

Session2-Numpy (Indexing, Operations)

DAwithPython S-2

Training Clarusway

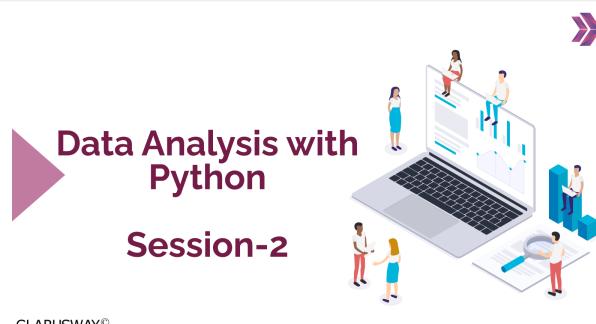
Pear Deck - January 20, 2022 at 7:42PM

Part 1 - Summary

Use this space to summarize your thoughts on the lesson

Part 2 - Responses

Slide 1



Use this space to take notes:

Slide 2



Use this space to take notes:

Slide 3

▶ Table of Contents

- ▶ Indexing of 1D Arrays
- ▶ Indexing of 2D Arrays
- ▶ Broadcasting
- ▶ Selection on a Condition
- ▶ NumPy Operations



Use this space to take notes:

Slide 4

Your Response

You Chose

I've completed the pre-class content?

True

False

 Pear Deck

 Students choose an option

Pear Deck Interactive Slide
Do not remove this bar

• True

Other Choices

- False

Use this space to take notes:

Slide 5

▶ Indexing of 1D Arrays

```
1 import numpy as np
2 a = np.arange(20,30)
array([20, 21, 22, 23, 24, 25, 26, 27, 28, 29])
1 a[0:3]
1 a[:3]
array([20, 21, 22])
1 a[3:]
array([23, 24, 25, 26, 27, 28, 29])
1 a[3::2]
array([21, 23, 25, 27, 29])
```

It's similar to normal indexing of a Python list.



5

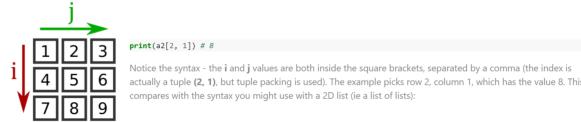
Use this space to take notes:

Slide 6

▶ Indexing of 2D Arrays



```
import numpy as np  
  
a2 = np.array([[1, 2, 3],  
              [4, 5, 6],  
              [7, 8, 9]])
```



We can index an element of the array using two indices - *i* selects the row, and *j* selects the column:

8

Use this space to take notes:

Slide 7

▶ Indexing of 2D Arrays

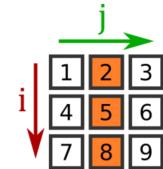


```
print(a2[:, 1]) # [2, 5, 8]
```

We are skipping ahead slightly to slicing, later in this tutorial, but what this syntax means is:

- for the *i* value, take all values (*i* is a full slice, from start to end)
- for the *j* value take 1

Giving this array [2, 5, 8]:



9

Use this space to take notes:

Slide 8

Your Response

Answer 1:
[1:4,2:4]

▶ Slicing of 2D Arrays



10	11	12	13	14
15	16	17	18	19
20	21	22	23	24
25	26	27	28	29

What is the code?



Students, write your response!

Peer Deck Interactive Slide
Do not remove this bar

Use this space to take notes:

Slide 9

▶ Slicing of 2D Arrays



10	11	12	13	14
15	16	17	18	19
20	21	22	23	24
25	26	27	28	29

The Answer

`arr[1:, 2:4]`

9

Use this space to take notes:

Slide 10

► Broadcasting

$$\begin{array}{c} \text{(3,3)} \\ \boxed{\begin{array}{ccc} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{array}} \end{array} \star \begin{array}{c} \text{(3,2) or (1,3)} \\ \boxed{\begin{array}{cc} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{array}} \end{array} = \begin{array}{c} \text{(3,3)} \\ \boxed{\begin{array}{ccc} -1 & 0 & 3 \\ -4 & 0 & 6 \\ -7 & 0 & 9 \end{array}} \end{array} \text{ multiplying several columns at once}$$

