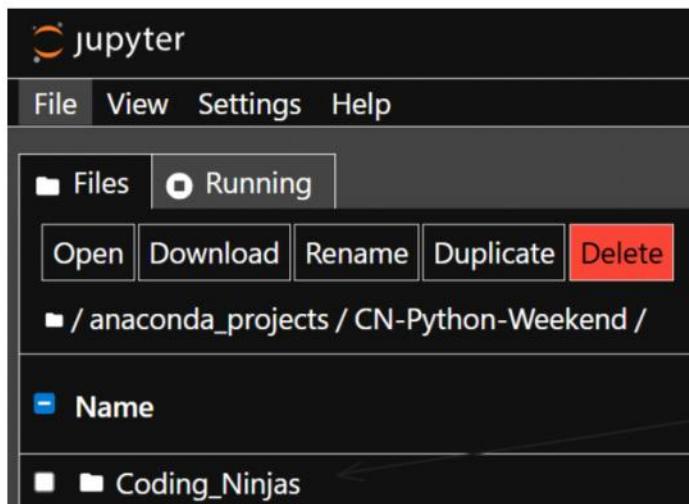


File Handling & NumPy - I

Session Objectives:

- ✓ Understand and use built-in modules
- ✓ Read content from files using `read()`, `readline()`, and `readlines()`
- ✓ Understand the importance of closing files
- ✓ Write data to files and create new files
- ✓ Delete files
- ✓ Understand what NumPy is and why it's important
- ✓ Understand what an array is and how it differs from lists/tuples
- ✓ Create NumPy arrays using various methods
- ✓ Explore the attributes of NumPy arrays
- ✓ Apply indexing and slicing on arrays



`mkdir` -> making a Directory [Folders]
`os.mkdir('Coding_Ninjas')`

File Handling

```
# Working with files and directories
# OS Module - Operating System Interface

import os
dir(os)

['DirEntry',
 'EX_OK',
 'F_OK',
 'GenericAlias',
 'Mapping',
 'MutableMapping',
 'O_APPEND',
 'O_BINARY',
 'O_CREAT',
 'O_EXCL',
 'O_NOINHERIT',
 'O_RANDOM',
 'O_RDONLY',
 'O_RDWR',
 'O_SEQUENTIAL',
 'O_SHORT_LIVED',
 'O_TEMPORARY',

anaconda_projects/CN-Python-Weekend/Modules&FileHandling.ipynb
getcwd() -> get current working directories.
```

```
os.getcwd()
'C:\\\\Users\\\\krish\\\\anaconda_projects\\\\CN-Python-Weekend'

os.getcwdb()
b'C:\\\\Users\\\\krish\\\\anaconda_projects\\\\CN-Python-Weekend'

os.listdir()
['.ipynb_checkpoints',
 'math_utils.py',
 'Modules&FileHandling.ipynb',
 'PythonSession.ipynb',
 'python_modules.py',
 'string_utils.py',
 '__pycache__']

# mkdir -> making a Directory [Folders]
os.mkdir('Coding_Ninjas')

os.rmdir('Coding_Ninjas')

# mkdir -> making a Directory [Folders]
os.mkdir('Python_Folder')
```

```
# mkdir -> making a Directory [Folders]
os.mkdir('Python_Folder')

# rmdir -> Removing a Directory [Folder]
os.rmdir('Python_Folder')

os.listdir()

['.ipynb_checkpoints',
 'math_utils.py',
 'Modules&FileHandling.ipynb',
 'PythonSession.ipynb',
 'Python_Folder',
 'python_modules.py',
 'string_utils.py',
 '__pycache__']

os.chdir("C:\\\\Users\\\\krish\\\\anaconda_projects\\\\CN-Python-Weekend\\\\Python_Folder")

os.listdir()

[]

open('coding.txt', 'x') # Creating a new file

<_io.TextIOWrapper name='coding.txt' mode='x' encoding='cp1252'>
```

```
os.listdir()

['coding.txt']

os.chdir("C:\\\\Users\\\\krish\\\\anaconda_projects\\\\CN-Python-Weekend")

os.listdir()

['.ipynb_checkpoints',
 'math_utils.py',
 'Modules&FileHandling.ipynb',
 'PythonSession.ipynb',
 'Python_Folder',
 'python_modules.py',
 'string_utils.py',
 '__pycache__']

# rmdir -> Removing a Directory [Folder]
os.rmdir('Python_Folder') # OSError: [WinError 145] The directory is not empty: 'Python_Folder'
```

