# *File Handling*

**Akhil Kaushik**
Asstt. Prof., CE Deptt.,
TIT Bhiwani

# What is a File?

- A **file** represents a sequence of bytes on disk, where a group of related data is stored.
- File is ready-made structure.
- File is created for permanent data storage (HDD).
- Syntax: file_object = open("filename", "mode")

# File Modes

- **r** – opens a file in reading mode.
- **w** – opens/ creates a file in writing mode.
- **x** - opens a file for exclusive creation. If the file already exists, the operation fails.
- **a** - opens a file in append mode.
- **t** - opens in text mode. (default)
- **b** - opens in binary mode.
- **+** - opens a file for updating (reading and writing)

# File Modes



- In '**append mode**', previous contents of file remain & we can write more data after it (if file exists).
- In '**write mode**', previous contents are cleared & user writes from scratch(if file exists).
- A '**Binary File**' is similar to text file. It deals with non-text files like images or executable files..

# Opening Files in Python

- Python has a built-in **open() function** to open a file.
- This function returns a file object, also called a **handle**, as it is used to read or modify the file accordingly.
- Ex: f = open("test.txt")  #file in current directory
- f = open("C:/Python38/README.txt")  # specifying full path of file

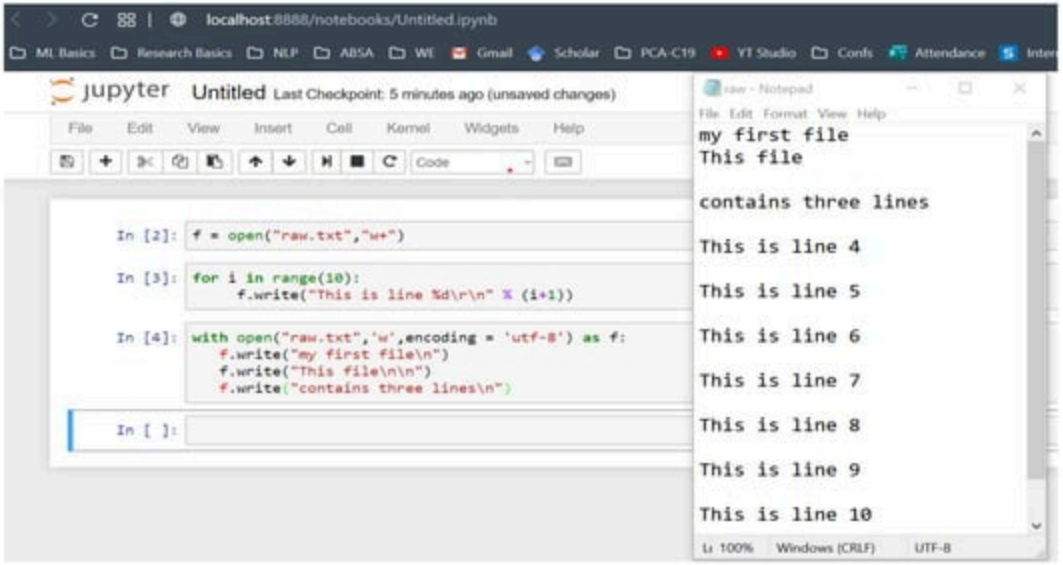
# Opening Files in Python


- f = open(“raw.txt")  # equivalent to 'r' or 'rt'
- f = open(“raw.txt",'w')  # write in text mode
- f = open("img.png",'r+b') # read and write in binary mode
- f= open(“raw.txt","w+")  #read and write permissions

# Create a Text File

- f = open(“raw.txt","w+")
- Here, f is a variable to open a file.
- open() function is used here.
- raw.txt is a text file that has name ‘raw’.
- Permission is read and write.
- If file does not exist, file will be created.

# Writing in a File



# *Reading from a File*

- f = open(“raw.txt", 'r', encoding = 'utf-8')
- f.read(4)  # read the first 4 data
- f.read(4)  # read the next 4 data
- f.read()  # read in the rest till end of file
- f.readline()  #read individual lines of a file

# Reading from a File


In [5]:
```
f = open("raw.txt",'r',encoding = 'utf-8')
f.read(4)
```

Out[5]:
```
'my f'
```

In [6]:
```
f.read(4)
```

Out[6]:
```
'irst'
```

In [7]:
```
f.read()
```

Out[7]:
```
' file\nThis file\n\ncontains three lines\n\nThis is line 4\n\nThis is line 5\n\nThis is line 6\n\nThis is line 7\n\nThis is li
ne 8\n\nThis is line 9\n\nThis is line 10\n\n'
```

