



Mahidol University  
*Wisdom of the Land*



# Lecture 14: String

Lecturers:

Aj. Jidapa Kraisangka

Aj. Akara Supratak

Aj. Tipajin Thaipisutikul



สตริง (String) คือ ลำดับของตัวอักษรหลายตัวเรียงต่อกัน

การที่จะประกาศ **string** ค่าของมันจะอยู่ระหว่างเครื่องหมาย **double quote (")** หรือ **single quote (')** เท่านั้น

ตัวอย่าง:

- `'hello'`
- `"hello"`
- `'ผมจะตั้งใจเรียนครับ'`
- `'This is a string value'`



เราสามารถแสดง **string** ได้โดยใช้ฟังก์ชัน **print()**

```
print('hello')  
print("hello")  
print('ผมจะตั้งใจเรียนครับ')  
print('This is a string value')
```

```
hello  
hello  
ผมจะตั้งใจเรียนครับ  
This is a string value
```



การประกาศค่าสตริงในตัวแปร

```
a = "Hello"
```

```
b = 'World'
```

```
c = ''      # string ว่าง (empty string)
```

```
print(a)
```

```
print(b)
```

```
print(c)
```

```
Hello
```

```
World
```



การประกาศค่า **string** ในตัวแปรแบบหลายบรรทัดจะอยู่ระหว่างเครื่องหมาย **double quote 3 ตัว ("")** หรือ **single quote ตัว (')** เท่านั้น

```
a = """ประโยคตัวอย่าง  
ที่กาศค่าใส่ในตัวแปรแบบหลาย  
บรรทัด xD"""  
print(a)
```

```
ประโยคตัวอย่าง  
ที่กาศค่าใส่ในตัวแปรแบบหลาย  
บรรทัด xD
```



การประกาศค่า **string** ในตัวแปรแบบหลายบรรทัดจะอยู่ระหว่างเครื่องหมาย **double quote 3 ตัว ("")** หรือ **single quote ตัว (")** เท่านั้น

```
a = '''ประโยคตัวอย่าง  
ที่กาสค่าใส่ในตัวแปรแบบหลาย  
บรรทัด xD'''  
print(a)
```

```
ประโยคตัวอย่าง  
ที่กาสค่าใส่ในตัวแปรแบบหลาย  
บรรทัด xD
```



เนื่องจาก **String** คือ ลำดับของตัวอักษรหลายตัวเรียงต่อกัน เราจึงสามารถใช้งานได้เหมือนว่าเป็น **list** ของตัวอักษร

## String indexing

```
x = 'foobar'  
print(x[0])  
print(x[3])  
print(x[-1])
```

-6	-5	-4	-3	-2	-1
f	o	o	b	a	r
0	1	2	3	4	5

```
f  
b  
r
```



เนื่องจาก **String** คือ ลำดับของตัวอักษรหลายตัวเรียงต่อกัน เราจึงสามารถใช้งานได้เหมือนว่าเป็น **list** ของตัวอักษร

```
for x in 'ITDS120':  
    print(x)
```

```
I  
T  
D  
S  
1  
2  
0
```





เนื่องจาก **String** คือ ลำดับของตัวอักษรหลายตัวเรียงต่อกัน เราจึงสามารถใช้งานได้เหมือนว่าเป็น **list** ของตัวอักษร

```
for x in 'ITDS120':  
    print(x)
```

```
a = 'อะไรนะ'  
for x in a:  
    print(x)
```

I  
T  
D  
S  
1  
2  
0

อ  
ะ  
ไ  
ร  
น  
ะ



สามารถใช้คำสั่ง `len()` กับ **String** เพื่อหาจำนวนตัวอักษรได้

```
print(len('ITDS120'))
```

7



สามารถใช้คำสั่ง `len()` กับ **String** เพื่อหาจำนวนตัวอักษรได้

```
print(len('ITDS120'))
```

7  
7

```
a = 'ITDS120'  
print(len(a))
```



สามารถใช้คำสั่ง `len()` กับ **String** เพื่อหาจำนวนตัวอักษรได้

```
print(len('ITDS120'))
```

```
a = 'ITDS120'  
print(len(a))
```

```
for i in range(len(a)):  
    print(a[i])
```

```
7  
7  
I  
T  
D  
S  
1  
2  
0
```



