

PYTHON TEXTFILE STATEMENT **TEXTFILE(2)**

Sirinthorn Cheyasak, Asst.Prof.

OUTLINE



Python File Mode (r+, w+, a+)

Multiples Input file

Multiples Output file

Formatted file output

Python File Examples

Class Activity

PYTHON TEXTFILE



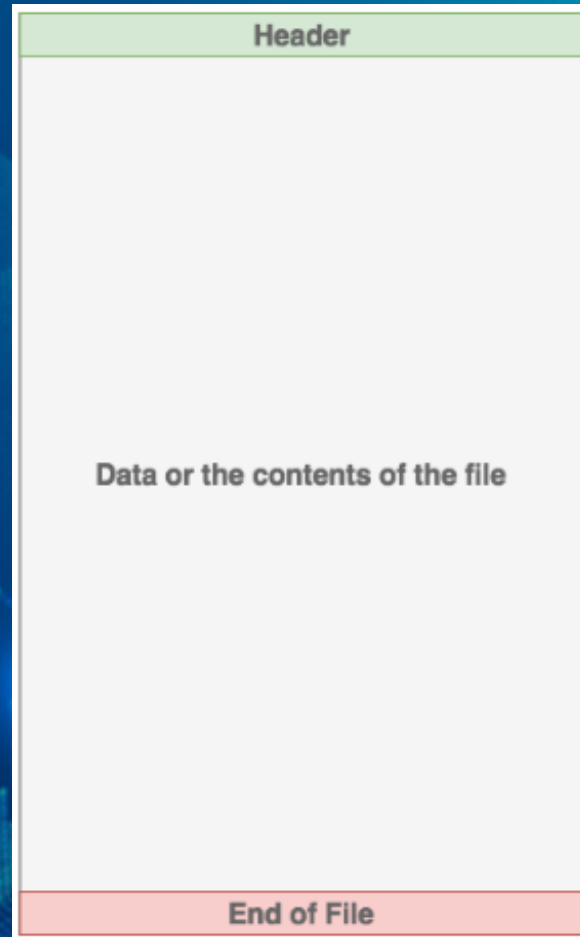
Real Python

TEXTFILE CONTENTS

Files on most modern file systems are composed of **three main parts**:

- 1. Header:** metadata about the contents of the file (file name, size, type, and so on)
- 2. Data:** contents of the file as written by the creator or editor
- 3. End of file (EOF):** special character that indicates the end of the file

THREE MAIN PARTS OF MODERN FILE SYSTEM





PYTHON FILE HANDLING MODE

OPEN A FILE IN PYTHON

- To open the file, use the built-in **open()** function
- Syntax:

```
file_name = open(file_name [, access_mode][, buffering])
```

file_name คือ ตัวแปรที่ใส่เก็บข้อมูลไฟล์

open คือ ฟังก์ชันที่ใช้สำหรับขอเชื่อมต่อเพื่อเปิดใช้งาน

file_name คือ การระบุชื่อไฟล์ที่ต้องการเปิด

access_mode คือ การระบุโหมดการเข้าถึงไฟล์ default = read only

buffering คือ กำหนดขนาดหน่วยความจำชั่วคราวสำหรับพักข้อมูลก่อนประมวลผล ไม่จำเป็นต้องระบุ

FILE ACCESSING MODE

โหมด	ความหมาย
r	เปิดไฟล์เพื่ออ่านอย่างเดียว (read only) ไฟล์ต้องมีอยู่จริงตามที่ระบุ
r+	เปิดไฟล์เพื่ออ่านและเขียนทับ แต่ไฟล์ต้องมีอยู่จริงตามที่ระบุ
w	เปิดไฟล์เพื่อเขียนอย่างเดียว (write only) โดยจะเขียนทับข้อมูลเดิมในไฟล์ที่มีอยู่แล้ว ถ้าไม่พบไฟล์ที่ระบุจะเปิดไฟล์ขึ้นมาใหม่
w+	เปิดไฟล์เพื่ออ่านและเขียน โดยจะเขียนทับข้อมูลเดิมในไฟล์ที่มีอยู่แล้ว ถ้าไม่พบไฟล์ที่ระบุจะเปิดไฟล์ขึ้นมาใหม่
a	เปิดไฟล์เพื่อเขียนต่อท้ายไฟล์เดิม ถ้าไม่พบไฟล์ที่ระบุจะสร้างไฟล์ขึ้นมาใหม่
a+	เปิดไฟล์เพื่ออ่านและเขียน ถ้าไม่พบไฟล์ที่ระบุจะสร้างไฟล์ขึ้นมาใหม่

