

# Visionary Course – Energy AI

## Week 06

Apr. 8, 2022  
Seokju Lee

# Week 06a – Image Processing Puzzle



# Python Functions

## How to define functions?

→ Plot image (show\_image) & close the plot (close\_image)

```
def show_image(i, img):  
    plt.figure(i)  
    plt.imshow(img)  
    plt.xticks([]); plt.yticks([])  
    plt.ion(); plt.show()  
  
def close_image(i):  
    if i == 0:  
        plt.close('all')  
    else:  
        plt.close(i)
```

Please try to implement these functions (in basic\_test.py) for summation, subtraction, and multiplication.

```
def do_sum(a, b):  
    output = a + b  
    return output  
  
def do_subtract(a, b):  
    output = a - b  
    return output  
  
def do_multiply(a, b):  
    output = a * b  
    return output
```

# Basic Python Library: NumPy

## Operations

<https://replit.com/team/VC-Spring-2022>  
(team-based project)

→ There are many strong operation functions for “**multi-dimensional arrays**”

```
import numpy as np

a = np.array([1,2,3])
print(a)          # [2 3 4]
print(a.dtype)    # int64
b = np.array([1.2, 3.5, 5.1])
print(b.dtype)    # float64
print(a**2)
print(a.sum())
print(a.mean())
print(a.min())
print(a.max())
```

Please try basic math operations.

```
a = np.arange(8)
print(a)          # [0 1 2 3 4 5 6 7]
b = a.reshape(2, 4)
print(b)
c = a.T
print(b)
```

```
print(a.shape)
print(b.shape)
print(c.shape)
print(b.sum(axis=0))
print(b.sum(axis=1))
```

→ **Matrix operation, linear algebra**

Please use shell commands in Replit,  
>> python basic\_test.py

Please discuss each print line.

# Basic Python Library: NumPy

## Array slicing

→ Along the first axis

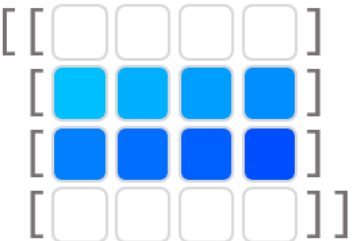
```
import numpy as np


arr = np.array([[1, 2, 3, 4],
                [5, 6, 7, 8],
                [9, 10, 11, 12],
                [13, 14, 15, 16]])


print(arr)
print(arr[1:3])
print(arr[:-1])
print(arr[:,2])
```

[1] codetorial.net

<https://replit.com/team/VC-Spring-2022>  
(team-based project)

`arr[1:3]` = 

`arr[:-1]` = 

`arr[:,2]` = 

# Basic Python Library: NumPy

## Array slicing

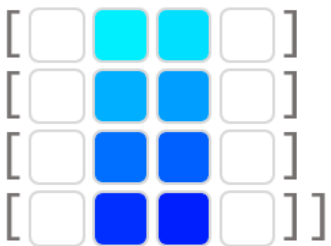
→ Along the second axis


<https://replit.com/team/VC-Spring-2022>  
(team-based project)

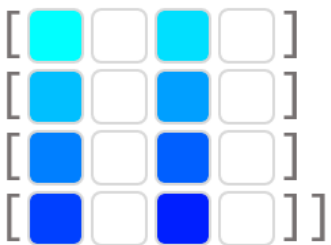
```
import numpy as np

arr = np.array([[1, 2, 3, 4],
                [5, 6, 7, 8],
                [9, 10, 11, 12],
                [13, 14, 15, 16]])

print(arr)
print(arr[:, 1:3])
print(arr[:, :-1])
print(arr[:, ::2])
```

`arr[:, 1:3]` = 

`arr[:, :-1]` = 

`arr[:, ::2]` = 

# Basic Python Library: NumPy

## Array slicing

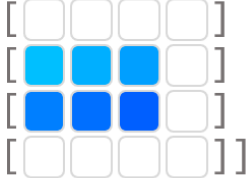
→ Along the first and second axis

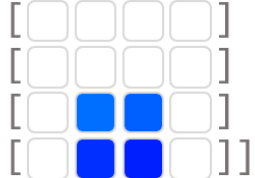
<https://replit.com/team/VC-Spring-2022>  
(team-based project)


```
import numpy as np

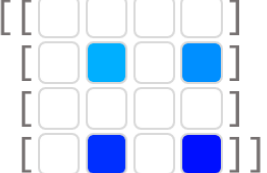
arr = np.array([[1, 2, 3, 4],
                [5, 6, 7, 8],
                [9, 10, 11, 12],
                [13, 14, 15, 16]])

print(arr)
print(arr[1:3, :-1])
print(arr[2:, 1:3])
print(arr[::2, ::2])
print(arr[1::2, 1::2])
```

`arr[1:3, :-1]` = 

`arr[2:, 1:3]` = 

`arr[::2, ::2]` = 

`arr[1::2, 1::2]` = 

# Basic Python Library: NumPy

## Changing specific values

→ Can be applied with a condition

<https://replit.com/team/VC-Spring-2022>  
(team-based project)

```
import numpy as np

a = np.array([[1, 2], [3, 1]])
b = np.where(a == 1, 10, a)

print(a)
print(b)
```

```
import numpy as np

a = np.array([[0.5, 1.2, 0.9], [1.1, 0.8, 1.4]])
b = np.where(a < 1.0, 0.0, a)

print(a)
print(b)
```



# Basic Python Library: NumPy

## Concatenate array

→ Along different axis

<https://replit.com/team/VC-Spring-2022>  
(team-based project)

```
import numpy as np

a = np.array([1, 2, 3])
b = np.array([4, 5, 6])
c = np.array([7, 8, 9])

ab = np.concatenate((a, b))
abc = np.concatenate((a, b, c))

print(ab)
print(abc)
```

```
import numpy as np

a = np.array([[1, 2], [3, 4]])
b = np.array([[5, 6], [7, 8]])

ab_0 = np.concatenate((a, b), axis=0)    # Default
ab_1 = np.concatenate((a, b), axis=1)

print(ab_0)
print(ab_1)
```

# Basic Python Library: NumPy

## Stack array

→ Along different axis

<https://replit.com/team/VC-Spring-2022>  
(team-based project)

```
import numpy as np

a = np.array([[1, 2], [3, 4]])
b = np.array([[5, 6], [7, 8]])
c = np.hstack([a, b])
d = np.vstack([a, b])

print(a)
print(b)
print(c)
print(d)
```

```
import numpy as np

a = np.array([[1, 2], [3, 4]])
b = np.array([[5, 6], [7, 8]])
c = np.stack([a, b], axis=0)
d = np.stack([a, b], axis=1)

print(a)
print(b)
print(c)
print(d)
```

Difference between  
concatenate and stack?

# Basic Python Library: NumPy

## Broadcasting

→ Flexible operations

<https://replit.com/team/VC-Spring-2022>  
(team-based project)

```
import numpy as np

array1 = np.array([1, 2, 3, 4]).reshape(2, 2)
array2 = np.array([1.5, 2.5])

add = array1 + array2

print(add)
```

What are the rules to enable broadcasting?

# Basic Python Library: NumPy

## Append array

→ Along different axis

<https://replit.com/team/VC-Spring-2022>  
(team-based project)

```
import numpy as np

arr = np.array([[ [1, 1], [2, 2] ],
                [ [3, 3], [4, 4] ]])
item = np.array([ [5, 5], [6, 6] ])

print(arr.shape)
print(item.shape)

append = np.append(arr, item.reshape(1, 2, 2), axis=0)

print(append)
```