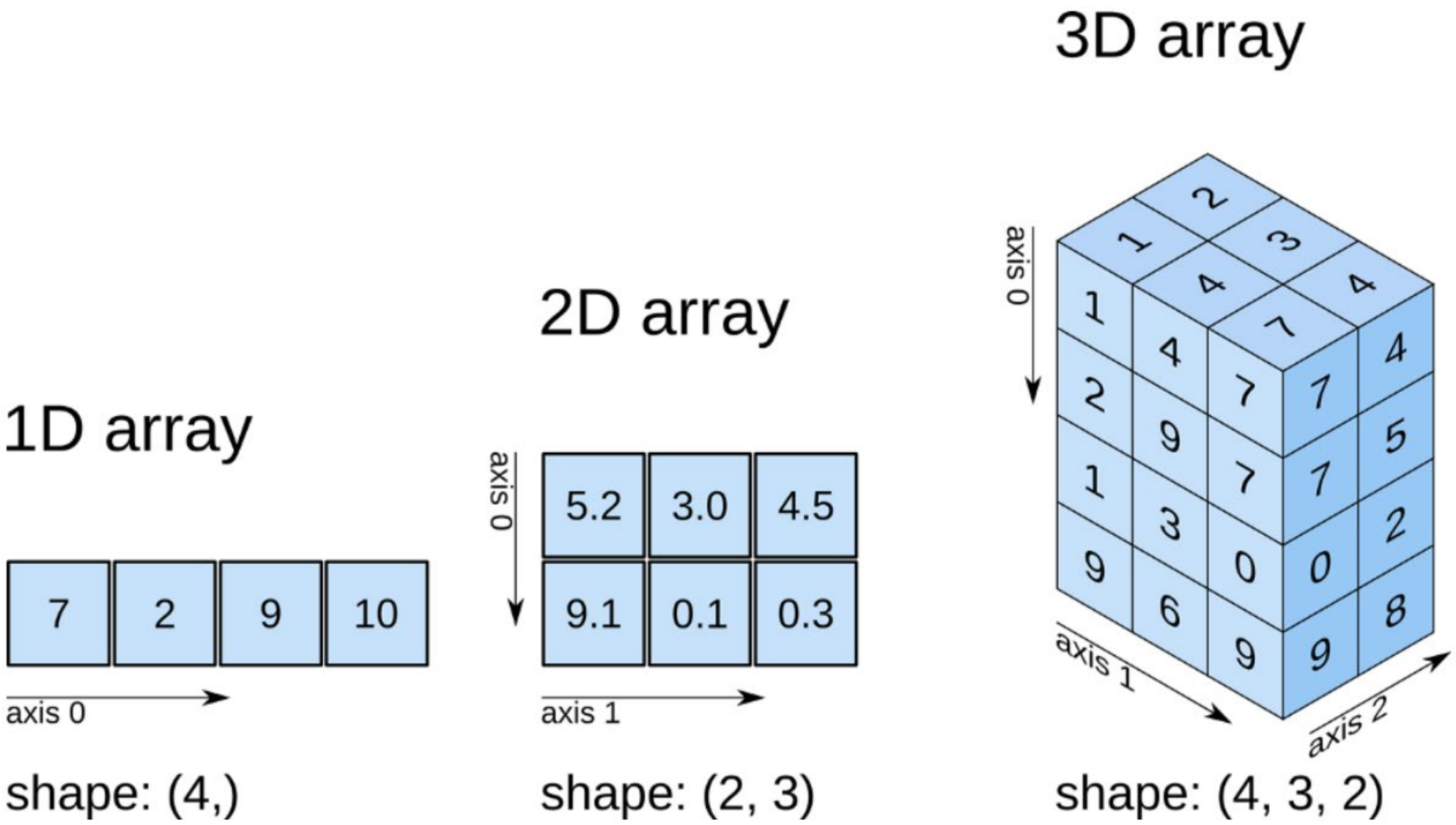


Shape

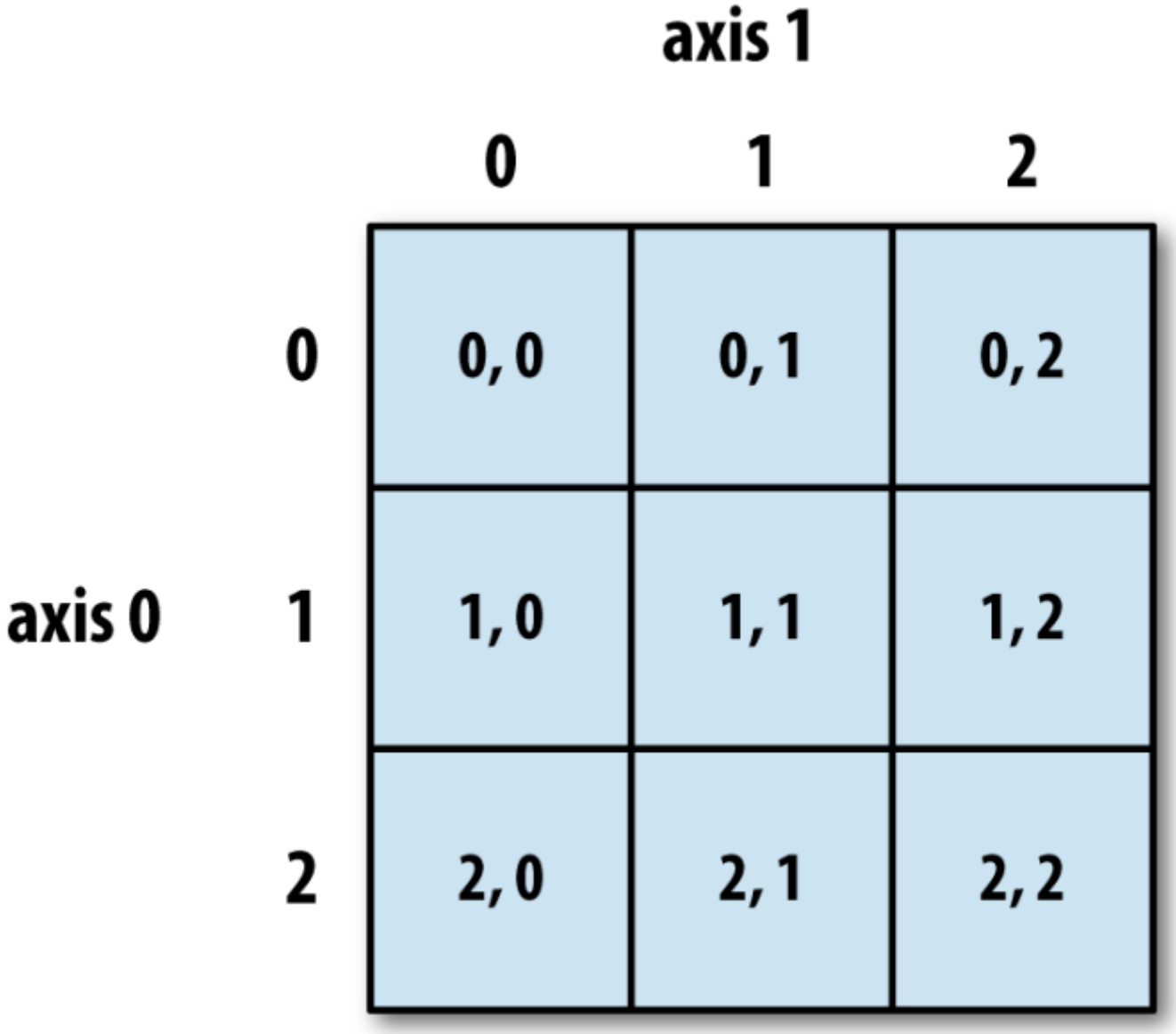
0	1	2	3	4	5	6	7	8	9	10	11

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

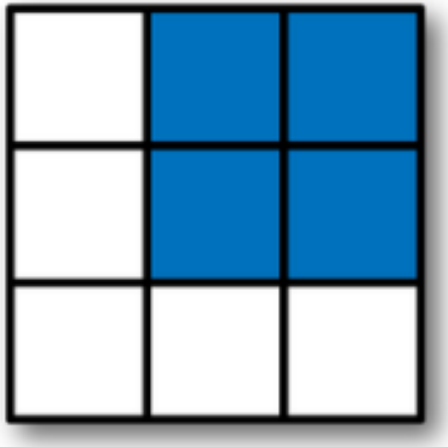