# Reading from a File

```
In [1]: f = open("input.txt", "r")
        print(f.readline())
        print(f.readline())
```

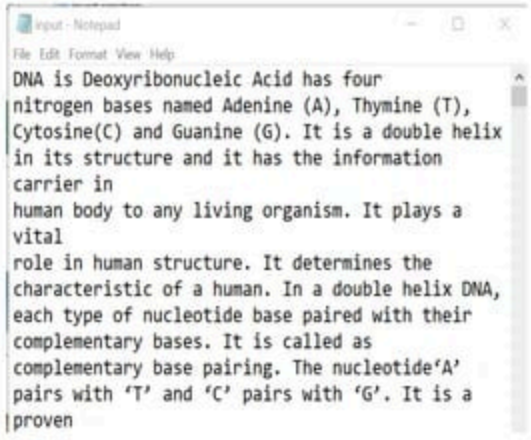
```
DNA is Deoxyribonucleic Acid has four

nitrogen bases named Adenine (A), Thymine (T),
```

```
In [2]: print(f.readline())
```

```
Cytosine(C) and Guanine (G). It is a double helix
```

```
In [3]: print(f.readline())
```

```
in its structure and it has the information carrier in
```



# Reading from a File

Read the whole file line by line:

```
In [4]: f = open("input.txt", "r")
        for x in f:
          print(x)
```

```
DNA is Deoxyribonucleic Acid has four
nitrogen bases named Adenine (A), Thymine (T),
Cytosine(C) and Guanine (G). It is a double helix
in its structure and it has the information carrier in
human body to any living organism. It plays a vital
role in human structure. It determines the
characteristic of a human. In a double helix DNA,
each type of nucleotide base paired with their
complementary bases. It is called as
complementary base pairing. The nucleotide'A'
```

# *Random Access - File*


- There is no need to access the records of file sequentially.
- To random access the files, there are 3 functions:
- f.seek()
- f.tell()

# Random Access - Files


- It is used to seek pointer position in file at specified byte. Its syntax is->

**seek(offset, from = SEEK_SET)**

- Changes the file position to offset bytes, in reference to from (start, current, end).

- **f.tell()** – It returns value of current position in file. The value is count from beginning of file. Its syntax-

**f.tell()**

# Other Functions



- **fileno()** - Returns an integer number (file descriptor) of the file.
- **flush()** – Flushes the write buffer of the file stream.
- **seekable()** - Returns True if the file stream supports random access.
- **truncate(size=None)** - Resizes the file stream to size bytes. If size is not specified, resizes to current location.
- **writable()** - Returns True if the file stream can be written to.

# Close a File


- When we are done with performing operations on the file, we need to properly close the file.
- Closing a file will free up the resources that were tied with the file. It is done using the **close() method** available in Python.
- Python has a garbage collector to clean up unreferenced objects but we must not rely on it to close the file.

# Close a File


In [7]:
```
f.read()
```

Out[7]:
```
' file\nThis file\n\ncontains three lines\n\nThis is line 4\n\nThis is line 5\n\n
ne 8\n\nThis is line 9\n\nThis is line 10\n\n'
```

In [8]:
```
f = open("raw.txt", encoding = 'utf-8')
# perform file operations
f.close()
```

In [ ]:
```

```

# Close a File

- This method is not entirely safe.
- If an exception occurs when we are performing some operation with the file, the code exits without closing the file.
- A safer way is to use a try...finally block.

```
try:
   f = open("test.txt", encoding = 'utf-8')
   # perform file operations
finally:
   f.close()
```

# Close a File


- This way, we are guaranteeing that the file is properly closed even if an exception is raised that causes program flow to stop.
- The best way to close a file is by using the **with statement**.
- We don't need to explicitly call the close() method. It is done internally.

```
with open("test.txt", encoding = 'utf-8') as f:
   # perform file operations
```

# File Handling using Colab


- **Upload the file:-**

```
from google.colab import files

upload = files.upload()
```

Browse... No files selected. Cancel upload

- **Write from Colab:-**

```
1 from google.colab import drive
2 drive.mount('/mntDrive')
```

```
Go to this URL in a browser: https://accounts.google.co

Enter your authorization code:
··········
Mounted at /mntDrive
```

# THANK YOU !!!

**CONTACT ME AT:**


Akhil Kaushik

akhilkaushik05@gmail.com

9416910303


Akhil Kaushik

Earnest