SYNTAX- File Handling:

```
open('<filename>', mode)

1. 'x' : 'create' a new file (error: if the file already exist)
2. 'w' : 'write' (overwrite if the file exist)
3. 'a' : 'append' (adding new content to the existing one)
4. 'r' : 'read' (default)

# File Handling
file = open('python.txt' , 'x') # Create a new file in the current working directory

file.close()

# .replace()
os.replace('python.txt', 'python-program.txt')

os.remove('python-program.txt') # Remove the file if exist.

file = open('python.txt', 'x') # Create a file for first time.

file = open('python.txt', 'r') # Read Mode
content = file.read()
content

::

file.close() # This will close the file from read mode.

file = open('python.txt', 'w') # Overwrite the content
content = file.write("We are learning File Handling in Python.") # 40 Length
print(content) # Length of a text

40

file.close() # This will close the current working file

file = open('python.txt', 'r') # Read Mode
content = file.read() # Single Line Read
content

'We are learning File Handling in Python.'

file.close()

file = open('python.txt', 'a') # Appending the new content within existing file
file.write("\nHello World.")
file.write("\nPython Programming.")
file.write("\nCoding Ninjas.....")

19
```

```
file = open('python.txt', 'r') # Read Mode
content = file.read() # Single Line Read
content

'We are learning File Handling in Python.\nHello World.\nPython Programming.\nCoding Ninjas.....'

print(content)

We are learning File Handling in Python.
Hello World.
Python Programming.
Coding Ninjas.....
```

```
file = open('python.txt', 'r') # Read Mode
line1 = file.readline()
line2 = file.readline()
line3 = file.readline()
line4 = file.readline()
print("Line 1: " , line1)
print("Line 2: " , line2)
print("Line 3: " , line3)
print("Line 4: " , line4)

Line 1: We are learning File Handling in Python.

Line 2: Hello World.

Line 3: Python Programming.

Line 4: Coding Ninjas.....
```

```
file = open('python.txt', 'r') # Read Mode
lines = file.readlines()
print(lines)

['We are learning File Handling in Python.\n', 'Hello World.\n', 'Python Programming.\n', 'Coding Ninja
s.....']

print(lines[0])
We are learning File Handling in Python.

print(lines[1])
Hello World.

for line in lines:
    print(line)

We are learning File Handling in Python.

Hello World.

Python Programming.

Coding Ninjas.....
```

```
file.close()
```

```

file = open('coding.txt', 'r') # Read Mode
content = file.read() # Single Line Read
print(content)

Hi EveryOne! Welcome to the Python Course

file = open('coding.txt' , 'w')
content = file.write("Bye Bye! Tata Tata Khatam")
print(content) # Length of a text

25

file = open('coding.txt', 'r') # Read Mode
content = file.read() # Single Line Read
print(content)

Bye Bye! Tata Tata Khatam

# try : except : finally
file = open('python.txt' , 'r')
try:
    content = file.read()
    print(content)
finally:
    file.close()

We are learning File Handling in Python.
Hello World.
Python Programming.
Coding Ninjas.....
```

```

# try : except : finally
try:
    file = open('coding.txt' , 'r')
    try:
        content = file.read()
        print(content)
    except IOError:
        print("IO Error Occurred.")
    finally:
        file.close()
except FileNotFoundError:
    print("Error: File Not Found.")

Bye Bye! Tata Tata Khatam

# try : except : finally
try:
    file = open('hello.txt' , 'r')
    try:
        content = file.read()
        print(content)
    except IOError:
        print("IO Error Occurred.")
    finally:
        file.close()
except FileNotFoundError:
    print("Error: File Not Found.")

Error: File Not Found.
```

```
# try : except : finally
try:
    file = open('coding.txt' , 'w') # Overwrite
    try:
        content = file.read()
        print(content)
    except IOError:
        print("IO Error Occurred.")
    finally:
        file.close()
except FileNotFoundError:
    print("Error: File Not Found.")
```

IO Error Occurred.

```
# try : except : finally
try:
    file = open('coding.txt' , 'r') # Loss of data
    try:
        content = file.read()
        print(content) # " " -> You know the reason
    except IOError:
        print("IO Error Occurred.")
    finally:
        file.close()
except FileNotFoundError:
    print("Error: File Not Found.")
```

```
# try : except : finally
try:
    file = open('coding.txt' , 'a') # Appending the new content within the existing file
    try:
        file.write("\nHello World.")
        file.write("\nPython Programming.")
        file.write("\nCoding Ninjas.....")
    except IOError:
        print("IO Error Occurred.")
    finally:
        file.close()
except FileNotFoundError:
    print("Error: File Not Found.")
```

```

# try : except : finally
try:
    file = open('coding.txt' , 'r') # Loss of data
    try:
        lines = file.readlines() # return list
        for line in lines:
            print(line)
    except IOError:
        print("IO Error Occurred.")
    finally:
        file.close()
except FileNotFoundError:
    print("Error: File Not Found.")