เราสามารถใช้คำสั่ง **in** ในการตรวจสอบว่ามีลำดับของตัวอักษรอยู่กับ **string** ได้

```
txt = 'The best things in life are free!'
print('free' in txt)      # True
print('thing' in txt)     # True
print('thins' in txt)     # False

print('h' in 'hello')     # True
print('i' in 'hello')     # False
```



เราสามารถใช้คำสั่ง **in** ในการตรวจสอบว่ามีลำดับของตัวอักษรอยู่กับ **string** ได้

```
txt = 'ผมรักการเรียนรู้ ITDS120'
print("ผม" not in txt)           # False
print('รัก' in txt)             # True
print('เรียนรู้ IT' not in txt)  # False
print("ITSD" in txt)           # False
print('ผมรักการเรียนรู้ ITDS120' in txt)  # True
```



เราสามารถใช้อำสั่ง **in** ในการตรวจสอบว่ามีลำดับของตัวอักษรอยู่กับ **string** ได้

```
txt1 = "อาหารอร่อยและถูก"  
txt2 = "ทำไมร้านนี้ขายแพงจัง"
```

```
ร้านแรกขายไม่แพง  
ร้านสองขายแพง
```

```
if "แพง" not in txt1: print("ร้านแรกขายไม่แพง")  
else: print("ร้านแรกขายแพง")
```

```
if "แพง" not in txt2: print("ร้านสองขายไม่แพง")  
else: print("ร้านสองขายแพง")
```



```
txt = 'ITDS 120 Fundamentals of Programming'
```

ข้อ 1: ตัวแรกของ **txt**

ข้อ 2: ตัวรองสุดท้ายของ **txt**

ข้อ 3: ตรวจสอบว่ามี '**of**' หรือไม่?

ข้อ 4: พิมพ์ที่ละตัวอักษรใน **txt**





การเข้าถึงบางส่วน **string (slicing string)** สามารถทำได้เช่นเดียวกันกับ **list**

**Print** ตัวอักษรที่ตำแหน่ง **1** ถึง **6** (จะไม่รวมตัวอักษรตำแหน่งที่ **6**)

```
txt = 'ITDS120 is fun'  
print(txt[1:6])
```

TDS12



การเข้าถึงบางส่วน **string (slicing string)** สามารถทำได้เช่นเดียวกันกับ **list**

```
txt = 'ITDS120 is fun'  
print(txt[1:6])  
print(txt[1:4])  
print(txt[4:6])
```

```
TDS12  
TDS  
12
```



การเข้าถึงบางส่วน **string (slicing string)** สามารถทำได้เช่นเดียวกันกับ **list**

```
txt = 'ITDS120 is fun'  
print(txt[1:6])  
print(txt[1:4])  
print(txt[4:6])  
print(txt[:2])  
print(txt[5:-1])  
print(txt[-5:])
```

```
TDS12  
TDS  
12  
IT  
20 is fu  
s fun
```



เปลี่ยนเป็น ตัวพิมพ์เล็ก (**lowercase**) (เฉพาะภาษาอังกฤษ)

```
a = "Hello, World! สวัสดีชาวโลก"
```

```
print(a.lower())
```

```
print("AEiuo".lower())
```

```
hello, world! สวัสดีชาวโลก  
aeiuo
```

เปลี่ยนเป็น ตัวพิมพ์ใหญ่ (**uppercase**) (เฉพาะภาษาอังกฤษ)

```
a = "Hello, World! สวัสดีชาวโลก"
```

```
print(a.upper())
```

```
print("AEiuo".upper())
```

```
HELLO, WORLD! สวัสดีชาวโลก  
AEIUO
```



สามารถลบ **whitespace** เช่น เว้นวรรค ก่อนและหลังตัวอักษรใด ๆ ได้โดยให้ฟังก์ชัน `strip()`

```
a = " Hello, World! "  
print(a)  
print(a.strip())
```

```
    Hello, World!  
Hello, World!
```



สามารถแทนที่ตัวอักษรที่กำหนด (ถ้ามี) ได้ด้วยฟังก์ชัน `replace (<old>, <new>)`

```
a = "Hello, World!"  
print(a.replace("H", "J"))  
print(a.replace("Hello", "สวัสดี"))  
print(a.replace("a", "ไม่มี"))
```

```
Jello, World!  
สวัสดี, World!  
Hello, World!
```



