แผนบริหารการสอนประจำบทที่ 7

แฟ้มข้อมูล

หัวข้อประจำบท

1. เบื้องต้นเกี่ยวกับแฟ้มข้อมูล
2. การจัดการข้อมูลในแฟ้มข้อมูล
3. การจัดการแฟ้มข้อมูลและตำแหน่งที่อยู่

วัตถุประสงค์เชิงพฤติกรรม

1. เพื่อให้ผู้เรียนสามารถอธิบายเกี่ยวกับเบื้องต้นเกี่ยวกับแฟ้มข้อมูลได้
2. เพื่อให้ผู้เรียนสามารถอธิบายเกี่ยวโครงสร้างการทำงานของแฟ้มข้อมูลได้
3. เพื่อให้ผู้เรียนสามารถอธิบายเกี่ยวกับการจัดการข้อมูลในแฟ้มข้อมูลได้
4. เพื่อให้ผู้เรียนสามารถอธิบายเกี่ยวกับการจัดการแฟ้มข้อมูลและตำแหน่งที่อยู่ได้
5. เพื่อให้ผู้เรียนสามารถนำความรู้เรื่อง แฟ้มข้อมูล และสามารถประยุกต์ใช้งานในการแก้โจทย์ปัญหาได้อย่างถูกต้อง

วิธีการสอนและกิจกรรม

1. ผู้สอนบรรยายในชั้นเรียน และโปรแกรมนำเสนอ ตามหัวข้อเนื้อหาในเอกสารประกอบการสอน
2. ผู้เรียนทำแบบฝึกหัดท้ายบท และฝึกเขียนโปรแกรมด้วยคอมพิวเตอร์

สื่อการเรียนการสอน

1. เอกสารประกอบการสอน วิชา หลักการเขียนโปรแกรมคอมพิวเตอร์
2. โปรแกรม Python รุ่น 3.8.10
3. โปรแกรม PowerPoint

การวัดผลและการประเมินผล

1. สังเกตจากการอภิปราย การตอบคำถาม และซักถามระหว่างเรียน
2. การปฏิบัติบนเครื่องคอมพิวเตอร์
3. การทำแบบฝึกหัดท้ายบท

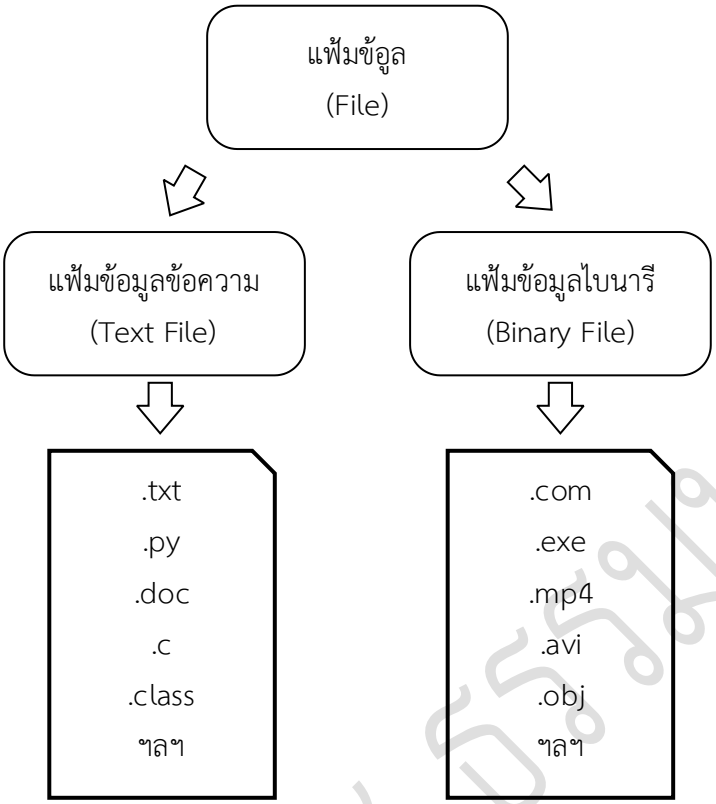
บทที่ 7
แฟ้มข้อมูล

การเก็บข้อมูลในคอมพิวเตอร์นั้น ข้อมูลจะถูกเก็บในรูปแบบของแฟ้มข้อมูล ที่กำหนดด้วยชื่อเฉพาะที่ใช้อ้างอิงใช้งาน แฟ้มข้อมูลถูกเก็บไว้ในอุปกรณ์ที่ใช้ในการเก็บข้อมูล ซึ่งอาจอยู่ในเครื่องคอมพิวเตอร์หรืออุปกรณ์ภายนอก เช่น รอม (ROM) จานบันทึกแบบแข็ง (Hard Disk) แผ่นซีดี (Compact Disk) หรืออุปกรณ์อิเล็กทรอนิกส์ในปัจจุบัน (ราชบัณฑิตยสภา, 2564) ลักษณะของแฟ้มข้อมูล จะเก็บไว้ที่ตำแหน่งที่อยู่ของชุดการทำงานที่มีการทำงานกลุ่มเดียวกัน สอดคล้องและมีการทำงานร่วมกัน ระหว่างการทำงานนั้น อาจจะมีการดำเนินการจัดการข้อมูลภายในแฟ้มข้อมูลรูปแบบต่าง ๆ เช่น การสร้าง การอ่าน การเขียน การลบ เป็นต้น เพื่อเปลี่ยนแปลงข้อมูลภายในแฟ้มข้อมูลตามวัตถุประสงค์ที่ต้องการ นอกเหนือจากนี้ยังอาจมีการจัดการแฟ้มข้อมูล เช่น การสร้างตำแหน่งที่อยู่เก็บข้อมูล การลบแฟ้มข้อมูล การเปลี่ยนชื่อแฟ้มข้อมูล การย้ายตำแหน่งที่อยู่ เป็นต้น ดังนั้นการดำเนินการเหล่านี้ ภาษาไพธอนได้เตรียมเครื่องมือที่ใช้ในการจัดการแฟ้มข้อมูลดังกล่าว ในรูปแบบของฟังก์ชันและโมดูล ประกอบด้วยการทำงานหลัก ดังนี้คือ การจัดการข้อมูลในแฟ้มข้อมูล การจัดการข้อมูลแฟ้มข้อมูลและตำแหน่งที่อยู่ของแฟ้มข้อมูล โดยมีรายละเอียดดังต่อไปนี้

เบื้องต้นเกี่ยวกับแฟ้มข้อมูล

แฟ้มข้อมูล (File) หมายถึง การเก็บรวบรวมรายละเอียดข้อมูล ที่ซึ่งถูกเก็บไว้ในคอมพิวเตอร์โดยให้ใช้ชื่อเฉพาะ (Lutz, 2013) ข้อมูลที่เก็บจะมีลักษณะขนาดเล็กที่สุดที่คอมพิวเตอร์จะแสดงให้ผู้ใช้งานได้เห็น โดยการเก็บข้อมูลจะอยู่ในรูปของระเบียน (Record) อาจถูกเก็บในรูปแบบและสถานที่ที่แตกต่างกัน เช่น แฟ้มเอกสาร แฟ้มรูปภาพ แฟ้มมัลติมีเดีย แฟ้มทำงานบนเว็บ แฟ้มสำหรับสั่งงานคอมพิวเตอร์ แฟ้มสั่งงานเชื่อมต่ออุปกรณ์รอบข้าง เป็นต้น ซึ่งแฟ้มข้อมูลประกอบไป 2 ชนิดหลัก ได้แก่ แฟ้มข้อความ (Text File) สำหรับเก็บข้อมูลที่สามารถเข้าใจได้ เช่น .txt .py .doc .c .class เป็นต้น และแฟ้มไบนารี (Binary File) ที่ใช้สำหรับล่างสุด เพื่อใช้สั่งงานคอมพิวเตอร์ ส่วนมากไม่สามารถอ่านทำความเข้าใจได้ เช่น .com .exe .mp4 .avi .obj เป็นต้น (สุชาติ คุ้มมณี, 2558)

จากความหมายเบื้องต้นที่เกี่ยวของกับแฟ้มข้อมูลที่แบ่งประเภทของแฟ้ม สามารถนำไปอธิบายเป็นโครงสร้าง มีรายละเอียดของแต่ละประเภทแฟ้มข้อมูลและชนิดแฟ้มข้อมูลตัวอย่างดังต่อไปนี้



ภาพที่ 7.1 โครงสร้างประเภทของแฟ้มข้อมูล

ความแตกต่างระหว่างระหว่างแฟ้มข้อมูลข้อความและแฟ้มข้อมูลไบนารี คือ แฟ้มข้อมูลแบบข้อความจะมีการเปลี่ยนแปลงข้อมูลก่อนเก็บลงในแฟ้มข้อมูล เช่น เมื่อพบสัญลักษณ์ \n จะทำการเพิ่มอักขระพิเศษ \r เพิ่มขึ้นมาด้านหน้าเป็น \r\n ใช้ในการแบ่งเรคคอร์ด เรียกว่า อักขระขึ้นบรรทัดใหม่ (Carriage-Return Line Feed) หรือ CRLF และทำการเก็บลงในแฟ้มข้อมูล เมื่อมีการอ่านแฟ้มข้อมูลและพบสัญลักษณ์ CRLF จะทำการตัด \r ให้เหลือ \n ส่วนไบนารีนั้นก่อนเก็บข้อมูลลงในแฟ้มข้อมูลจะไม่การเปลี่ยนแปลงข้อมูลเกิดขึ้น (จักรกฤษณ์ แสงแก้ว, 2549)

การจัดการข้อมูลในแฟ้มข้อมูล

การทำงานของคอมพิวเตอร์ ขณะที่โปรแกรมคอมพิวเตอร์ทำงาน ข้อมูลจะถูกเก็บไว้ที่ แรม (RAM : Random Access Memory) ซึ่งสามารถเข้าถึงได้อย่างรวดเร็ว แต่ข้อมูลที่เก็บสามารถสามารถโดนลบทิ้งได้ นั่นคือ เมื่อคอมพิวเตอร์หยุดการทำงานหรือปิดเครื่อง ข้อมูลที่อยู่ในแรมจะหายไป ถ้าหากต้องการเก็บไว้ให้สามารถใช้งานได้ในภายหลังได้ จะต้องเก็บข้อมูลไว้ในอุปกรณ์ที่เมื่อหยุดการทำงานข้อมูลจะไม่หายไป เช่น ฮาร์ดไดร์ฟ (Hard Drive) ยูเอสบีไดร์ฟ (USB Drive)

หรืออุปกรณ์ที่ใช้เก็บข้อมูลต่าง ๆ ข้อมูลจะถูกเก็บโดยมีชื่อและตำแหน่งที่อยู่ของแฟ้มข้อมูลเพื่อระบุการเรียกใช้ผ่านการอ่านและเขียน ขณะที่โปรแกรมกำลังทำงาน (Elkner, 2014)

กระบวนการจัดการแฟ้มข้อมูล ประกอบไปด้วย สามารถจัดการได้ตั้งแต่ขบวนการของการเปิดแฟ้มข้อมูล อ่านแฟ้มข้อมูล เขียนแฟ้มข้อมูล (Fior, 2016) ซึ่งมีรายละเอียดดังต่อไปนี้

1. การเปิดแฟ้มข้อมูล

การเปิดแฟ้มข้อมูล หมายถึง การใช้คำสั่ง สั่งงานให้ระบบปฏิบัติการจองเนื้อที่ในหน่วยความจำให้กับแฟ้มข้อมูล หากไม่มีข้อผิดพลาดในการเปิดแฟ้มข้อมูล ระบบปฏิบัติการจะสร้างส่วนควบคุมแฟ้มข้อมูล ขึ้นมาให้โดยอัตโนมัติ (Sheppard, 2014)

การใช้คำสั่งในการเปิดแฟ้มข้อมูล open() เป็นคำสั่งที่อยู่ในภาษาไพธอน โดยมีรูปแบบการใช้คำสั่งดังนี้

> “ชื่อตัวแปรแฟ้มข้อมูล = open(ชื่อแฟ้มข้อมูล, โหมดการดำเนินการ, หน่วยความจำชั่วคราว)”

การเปิดการดำเนินการแฟ้มข้อมูล มีการใช้สัญลักษณ์ในการบอกประเภทของการอนุญาตดำเนินงานต่อข้อมูลภายในแฟ้มข้อมูล ประกอบด้วยการอ่านข้อมูล การเขียนข้อมูล การอ่านและเขียนข้อมูลพร้อมกัน ดำเนินการกับแฟ้มข้อมูลแบบข้อความและแฟ้มข้อมูลแบบไบนารี ซึ่งมีรายละเอียดดังตารางที่ 7.1

ตารางที่ 7.1 โหมดการดำเนินการและความหมาย

โหมดการดำเนินการ	ความหมาย
a	การดำเนินการเปิดแฟ้มข้อมูลเพื่อเขียน โดยตัวชี้ตำแหน่งจะอยู่สุดท้ายของข้อมูลในแฟ้ม และสามารถทำการเขียนข้อมูลลงตำแหน่งสุดท้ายข้อมูลที่อยู่ในแฟ้ม หากไม่มีแฟ้มข้อมูล จะทำการสร้างขึ้นมาใหม่
a+	การดำเนินการเปิดแฟ้มข้อมูลเพื่ออ่านและเขียน โดยตัวชี้ตำแหน่งจะอยู่สุดท้ายของข้อมูลในแฟ้มข้อมูล หากไม่มีแฟ้มข้อมูล จะทำการสร้างขึ้นมาใหม่

ตารางที่ 7.1 โหมดการดำเนินการและความหมาย (ต่อ)

โหมดการดำเนินการ	ความหมาย
ab	การดำเนินการเปิดแฟ้มข้อมูลที่เป็นประเภทไบนารีเพื่อเขียน โดยตัวชี้ตำแหน่งจะอยู่สุดท้ายของข้อมูลในแฟ้ม และสามารถทำการเขียนข้อมูลลงตำแหน่งสุดท้ายข้อมูลที่อยู่ในแฟ้ม หากไม่มีแฟ้มข้อมูล จะทำการสร้างขึ้นมาใหม่
ab+	การดำเนินการเปิดแฟ้มข้อมูลที่เป็นประเภทไบนารีเพื่ออ่านและเขียน โดยตัวชี้ตำแหน่งจะอยู่สุดท้ายของข้อมูลในแฟ้ม และสามารถทำการเขียนข้อมูลลงตำแหน่งสุดท้ายข้อมูลที่อยู่ในแฟ้ม หากไม่มีแฟ้มข้อมูล จะทำการสร้างขึ้นมาใหม่
b	การดำเนินการเปิดแฟ้มข้อมูลเพื่อเขียน โดยไม่สนใจคำสั่งปิดท้าย \n
r	การดำเนินการเปิดแฟ้มข้อมูลเพื่ออ่าน โดยตัวชี้ตำแหน่งจะอยู่แรกสุดของข้อมูลในแฟ้ม ถ้าหากไม่ได้กำหนดโหมดของการดำเนินการ จะทำการในโหมดนี้เป็นค่ามาตรฐาน
r+	การดำเนินการเปิดแฟ้มข้อมูลเพื่ออ่านและเขียน โดยตัวชี้ตำแหน่งจะอยู่แรกสุดของข้อมูลในแฟ้มข้อมูล ข้อมูลเดิมยังคงอยู่
rb	การดำเนินการเปิดแฟ้มข้อมูลเที่เป็นประเภทไบนารีเพื่ออ่าน โดยตัวชี้ตำแหน่งจะอยู่แรกสุดของข้อมูลในแฟ้มข้อมูล
rb+	การดำเนินการเปิดแฟ้มข้อมูลเที่เป็นประเภทไบนารีเพื่ออ่านและเขียน โดยตัวชี้ตำแหน่งจะอยู่แรกสุดของข้อมูลในแฟ้มข้อมูล ข้อมูลเดิมยังคงอยู่
U	การดำเนินการเปิดแฟ้มข้อมูลประเภท ยูนิโค้ด (Unicode) เพื่ออ่าน
w	การดำเนินการเปิดแฟ้มข้อมูลเพื่อเขียน จะเขียนข้อมูลใหม่แทนที่ข้อมูลเดิม หากไม่มีแฟ้มข้อมูล จะทำการสร้างขึ้นมาใหม่
w+	การดำเนินการเปิดแฟ้มข้อมูลเพื่ออ่านและเขียน จะเขียนข้อมูลใหม่แทนที่ข้อมูลเดิม หากไม่มีแฟ้มข้อมูล จะทำการสร้างขึ้นมาใหม่
wb	การดำเนินการเปิดแฟ้มข้อมูลประเภทไบนารีเพื่อเขียน จะเขียนข้อมูลใหม่แทนที่ข้อมูลเดิม หากไม่มีแฟ้มข้อมูล จะทำการสร้างขึ้นมาใหม่
wb+	การดำเนินการเปิดแฟ้มข้อมูลประเภทไบนารีเพื่ออ่านและเขียน จะเขียนข้อมูลใหม่แทนที่ข้อมูลเดิม หากไม่มีแฟ้มข้อมูล จะทำการสร้างขึ้นมาใหม่

ที่มา : สุชาติ คุ้มมณี (2558)

การเข้าถึงแฟ้มข้อมูลนั้น จะต้องกำหนดโหมดการดำเนินการ ซึ่งสัญลักษณ์ของโหมดการดำเนินการจะอนุญาตให้ดำเนินการกับข้อมูลภายในแฟ้มข้อมูล ตามความหมายและขอบเขตการทำงานเท่านั้น

2. การดำเนินการแฟ้มข้อมูล

การดำเนินการแฟ้มข้อมูล เพื่ออธิบายการใช้คำสั่งในการดำเนินการจัดการข้อมูลในแฟ้มข้อมูล ได้ยกตัวอย่างแฟ้มข้อมูล โดยมีชื่อ ตำแหน่งและข้อมูลในแฟ้มข้อมูล เพื่อตรวจสอบผลลัพธ์ที่ได้ของการดำเนินการจัดการของคำสั่ง ดังตัวอย่างที่ 7.1

ตัวอย่างที่ 7.1 แฟ้มข้อมูลประเภทข้อความ

d:\\python_file\\file_test.txt
computer Technology Industrial Technology UBRU

ภาษาไพธอน มีคำสั่งที่ใช้ในการดำเนินการจัดการกับแฟ้มข้อมูลในรูปของการอ่านเขียนและนำคำสั่งที่มี ใช้ประมวลการทำงานของคำสั่งร่วมกับแฟ้มข้อมูลตัวอย่างที่ 7.1 มีรายละเอียดการทำงานดังต่อไปนี้

2.1 การอ่านข้อมูล

การอ่านข้อมูลในแฟ้มข้อมูล คือ คำสั่งและวิธีการที่ใช้ในการเข้าถึงข้อมูล มีรายละเอียดดังต่อไปนี้

2.1.1 การอ่านข้อมูลทั้งหมดด้วยใช้คำสั่ง read() คือ การใช้คำสั่งในการกำหนดขนาดการอ่านข้อมูลในแฟ้มข้อมูล ถ้าหากไม่มีการกำหนดขนาด จะอ่านข้อมูลทั้งหมดในแฟ้มข้อมูล มีรูปแบบของคำสั่งดังต่อไปนี้

```
ตัวแปรแฟ้มข้อมูล.read([ขนาด])
```

จากรูปแบบคำสั่ง read() สามารถนำไปใช้ในการเขียนโปรแกรม สำหรับการอ่านข้อมูลในแฟ้มข้อมูล มีรายละเอียดการทำงานของคำสั่ง ดังโปรแกรมตัวอย่างที่ 7.2

ตัวอย่างที่ 7.2 การใช้คำสั่ง read()

บรรทัดที่	คำสั่ง
1	obj_file1=open("d:\\python_file\\file_test.txt",'r')
2	a=obj_file1.read(10)
3	b=obj_file1.read()
4	print(a)
5	print(b)
6	obj_file1.close()
7	print('-'*20)
8	obj_file2=open("d:\\python_file\\file_test.txt",'r')
9	x=obj_file2.read(10)
10	print(x)
11	obj_file2.close()
12	obj_file2=open("d:\\python_file\\file_test.txt",'r')
13	y=obj_file2.read()
14	print(y)
15	obj_file2.close()
ผลลัพธ์	
computer T echnology Industrial Technology UBRU ---------- computer T computer Technology Industrial Technology UBRU	

การทำงานของโปรแกรมตัวอย่างที่ 7.2 ได้กำหนดตัวแปรแฟ้มข้อมูลประเภทอ่านข้อมูลของแฟ้มข้อมูลได้อย่างเดียว ข้อมูลจะถูกเก็บไว้ที่ obj_file1 ในบรรทัดที่ 1 จากนั้นกำหนดให้ตัวแปร a ทำหน้าที่ในการอ่านข้อมูลผ่านคำสั่ง read() จำนวน 10 ตัวอักษร โปรแกรมจะทำการนำข้อมูลใน 10 ตัวอักษรแรกมาไว้ที่ตัวแปร a ในบรรทัดที่ 2 และตัวแปร b อ่านข้อมูลไม่ได้ระบุขนาดของข้อมูล

ใน obj_file1 ซึ่งถ้าไม่ระบุขนาดของข้อมูล โปรแกรมจะแปลความหมายเบื้องต้น โดยอ่านข้อมูลทั้งหมด ซึ่งโปรแกรมจะนำข้อมูลตัวอักษรทั้งหมดที่เหลือของ obj_file1 มาเก็บไว้ในตัวแปร b ในบรรทัดที่ 3 ของโปรแกรม

การอ่านข้อมูลของตัวแปรแฟ้มข้อมูล เมื่อมีการอ่านด้วยคำสั่ง read() ต่อเนื่องกัน การอ่านแต่ละครั้งข้อมูลจะถูกดึงออกไป ดังนั้นเพื่อไม่ให้ข้อมูลขาดหายไป จะต้องทำการสร้างตัวแปรแฟ้มข้อมูลในการอ่านข้อมูลแต่ละครั้ง ดังบรรทัดที่ 8 ถึง 14

2.1.2 การอ่านข้อมูลทีละบรรทัดด้วยคำสั่ง readline() คือ การอ่านข้อมูลในแฟ้มข้อมูล ครั้งละ 1 บรรทัด โดยถ้ากำหนดจำนวนอักษรที่อ่านในแฟ้มข้อมูล จะอ่านตามขนาดที่ระบุ ถ้าหากไม่มีการกำหนดขนาด จะการอ่านข้อมูลทั้งหมดในบรรทัดนั้น มีรูปแบบของคำสั่งดังต่อไปนี้

```
ตัวแปรแฟ้มข้อมูล.readline(ขนาดตัวอักษร)
```

จากรูปแบบคำสั่ง readline() สามารถนำไปใช้ในการเขียนโปรแกรม สำหรับการอ่านข้อมูลในแฟ้มข้อมูล มีรายละเอียดการทำงานของคำสั่ง ดังโปรแกรมตัวอย่างที่ 7.3

ตัวอย่างที่ 7.3 การอ่านข้อมูลทีละบรรทัดคำสั่ง readline()

บรรทัดที่	คำสั่ง
1	obj_file=open("d:\\python_file\\file_test.txt",'r')
2	a=obj_file.readline()
3	b=obj_file.readline()
4	c=obj_file.readline(3)
5	print(a)
6	print(b)
7	print(c)
8	obj_file.close()
ผลลัพธ์	
computer Technology Industrial Technology UBR	

โปรแกรมตัวอย่างที่ 7.3 เริ่มจากการเปิดแฟ้มข้อมูลและใช้คำสั่ง readline() ในการอ่านข้อมูล ซึ่งการอ่านข้อมูลแต่ละครั้งจะทำการอ่านข้อมูลทีละบรรทัดต่อเนื่องกันไป ถ้าหากมีการระบุตัวเลขในคำสั่ง readline() จะเป็นการระบุจำนวนตัวอักษรที่อ่านในบรรทัดนั้น

2.1.3 การอ่านข้อมูลโดยระบุบรรทัดด้วยคำสั่ง getline() คือ การใช้คำสั่งในโมดูล linecach() ที่มีอยู่ในภาษาโปรแกรมในการอ่านข้อมูลในแฟ้มข้อมูล โดย ซึ่งมีรูปแบบการใช้คำสั่งดังต่อไปนี้

```
ตัวแปรแฟ้มข้อมูล.getline (ขนาดตัวอักษร)
```

จากรูปแบบคำสั่ง getline() สามารถนำไปใช้ในการเขียนโปรแกรม สำหรับการอ่านข้อมูลในแฟ้มข้อมูลโดยระบุบรรทัด มีรายละเอียดการทำงานของคำสั่ง ดังโปรแกรมตัวอย่างที่ 7.4

ตัวอย่างที่ 7.4 การอ่านข้อมูลโดยระบุบรรทัดด้วยคำสั่ง getline()

บรรทัดที่	คำสั่ง
1	from linecache import getline
2	obj_file="d:\\python_file\\file_test.txt"
3	read_file = getline(obj_file, 2)
4	print(read_file)
5	read_file = getline(obj_file, 3)
6	print(read_file)
ผลลัพธ์	
Industrial Technology UBRU	

การเข้าถึงข้อมูลในแฟ้มข้อมูล จะมีความซับซ้อนในการสร้างคำสั่ง ดังนั้น ในภาษาไพธอนจึงได้เตรียมโมดูลเพื่อให้ใช้งาน ดังโปรแกรมตัวอย่างที่ 7.4 นำเข้าฟังก์ชัน getline() จากโมดูล linecache() ทำการอ่านข้อมูลในตัวแปรแฟ้มข้อมูล obj_file() ที่มีข้อมูลในแฟ้มข้อมูลและทำการระบุตำแหน่งของบรรทัดที่อ่านข้อมูล ดังผลลัพธ์ที่ได้โปรแกรมตัวอย่างที่ 7.4

2.1.4 การอ่านข้อมูลทีละข้อความด้วยคำสั่ง split() คือ การอ่านข้อมูลทีละข้อความในแฟ้มข้อมูล โดยสามารถแยกเป็นข้อความในแฟ้มข้อมูล มีรูปแบบดังต่อไปนี้

```
ตัวแปรแฟ้มข้อมูล.split()
```

จากรูปแบบคำสั่ง split() สามารถนำไปใช้ในการเขียนโปรแกรม สำหรับการอ่านข้อมูลทีละข้อความ มีรายละเอียดการทำงานของคำสั่ง ดังโปรแกรมตัวอย่างที่ 7.5

ตัวอย่างที่ 7.5 การอ่านข้อมูลทีละข้อความด้วยคำสั่ง split()

บรรทัดที่	คำสั่ง
1	obj_file=open("d:\\python_file\\file_test.txt",'r')
2	a=obj_file.read()
3	print(a.split())
ผลลัพธ์	
['computer', 'Technology', 'Industrial', 'Technology', 'UBRU']	

โปรแกรมเริ่มต้นการทำงานด้วยการเปิดแฟ้มข้อมูลและทำการสร้างตัวแปร a เก็บข้อมูลที่ได้จากตัวแปรแฟ้มข้อมูล obj_file อ่านข้อมูลภายในแฟ้มข้อมูล จากนั้นให้แสดงข้อมูลที่อยู่ตัวแปร a โดยใช้คำสั่ง split() ทำการตรวจสอบข้อมูลในแฟ้มข้อมูล ซึ่งจะทำการตรวจสอบข้อความและช่องว่างระหว่างคำในข้อมูลที่มี ทำการตัดข้อความพร้อมทั้งทำการดึงข้อมูลออกมาทีละข้อความของข้อมูล ดังผลลัพธ์ที่ได้โปรแกรมตัวอย่างที่ 7.5

2.2 การเขียนข้อมูล

การดำเนินการเปลี่ยนแปลงข้อมูลในแฟ้มข้อมูล แฟ้มข้อมูลจะสามารถจัดการข้อมูลได้จะต้องระบุเงื่อนไขตำแหน่งในการเขียนข้อมูล ดังนั้นจึงต้องประกาศควบคุมกับโหมดการดำเนินการเพื่อบอกตำแหน่งในการเขียนข้อมูล การเขียนข้อมูลมีคำสั่งที่ใช้ในการดำเนินการหลักที่สำคัญดังต่อไปนี้

2.2.1 การเขียนข้อมูลด้วยคำสั่ง write() คือ คำสั่งที่ใช้ในการเขียนข้อมูล โดยจะใช้ควบคู่กับโหมดการดำเนินการ จะต้องอยู่ในโหมดของการเขียนได้ตามเงื่อนไข ซึ่งมีรูปแบบการใช้คำสั่งดังโปรแกรมตัวอย่างที่ 7.6

ตัวอย่างที่ 7.6 การเขียนข้อมูลทีละข้อความด้วยคำสั่ง write()

บรรทัดที่	คำสั่ง
1	obj_file = open("d:\\python_file\\file_test.txt",'a')
2	str1 = '\n----------'
3	str2 = '\nLocal University'
4	obj_file.write(str1)
5	obj_file.write(str2)
6	obj_file.close()
ผลลัพธ์ในแฟ้มข้อมูล file_test.txt	
computer Technology Industrial Technology UBR ---------- Local University	

โปรแกรมตัวอย่างที่ 7.6 มีการเปิดแฟ้มข้อมูลใช้งาน โดยโหมดการดำเนินการอนุญาตให้ทำการเขียนข้อมูลลงตำแหน่งสุดท้ายต่อจากข้อมูลที่มีอยู่ในแฟ้มข้อมูล ใช้สัญลักษณ์ a คือ โหมดในการดำเนินการดังกล่าว เมื่อมีการประมวลผลคำสั่ง โปรแกรมจะดำเนินการเขียนข้อมูลในแฟ้มข้อมูลในตำแหน่งสุดท้ายที่กำหนด เมื่อทำการเขียนเสร็จสิ้นและทำการเปิดแฟ้มข้อมูล file_test.txt จะมีข้อมูลที่เกิดขึ้นตามกำหนดเงื่อนไข