```

Hello World.

Python Programming.

Coding Ninjas.....

```

# Deleting a file
os.getcwd()

'C:\\\\Users\\\\krish\\\\anaconda_projects\\\\CN-Python-Weekend'

```

```

file_path = "anaconda_projects/CN-Python-Weekend/coding.txt"

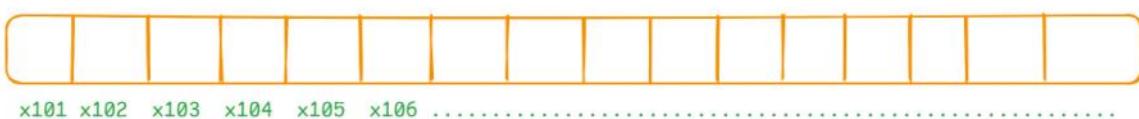
file_path = "C:\\\\Users\\\\krish\\\\anaconda_projects\\\\CN-Python-Weekend\\\\coding.txt"
try:
    os.remove(file_path)
    print(f"{file_path} is removed successfully.")
except FileNotFoundError:
    print("Error: File Not Found.")

C:\\Users\\krish\\anaconda_projects\\CN-Python-Weekend\\coding.txt is removed successfully.

```

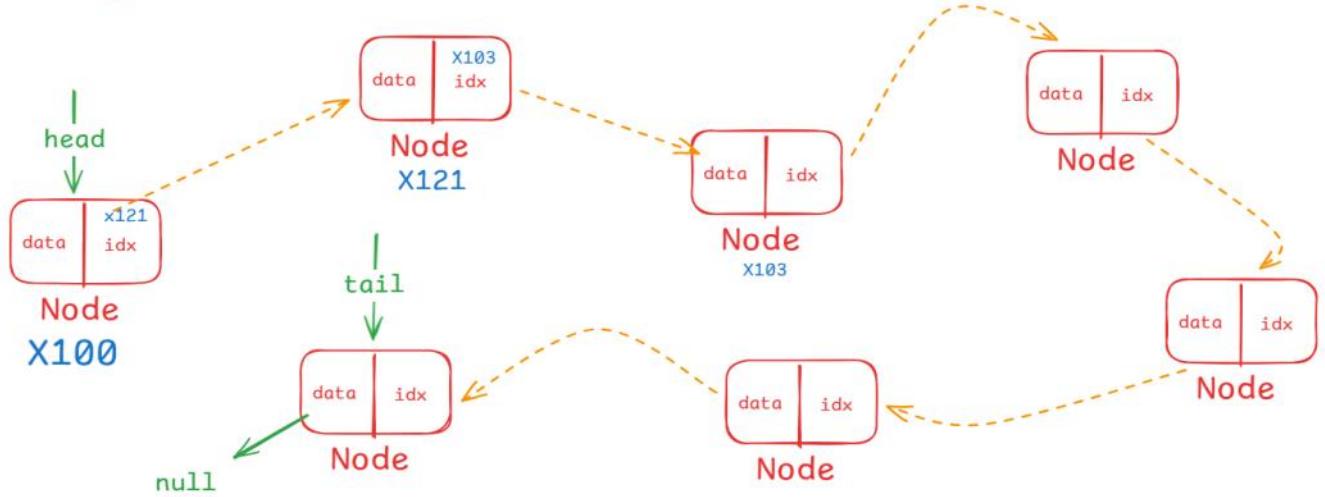
Numpy - np.array

Array



Contiguous in Nature

Linked_List



What is Numpy?

Numpy (Numerical Python) is a core library for scientific and numerical computing in Python.

Why use Numpy?

- Fast : Works with Large, multi Dimensional Arrays stored in contiguous memory blocks
- Efficient : Operations are optimized via 'C' under the hood.
- Foundational : Powered with many Scientific Libraries (eg. Pandas, Scipy , Tensorflow)

Numpy Also:

- offers the statistical tools like - mean, median, std, variance , etc.
- Integrates well with Visualization Libraries Like - Matplotlib.



[1,2,4.4,5,False,True] -> [1.0,2.0,4.4,5.0,0.0,1.0]



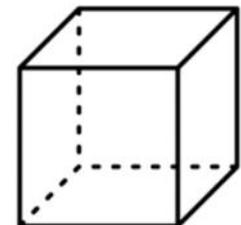
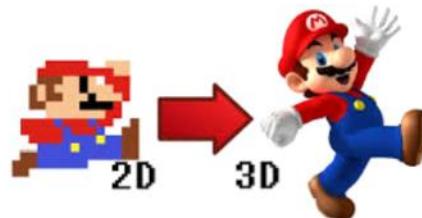
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

depth1

1	2
3	4
5	6
7	8

3D

[2, 4, 2] = 16



depth2

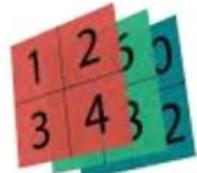
9	10
11	12
13	14
15	16



Vector



Matrix



3D Matrix

```
np.array([1, 2])           np.array([[1, 2], [3, 4]])          np.array([[[1, 2], [3, 4]], [[5, 6], [7, 8]], [[9, 10], [11, 12]]])
```

Importing NumPy in Jupyter Notebook [1](#) [2](#)

To use NumPy in a Jupyter Notebook, you first need to ensure that the library is installed in your Python environment. Once installed, you can import it using the following code:

```
import numpy as np
```

This imports NumPy and assigns it the alias `np`, which is a common convention.

Installation Steps (if not already installed)

If you encounter a `ModuleNotFoundError`, it means NumPy is not installed. You can install it directly from within Jupyter Notebook by running:

```
!pip install numpy
```

Alternatively, if you're using Python 3, you can use:

```
!pip3 install numpy
```

After installation, restart the kernel and try importing NumPy again.



$$[3 \times 3] = 9$$

$$6 \times [9] = 54$$

$$6 \times 3 \times 3 = \text{depth, row, col}$$

```
# Install
conda install numpy
OR
pip install numpy

!pip3 install numpy
Requirement already satisfied: numpy in c:\users\krish\.anaconda\new.anaconda\lib\site-packages (1.26.4)

import numpy as np
dir(np)

['ALLOW_THREADS',
 'BUFSIZE',
 'CLIP',
 'DataSource',
 'ERR_CALL',
 'ERR_DEFAULT',
 'ERR_IGNORE',
 'ERR_LOG',
 'ERR_PRINT',
 'ERR_RAISE',
 'ERR_WARN',
 'FLOATING_POINT_SUPPORT',
 'FPE_DIVIDEBYZERO',
 'FPE_INVALID',
 'FPE_OVERFLOW',
 'FPE_UNDERFLOW',
 'False']
```

```
np.__version__
'1.26.4'

print(np.__doc__)
NumPy
=====

Provides
1. An array object of arbitrary homogeneous items
2. Fast mathematical operations over arrays
3. Linear Algebra, Fourier Transforms, Random Number Generation

How to use the documentation
-----
Documentation is available in two forms: docstrings provided
with the code, and a loose standing reference guide, available from
`the NumPy homepage <https://numpy.org>`_.

We recommend exploring the docstrings using
`IPython <https://ipython.org>`, an advanced Python shell with
TAB-completion and introspection capabilities. See below for further

# num = np.array(<Iterable>
num = np.array([11,22,33,44,55,66,77,88,99])
num

array([11, 22, 33, 44, 55, 66, 77, 88, 99])
```

```

print(num)
[11 22 33 44 55 66 77 88 99]

# Indexation in numpy : start with 0
num[0]

11

num[6] # 77

77

num[-1] # 99

99

arr_from_list = np.array([11,22,33,44,55,66,77,88,99])
arr_from_tuple = np.array((11,22,33,44,55,66,77,88,99))
print(arr_from_list)
print(arr_from_tuple)

[11 22 33 44 55 66 77 88 99]
[11 22 33 44 55 66 77 88 99]

# Common Mistake
arr_from_tuple = np.array(11,22,33,44,55,66,77,88,99)
print(arr_from_tuple)
# TypeError: array() takes from 1 to 2 positional arguments but 9 were given

```

```

np_arr = np.array(11)
np_arr

array(11)

print(np_arr)

11

type(arr_from_list)

numpy.ndarray

# ndarray -> N Dimensional Array
print(type(arr_from_list)) # class <numpy.ndarray>
<class 'numpy.ndarray'>

# Properties [ndim]
# print("Hello World" , end = " ") -> end -> attribute
arr_from_list.ndim

1

# np_arr # 11
np_arr.ndim # 0

0

