

File Handling (การจัดการไฟล์)

What Is a File? (ไฟล์คืออะไร)

ไฟล์คือที่เก็บข้อมูลแบบถาวร ซึ่งไฟล์ต่างจากหน่วยความจำ (memory) ตรงที่หน่วยความจำเป็นที่เก็บข้อมูลชั่วคราว เมื่อปิดเครื่องคอมพิวเตอร์แล้วข้อมูลที่เก็บอยู่ในหน่วยความจำก็จะหายไปด้วยทุกครั้ง ต่างจากไฟล์ที่เมื่อทำการเก็บข้อมูลลงไฟล์แล้วข้อมูลจะถูกเก็บอย่างถาวรไม่ว่าจะปิดหรือเปิดเครื่องกี่ครั้งก็ตาม ข้อมูลก็จะยังคงไม่สูญหาย

Working with files (การทำงานกับไฟล์)

Opening and Closing a File in Python (การเปิดและปิดไฟล์ในไพธอน)

เมื่อคุณต้องการทำงานกับไฟล์ สิ่งแรกที่ต้องทำคือเปิดไฟล์ โดยเรียกใช้ `open()` built-in function.

```
In [1]: file = open('dog_breeds.txt')
```

หลังจากเปิดไฟล์แล้ว สิ่งต่อไปที่ต้องเรียนรู้คือวิธีปิดไฟล์ สิ่งสำคัญคือต้องจำไว้ว่าเป็นความรับผิดชอบของคุณในการปิดไฟล์ ในกรณีส่วนใหญ่ เมื่อแอปพลิเคชันหรือสคริปต์สิ้นสุดลง ไฟล์จะถูกปิดในที่สุด อย่างไรก็ตาม ไม่มีการรับประกันว่าจะเกิดขึ้นเมื่อใด ซึ่งอาจนำไปสู่พฤติกรรมที่ไม่พึงประสงค์

```
In [2]: reader = open('dog_breeds.txt')

# Further file processing goes here
# ...

reader.close()
```

Manage files with commands `with` (จัดการไฟล์ด้วยคำสั่ง `with`)

โดยปกติเมื่อเราเรียกฟังก์ชัน `open()` เพื่อเปิดใช้งานไฟล์ เราจะต้องเรียกฟังก์ชัน `close()` เพื่อปิดการใช้งานไฟล์ด้วยทุกครั้งที่เราทำงานกับไฟล์เสร็จ เพราะหากไม่สั่งปิดไฟล์ ทรัพยากรต่างๆ ที่ใช้สำหรับการทำงานกับไฟล์นั้นก็จะไม่ถูกคืนให้กับระบบ

กรณีที่เรากลัวลืมปิดไฟล์ หรือไม่อยากรั่งกังวลว่าจะลืมปิดไฟล์ที่เปิดหรือไม่ เราสามารถนำคำสั่ง `with` มาใช้ในการเปิดไฟล์ได้

```
In [3]: with open('dog_breeds.txt') as reader:
        # Further file processing goes here
```

pass

สำหรับโหมด (mode) การทำงานกับไฟล์แบ่งออกได้ดังนี้:

Character	Meaning
'r'	เปิดไฟล์เพื่ออ่านข้อมูลเท่านั้น
'w'	เปิดไฟล์เพื่อเขียนข้อมูลเท่านั้น
'a'	เปิดไฟล์เพื่อเขียนข้อมูลต่อท้ายไฟล์เดิม (append)

```
with open('abc.txt', 'r')
```

```
with open('abc.txt', 'w')
```

```
with open('abc.txt', 'a')
```

Reading and Writing Opened Files (การอ่านและการเขียนไฟล์ที่เปิดอยู่)

เมื่อคุณเปิดไฟล์ขึ้นมาแล้ว คุณจะต้องการอ่านหรือเขียนไฟล์ ก่อนอื่นมาดูการอ่านไฟล์กันก่อน มีวิธีการต่างๆ ที่สามารถเรียกใช้บน file objects ดังนี้:

Method	What It Does
<code>.read()</code>	ใช้สำหรับการอ่านไฟล์ตามขนาดข้อมูล(byte)ที่ระบุ หากไม่ได้ส่งอาร์กิวเมนต์ใดๆ ระบบจะอ่านไฟล์ทั้งหมด
<code>.readline()</code>	ใช้สำหรับการอ่านไฟล์จากบรรทัดได้ไม่เกินจำนวนอักขระโดยจะอ่านต่อจนจบบรรทัดแล้วจึงวนกลับมาใหม่ หากไม่มีการส่งอาร์กิวเมนต์ก็จะอ่านทั้งบรรทัด (หรือส่วนที่เหลือของบรรทัด)
<code>.readlines()</code>	ใช้สำหรับการอ่านข้อความทุกบรรทัดในไฟล์ ผลลัพธ์ที่ได้เป็นชนิดข้อมูลลิสต์

ต่อไปเราจะลองใช้ไฟล์ `dog_breeds.txt` เพื่อดูตัวอย่างการใช้งานเมธอดเหล่านี้ นี่คือการเปิดและอ่านไฟล์ทั้งหมดโดยใช้ `.read()` :

```
In [4]: with open('dog_breeds.txt', 'r') as reader:
        print(reader.read())
```

```
Pug
Jack Russell Terrier
English Springer Spaniel
German Shepherd
Staffordshire Bull Terrier
Cavalier King Charles Spaniel
Golden Retriever
West Highland White Terrier
Boxer
Border Terrier
```

นี่คือตัวอย่างการอ่านไฟล์ 5 ไบต์ต่อบรรทัดในแต่ละครั้งโดยใช้เมธอด Python `.readline()` :

```
In [5]: with open('dog_breeds.txt', 'r') as reader:
        # Read & print the first 5 characters of the line 5 times
        print(reader.readline(5))
        # Notice that line is greater than the 5 chars and continues
        # down the line, reading 5 chars each time until the end of the
        # line and then "wraps" around
        print(reader.readline(5))
        print(reader.readline(5))
        print(reader.readline(5))
        print(reader.readline(5))
```

Pug

Jack

Russe

ll Te

rrier

นี่คือตัวอย่างการอ่านไฟล์ทั้งหมดเป็นลิสต์โดยใช้เมธอด Python `.readlines()` :

```
In [6]: with open('dog_breeds.txt', 'r') as reader:
        print(reader.readlines())
```

```
['Pug\n', 'Jack Russell Terrier\n', 'English Springer Spaniel\n', 'German Sh
epherd\n', 'Staffordshire Bull Terrier\n', 'Cavalier King Charles Spaniel
\n', 'Golden Retriever\n', 'West Highland White Terrier\n', 'Boxer\n', 'Bord
er Terrier']
```

ตอนนี้เรามาเริ่มเขียนไฟล์กันเลย เช่นเดียวกับการอ่านไฟล์ file objects มีวิธีการต่างๆ ที่เป็นประโยชน์ในการเขียนลงในไฟล์ดังนี้:

Method	What It Does
<code>.write(string)</code>	ใช้สำหรับเขียนข้อมูล(string)ลงในไฟล์
<code>.writelines(list)</code>	ใช้สำหรับเขียนข้อมูลประเภทลิสต์ลงในไฟล์

```
In [7]: with open('dog_breeds.txt', 'r') as reader:
        dog_breeds = reader.readlines()
        # print(dog_breeds)

        with open('dog_breeds_reversed.txt', 'w') as writer:
            for breed in reversed(dog_breeds):
                writer.write(breed)
```

บางครั้งคุณอาจต้องการผนวกไฟล์หรือเริ่มเขียนที่ท้ายไฟล์ที่มีข้อมูลอยู่แล้ว ซึ่งทำได้ง่ายๆ โดยใช้ตัวอักษร 'a' เป็น mode argument:

```
In [8]: with open('dog_breeds.txt', 'a') as a_writer:
        a_writer.write('\nBeagle')
```

เมื่อคุณตรวจสอบ `dog_breeds.txt` อีกครั้ง คุณจะเห็นว่าจุดเริ่มต้นของไฟล์ไม่มีการเปลี่ยนแปลง และตอนนี้ `Beagle` จะถูกเพิ่มไปที่ตอนท้ายของไฟล์

```
In [9]: with open('dog_breeds.txt', 'r') as reader:  
        print(reader.read())
```

```
Pug  
Jack Russell Terrier  
English Springer Spaniel  
German Shepherd  
Staffordshire Bull Terrier  
Cavalier King Charles Spaniel  
Golden Retriever  
West Highland White Terrier  
Boxer  
Border Terrier  
Beagle
```

Your turn!

It is your turn to test how much you understand the ways we work with files in Python.
Try to complete [the exercise](#).

```
In [ ]:
```