

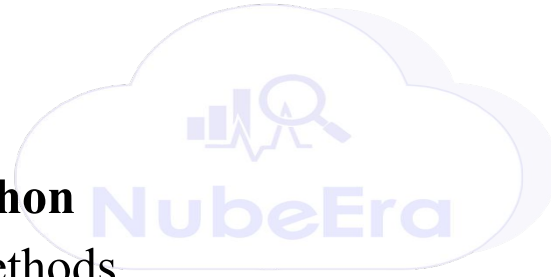
File Handling



Session Objectives



- **File Handling**
 - **Path & Directory Settings**
 - Absolute
 - Relative
 - **Open & Close file**
 - **File Modes(r,w,a)**
 - **Reading File using Python**
 - read(),readline() methods
 - **Writing Text File using Python**
 - Write and Append Mode using write() method
 - **with() ,split() ,remove() function**
- **Summary**





• Concept

- File handling is an important part of any web application.
- Python too supports file handling and allows users to handle files .
- Python has several functions for creating, reading, updating, and deleting files.
- Python has many built in methods to do this like
 - `open()`
 - `read()`
 - `readline()`
 - `write()`
 - `close()`

NubeEra





● Path & Directory Settings

○ Absolute

- An absolute file path **describes how to access a given file or directory**, starting from the root of the file system. A file path is also called a pathname.
- When accessing file is in different folder or drive then we use absolute path.
- Example: **“D:\\my_files\\python_files\\a1.txt”**

○ Relative

- Relative file paths are notated by a lack of a leading forward slash. For example, example_directory.
- When accessing file is in same folder then we use relative path.
- Example: **“a1.txt”**

NubeEra



CORPORATE



RECORDED



INTERACTIVE



● Open and close file

○ open()

- We use open () function in Python to **open a file** in read or write mode.
- Open () will return a **file object**. To return a file object we use open() function along with two arguments, that accepts file name and the mode, whether to read or write.
- **syntax :**
 - open(filename, mode)

Example:

- f = open("demofile.txt")
- f = open ("D:\\my_files\\python_files\\a1.txt")



CORPORATE



RECORDED



INTERACTIVE



- There are four kinds of mode
 - **r" - Read - Default** value. Opens a file for reading, error if the file does not exist
 - **"a" - Append** - Opens a file for appending, creates the file if it does not exist
 - **"w" - Write** - Opens a file for writing, creates the file if it does not exist
 - **"x" - Create** - Creates the empty file, error if file already exists.
 - `f = open("demofile.txt","x")`
 - `f = open ("D:\\my_files\\python_files\\a1.txt", "w")`
- In addition you can specify if the file should be handled as binary or text mode
 - **"t" - Text - Default value. Text mode**
 - **"b" - Binary - Binary mode (e.g. images)**





○ **close()**

- The **filehandle.close()** method is used to asynchronously *close the given file descriptor* thereby clearing the file that is associated with it. This will allow the file descriptor to be reused for other files.
- Calling **filehandle.close()** method on a file descriptor while some other operation is being performed on it may lead to undefined behavior.

■ **Syntax:**

`filehandle.close();`

- **Parameters:** This method does not accept any parameters.

■ **Example:**

- **f.close()**

NubeEra



CORPORATE



RECORDED



INTERACTIVE



● Reading File

○ read()

- If you need to extract a string that contains all characters in the file then we can use `file.read()`

- `file = open("myfile.txt", "r")`

`print (file.read())` #read **all content** of file

- `file = open("myfile.txt", "r")`

`print (file.read(5))` #read first 5 **characters** of file

○ readline()

- You can return one line by using the `readline()` method.

- `f = open("demofile.txt", "r")`

`print(f.readline())` #read first **line** from file





● Writing Text File

- To write to an existing file, you must add a parameter to the `open()` function:

`"a"` - Append - will append to the end of the file

`"w"` - Write - will overwrite any existing content

- `write()`

- ```
f = open("myfile.txt", "w") #create new file and write some content
f.write("Welcome!!!This is file system in python")
f.close()
```

**\*If we again open the same file using “w” mode, previous content will be deleted.**

- ```
f = open("myfile.txt", "a") #open existing file and append some content
f.write("now file has some more content")
f.close()
```





• with() function

- # Python code to illustrate with()

with open("file.txt") as file:

```
data = file.read()
```

• split() function

with open("file.text", "r") as file:

```
data = file.readlines()
```

```
for line in data:
```

```
word = line.split()
```

```
print (word)
```

#all content(lines) stores in “data”

#each line stores in”line”

#each line splits at white space and stores in word



CORPORATE



RECORDED



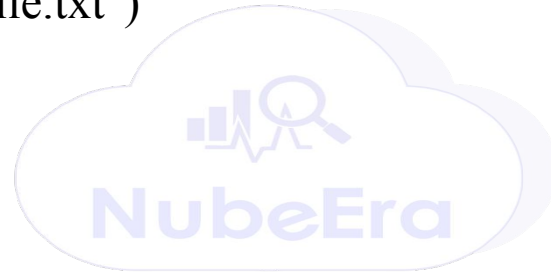
INTERACTIVE



● Delete file

○ To delete a file, you must import the OS module, and run its `os.remove()` function:

- `import os`
`os.remove("demofile.txt")`



CORPORATE



RECORDED



INTERACTIVE



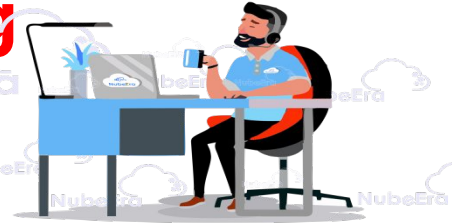
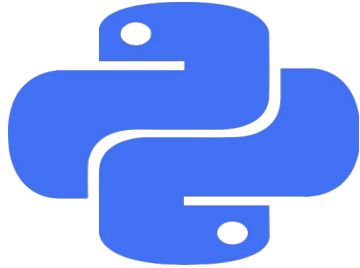
- **File Handling**
 - **Path & Directory Settings**
 - Absolute
 - Relative
 - **Open & Close file**
 - **File Modes(r,w,a)**
 - **Reading File using Python**
 - read(),readline() methods
 - **Writing Text File using Python**
 - Write and Append Mode using write() method
 - **with() ,split() ,remove() function**
- **Demo**



???

The Important thing is not to
stop

Questioning



RECORDED



CORPORATE



INTERACTIVE



CORPORATE



RECORDED



INTERACTIVE