



# CSS 112 Computer Programming

## Lecture VI: File I/O

Suvil Chomchaiya (Ph.D.)



# Contents

- การจัดการไฟล์ **Python**
- การเปิดไฟล์
- การปิดไฟล์
- การเขียนไฟล์
- การสร้างไฟล์ใหม่



## การจัดการไฟล์ Python

- **Python** อนุญาตให้ผู้ใช้จัดการไฟล์ (**File Handling**) ควบคู่ไปกับตัวเลือกการจัดการไฟล์อื่นๆ เพื่อดำเนินการกับไฟล์
- ใน **Python** ไม่จำเป็นต้องนำเข้าไลบรารีภายนอกเพื่ออ่านและเขียนไฟล์
- **Python** มีฟังก์ชัน **built-in** สำหรับสร้าง เขียน และอ่านไฟล์



## การจัดการไฟล์ Python

- ลักษณะของไฟล์ข้อมูลที่ Python รองรับ
  - ไฟล์ข้อความ (Text File):
    - ในไฟล์ประเภทนี้ ข้อความแต่ละบรรทัดจะสิ้นสุดด้วยอักขระพิเศษที่เรียกว่า EOL (สิ้นสุดบรรทัด) ซึ่งเป็นอักขระขึ้นบรรทัดใหม่ ('\n') โดยเป็นค่าเริ่มต้นใน Python
    - ในกรณีของไฟล์ CSV หรือ Comma Separate Value ซึ่งเป็นไฟล์ที่คั่นด้วยเครื่องหมายจุลภาค EOF จะใช้เครื่องหมายจุลภาคเป็นค่าเริ่มต้น
  - ไฟล์ไบนารี (Binary File):
    - ในไฟล์ประเภทนี้ ไม่มีตัวสิ้นสุดสำหรับบรรทัด
    - ข้อมูลจะถูกจัดเก็บหลังจากแปลงเป็นภาษาไบนารีที่เครื่องเข้าใจได้ เช่น รูปแบบ 0 และ 1





## การจัดการไฟล์ Python: การเปิดไฟล์

- Python ใช้ฟังก์ชัน `open()` ในการเปิดไฟล์ ซึ่งฟังก์ชันนี้จะคืนค่าเป็นไฟล์ออกมา
- ฟังก์ชันเปิดรับสองพารามิเตอร์ คือ ชื่อของไฟล์และโหมดการเข้าถึง
- โหมดการเข้าถึงจะระบุการดำเนินการที่เราจะดำเนินการกับไฟล์ไม่ว่าจะอ่านหรือเขียน ฟังก์ชัน `open ()` จะส่งคืนอ็อบเจกต์ไฟล์ / แชนเนล
- สามารถดำเนินการไฟล์ตามโหมดการเข้าถึง (`access mode`)
- ในการเปิดไฟล์แบบไม่ได้ระบุ `mode (argument` ตัวที่ 2) จะหมายถึงการอ่านเพียงอย่างเดียว (`reading`)



## การจัดการไฟล์ Python: การเปิดไฟล์

- Mode ในการเปิดไฟล์

r	โหมดอ่านอย่างเดียว ตัวชี้อยู่ที่จุดเริ่มต้นของไฟล์
rb	โหมดอ่านอย่างเดียวในรูปแบบไบนารี ตัวชี้อยู่ที่จุดเริ่มต้นของไฟล์
r+	อนุญาตทั้งอ่านและเขียน ตัวชี้อยู่ที่จุดเริ่มต้นของไฟล์
rb +	อนุญาตทั้งการอ่านและเขียนในรูปแบบไบนารี ตัวชี้อยู่ที่จุดเริ่มต้นของไฟล์



## การจัดการไฟล์ Python: การเปิดไฟล์

- Mode ในการเปิดไฟล์

w	โหมดเขียนอย่างเดียว เขียนทับไฟล์หากมีอยู่แล้วจะสร้างไฟล์ใหม่ ตัวชี้อยู่ที่จุดเริ่มต้นของไฟล์
wb	โหมดเขียนอย่างเดียวในรูปแบบไบนารี เขียนทับไฟล์หากมีอยู่แล้วจะสร้างไฟล์ใหม่ ตัวชี้อยู่ที่จุดเริ่มต้นของไฟล์
w+	อนุญาตให้ใช้ทั้งการอ่านและเขียน w + เขียนทับหากไฟล์มีอยู่แล้วในขณะที่ r + ไม่ได้เขียนทับไฟล์ที่มีอยู่ ตัวชี้อยู่ที่จุดเริ่มต้นของไฟล์



## การจัดการไฟล์ Python: การเปิดไฟล์

- Mode ในการเปิดไฟล์

wb +	อนุญาตให้ใช้ทั้งการอ่านและเขียนในรูปแบบไบนารี wb + เขียนทับหากไฟล์มีอยู่แล้วในขณะที่ rb + ไม่ได้เขียนทับไฟล์ที่มีอยู่ ตัวชี้อยู่ที่จุดเริ่มต้นของไฟล์
a	โหมดผนวก หากไฟล์มีอยู่ตัวชี้จะอยู่ที่ท้ายไฟล์ มิฉะนั้นจะสร้างไฟล์ใหม่
ab	ผนวกโหมดในรูปแบบไบนารี หากไฟล์มีอยู่ตัวชี้จะอยู่ที่ท้ายไฟล์ มิฉะนั้นจะสร้างไฟล์ใหม่





## การจัดการไฟล์ Python: การเปิดไฟล์

- Mode ในการเปิดไฟล์

a+	อนุญาตให้มีการดำเนินการต่อท้ายและอ่าน หากไฟล์มีอยู่ตัวชี้จะอยู่ที่ท้ายไฟล์ มิฉะนั้นจะสร้างไฟล์ใหม่
ab +	อนุญาตให้ทั้งผนวกและอ่านการดำเนินการในรูปแบบไบนารี หากไฟล์มีอยู่ตัวชี้จะอยู่ที่ท้ายไฟล์ มิฉะนั้นจะสร้างไฟล์ใหม่



## การจัดการไฟล์ Python: การเปิดไฟล์

- File Opening Syntax in Python

```
file_object=open(filename, access_mode)
```

- Example:

- `f = open("demofile.txt", "r")` `print(f.read())` #อ่านไฟล์และแสดงผล

- `f = open("D:\\myfiles\\welcome.txt", "r")` `print(f.read())` #อ่านไฟล์จาก Location อื่นๆ



## การจัดการไฟล์ Python: การเปิดไฟล์

- Example:

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

#อ่านไฟล์บรรทัดเดียว

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

#อ่าน 2 บรรทัดแรกของไฟล์



## การจัดการไฟล์ Python: การเปิดไฟล์

- Example:

- `f = open("demofile.txt", "r")`      #เปิดไฟล์บรรทัดต่อบรรทัด โดยใช้คำสั่ง `for Loop`  
    `for x in f:`  
        `print(x)`





## การจัดการไฟล์ Python: การปิดไฟล์

- เมื่อดำเนินการกับไฟล์เสร็จแล้ว จำเป็นต้องปิดไฟล์อย่างถูกต้องเพื่อคืนทรัพยากรที่ผูกกับการเปิดไฟล์นั้นให้กับระบบ
- Example:

```
f = open("demofile.txt", "r")  
print(f.readline())  
f.close()
```



## การจัดการไฟล์ Python: การปิดไฟล์

- เมื่อดำเนินการกับไฟล์เสร็จแล้ว จำเป็นต้องปิดไฟล์อย่างถูกต้องเพื่อคืนทรัพยากรที่ผูกกับการเปิดไฟล์นั้นให้กับระบบ
- Example:

```
f = open("demofile.txt", "r")  
print(f.readline())  
f.close()
```