สามารถแยก **string** ออกเป็นส่วน ๆ ได้โดยใช้คำสั่ง `split (<val>)`

```
a = "Hello, World!"  
print(a.split(","))  
print(a.split("lo"))
```

```
['Hello', ' World!']  
['Hel', ', World!']
```



สามารถแยก **string** ออกเป็นส่วน ๆ ได้โดยใช้คำสั่ง `split (<val>)`

```
a = "Hello, World!"  
print(a.split(","))  
print(a.split("lo"))
```

```
['Hello', ' World!']  
['Hel', ', World!']
```

```
words = 'This is a sentence'.split(' ')  
print(words)  
print(words[0])  
print(words[1])  
print(len(words))
```

```
['This', 'is', 'a', 'sentence']  
This  
is  
4
```





สามารถนับจำนวนตัวอักษรใน **string** ได้โดยใช้คำสั่ง `count (<val>)`

```
txt = "Hello, World! Hello hello!"  
print(txt.count("e"))  
print(txt.count("Hello"))  
print(txt.count("world!"))  
print(txt.count(" "))
```

```
3  
2  
0  
3
```



ใช้ตรวจสอบ **string** ว่ามีตัวอักษรอะไรอยู่บ้าง

- **s.isalnum()** : return True ถ้าทั้งหมดเป็น **A-Z, a-z** และ **0-9**
- **s.isalpha()** : return True ถ้าทั้งหมดเป็น **A-Z** และ **a-z**
- **s.isdecimal()** : return True ถ้าทั้งหมดเป็น **0-9**
- **s.islower()** : return True ถ้าทั้งหมดเป็นตัวพิมพ์เล็ก
- **s.isupper()** : return True ถ้าทั้งหมดเป็นตัวพิมพ์ใหญ่



สามารถใช้เครื่องหมาย + เพื่อรวม **string** หลายตัวได้เป็น **string** ตัวเดียวได้ (**string concatenation**)

```
a = "Hello"  
b = "World"  
c = a + b  
print(c)  
d = a + " " + b  
print(d)
```

```
HelloWorld  
Hello World
```



สามารถใช้เครื่องหมาย + เพื่อรวม **string** หลายตัวได้เป็น **string** ตัวเดียวได้ (**string concatenation**)

```
txt = "Hello! How are you?"  
print(txt)  
txt = txt[:5] + " Akara" + txt[5:]  
print(txt)
```

```
Hello! How are you?  
Hello Akara! How are you?
```



```
txt = ' ITDS 120 Fundamentals of Programming '
```

ข้อ **1:** ลบ **whitespace** หน้าหลังตัวอักษรใน **txt**

ข้อ **2:** นับว่าใน **txt** มีทั้งหมดกี่คำ

ข้อ **3:** ทำให้ **txt** เป็นตัวพิมพ์ใหญ่ทุกตัวอักษร

ข้อ **4:** ทำให้ 'Fundamentals' เท่านั้นเป็นเป็นตัวพิมพ์ใหญ่

ข้อ **5:** แทน 'Fundamentals' ด้วย 'ขั้นพื้นฐาน'



สามารถแทรกค่าของตัวแปรไปใน **string** ได้ 2 วิธี

วิธีที่ **1** ใช้ { } เพื่อระบุตำแหน่งที่จะใส่ค่า และใส่ค่าของตัวแปรผ่านฟังก์ชัน `format()`

```
age = 36  
txt = "My name is John, I am {} years old"  
print(txt.format(age))
```

```
My name is John, I am 36 years old
```



สามารถแทรกค่าของตัวแปรไปใน **string** ได้ 2 วิธี

วิธีที่ **1** ใช้ { } เพื่อระบุตำแหน่งที่จะใส่ค่า และใส่ค่าของตัวแปรผ่านฟังก์ชัน `format()`

```
age = 36  
name = 'John'  
txt = "My name is {}, I am {} years old"  
print(txt.format(name, age))
```

```
My name is John, I am 36 years old
```



สามารถแทรกค่าของตัวแปรไปใน **string** ได้ 2 วิธี

วิธีที่ **1** ใช้ { } เพื่อระบุตำแหน่งที่จะใส่ค่า และใส่ค่าของตัวแปรผ่านฟังก์ชัน `format()`

```
age = 36  
name = 'John'  
txt = "My name is {}, I am {} years old"  
print(txt.format(name, age))
```

```
My name is John, I am 36 years old
```