Rule

$$\begin{array}{c} \text{(3,3)} \\ \boxed{\begin{array}{ccc} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{array}} \end{array} / \begin{array}{c} \text{(3,1)} \\ \boxed{\begin{array}{c} 3 \\ 6 \\ 9 \end{array}} \end{array} = \begin{array}{c} \text{(3,3)} \\ \boxed{\begin{array}{ccc} .3 & .7 & 1 \\ .6 & .8 & 1 \\ .8 & .9 & 1 \end{array}} \end{array} \text{ row-wise normalization}$$

$$\begin{array}{c} \text{(3,2) or (1,3)} \\ \boxed{\begin{array}{ccc} 1 & 2 & 3 \\ 1 & 2 & 3 \\ 1 & 2 & 3 \end{array}} \end{array} \star \begin{array}{c} \text{(3,1)} \\ \boxed{\begin{array}{c} 1 \\ 2 \\ 3 \end{array}} \end{array} = \begin{array}{c} \text{(3,3)} \\ \boxed{\begin{array}{ccc} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 3 & 6 & 9 \end{array}} \end{array} \text{ outer product}$$

In order to broadcast, the size of the trailing axes for both arrays in an operation must either be the same size or one of them must be one.

10

Use this space to take notes:

Slide 11

► Broadcasting

```
import numpy as np
a = np.arange(10,20)
array([10, 11, 12, 13, 14, 15, 16, 17, 18, 19])
a[1] = 5
array([ 5,  5,  5, 13, 14, 15, 16, 17, 18, 19])
l = list(range(10))
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
l[1] = 100
-----
TypeError: 'int' object cannot be interpreted as an iterator
-----> 1[1] = 100
Traceback (most recent call last):
-----> 1[1] = 100
TypeError: can only assign an iterable
-----> 1[1] = [100, 100, 100]
[100, 100, 100, 3, 4, 5, 6, 7, 8, 9]
```

11

Use this space to take notes:

Slide 12

► Broadcasting

```
a = np.arange(20,30)
a
array([20, 21, 22, 23, 24, 25, 26, 27, 28, 29])

a_slice = a[:5]
a_slice
array([20, 21, 22, 23, 24])

a_slice[:3] = 5
a_slice
array([ 5,  5,  5, 23, 24])

a
array([ 5,  5,  5, 23, 24, 25, 26, 27, 28, 29])
```

```
a = np.arange(20,30)
a
array([20, 21, 22, 23, 24, 25, 26, 27, 28, 29])

a_copy = a.copy()
a_slice = a_copy[:5]
a_slice
array([20, 21, 22, 23, 24])

a_slice[:3] = 5
a_slice
array([ 5,  5,  5, 23, 24])

a
array([20, 21, 22, 23, 24, 25, 26, 27, 28, 29])
```

12

Use this space to take notes:

Slide 13

► Selection on a Condition

```
a = np.random.randint(1, 100, 15)
a
array([66, 93, 22, 91, 24, 66, 72, 67, 54, 85, 90, 50, 28, 67, 34])

a < 50
array([False, False, True, False, True, False, False, False,
       False, False, False, True, False, True])

a[a < 50]
array([22, 24, 28, 34])

a[(a < 25) | (a > 75)]
array([93, 22, 91, 24, 85, 90])

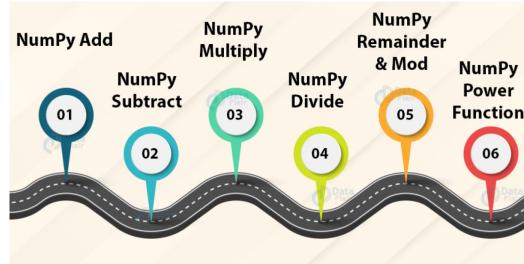
a[(a > 25) & (a < 75)]
array([66, 66, 72, 67, 54, 50, 28, 67, 34])
```

13

Use this space to take notes:

Slide 14

▶ NumPy Operations



14

Use this space to take notes:

Slide 15

▶ NumPy Operations

Arithmetic Operations & Universal Array Functions

<pre>a = np.array([10,20,30,40]) b = np.array([5,5,5,5])</pre>	<pre>a2 = np.array([4,9,16,25]) b2 = np.array([2,2,2,2])</pre>
<pre>Addition np.add(a,b) = a+b =[15,25,35,45]</pre>	<pre>Square Root np.sqrt(a2) =[2,3,4,5]</pre>
<pre>Subtraction np.subtract(a,b) = a-b =[5,15,25,35]</pre>	<pre>Power np.power(a2,b2) =[16,81,256,625]</pre>
<pre>Multiplication np.multiply(a,b) = a*b =[50,100,150,200]</pre>	<pre>Mod/Remainder np.mod(a2,b2) = np.remainder(a2+b2) =[0,1,0,1]</pre>
<pre>Division np.divide(a,b) = a/b =[2,4,6,8]</pre>	<pre>Exponential np.exp([1, 3, 5]) =[2.71828183 20.08553692 148.4131591]</pre>

15

Use this space to take notes:

Slide 16

▶ NumPy Operations



► Statistical Calculations

- `np.mean(arr, axis=0)` => Returns mean along specific axis
- `arr.sum()` => Returns sum of arr
- `arr.min()` => Returns minimum value of arr
- `arr.max(axis=0)` => Returns maximum value of specific axis
- `np.var(arr)` => Returns the variance of array
- `np.std(arr, axis=1)` => Returns the standard deviation of specific axis
- `np.corrcoef(arr)` => Returns correlation coefficient of array

18

Use this space to take notes:

Slide 17

▶ Data Analysis with Python



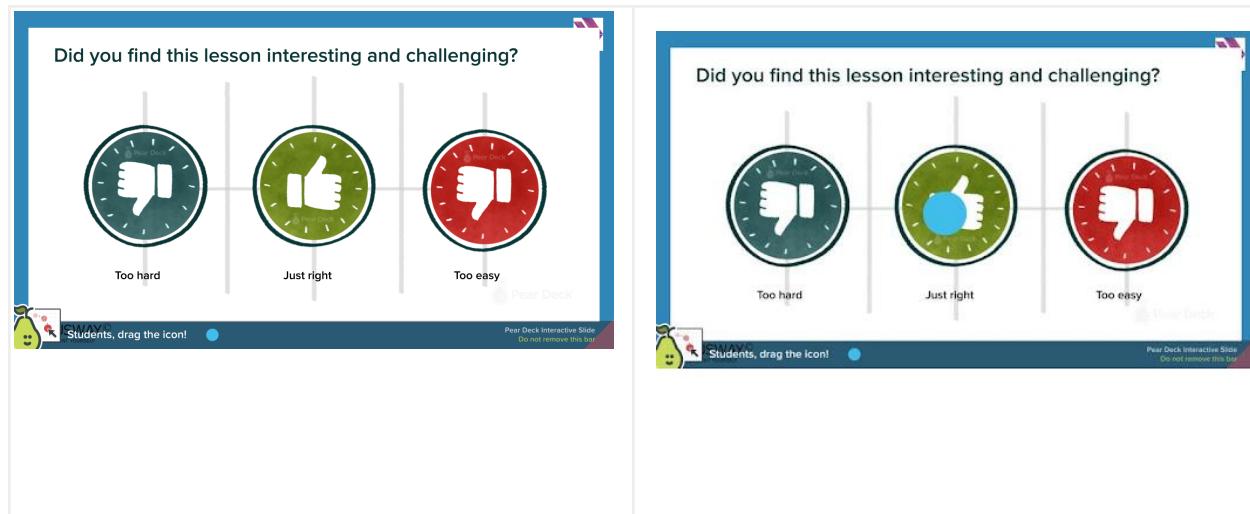
let's start the
hands-on phase

17

Use this space to take notes:

Slide 18

Your Response



Use this space to take notes:

Slide 19

THANKS!

Any questions?

You can find us at:

steve_w@clarusway.com
michael_g@clarusway.com

CLARUSWAY®
way to success



19

Use this space to take notes: