Python-Numpy

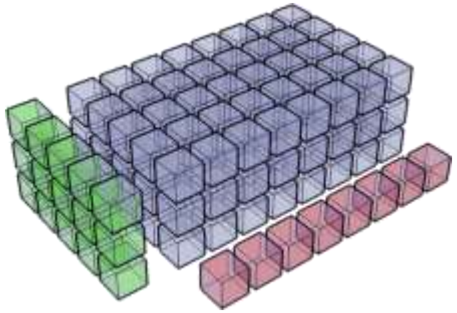
Dr. Sarwan Singh

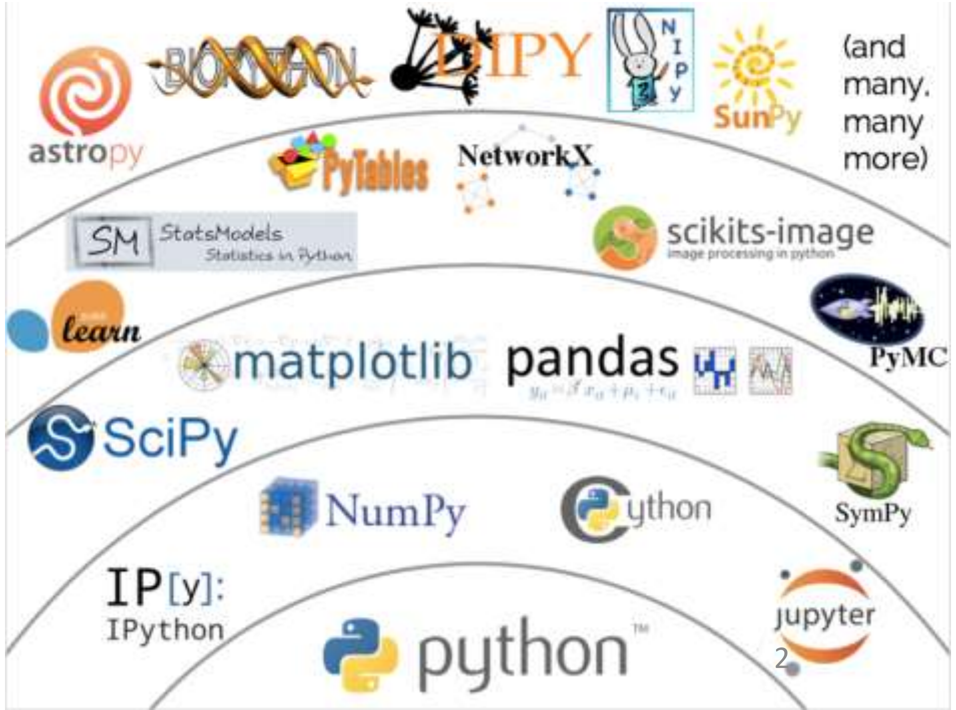
Agenda

- Reading from csv files
- Sorting

NumPy

Numerical Python

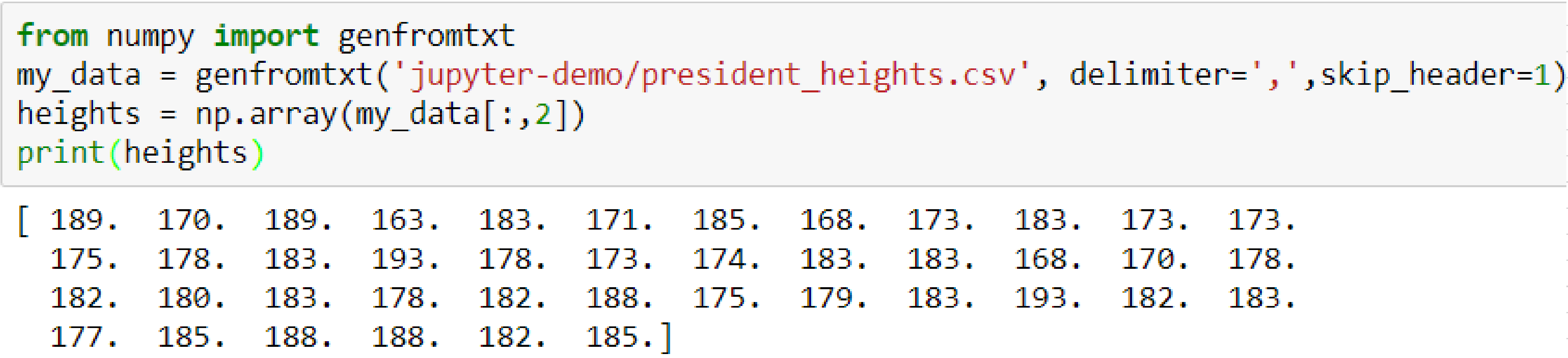




Processing data using Numpy

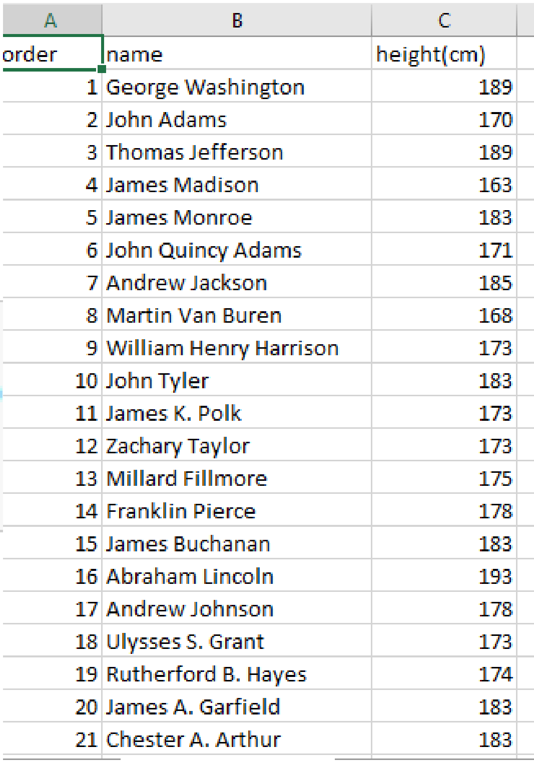
Reading data from csv file

- Using – genfromtxt , csv.reader

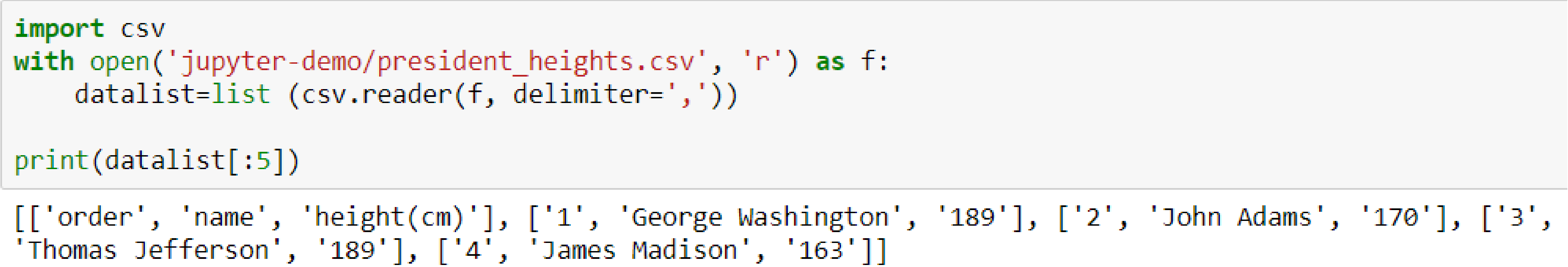


```
from numpy import genfromtxt
my_data = genfromtxt('jupyter-demo/president_heights.csv', delimiter=',',skip_header=1)
heights = np.array(my_data[:,2])
print(heights)
```

```
[ 189.  170.  189.  163.  183.  171.  185.  168.  173.  183.  173.  173.
  175.  178.  183.  193.  178.  173.  174.  183.  183.  168.  170.  178.
  182.  180.  183.  178.  182.  188.  175.  179.  183.  193.  182.  183.
  177.  185.  188.  188.  182.  185.]
```



A	B	C
order	name	height(cm)
1	George Washington	189
2	John Adams	170
3	Thomas Jefferson	189
4	James Madison	163
5	James Monroe	183
6	John Quincy Adams	171
7	Andrew Jackson	185
8	Martin Van Buren	168
9	William Henry Harrison	173
10	John Tyler	183
11	James K. Polk	173
12	Zachary Taylor	173
13	Millard Fillmore	175
14	Franklin Pierce	178
15	James Buchanan	183
16	Abraham Lincoln	193
17	Andrew Johnson	178
18	Ulysses S. Grant	173
19	Rutherford B. Hayes	174
20	James A. Garfield	183
21	Chester A. Arthur	183