PYTHON OPEN FILE

```
file = open("data2.txt","r+")  
file.close()
```

mode r+
ต้องสร้างไฟล์ก่อนเสมอ






















```
with open("data3.txt","w+") as file :  
    print("Mode w+")
```

mode w+
ไม่จำเป็นต้องสร้างไฟล์ก่อน

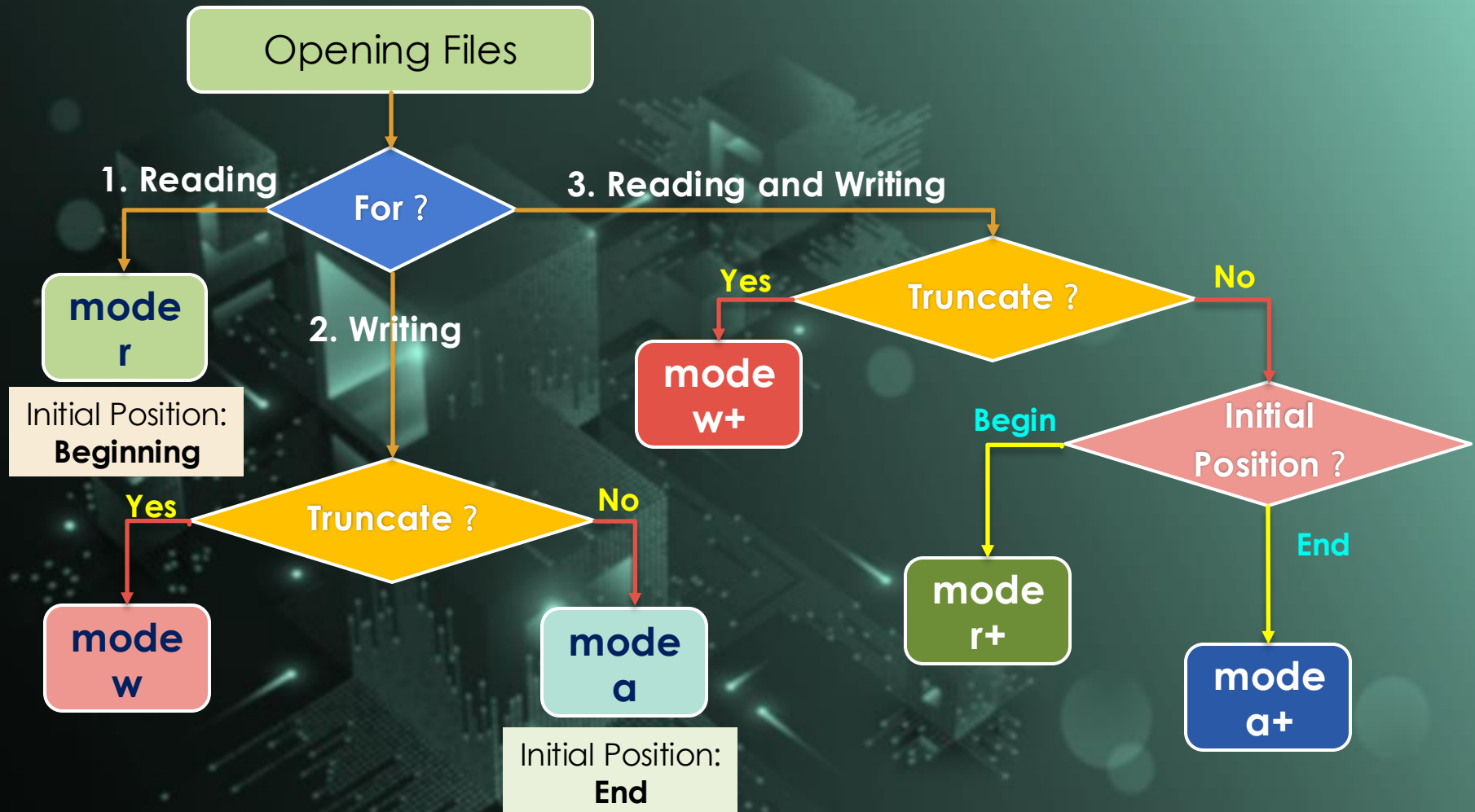
```
with open("data4.txt","a+") as file:  
    print("Mode a+")
```

mode a+
ไม่จำเป็นต้องสร้างไฟล์ก่อน

SUMMARY TABLE OF PYTHON FILE

Mode	r	r+	w	w+	a	a+
read						
write						
create						
truncate						
position at start						
position at end						