Shape

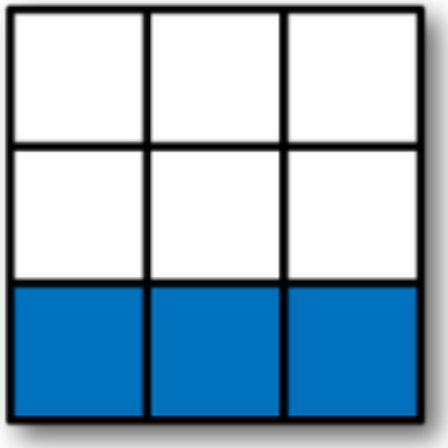


Indices



Selecting Entries

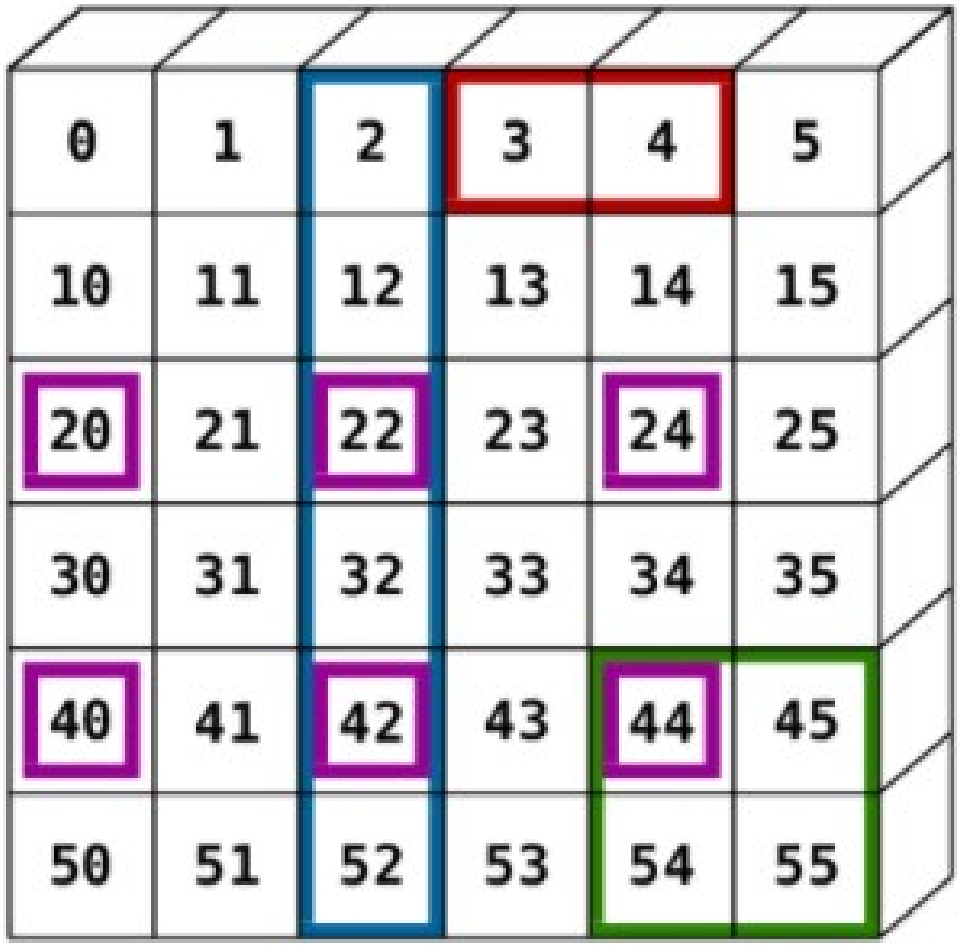




Expression	Shape
`arr[:2, 1:]`	`(2, 2)`
`arr[2]`	`(3,)`
`arr[2, :]`	`(3,)`
`arr[2:, :]`	`(1, 3)`

Selecting Entries

```
>>> a[0, 3:5]
array([3, 4])

>>> a[4:, 4:]
array([[44, 55],
       [54, 55]])

>>> a[:, 2]
a([2, 12, 22, 32, 42, 52])

>>> a[2::2, ::2]
array([[20, 22, 24],
       [40, 42, 44]])
```