2.2.2 การเขียนข้อมูลด้วยคำสั่ง writeline() คือ คำสั่งในการเขียนข้อมูลที่ถูกเก็บในลักษณะของชุดข้อมูล โดยจะทำการเรียงลำดับตำแหน่งของชุดข้อมูลและนำข้อมูลที่อยู่ในตำแหน่งไปทำการเขียนในแฟ้มข้อมูล ต่อเนื่องไปจนสิ้นสุดชุดข้อมูลสุดท้าย จึงหยุดการทำงาน ดังนี้

ตัวอย่างที่ 7.7 การเขียนข้อมูลชุดด้วยคำสั่ง writelines()

บรรทัดที่	คำสั่ง
1	obj_file = open("d:\\python_file\\file_test.txt",'a')
2	str1 = ['\n----------','\nLocal University','\nResearch']
3	obj_file.writelines(str1)
4	obj_file.close()
ผลลัพธ์ในแฟ้มข้อมูล file_test.txt	

```
computer Technology
Industrial Technology
UBR
----------
Local University
Research
```

การเขียนข้อมูลด้วยคำสั่ง writelines() โดยข้อมูลนั้นถูกเก็บในรูปแบบของลีสต์ ในตัวแปร str1 ซึ่งมีการเก็บข้อมูลเป็นชุดและการเข้าถึงอนุญาตให้เขียนข้อมูลในตำแหน่งสุดท้ายของข้อมูล เมื่อใช้คำสั่ง writelines() โปรแกรมจะทำการดึงข้อมูลแต่ละชุดไปเขียนที่แฟ้มข้อมูล file_test.txt ดังผลลัพธ์ที่ได้โปรแกรมตัวอย่างที่ 7.7

3. การปิดแฟ้มข้อมูล

กระบวนการของการดำเนินการต่อแฟ้มข้อมูล ตั้งแต่การเปิดแฟ้มข้อมูล การจัดการข้อมูลในแฟ้มข้อมูล จะมีการใช้งานเกี่ยวข้องกับหน่วยความจำ โดยจะทำการจองพื้นที่ในหน่วยความจำไว้ใช้งานกับตัวแปรแฟ้มข้อมูลหรือฟังก์ชัน เมื่อเสร็จสิ้นกระบวนการ จะต้องมีการคืนพื้นที่ในหน่วยความจำคืนสู่ระบบ หากไม่คืนจะทำให้ระบบมีพื้นที่ใช้งานในหน่วยความจำมากเกินไป โดยบางส่วนจองไว้ไม่ได้ทำการปิด หลังเสร็จสิ้นการทำงาน ทำให้เกิดประสิทธิภาพในการทำงานของโปรแกรมต่ำ ระบบทำงานช้าลง ดังนั้น เมื่อมีการเปิดใช้งานแฟ้มข้อมูล จะต้องการทำการปิดแฟ้มข้อมูลหรือคืนพื้นที่ในหน่วยความจำสู่ระบบ จะได้นำพื้นที่นั้นไปใช้งานต่อไป การปิดการทำงานของแฟ้มข้อมูล มีคำสั่งและรูปแบบการใช้งาน ดังรายละเอียดต่อไปนี้

```
ตัวแปรแฟ้มข้อมูล.close()
```

จากรูปแบบคำสั่งการเปลี่ยนปิดแฟ้มข้อมูลด้วยคำสั่ง close() สามารถนำไปเขียนโปรแกรม ซึ่งมีรายละเอียดการทำงานของคำสั่ง ดังโปรแกรมตัวอย่างที่ 7.8

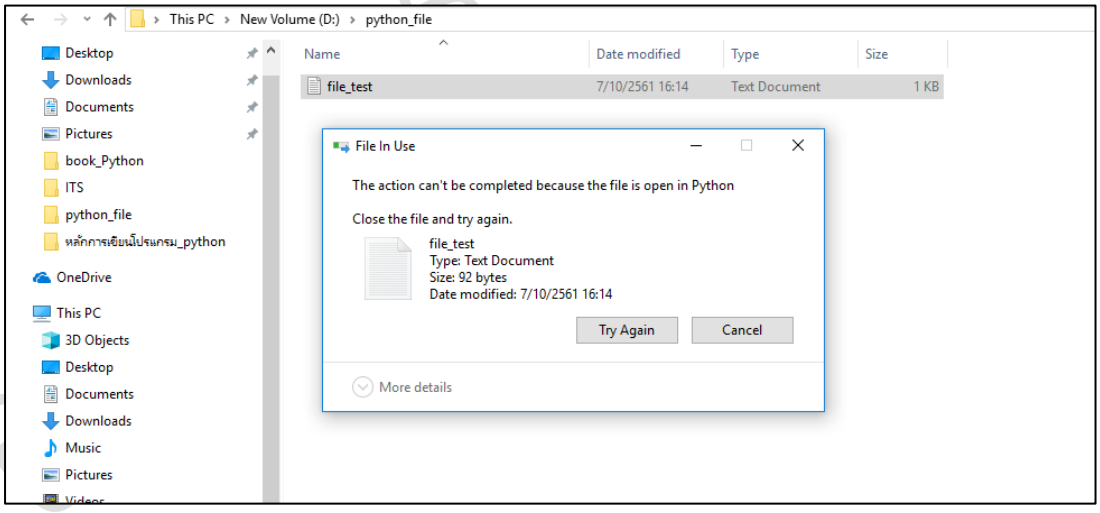
ตัวอย่างที่ 7.8 การปิดแฟ้มข้อมูล

บรรทัดที่	คำสั่ง
1	import os
2	obj_file=open("d:\\python_file\\file_test2.txt",'r')

3	a=obj_file.read()
4	os.remove("d:\\python_file\\file_test2.txt")
ผลลัพธ์	
Traceback (most recent call last): File "C:\Python36\t.py", line 4, in <module> os.remove("d:\\python_file\\file_test2.txt") PermissionError: [WinError 32] The process cannot access the file because it is being used by another process: 'd:\\python_file\\file_test2.txt'	

การเปิดแฟ้มข้อมูลและได้ดำเนินการอ่านข้อมูลจากแฟ้มข้อมูล โดยไม่ได้ทำการปิดและดำเนินการต่อด้วยการลบแฟ้มข้อมูลจากคำสั่ง remove() แฟ้มข้อมูลจะไม่อนุญาตให้ลบและแจ้งให้ทราบถึงสถานะแฟ้มข้อมูลที่ถูกเปิดใช้งานอยู่ จึงไม่สามารถดำเนินการได้ ดังผลลัพธ์ที่ได้โปรแกรมตัวอย่างที่ 7.8

การปิดแฟ้มข้อมูลมีความสำคัญมาก เนื่องจากเมื่อเสร็จสิ้นกระบวนการทำงานและไม่ทำการปิดแฟ้มข้อมูล สถานะของแฟ้มนั้นจะอยู่ในสถานะที่เปิดใช้งาน ผลลัพธ์ที่ได้จะไม่สามารถดำเนินการกับแฟ้มข้อมูลได้ ดังภาพที่ 7.2



ภาพที่ 7.2 ผลจากการเปิดแฟ้มข้อมูลและไม่ได้ทำการปิด

จากภาพที่ 7.2 แฟ้มข้อมูล file_test.txt ได้มีการเรียกใช้ โดยทำการเปิดแฟ้มข้อมูลและไม่ได้ทำการปิดแฟ้มข้อมูล เมื่อต้องการลบแฟ้มข้อมูลนี้ จะไม่สามารถกระทำได้ โปรแกรมจะทำการแจ้งสถานะของแฟ้มข้อมูลอยู่ในสถานะที่กำลังเปิดใช้งานในปัจจุบัน

จากกระบวนการที่กล่าวมาในข้างต้น จะพบว่ากระบวนการทั้ง 3 ขั้นตอน มีความสำคัญต่อการดำเนินการเพื่อการจัดการข้อมูลภายในแฟ้มข้อมูล ซึ่งประกอบไปด้วย การเปิดแฟ้มข้อมูล การจัดการข้อมูลแฟ้มข้อมูลและการปิดแฟ้มข้อมูล ซึ่งหากขาดกระบวนใดกระบวนการหนึ่ง จะส่งกระทบต่อการทำงานของแฟ้มข้อมูลนั้น เช่น ถ้าต้องการจัดการข้อมูลในแฟ้มข้อมูล โดยไม่ได้เปิดแฟ้มก่อน จะไม่สามารถดำเนินการได้ และหากต้องการใช้งานแฟ้มข้อมูลเดิม ที่ได้ถูกใช้ก่อนหน้าและยังไม่ได้ทำการปิดแฟ้มข้อมูล ก็จะไม่สามารถดำเนินการต่อแฟ้มข้อมูลนั้นได้ ดังนั้นการจัดการแฟ้มข้อมูลก่อนใช้งานจะต้องพิจารณาถึงขั้นตอนการดำเนินการ เพื่อไม่ให้เกิดผลกระทบต่อการทำงานของแฟ้มข้อมูลตามวัตถุประสงค์ที่ต้องการ

การจัดการแฟ้มข้อมูลและตำแหน่งที่อยู่

การใช้งานเกี่ยวกับแฟ้มข้อมูลและตำแหน่งที่อยู่มีประโยชน์มาก ซึ่งถูกอ้างอิงไปใช้ในการจัดการภายใต้โปรแกรมที่มีความซับซ้อนและขนาดใหญ่ แฟ้มข้อมูลมีความสำคัญในการจัดการระบบถูกเก็บไว้ในโมดูลแฟ้มข้อมูลของระบบปฏิบัติการ (Sheppard, 2014) นอกเหนือจากการเข้าถึงแฟ้มข้อมูล การจัดการข้อมูลภายในรูปแบบของการอ่านและการเขียน ยังมีการทำงานในส่วนอื่น ที่เกี่ยวข้องกับการจัดการแฟ้มข้อมูลและตำแหน่งที่อยู่ของแฟ้มข้อมูล ซึ่งเรียกว่า “ไดเรกทอรี” (Directory) (สุชาติ คุ้มมณี, 2558) ในภาษาไพธอน ได้มีฟังก์ชันที่อยู่ในโมดูล os ที่ใช้ในการจัดการแฟ้มข้อมูลและไดเรกทอรี ประกอบด้วยคำสั่งพื้นฐาน รายละเอียดของการทำงานและรูปแบบ ของคำสั่งดังต่อไปนี้

ตารางที่ 7.2 คำสั่งที่ใช้ในการจัดการแฟ้มข้อมูลและไดเรกทอรี

คำสั่ง	รายละเอียด	รูปแบบ
chdir()	การเปลี่ยนไดเรกทอรีของแฟ้มข้อมูล	os.chdir (“ไดเรกทอรีและชื่อแฟ้มข้อมูล”)
getcwd()	การแสดงไดเรกทอรีปัจจุบัน	os.getcwd()
mkdir()	การสร้างไดเรกทอรีของแฟ้มข้อมูล	os.mkdir(“ไดเรกทอรีและชื่อแฟ้มข้อมูล”)
rename()	การเปลี่ยนชื่อแฟ้มข้อมูล	os.rename (“ไดเรกทอรีและชื่อแฟ้มข้อมูลเดิม”, “ไดเรกทอรีและชื่อแฟ้มข้อมูลใหม่”)
remove()	การลบแฟ้มข้อมูล	os.remove(“ไดเรกทอรีและชื่อแฟ้มข้อมูล”)

ตารางที่ 7.2 คำสั่งที่ใช้ในการจัดการแฟ้มข้อมูลและไดเรกทอรี (ต่อ)

คำสั่ง	รายละเอียด	รูปแบบ
rmdir()	การลบไดเรกทอรีแฟ้มข้อมูล	os.rmdir("ไดเรกทอรีและ ชื่อแฟ้มข้อมูล")

ที่มา : Python Directory and Files Management (2016)

จากคำสั่งที่ใช้ในการจัดการแฟ้มข้อมูลและไดเรกทอรีพื้นฐาน สามารถนำไปใช้ในทำงานของโปรแกรม ดังรายละเอียดของแต่ละคำสั่งต่อไปนี้

1. การแสดงไดเรกทอรีแฟ้มข้อมูล

การจัดการแฟ้มข้อมูลในรูปแบบต่าง ๆ ก่อนการจัดการแฟ้มข้อมูล อาจจะต้องตรวจสอบไดเรกทอรีปัจจุบันของแฟ้มข้อมูล เพื่อไม่ให้เกิดการผิดพลาดของชุดคำสั่ง ดังนั้นคำสั่งที่ใช้ในการแสดงไดเรกทอรีแฟ้มข้อมูล คือคำสั่ง getcwd() มีรูปแบบการใช้คำสั่งดังต่อไปนี้

```
os.getcwd()
```

จากรูปแบบคำสั่งการเปลี่ยนไดเรกทอรีแฟ้มข้อมูลด้วยคำสั่ง getcwd() สามารถนำไป เขียนโปรแกรม ซึ่งมีรายละเอียดการทำงานของคำสั่ง ดังโปรแกรมตัวอย่างที่ 7.9

ตัวอย่างที่ 7.9 การแสดงไดเรกทอรีแฟ้มข้อมูล

บรรทัดที่	คำสั่ง
1 2	import os print (os.getcwd())
ผลลัพธ์	
C:\Python36	

โปรแกรมตัวอย่างที่ 7.9 แสดงไดเรกทอรีปัจจุบันของแฟ้มข้อมูลด้วยคำสั่ง getcwd() ซึ่งเป็นไดเรกทอรีแฟ้มข้อมูล ที่ได้ทำการเลือกตำแหน่งที่ใช้ในการติดตั้งโปรแกรมในครั้งแรก

2. การเปลี่ยนไดเรกทอรีแฟ้มข้อมูล

แฟ้มข้อมูลในระบบคอมพิวเตอร์ที่ถูกเก็บไว้ในรูปแบบไดเรกทอรีที่แตกต่างกัน ดังนั้น การที่เข้าถึงแต่ละไดเรกทอรีแต่ละตำแหน่งนั้น จะต้องมีการย้ายตำแหน่งการทำงาน คำสั่งที่ใช้ ในการเปลี่ยนไดเรกทอรีแฟ้มข้อมูล คือ chdir() มีรูปแบบการใช้คำสั่งดังต่อไปนี้

```
os.chdir (“ไดเรกทอรีและชื่อแฟ้มข้อมูล”)
```

จากรูปแบบคำสั่งการเปลี่ยนไดเรกทอรีแฟ้มข้อมูล สามารถนำไปเขียนโปรแกรม ซึ่งมีรูปแบบการทำงานของคำสั่ง ดังโปรแกรมตัวอย่างที่ 7.10

ตัวอย่างที่ 7.10 การเปลี่ยนไดเรกทอรีแฟ้มข้อมูล

บรรทัดที่	คำสั่ง
1	import os
2	print (os.getcwd())
3	os.chdir("d:\\python_file")
4	print (os.getcwd())
ผลลัพธ์	
C:\Python36 d:\python_file	

การแสดงไดเรกทอรีปัจจุบัน ใช้คำสั่ง chdir() เป็นที่อยู่เริ่มต้นของโปรแกรมภาษาไพธอน เมื่อทำการเปลี่ยนไดเรกทอรีของแฟ้มข้อมูลด้วยคำสั่ง chdir() โดยให้ไปอยู่ที่ตำแหน่งที่ d:\\python_file และใช้คำสั่ง getcwd() แสดงไดเรกทอรีอีกครั้ง ตำแหน่งจะเปลี่ยนไปดังผลลัพธ์ที่ได้โปรแกรมตัวอย่างที่ 7.10

3. การสร้างไดเรกทอรีแฟ้มข้อมูล

การทำงานเก็บข้อมูลในระบบคอมพิวเตอร์ เพื่อให้มีความเป็นระเบียบในการเก็บข้อมูล จะต้องมีการสร้างไดเรกทอรีในการเก็บข้อมูลที่แตกต่างกัน ดังนั้นการสร้างไดเรกทอรีในภาษา ไพธอน มีคำสั่งที่ใช้ในการสร้างไดเรกทอรีแฟ้มข้อมูล คือ mkdir() มีรูปแบบการใช้คำสั่งดังต่อไปนี้

os.chdir (“ไดเรกทอรีและชื่อแฟ้มข้อมูล”)

จากรูปแบบคำสั่งการสร้างไดเรกทอรีแฟ้มข้อมูล สามารถนำไปเขียนโปรแกรมดังโปรแกรมตัวอย่างที่ 7.11

ตัวอย่างที่ 7.11 การสร้างไดเรกทอรีแฟ้มข้อมูล

บรรทัดที่	คำสั่ง
1	import os
2	os.chdir("d:\\python_file")
3	print (os.getcwd())
4	print(os.listdir())
5	os.mkdir("content3")
6	print(os.listdir())
ผลลัพธ์	
d:\python_file ['content1', 'content2', 'file_test2.txt'] ['content1', 'content2', 'content3', 'file_test2.txt']	

การดำเนินการของโปรแกรม เริ่มต้นด้วยการย้ายไดเรกทอรีแฟ้มข้อมูลไปยัง d:\python_file และให้แสดงข้อมูลที่อยู่ในตำแหน่งด้วยคำสั่ง listdir() ทำการสร้างไดเรกทอรีใหม่ ชื่อ content3 และทำการแสดงข้อมูลไดเรกทอรีอีกครั้ง จะเห็นได้ว่ามีไดเรกทอรี content3 ที่ถูกสร้างขึ้นมาใหม่ภายในแสดงในรูปแบบของลิสต์ ดังผลลัพธ์ที่ได้โปรแกรมตัวอย่างที่ 7.11

4. การเปลี่ยนชื่อแฟ้มข้อมูล

แฟ้มข้อมูลบางรายการ ที่อาจมีความต้องการให้เปลี่ยนจากชื่อเดิมเป็นชื่อใหม่ สามารถกระทำได้ โดยใช้คำสั่งที่ใช้ในการเปลี่ยนชื่อแฟ้มข้อมูล คือ คำสั่ง rename() มีรูปแบบการใช้คำสั่งดังต่อไปนี้

os.rename(“ไดเรกทอรีและชื่อแฟ้มข้อมูลเดิม”,“ไดเรกทอรีและชื่อแฟ้มข้อมูลใหม่”)

จากรูปแบบคำสั่งการเปลี่ยนชื่อแฟ้มข้อมูล rename() สามารถนำไปเขียนโปรแกรม มีรายละเอียดการทำงานของคำสั่ง ดังโปรแกรมตัวอย่างที่ 7.12

ตัวอย่างที่ 7.12 การเปลี่ยนชื่อแฟ้มข้อมูล

บรรทัดที่	คำสั่ง
1	import os
2	os.chdir("d:\\python_file")
3	print (os.getcwd())
4	print(os.listdir())
5	os.rename("file_test.txt","file_test2.txt")
6	print(os.listdir())
ผลลัพธ์	
d:\python_file ['content1', 'content2', 'content3', 'file_test.txt'] ['content1', 'content2', 'content3', 'file_test2.txt']	

การดำเนินการของโปรแกรม เริ่มต้นได้มีการแสดงรายละเอียดที่มีอยู่ในไดเรกทอรีปัจจุบัน และให้มีการเปลี่ยนชื่อแฟ้มข้อมูลข้อมูลจาก file_test2.txt เป็น file_test.txt ด้วยคำสั่ง rename() เมื่อทำการแสดงข้อมูลที่มีอยู่ในไดเรกทอรีปัจจุบันอีกครั้ง จะเห็นได้ว่าแฟ้มข้อมูลได้ถูกเปลี่ยนชื่อจากชื่อเดิมเป็นชื่อใหม่ ดังผลลัพธ์ที่ได้โปรแกรมตัวอย่างที่ 7.12

5. การลบแฟ้มข้อมูล

บางกรณีที่มีแฟ้มข้อมูลที่ไม่ต้องการหรือไม่ได้ใช้งาน คำสั่งที่ใช้ในการลบแฟ้มข้อมูลที่ไม่ต้องการออกจากคอมพิวเตอร์ คือคำสั่ง remove() มีรูปแบบการใช้คำสั่งดังต่อไปนี้

```
os.remove(“ไดเรกทอรีและชื่อแฟ้มข้อมูล”)
```

จากรูปแบบคำสั่งการลบแฟ้มข้อมูล สามารถนำไปเขียนโปรแกรม มีรายละเอียดการทำงานของคำสั่ง ดังโปรแกรมตัวอย่างที่ 7.13

ตัวอย่างที่ 7.13 การลบแฟ้มข้อมูล

บรรทัดที่	คำสั่ง
1	import os
2	os.chdir("d:\\python_file")
3	print (os.getcwd())
4	print(os.listdir())
5	os.remove("file_test2.txt")
6	print(os.listdir())
ผลลัพธ์	
d:\python_file ['content1', 'content2', 'content3', 'file_test.txt', 'file_test2.txt'] ['content1', 'content2', 'content3', 'file_test.txt']	

การดำเนินการของโปรแกรมก่อนการใช้คำสั่งลบ ได้แสดงรายละเอียดในไดเรกทอรีปัจจุบัน และเมื่อได้ใช้คำสั่ง remove() ในการลบ file2_test.txt ออกจากไดเรกทอรีปัจจุบัน จะพบว่า เมื่อแสดงข้อมูลไดเรกทอรีปัจจุบันอีกครั้ง แฟ้มข้อมูลที่ถูกใช้คำสั่งลบออกไปจะไม่แสดง ดังผลลัพธ์ที่ได้โปรแกรมตัวอย่างที่ 7.13

6. การลบไดเรกทอรีแฟ้มข้อมูล

ไดเรกทอรีที่มีอยู่ในระบบ หากไม่ต้องการใช้งานหรือลบออกจากระบบ คำสั่งที่ใช้ในการลบคือ remove() มีรูปแบบการใช้คำสั่งดังต่อไปนี้

```
os.rmdir(“ไดเรกทอรีและชื่อแฟ้มข้อมูล”)
```

จากรูปแบบคำสั่งการลบไดเรกทอรีแฟ้มข้อมูล สามารถนำไปใช้โปรแกรม มีรายละเอียด การทำงานของคำสั่ง ดังโปรแกรมตัวอย่างที่ 7.14

ตัวอย่างที่ 7.14 การลบไดเรกทอรีแฟ้มข้อมูล

บรรทัดที่	คำสั่ง
1	import os

2	os.chdir("d:\\python_file")
3	print (os.getcwd())
4	print(os.listdir())
5	os.rmdir("content3")
6	print(os.listdir())
ผลลัพธ์	
d:\python_file ['content1', 'content2', 'content3', 'file_test.txt'] ['content1', 'content2', 'file_test.txt']	

โปรแกรมตัวอย่างที่ 7.14 ในเบื้องต้นได้แสดงข้อมูลที่มีในไดเรกทอรีปัจจุบันและเมื่อทำการลบไดเรกทอรี content3 ออกจากไดเรกทอรีในปัจจุบัน เมื่อแสดงข้อมูลอีกครั้ง จะพบว่าไม่มีไดเรกทอรีนี้อีก แต่การลบในลักษณะนี้ ไดเรกทอรีที่ต้องการลบ จะต้องไม่มีไดเรกทอรีหรือแฟ้มข้อมูลอยู่ภายใน ซึ่งถ้ามีคำสั่งนี้จะไม่สามารถทำการลบไดเรกทอรีนี้ได้

สรุป

แฟ้มข้อมูล คือ การเก็บข้อมูลในคอมพิวเตอร์ ที่ใช้ชื่อเฉพาะอ้างอิงการเก็บข้อมูล ที่เป็นหน่วยเล็กที่สุดที่ผู้ใช้งานเข้าใจ แฟ้มข้อมูลประกอบด้วย 2 ประเภท ได้แก่ แฟ้มแบบข้อความและแฟ้มไบนารี ซึ่งมีความแตกต่างกันตรงที่แฟ้มข้อความจะมีการเปลี่ยนแปลงรายละเอียดข้อมูลก่อนเก็บลงที่แฟ้มข้อมูลแต่แฟ้มแบบไบนารีจะไม่มีการดำเนินการที่เกี่ยวข้องกับแฟ้มข้อมูล ประกอบไปด้วยการเข้าถึงแฟ้มข้อมูลเพื่อดำเนินการเปลี่ยนแปลงรายละเอียด จะมีฟังก์ชันที่ภาษาไพธอนได้เตรียมไว้ให้ใช้งานเกี่ยวกับแฟ้มข้อมูล ประกอบไปด้วยกระบวนการทำงานหลัก คือ การเปิดแฟ้มข้อมูล การดำเนินการแฟ้มข้อมูลและการปิดแฟ้มข้อมูล ซึ่งการดำเนินการแฟ้มข้อมูลประกอบด้วยการอ่านข้อมูลและการเขียนข้อมูล ซึ่งประกอบด้วยคำสั่งดำเนินการดังนี้ คือ read() ใช้ในการอ่านข้อมูล, readline() ใช้ในการอ่านข้อมูลทีละบรรทัด, getline() การอ่านข้อมูลโดยระบุบรรทัด, write() การเขียนข้อมูล, writeline() การเขียนข้อมูลที่เก็บอยู่ในรูปของชุดข้อมูลและการปิดแฟ้มข้อมูลด้วยคำสั่ง close() นอกเหนือจากนี้ยังมีการจัดการเกี่ยวกับแฟ้มข้อมูลและไดเรกทอรี โดยมีคำสั่งหรือฟังก์ชันมากมายในโมดูล os ซึ่งโมดูลที่อยู่ในภาษาไพธอน คำสั่งที่ใช้ในการจัดการแฟ้มข้อมูลประกอบไปด้วยคำสั่ง chdir() ใช้การเปลี่ยนไดเรกทอรีของแฟ้มข้อมูล, getcwd() ใช้ในการแสดงไดเรกทอรีปัจจุบัน, mkdir() ใช้ในการสร้างไดเรกทอรีของแฟ้มข้อมูล, rename() ใช้ในการเปลี่ยนชื่อแฟ้มข้อมูล, remove() ใช้ในการลบแฟ้มข้อมูลและ rmdir() ใช้ในการลบไดเรกทอรีแฟ้มข้อมูล

แบบฝึกหัด

1. ให้ผู้เรียนอธิบายความหมาย หน้าที่ และหลักการทำงานของแฟ้มข้อมูล

2. ให้ผู้เรียนบอกแฟ้มข้อมูลประกอบมี่ทั้งหมดกี่ประเภท แต่ละประเภทมีลักษณะอย่างไร พร้อมอธิบายความแตกต่างระหว่างประเภทของแฟ้มข้อมูล

3. ให้ผู้เรียนบอกวิธีการจัดการแฟ้มข้อมูลมีอะไรบ้าง พร้อมทั้งอธิบายแต่ละรูปแบบมีวิธีการและมีความสำคัญอย่างไร

4. ให้ผู้เรียนบอกการดำเนินการกับแฟ้มข้อมูลมีทั้งหมดกี่ประเภท และอธิบายแต่ละประเภทมีรูปแบบการใช้คำสั่งอย่างไร

5. ให้ผู้เรียนบอกการเปิดแฟ้มข้อมูลมีทั้งหมดกี่ประเภท และอธิบายแต่ละประเภทมีความแตกต่างกันอย่างไร

6. ให้ผู้เรียนบอกการเขียนแฟ้มข้อมูลมีทั้งหมดกี่ประเภท และอธิบายแต่ละประเภทมีความแตกต่างกันอย่างไร

7. ให้ผู้เรียนบอกการจัดการแฟ้มข้อมูลและตำแหน่งที่อยู่ มีคำสั่งที่ใช้ในการดำเนินการอะไรบ้าง และอธิบายแต่ละคำสั่งมีหน้าที่อย่างไร

8. ให้ผู้เรียนเขียนโปรแกรมสร้างแฟ้มข้อมูลชื่อ user และสร้างข้อมูลดังต่อไปนี้

นายสมหมาย	ใจดี	สาขาเทคโนโลยีคอมพิวเตอร์
นางสาวสุดใจ	การัมย์	สาขาวิทยาการคอมพิวเตอร์
นายวิชิต	สถิตถาวร	สาขาเทคโนโลยีสารสนเทศ

9. ให้ผู้เรียนเขียนโปรแกรมยกตัวอย่างของการแก้ไขข้อมูลในแฟ้มข้อมูล

นายสมหมาย	ใจดี	สาขาวิศวกรรมเครือข่ายคอมพิวเตอร์
นางสาวสุดสวาท	การัมย์	สาขาวิทยาการคอมพิวเตอร์
นายปรัชญา	เจริญเนตร	สาขาเทคโนโลยีสารสนเทศ

10. ให้ผู้เรียนเขียนโปรแกรมยกตัวอย่างการจัดการข้อมูลในแฟ้มข้อมูล โดยเลือกการดำเนินการอย่างใดอย่างหนึ่งกับข้อมูลในแฟ้มข้อมูล

เอกสารอ้างอิง

ราชบัณฑิตยสภา. ศัพท์บัญญัติสำนักงานราชบัณฑิตยสภา. (ออนไลน์) 10 เมษายน 2564 (อ้างเมื่อ 10 เมษายน 2564). จาก: https://coined-word.orst.go.th/

สุชาติ คุ้มมณี. (2558). **เชี่ยวชาญการเรียนด้วยภาษาไพธอน**. มหาสารคาม. คณะวิทยาการสารสนเทศ: มหาวิทยาลัยมหาสารคาม.

จักรกฤษณ์ แสงแก้ว. (2549). **การเขียนโปรแกรมด้วยภาษาไพธอนด้วยตนเอง**. กรุงเทพฯ: สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น).

Lutz, Mark. (2013). **Learning Python**. 5thed. Massachusetts: O'Reilly Media.

Elkner, Jeffrey. (2014). **Practical Programming (in Python)**. 3rded. London: Pearson.

Fior, James M.(2016).**Computer Programming with Python and Multisim**. 3rded. London: Pearson.

Sheppard, Kevin. (2014). **Introduction to Python for Econometrics, Statistics and Data Analysis**. Oxford: University of Oxford.

Python Directory and Files Management. (2016) (online) (cited 15 July 2016). Available from: https://www.programiz.com/python-programming/directory