OPENING FILES CONCEPT



CLASS ACTIVITY - 1

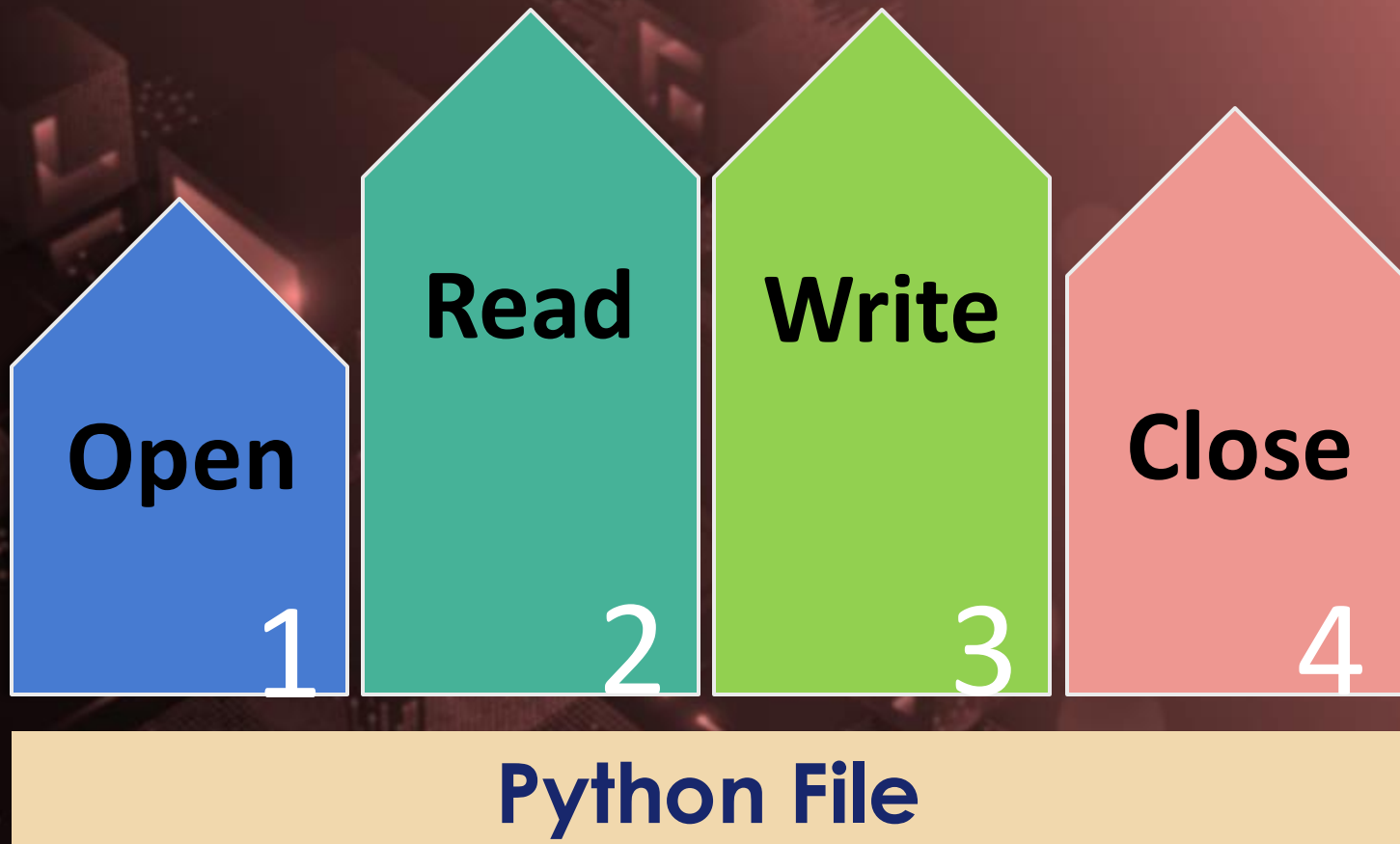
Instruction : write python a command of the following :



Google Form

SCORE CHECKING

Instruction : teacher explain the activity-1



ITERATING OVER EACH LINE IN THE FILE

customer1.txt

```
1001 Name1 Surname1 M 24 25000  
1002 Name2 Surname2 F 35 32000  
1003 Name3 Surname3 F 29 24000  
1004 Name4 Surname4 F 45 33000  
1005 Name5 Surname5 M 22 17800
```



```
['1001', 'Name1', 'Surname1', 'M', '24', '25000']  
['1002', 'Name2', 'Surname2', 'F', '35', '32000']  
['1003', 'Name3', 'Surname3', 'F', '29', '24000']  
['1004', 'Name4', 'Surname4', 'F', '45', '33000']  
['1005', 'Name5', 'Surname5', 'M', '22', '17800']
```