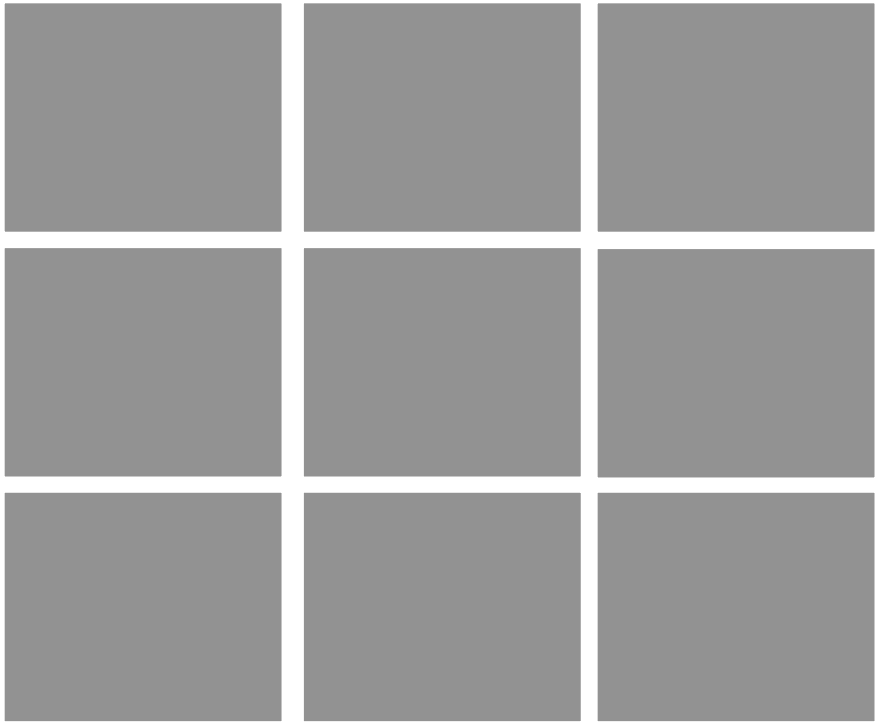


Breakout Room

Consider the (3,3) array **a**. Determine the shape of the following arrays

1. **a**[:2, 1:]
2. **a**[2]
3. **a**[2, :]
4. **a**[2:, :]
5. **a**[:, :2]
6. **a**[1, :2]
7. **a**[1:2, :2]

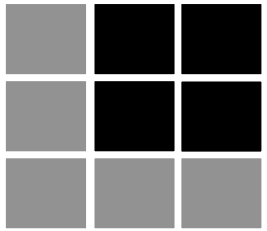
a[:,:]



Breakout Room

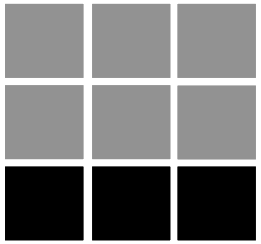
Consider the (3,3) array **a**. Determine the shape of the following arrays

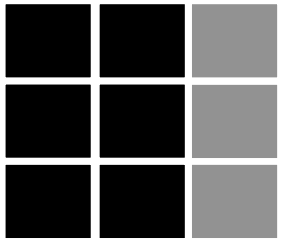
1. **a**[:2, 1:]
2. **a**[2]
3. **a**[2, :]
4. **a**[2:, :]
5. **a**[:, :2]
6. **a**[1, :2]
7. **a**[1:2, :2]



1

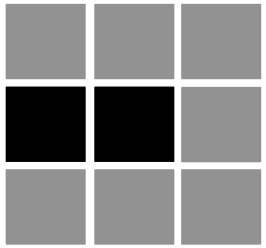
2,3,4





5

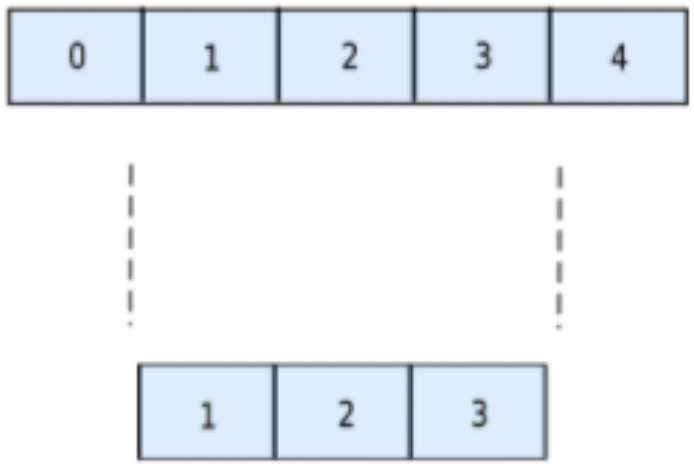
6,7



Views and Copies

View

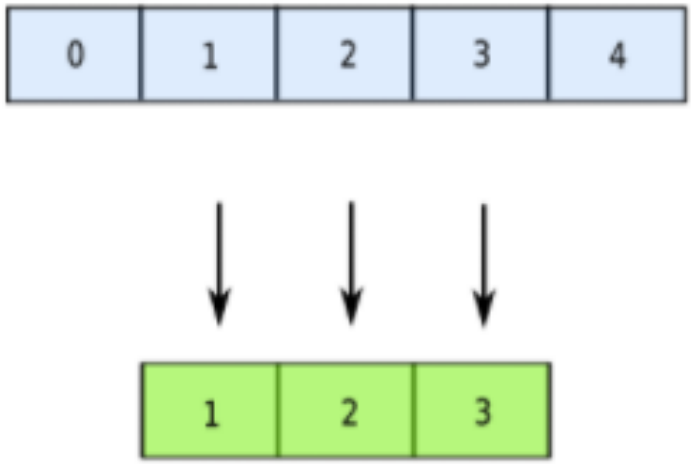
arr[1:4]



Merely offset is changed

Copy

arr[[1,2,3]]



