# 8. Numpy Array Indexing, Reshaping, Slicing, Join, Split

PYTHON COURSE

SIN YONG TENG

1

## Do you understand ndarray index/slicing?

Index the highlighted numbers/arrays

Basic:

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | **8** | 9 |

**8**

| | | |
|---|---|---|
| 1 | **2** | **3** |
| 4 | **5** | **6** |
| 7 | **8** | **9** |

**[[2,3], [5,6], [8,9]]**

Advanced:

**Value<4**

| | | | | | | |
|---|---|---|---|---|---|---|
| **1** | 7 | **2** | 5 | **3** | 5 | 6 |

**[1,2,3]**

| | | | |
|---|---|---|---|
| **1** | **2** | **3** | **Sum<7** |
| 4 | 5 | 6 | **[1,2,3]** |
| 7 | 8 | 9 | |

2

# Ndarray Slicing in more dimension

| | |
|---|---|
| [1, 2] | [3, 4] |
| [5, 6] | [7, 8] |

```
[[2 4]
 [6 8]]
```

```
[[[1,2],[3,4]],[[5,6],[7,8]]]
```

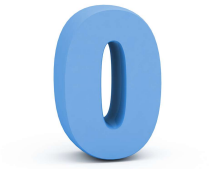
3

# Challenge 1: Show the zeros

You are given a list **[-1,0,-1,0,-1,0,-1,1]**

Convert it to an ndarray with:

- -1 being represented by “Negative One”
- 0 being represented by “Zero”
- 1 being represented by “One”



In other words

```
['Negative One' 'Zero' 'Negative One' 'Zero' 'Negative One' 'Zero'
 'Negative One' 'One']
```

Then remove all elements which are **NOT** “Zero”.

4

# Do you understand ndarray reshaping?

Basic:

| | |
|---|---|
| 1 | 2 |
| 3 | 4 |
| 5 | 6 |

→

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |

Advanced:

```
[[[1 2]
  [3 4]]

 [[5 6]
  [7 8]]]
```

→

```
[[[1 2]
  [3 4]
  [5 6]
  [7 8]]]
```

```
[[[1,2],[3,4]],[[5,6],[7,8]]]
```

5

# Ndarrays Joining

| Command | Axis? | Remarks |
|---|---|---|
| np.concatenate() | Yes | Best used |
| np.stack() | Yes | General stack |
| np.hstack() | No | Horizontal stack |
| np.vstack() | No | Vertical stack |
| np.dstack() | No | Depth stack |

6

# Ndarrays Splitting

| Command | Axis? | Remarks |
|---|---|---|
| np.array_split() | Yes | Only equal splitting |
| np.split() | Yes | Any split (best) |
| np.hsplit() | No | Horizontal splitting |
| np.vsplit | No | Vertical splitting |
| np.dplit | No | Depth Splitting |

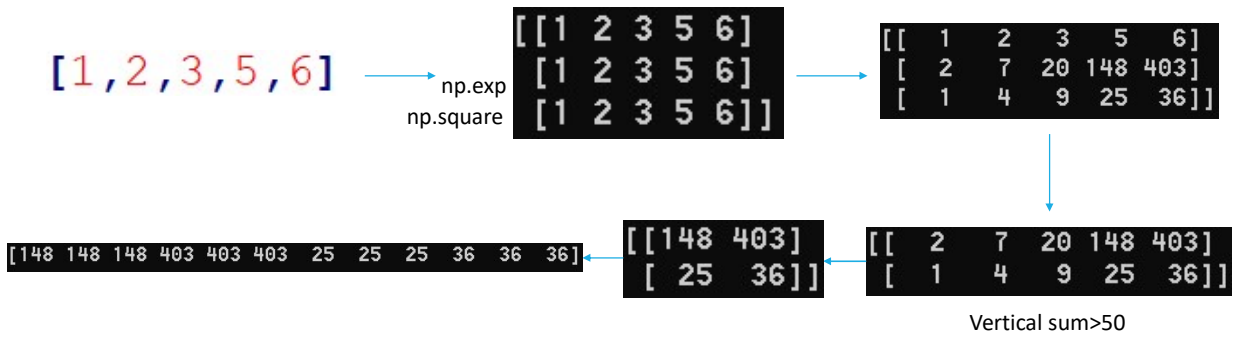
7

# Replicating ndarray

| Command | Axis? | Remarks |
|---|---|---|
| np.repeat() | Yes | Replicate by element sequence |
| np.tile() | No | Replicate by tile (inner array) sequence |

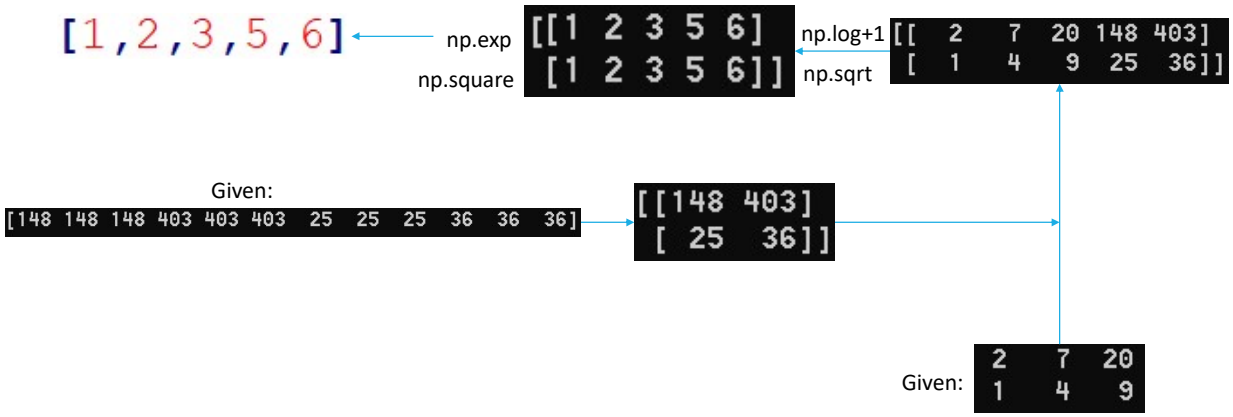
8

# Challenge 2: Numpy Transformation




9

# HOMEWORK: Reversed Challenge 2




10

# Conclusion

1. Index and Slicing
2. Slicing in more dimensions
3. Conditional transformation/indexing
4. Reshaping
5. Joining ndarray
6. Splitting ndarray
7. Replicating ndarray
8. Transformation of array
9. Reversed transformation of array

11