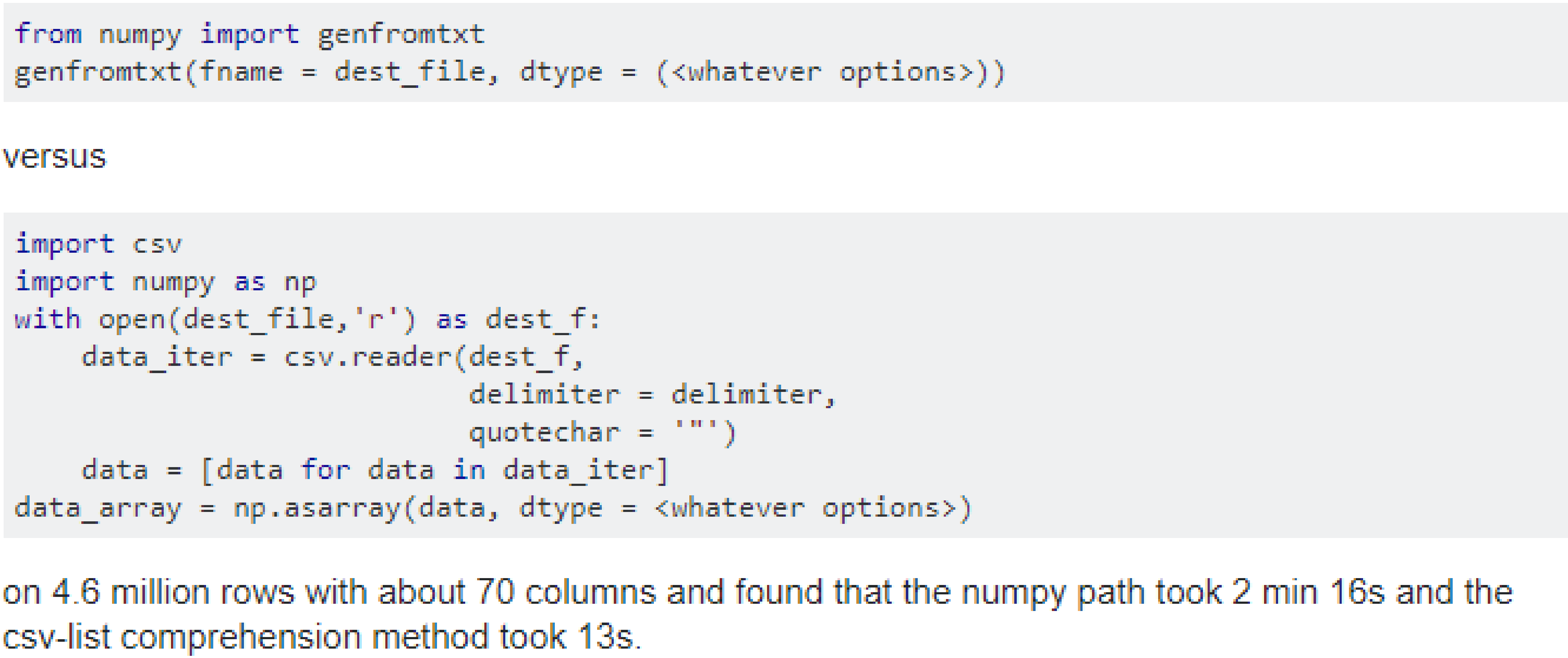


```
import csv
with open('jupyter-demo/president_heights.csv', 'r') as f:
    datalist=list (csv.reader(f, delimiter=','))

print(datalist[:5])
```

```
[['order', 'name', 'height(cm)'], ['1', 'George Washington', '189'], ['2', 'John Adams', '170'], ['3',
'Thomas Jefferson', '189'], ['4', 'James Madison', '163']]
```

Genfromtxt vs csv.reader



```
from numpy import genfromtxt
genfromtxt(fname = dest_file, dtype = (<whatever options>))
```

versus

```
import csv
import numpy as np
with open(dest_file,'r') as dest_f:
    data_iter = csv.reader(dest_f,
                           delimiter = delimiter,
                           quotechar = '"')
    data = [data for data in data_iter]
data_array = np.asarray(data, dtype = <whatever options>)
```

on 4.6 million rows with about 70 columns and found that the numpy path took 2 min 16s and the csv-list comprehension method took 13s.

exercise

Calculate following using data from presidents_heights.csv

- Mean height
- Standard deviation
- Minimum height
- Maximum height
- 25th percentile
- Median
- 75th percentile

Using seattle2014.csv file :