WAYS TO ITERATING OVER EACH LINE

1

Using for loop and `split()`

2

Using for loop and `splitlines()`

3

Using while loop and `readline()`

For + split() : Case of excluded the `\n` **read()** provide attribute **split()**

```
with open("customer1.txt","r") as file :  
    data = file.read().split("\n")  
    for line in data :  
        item = line.split()  
        print(item)
```

```
['1001', 'Name1', 'Surname1', 'M', '24', '25000']  
['1002', 'Name2', 'Surname2', 'F', '35', '32000']  
['1003', 'Name3', 'Surname3', 'F', '29', '24000']  
['1004', 'Name4', 'Surname4', 'F', '45', '33000']  
['1005', 'Name5', 'Surname5', 'M', '22', '17800']
```


For + splitlines() : Case of excluded the `\n` **read()**
provide attribute **splitlines()**

```
with open("customer1.txt","r") as file :  
    data = file.read().splitlines()  
    for line in data :  
        item = line.split()  
        print(item)
```

```
['1001', 'Name1', 'Surname1', 'M', '24', '25000']  
['1002', 'Name2', 'Surname2', 'F', '35', '32000']  
['1003', 'Name3', 'Surname3', 'F', '29', '24000']  
['1004', 'Name4', 'Surname4', 'F', '45', '33000']  
['1005', 'Name5', 'Surname5', 'M', '22', '17800']
```

3

ITERATING OVER EACH LINE IN THE FILE

while + readline() + split() : Case of excluded the `\n`
readline()

```
with open("customer1.txt","r") as file :  
    line = file.readline()  
    while line != "" :  
        item = line.split()  
        print(item)  
        line = file.readline()
```

```
['1001', 'Name1', 'Surname1', 'M', '24', '25000']  
['1002', 'Name2', 'Surname2', 'F', '35', '32000']  
['1003', 'Name3', 'Surname3', 'F', '29', '24000']  
['1004', 'Name4', 'Surname4', 'F', '45', '33000']  
['1005', 'Name5', 'Surname5', 'M', '22', '17800']
```

PYTHON FILE WRITE

- The **write()** writes a single string line to file descriptor:

```
file_name.write( )
```

file_name คือ ตัวแปรที่ใช้เก็บข้อมูลไฟล์

write คือ เมธอดที่ใช้สำหรับการเขียนข้อมูลที่เป็นสตริงลงไฟล์

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

"w" - Write - will overwrite any existing content

FORMATTED OUTPUT FILE

How to formatted output file?

```
code = ['ID0001', 'ID0002', 'ID0003', 'ID0004', 'ID0005']
name = ['thefrist surnamefirst', 'thesecond surnamessecond', 'thethird
surnamethird', 'theforth surnameforth', 'thefifth surnamefifth']
gpa = ['2.33', '3.45', '1.89', '2.89', '2.99']

with open("output.txt", "w") as outfile :
    for i in range(len(code)) :
        outfile.write(f"{code[i]:<8} {name[i]:<28} {gpa[i]:<6.2f}")
```


FORMATTED OUTPUT FILE

How to formatted output file?

```
with open("customer1.txt","r") as file :  
    data = file.readline()  
    with open("mycontent.txt","w") as outfile :  
        while data != "" :  
            item = line.split()  
            print(item)  
            outfile.write(f" {item[0]:<8} {item[1]:<10}  
{item[2]:<12}{item[3]:>2}{item[4]:>2} {item[5]:<12}\n")  
            data = file.readline()
```

CLASS ACTIVITY#1

Read data from input files for processing and display the output file name is report.txt

score.txt

Alice, 85
Bob, 78
Charlie, 90
David, 67
Eva, 92
Frank, 75
Grace, 88
Hannah, 82
Isaac, 95
Jack, 73

report.txt

Alice	85	A
Bob	78	-
Charlie	90	A
David	67	-
Eva	92	A
Frank	75	-
Grace	88	A
Hannah	82	A
Isaac	95	A
Jack	73	-

