

# Programming Fundamentals

---

Lecture 10 – File Handling

Iresh Bandara

# Learning Outcomes

---

- This lecture addresses LO2 for the module
- On completion of this lecture, students are expected to explain and apply
  - File read, apply
  - File modes
  - File handling with exceptions

# Agenda

---

- Python file handling
  - Opening
  - writing
  - reading
- File modes
- Exceptions and file handling
- Appending and file replacement

# Introduction

---

- It is a common task to
  - read some input data file
  - do some processing with the data
  - write some output data file with results
- Python distinguishes between
  - *text* files ( 't' )
  - *binary* files ( 'b' )

# File modes

---

Mode	Description
'r'	default mode. It Opens file for reading.
'w'	opens file for writing. If file does not exist, it creates a new file. If file exists it replaces
'x'	Creates a new file. If file already exists, the operation fails.
'a'	Open file in append mode. If file does not exist, it creates a new file.
't'	This is the default mode. It opens in text mode.
'b'	This opens in binary mode.
'+'	This will open a file for reading and writing (updating)

# Open, write and close

---

- File opening syntax

```
file_object = open("filename", "mode")
```

- **filename:** gives name of the file that the file object has opened.
- **mode:** attribute of a file object tells you which mode a file was opened. Check the file modes slide.

- File writing syntax

```
file_object.write("anything you want to add here")
```

- Closing the instance

```
File_object.close()
```

# Example – file writing

```
f=open("test.txt", "w+")
for i in range(10):
    f.write("This is line %d \n" % (i+1))
f.close()
```

```
This is line 1
This is line 2
This is line 3
This is line 4
This is line 5
This is line 6
This is line 7
This is line 8
This is line 9
This is line 10
```

Output

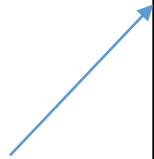


# Appending example

```
f=open("test.txt", "a+")
for i in range(10,20):
    f.write("This is line %d \n" % (i+1))
f.close()
```

- Spot the difference between w+ and a+

Output



```
This is line 1
This is line 2
This is line 3
This is line 4
This is line 5
This is line 6
This is line 7
This is line 8
This is line 9
This is line 10
This is line 11
This is line 12
This is line 13
This is line 14
This is line 15
This is line 16
This is line 17
This is line 18
This is line 19
This is line 20
```



# File open, read and close

---

- File opening syntax

```
file_object = open("filename", "read_mode")
```

- File reading syntax

```
file_object.read() or file_object.readlines()
```

- Closing the instance

```
file_object.close()
```

# Example – file reading

```
f=open("test.txt", "r")  
content= f.read()  
print(content)  
f.close()
```

read()

```
f=open("test.txt", "r")  
content= f.readlines()  
for line in content:  
    print(line)  
f.close()
```

readlines()

# Exercise

---

- The following file called `studentdata.txt` contains one line for each student in a class. The student's name is the first thing on each line, followed by some exam scores. The number of scores might be different for each student.

```
joe 10 15 20 30 40
bill 23 16 19 22
sue 8 22 17 14 32 17 24 21 2 9 11 17
grace 12 28 21 45 26 10
john 14 32 25 16 89
```

- Using the text file `studentdata.txt` write a program that prints out **the names of students that have more than six quiz scores.**

# File handling and exceptions

---

- If the file doesn't exist, then the `open()` raises the `FileNotFoundError` exception

```
import sys
try:
    f = open('myfilename.txt', 'r')
except FileNotFoundError:
    print("The file couldn't be found. ")
    sys.exit(1)  # a way to exit the program

for line in f:
    print(line, end='')
f.close()
```

# Summary

---

- Python allows you to read, write and delete files
- Use the function `open("filename","w+")` to create a file. The + tells the python interpreter to open file with read and write permissions.
- To append data to an existing file use the command `open("Filename", "a")`
- Use the read function to read the ENTIRE contents of a file
- Use the readlines function to read the content of the file one by one.