- extract rainfall inches
- Max rainfall

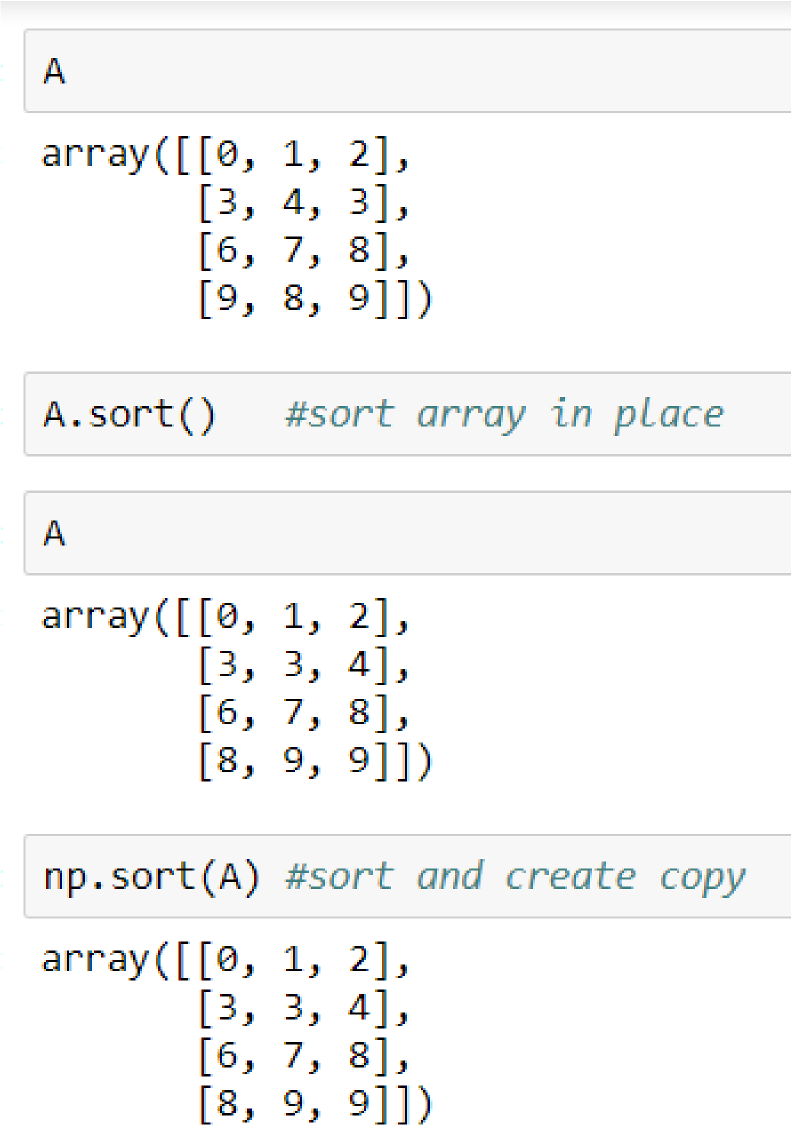
Sort, Search & Counting Functions

- Various sorting functions are available in NumPy having different sorting algorithms.
- Every algorithm is characterized by the speed of execution, worst case performance, the workspace required and the stability.

kind	speed	worst case	work space	stable
‘quicksort’	1	O(n^2)	0	no
‘mergesort’	2	O(n*log(n))	~n/2	yes
‘heapsort’	3	O(n*log(n))	0	no

Sorting

- numpy.sort (array , axis,   kind, order)
  - array- to be sorted
  - axis- axis of array to be sorted. If none, the array is flattened, sorting on the last axis
  - kind - Default is quicksort
  - order - If the array contains fields, the order of fields to be sorted



```
A
```

```
array([[0, 1, 2],
       [3, 4, 3],
       [6, 7, 8],
       [9, 8, 9]])
```

```
A.sort()   #sort array in place
```

```
A
```