## การจัดการไฟล์ Python: การปิดไฟล์

- Example:

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



## การจัดการไฟล์ Python: การเขียนไฟล์

- ในการเขียนไฟล์ใน **Python** ต้องเปิดไฟล์ในโหมดการเขียน (**w**) หรือโหมดการเขียนเพิ่มเติม (**a**)
- ต้องระวังด้วยโหมด **w** เพราะจะเป็นการเขียนทับลงในไฟล์หากมีอยู่แล้ว ซึ่งข้อมูลก่อนหน้านี้ทั้งหมดจะเขียนทับหมด





## การจัดการไฟล์ Python: การเขียนไฟล์

- Example:

```
with open("test.txt", 'w', encoding = 'utf-8') as f:  
    f.write("my first file\n")  
    f.write("This file\n\n")  
    f.write("contains three lines\n")
```

- โปรแกรมนี้จะสร้างไฟล์ใหม่ชื่อ **test.txt** ในไดเรกทอรีปัจจุบันหากไม่มีอยู่ หากมีอยู่ก็就会被เขียนทับทั้งหมด



## การจัดการไฟล์ Python: การเขียนไฟล์

- Example:

```
f = open("demofile2.txt", "a")
f.write("Now the file has more content!")
f.close()

#open and read the file after the appending:
f = open("demofile2.txt", "r")
print(f.read())
```

- ตัวอย่าง **code** จะเป็นการเพิ่มเนื้อหาให้กับไฟล์ **demofile2.txt** โดยที่ข้อมูลเก่ายังอยู่



## การจัดการไฟล์ Python: การเขียนไฟล์

- Example:

```
f = open("demofile3.txt", "w")
f.write("Woops! I have deleted the content!")
f.close()

#open and read the file after the appending:
f = open("demofile3.txt", "r")
print(f.read())
```

- ตัวอย่าง code จะเป็นการเพิ่มเนื้อหาให้กับไฟล์ demofile3.txt โดยที่ข้อมูลเก่าจะถูกเขียนทับทั้งหมด



## การจัดการไฟล์ Python: การสร้างไฟล์ใหม่

- ในการสร้างไฟล์ใหม่ใน Python ใช้ฟังก์ชัน `open()` ตามด้วยหนึ่งในพารามิเตอร์ต่อไปนี้:

```
f = open("<file name>", "x") # Text create
```

```
f = open("<file name>", "xt") # Same as above
```

```
f = open("<file name>", "xb") # Binary create
```





## การจัดการไฟล์ Python: การสร้างไฟล์ใหม่

- Example:

```
file = open('medium.txt','w')  
file.write("Medium is a platform for readers and writers")  
file.write("It gives writers full freedom to speak out loud with their  
thoughts and opinions.")  
file.close()
```

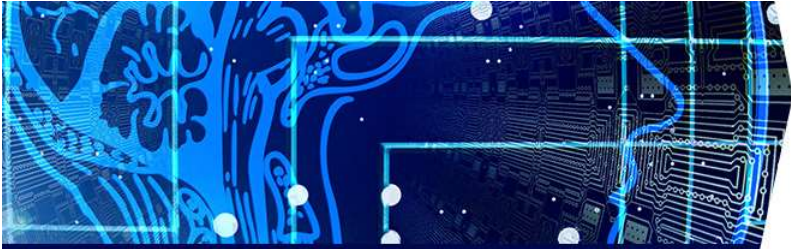
### Output

Look into the directory you're working on, there will be file created recently, open up this file and you'll find the text information.



- Read more:

- <https://www.pythontutorial.net/python-basics/python-write-text-file/>
- <https://www.digitalocean.com/community/tutorials/python-read-file-open-write-delete-copy>
- <https://www.digitalocean.com/community/tutorials/python-read-file-open-write-delete-copy>



**The End.....**  
**Questions????**