สามารถแทรกค่าของตัวแปรไปใน **string** ได้ 2 วิธี

วิธีที่ **2** เติม **f** ไว้ด้านหน้า **string** และใช้ { var } ที่ระบุชื่อของตัวแปรที่จะใส่ค่าลงไป  
ระหว่าง { } ได้เลย

```
age = 36  
name = 'John'  
txt = f"My name is {name}, I am {age} years old"  
print(txt)
```

```
My name is John, I am 36 years old
```



สาเหตุที่เราต้องใช้ **f-string** เนื่องจากเรา ไม่สามารถ รวมค่าที่ไม่ใช่ **string** เข้ากับ **string** โดยตัวได้

```
age = 36  
txt = "My name is John, I am " + age + " years old"  
print(txt)
```

```
Traceback (most recent call last):  
  File "/mnt/d/Workspace/itds120/main.py", line 2, in <module>  
    txt = "My name is John, I am " + age + " years old"  
TypeError: can only concatenate str (not "int") to str
```



สาเหตุที่เราต้องใช้ **f-string** เนื่องจากเรา ไม่สามารถ รวมค่าที่ไม่ใช่ **string** เข้ากับ **string** โดยตัวได้

อีกวิธีที่ใช้ได้นอกจาก **f-string** คือ  
ใช้ฟังก์ชัน **str()** เพื่อเปลี่ยนจากตัวเลขให้เป็น **string**

```
age = 36  
txt = "My name is John, I am " + str(age) + " years old"  
print(txt)
```

```
Traceback (most recent call last):  
  File "/mnt/d/Workspace/itds120/main.py", line 2, in <module>  
    txt = "My name is John, I am " + age + " years old"  
TypeError: can only concatenate str (not "int") to str
```



ในบางโอกาสที่เราจะใส่ตัวอักษรพิเศษไปใน **string** จะไม่สามารถทำได้ถ้าตัวอักษรนั้นได้ถูก  
ภาษา **Python** จองไว้ใช้ในโอกาสอื่นแล้ว เช่น เครื่องหมาย **double quote (")** หรือ  
**single quote (')** ที่ใช้สร้าง **string**

```
txt = "We are the so-called "Vikings" from the north."  
print(txt)
```

```
txt = 'We are the so-called 'Vikings' from the north.'  
print(txt)
```



ปัญหานี้สามารถแก้ไขได้โดยใช้สิ่งที่เรียกว่า **escape character** ซึ่งจะขึ้นต้นด้วย **backslash (\)** แล้วตามด้วยตัวอักษรพิเศษนั้น ๆ

```
txt = "We are the so-called \"Vikings\" from the north."  
print(txt)  
txt = "We are the so-called \nVikings\n from the north."  
print(txt)
```

```
We are the so-called "Vikings" from the north.  
We are the so-called  
Vikings  
from the north.
```



## Escape character ที่พบได้บ่อย ๆ

Code	Result
\'	Single quote
\"	Double quote
\\	Backslash
\n	New Line (ขึ้นบรรทัดใหม่)
\t	Tab



ลบ **whitespace** หน้าหลัง และ ตัวอักษรที่เป็นตัวเลขออกจาก **txt**

```
txt = ' ITDS 120 Fundamentals of Programming '  
new_txt = ''  
for x in txt.strip(' '): #remove whitespace  
    if not x.isdecimal():  
        new_txt = new_txt + x  
print(new_txt)
```

```
ITDS Fundamentals of Programming
```



## นับจำนวนตัวอักษรใน **string**

```
txt = ' ITDS 120 Fundamentals of Programming '  
counter = {}  
for c in txt:  
    if c in counter:  
        counter[c] += 1  
    else:  
        counter[c] = 1  
for k in counter:  
    print(f'{k}: {counter[k]}')
```

```
: 7  
I: 1  
T: 1  
D: 1  
S: 1  
1: 1  
2: 1  
0: 1  
F: 1  
u: 1  
n: 3  
d: 1  
a: 3  
m: 3  
e: 1  
t: 1  
l: 1  
s: 1  
o: 2  
f: 1  
P: 1  
r: 2  
g: 2  
i: 1
```





ยังมีฟังก์ชันที่สามารถเรียกใช้ได้อีกมากมาย ซึ่งสามารถศึกษาเพิ่มเติมผ่านเว็บไซต์ด้านล่างนี้ได้