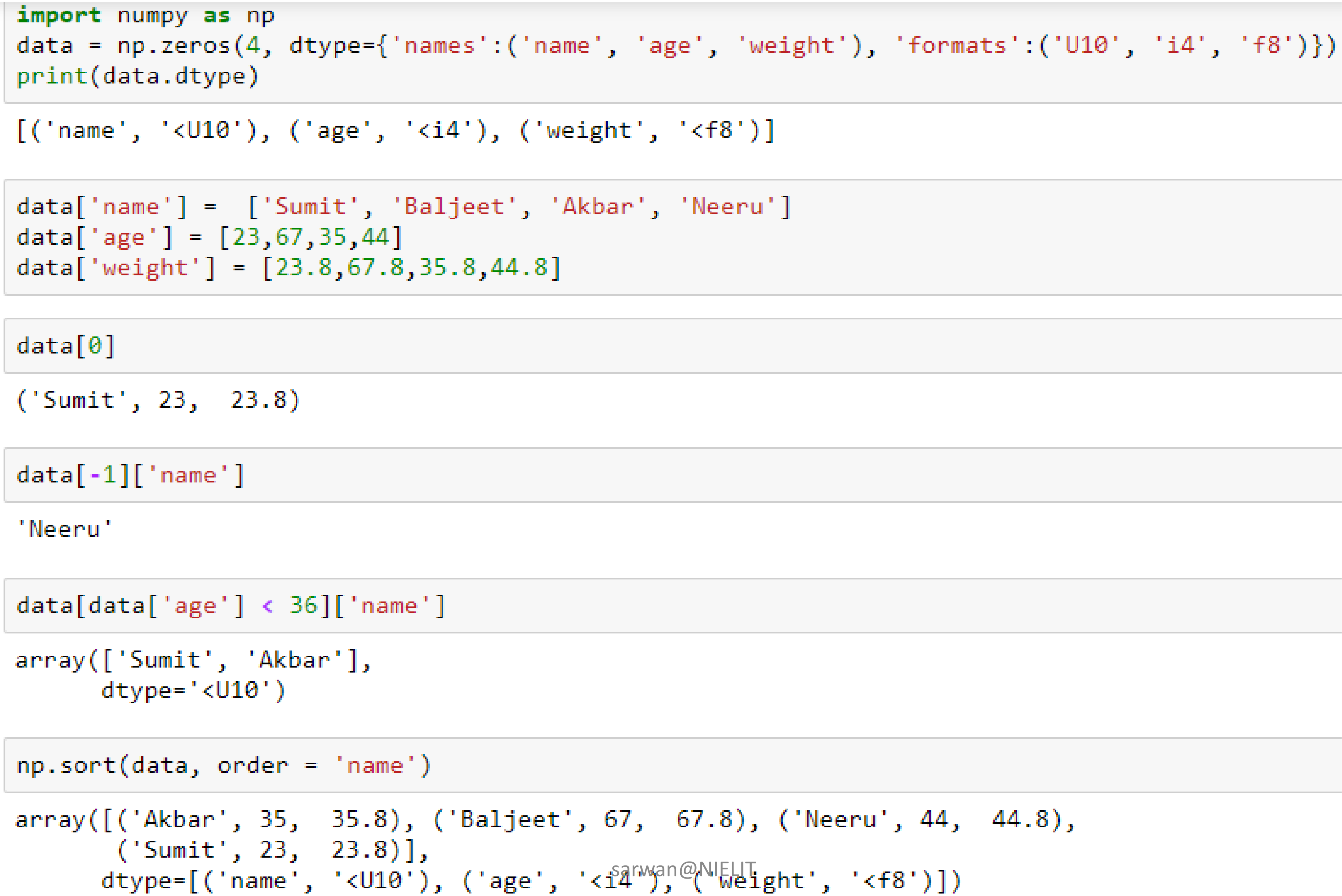
```
array([[0, 1, 2],
       [3, 3, 4],
       [6, 7, 8],
       [8, 9, 9]])
```

```
np.sort(A) #sort and create copy
```

```
array([[0, 1, 2],
       [3, 3, 4],
       [6, 7, 8],
       [8, 9, 9]])
```

Sorting

- np.sort(a, order = 'name')



```
import numpy as np
data = np.zeros(4, dtype={'names':('name', 'age', 'weight'), 'formats':('U10', 'i4', 'f8')})
print(data.dtype)
```

```
[('name', '<U10'), ('age', '<i4'), ('weight', '<f8')]
```

```
data['name'] =  ['Sumit', 'Baljeet', 'Akbar', 'Neeru']
data['age'] = [23,67,35,44]
data['weight'] = [23.8,67.8,35.8,44.8]
```

```
data[0]
```

```
('Sumit', 23,  23.8)
```

```
data[-1]['name']
```

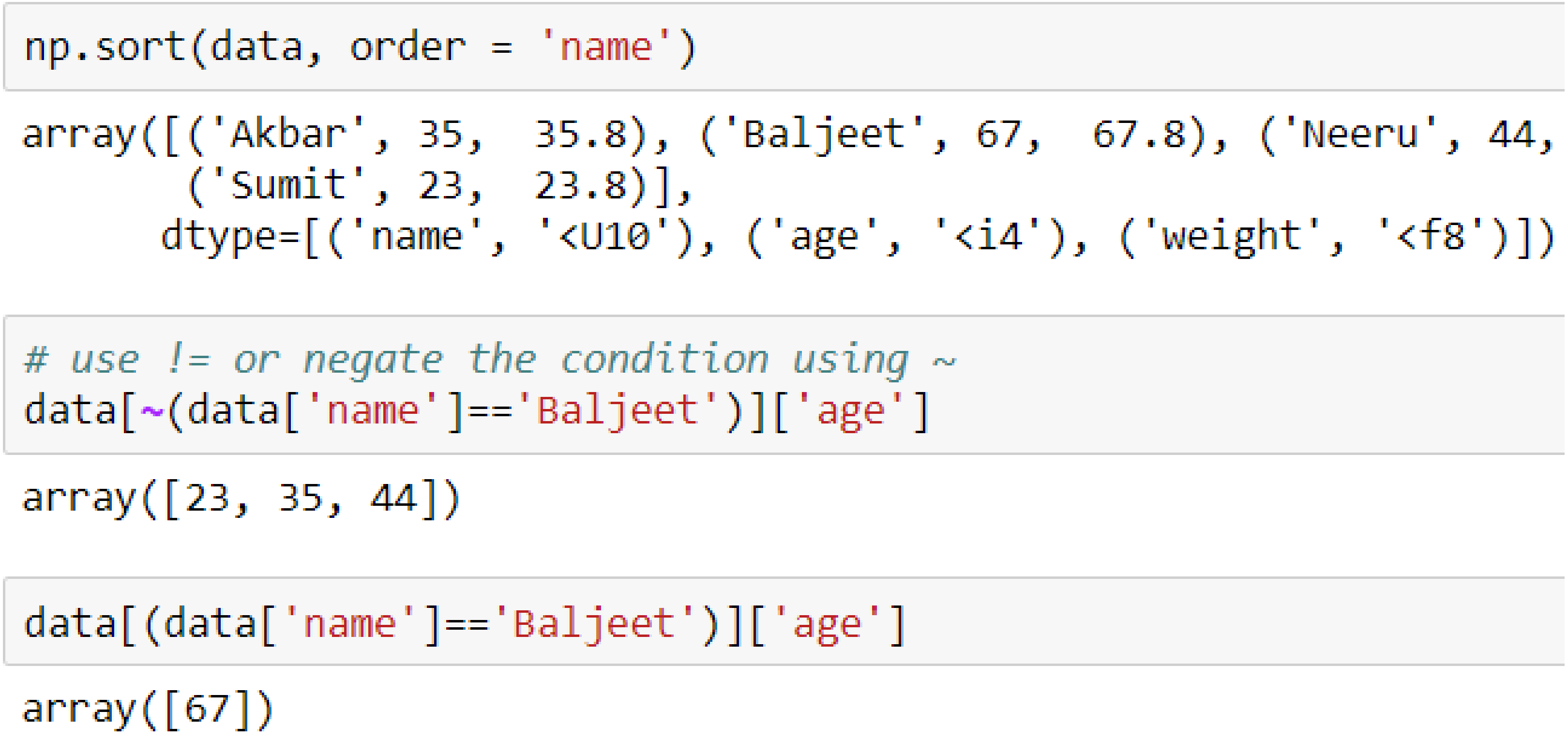
```
'Neeru'
```

```
data[data['age'] < 36]['name']
```

```
array(['Sumit', 'Akbar'],
      dtype='<U10')
```

```
np.sort(data, order = 'name')
```

```
array([('Akbar', 35,  35.8), ('Baljeet', 67,  67.8), ('Neeru', 44,  44.8),
       ('Sumit', 23,  23.8)],
      dtype=[('name', '<U10'), ('age', '<i4'), ('weight', '<f8')])
```



```python
np.sort(data, order = 'name')
```

```
array([('Akbar', 35,  35.8), ('Baljeet', 67,  67.8), ('Neeru', 44,
       ('Sumit', 23,  23.8)],
      dtype=[('name', '<U10'), ('age', '<i4'), ('weight', '<f8')])
```

```python
# use != or negate the condition using ~
data[~(data['name']=='Baljeet')]['age']
```

```
array([23, 35, 44])
```

```python
data[(data['name']=='Baljeet')]['age']
```

```
array([67])
```