Elements are copied and a new object is created

Reshaping

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

```
arr.reshape((4, 3), order=?)
```

C order (row major)

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

`order='C'`

Fortran order (column major)

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

`order='F'`

Reshaping

```
a1 = np.arange(1, 13)
```

1	2	3	4	5	6	7	8	9	10	11	12

```
a2 = np.arange(13, 25)
```

13	14	15	16	17	18	19	20	21	22	23	24

```
np.stack((a1, a2))
```

1	2	3	4	5	6	7	8	9	10	11	12
13	14	15	16	17	18	19	20	21	22	23	24

```
np.hstack((a1, a2)
```

1	2	3	4	5	…	20	21	22	23	24

```
np.stack((a1, a2), axis=1)
```

1	13
2	14
3	15
4	16
…	…
9	21
10	22
11	23
12	24

Filtering

- We can compare the entries of arrays

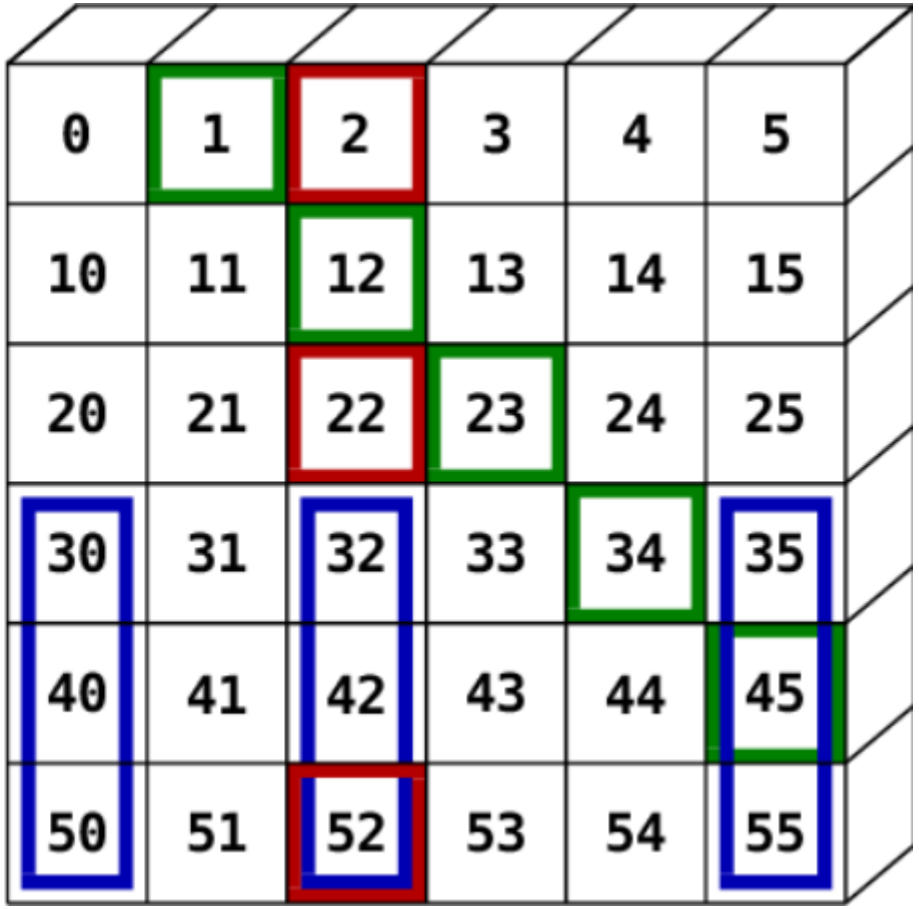
```
>>> a = np.array([1, 2, 3, 4])
>>> b = np.array([4, 2, 2, 4])
>>> a > b
array([False, False,  True, False], dtype=bool)
```

- We can check arrays with values True/False

```
>>> np.all([True, True, False])

False
>>> np.any([True, True, False])
True
```

Filtering

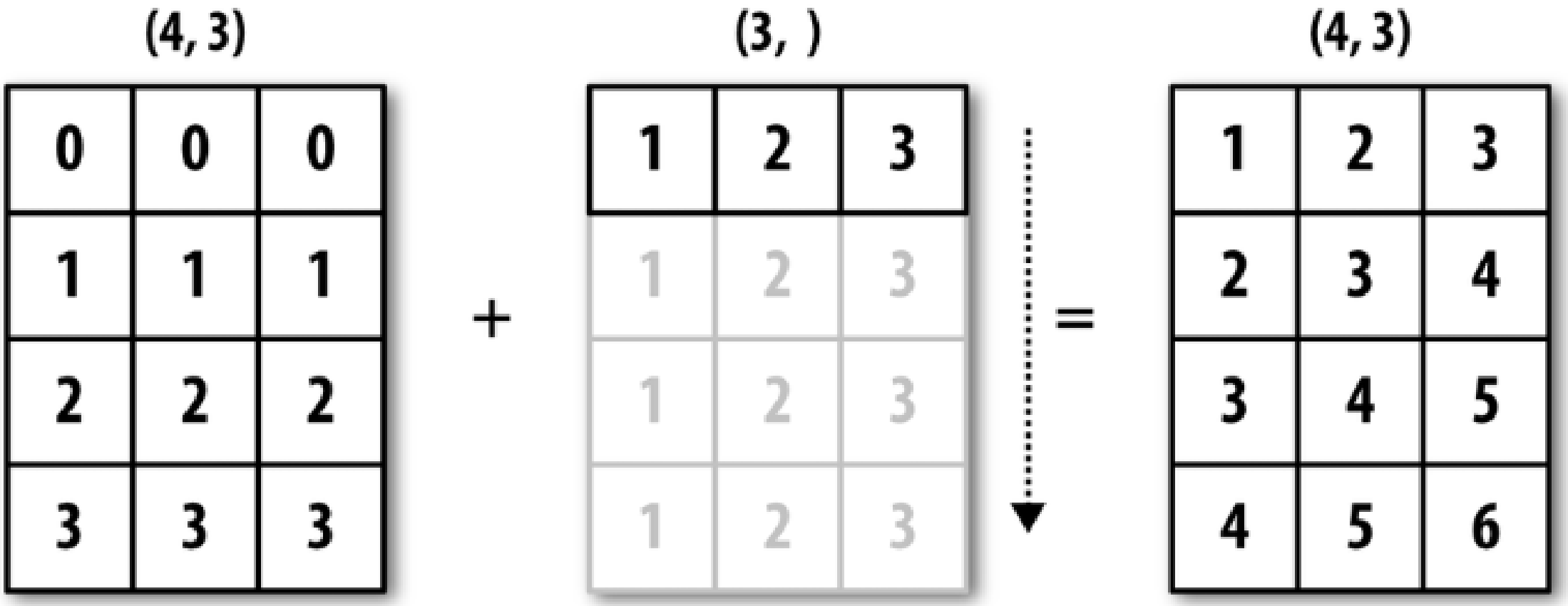
```
>>> a[(0,1,2,3,4), (1,2,3,4,5)]
array([1, 12, 23, 34, 45])

>>> a[3:, [0,2,5]]
array([[30, 32, 35],
       [40, 42, 45],
       [50, 52, 55]])

>>> mask = np.array([1,0,1,0,0,1], dtype=bool)
>>> a[mask, 2]
array([2, 22, 52])
```



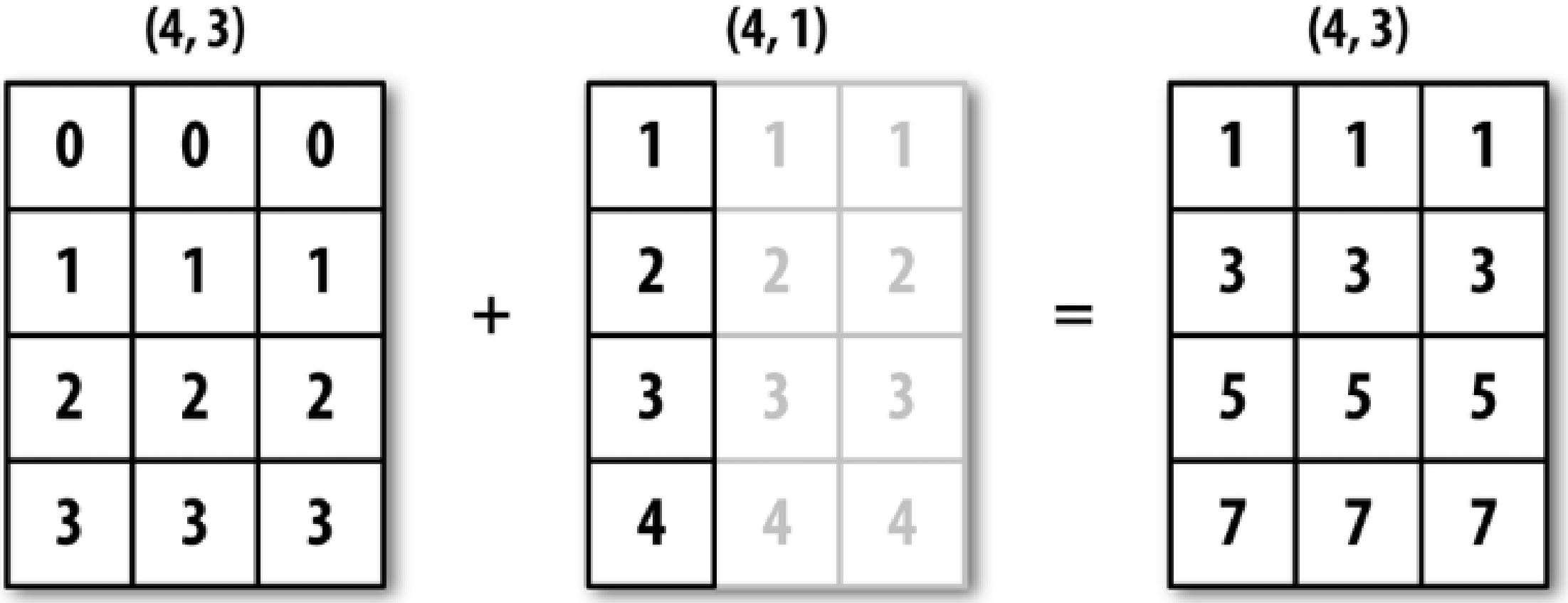
Broadcasting

	20.12		.2*20.12
.2 *	39.90	=	.2*39.90
	31.01		.2*31.01

Broadcasting



Broadcasting



Poll

True or False : We can subtract a shape (3,) array from a shape (4,1) array?

A. True
B. False

(4, 1)

0
3
6
9

-

(3,)

3	4	5

True or False : We can subtract a shape (3,) array from a shape (4,1) array?

A. True
B. False

(4, 1)

0	0	0
3	3	3
6	6	6
9	9	9

-

(3,)