CLASS ACTIVITY#2

Read data from input files for processing and display the output file name is gradereport.txt

exam_scores.txt

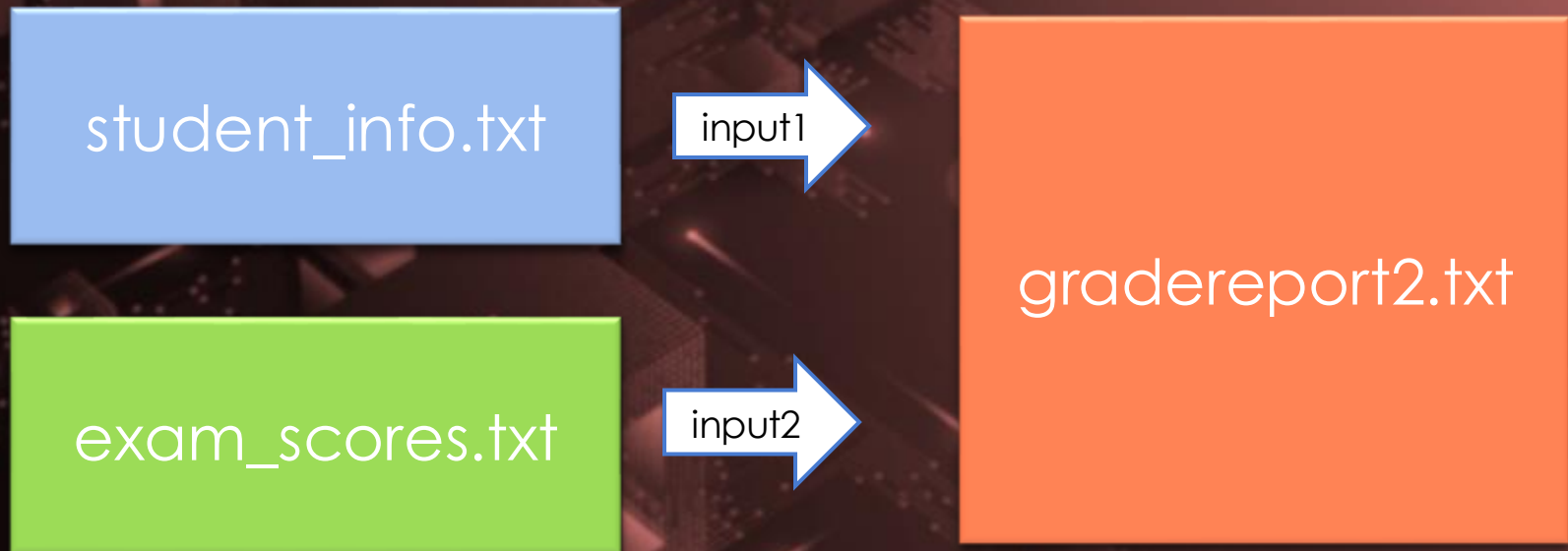
```
1012023001, Alice Johnson, 28, 50
1012023002, Bob Smith, 30, 55
1012023003, Carol Davis, 22, 40
1012023004, David Wilson, 25, 45
1012023005, Eva Brown, 30, 60
1012023006, Frank Jones, 20, 42
1012023007, Grace Miller, 27, 48
1012023008, Hannah Taylor, 24, 43
1012023009, Isaac Anderson, 18, 38
1012023010, Jack Thomas, 26, 47
1012023011, Karen Jackson, 29, 53
1012023012, Leo White, 21, 40
1012023013, Maria Harris, 33, 61
1012023014, Nick Martin, 28, 50
1012023015, Olivia Lee, 24, 45
1012023016, Peter Perez, 27, 49
1012023017, Quinn King, 32, 55
1012023018, Rachel Wright, 23, 41
1012023019, Sam Green, 29, 52
1012023020, Tina Scott, 25, 46
```


gradereport.txt

```
1012023001 Alice Johnson 78.00 B
1012023002 Bob Smith 85.00 A
1012023003 Carol Davis 62.00 C
1012023004 David Wilson 70.00 B
1012023005 Eva Brown 90.00 A
1012023006 Frank Jones 62.00 C
1012023007 Grace Miller 75.00 B
1012023008 Hannah Taylor 67.00 C
1012023009 Isaac Anderson 56.00 D
1012023010 Jack Thomas 73.00 B
1012023011 Karen Jackson 82.00 A
1012023012 Leo White 61.00 C
1012023013 Maria Harris 94.00 A
1012023014 Nick Martin 78.00 B
1012023015 Olivia Lee 69.00 C
1012023016 Peter Perez 76.00 B
1012023017 Quinn King 87.00 A
1012023018 Rachel Wright 64.00 C
1012023019 Sam Green 81.00 A
1012023020 Tina Scott 71.00 B
```

LAB10 : MULTIPLE INPUT FILE

Instruction : At Microsoft Teams in week10





THANK YOU
Q/A