```

```

# 1D Array
# Numpy -> An array object of arbitrary homogeneous items
arr_1d = np.array([11,121.9,True,'k','Coding',11+5j,99]) # Mixed Type [Upcasting]
print(arr_1d)
['11' '121.9' 'True' 'k' 'Coding' '(11+5j)' '99']

type(arr_1d)
numpy.ndarray

arr_1d.dtype # property data type
dtype('

arr_1d.ndim
1

# 2D Array [2 Dimensional] [row * col] n*m (5,3)
# Numpy -> An array object of arbitrary homogeneous items
arr_2d = np.array([
    [11,22,33],
    ['a','b','c'],
    [True,False,True],
    ['learn','python','programming'],
    [9.99,11.11,12.99]
]) # Mixed Type [Heterogenous]
print(arr_2d)

```

```

[[11' '22' '33']
 ['a' 'b' 'c']
 ['True' 'False' 'True']
 ['learn' 'python' 'programming']
 ['9.99' '11.11' '12.99']]

type(arr_2d)
numpy.ndarray

arr_2d.ndim # property
2

# 3D [3 Dimensional] [Depth,row,col] (d*r*c) = Total Number of Elements
# 3 Depth , 3 Rows , 3 Cols = 27
arr_3d = np.array([
    [[1,2,3],
     [4,5,6],
     [7,8,9]],

    [[[1.1,2.1,3.1],
      [4.1,5.1,6.1],
      [7.1,8.1,9.1]],

     [[[False,True,True],
       [True,False,True],
       [False,True,False]]],

    ]]) # 'upcasting' -> 'float'
arr_3d

```

```

array([[[1. , 2. , 3. ],
       [4. , 5. , 6. ],
       [7. , 8. , 9. ]],

      [[1.1, 2.1, 3.1],
       [4.1, 5.1, 6.1],
       [7.1, 8.1, 9.1]],

      [[0. , 1. , 1. ],
       [1. , 0. , 1. ],
       [0. , 1. , 0. ]]]))

type(arr_3d)
numpy.ndarray
arr_3d.ndim # n- dimension [3D]
3

# 27 => 3*3*3
# 1D -> 3D
arr_1d = np.array([1,2,3,4,5,6,7,8,9,'a','b','c','d','e','f','g','h','i',9,8,7,6,5,4,3,2,1])
arr_1d

array(['1', '2', '3', '4', '5', '6', '7', '8', '9', 'a', 'b', 'c', 'd',
       'e', 'f', 'g', 'h', 'i', '9', '8', '7', '6', '5', '4', '3', '2',
       '1'], dtype='<U11')

```

```

# 3D Shape [3,3,3] -> depth-> 3 , row -> 3 , cols -> 3
arr_1d.reshape(3,3,3)

array([[['1', '2', '3'],
       ['4', '5', '6'],
       ['7', '8', '9']],

      [['a', 'b', 'c'],
       ['d', 'e', 'f'],
       ['g', 'h', 'i']],

      [['9', '8', '7'],
       ['6', '5', '4'],
       ['3', '2', '1']]], dtype='<U11')

arr_1d = np.array([1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16])
arr_1d

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

# 1D -> 2D
# 4*4 , 8*2, 2*8
arr_1d.reshape(4,4)

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

arr_1d.reshape(4,3) # 12 Error
ValueError: cannot reshape array of size 16 into shape (4,3)

```

```

arr_1d.reshape(8,2)

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

arr_1d.reshape(2,8)

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

# 3D -> 2*4*2 , 2*2*4, 1*4*4
arr_1d.reshape(2,4,2)

array([[[ 1,  2],
         [ 3,  4],
         [ 5,  6],
         [ 7,  8]],

       [[ 9, 10],
        [11, 12],
        [13, 14],
        [15, 16]]])

```

```

arr_1d.reshape(2,2,4)

array([[[[ 1,  2,  3,  4],
          [ 5,  6,  7,  8]],

         [[ 9, 10, 11, 12],
          [13, 14, 15, 16]]])

# 4D -> 2*2*2*2 # 3D[depth,rows,cols]
arr_4d = arr_1d.reshape(2,2,2,2) # 4D -> 2 depth of 3D
arr_4d

array([[[[[ 1,  2],
            [ 3,  4]],

           [[ 5,  6],
            [ 7,  8]]],

         [[[ 9, 10],
           [11, 12]],

           [[13, 14],
            [15, 16]]]]]

arr_4d.ndim # property

4

```