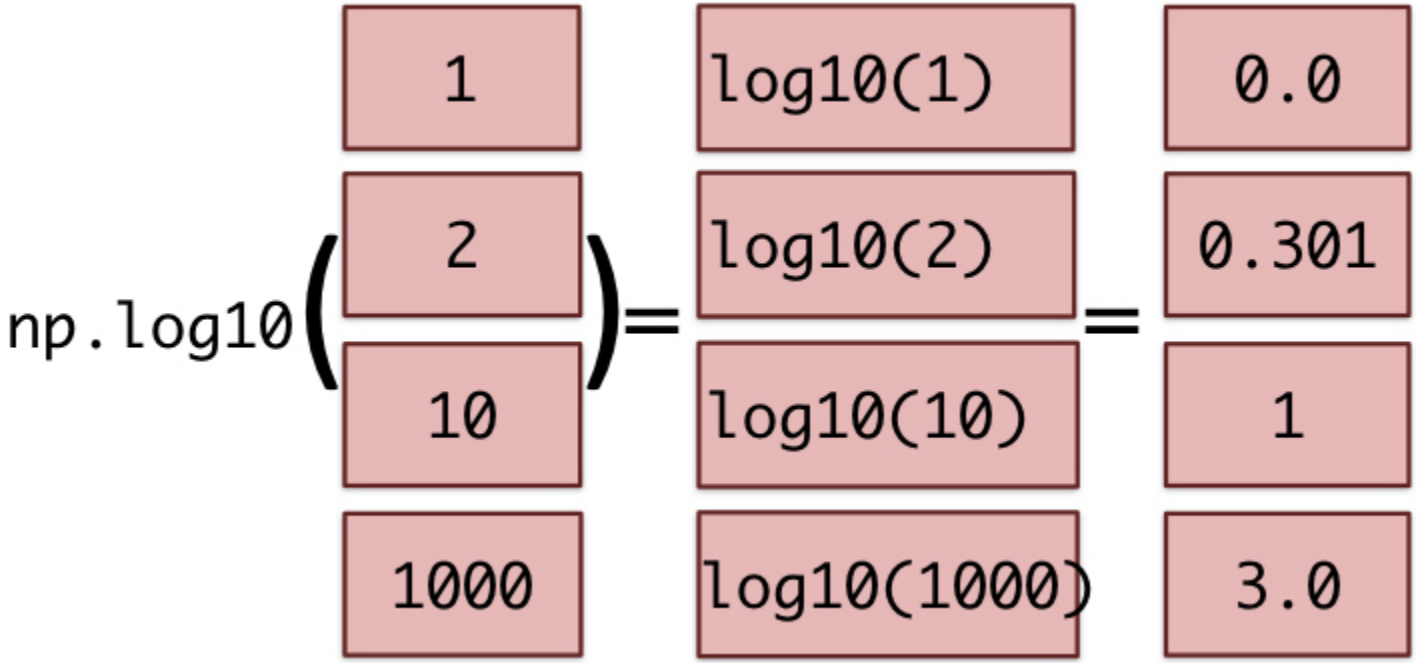
3	4	5
3	4	5
3	4	5
3	4	5

=

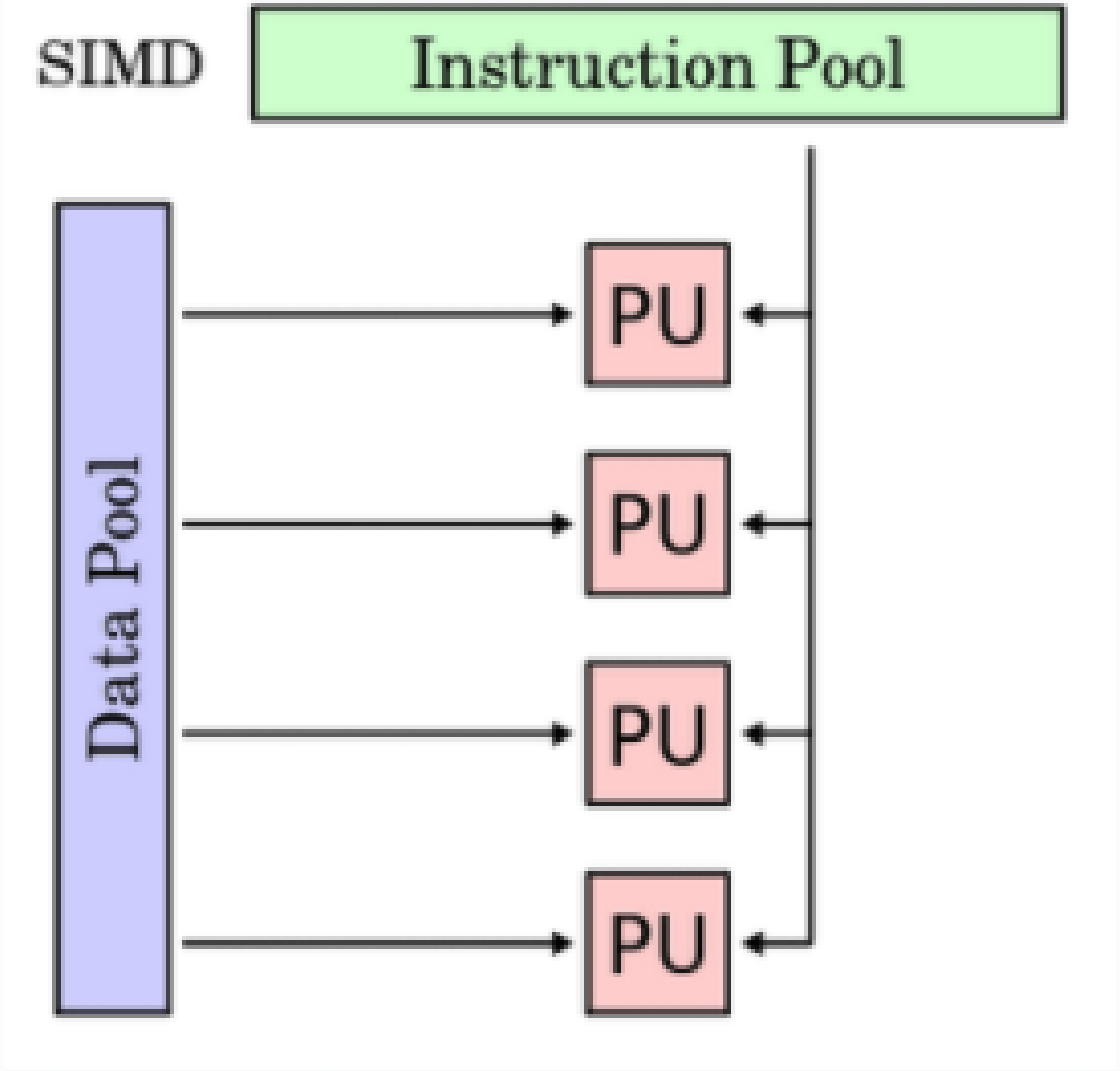
(4, 3)

-3	-4	-5
0	-1	-2
3	2	1
6	5	4

Vectorization



Vectorization



NYU TANDON SCHOOL OF ENGINEERING

Classes

- Package containers + functions

Arrays

- Accessing
- Manipulating
- Calculating

NYU TANDON SCHOOL OF ENGINEERING

References

- Lubanovic, *Introducing Python* (Ch. 14)
- McKinney, Python for Data Analysis (Ch. 4.1-4.4)

Questions

- Describe the learning objectives.
- Summarize the relevant take-aways.
- Ask about unclear information.