[https://www.w3schools.com/python/python\\_strings\\_methods.asp](https://www.w3schools.com/python/python_strings_methods.asp)

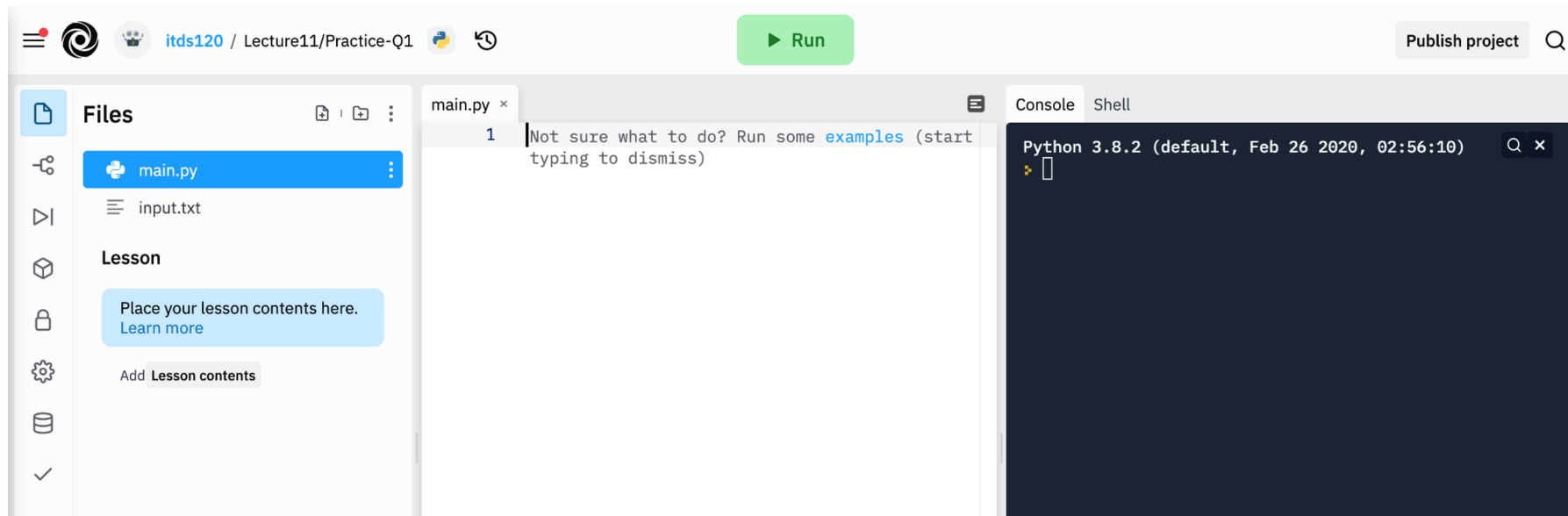
<https://realpython.com/python-strings/>



- ข้อมูลไม่หายเมื่อเราปิดโปรแกรม
  - ไฟล์สำหรับคอมพิวเตอร์เป็นเสมือนที่เก็บข้อมูลของแต่ละโปรแกรม ถ้าไม่มีการบันทึกข้อมูลลงในไฟล์ เวลาที่โปรแกรมสิ้นสุดการทำงานข้อมูลจะหายไป
- ไม่ต้องเสียเวลาใส่ข้อมูลเพื่อเริ่มโปรแกรม
  - กรณีที่โปรแกรมของเราต้องมีการประมวลผลข้อมูลเป็นปริมาณมาก จะเป็นการเสียเวลามากถ้าเราต้องใส่ข้อมูลดังกล่าวด้วยตัวเอง เช่น ผ่าน `input()` ฟังก์ชัน ถ้าเรามีไฟล์ที่เก็บรวบรวมข้อมูลไว้ เราสามารถให้โปรแกรมอ่านข้อมูลจากไฟล์ได้เลย
- ส่งข้อมูลไปมาระหว่างเครื่อง
  - กรณีที่เราต้องส่งข้อมูลไปมาระหว่างเครื่องคอมพิวเตอร์เราสามารถส่งข้อมูลให้กันและกันผ่านไฟล์ได้



สามารถทำตามอาจารย์โดยใช้แบบฝึกหัดตั้งต้นใน **replit** ที่มีชื่อว่า **Lecture14/InclassQ1** ได้





input.txt

≡ input.txt

```
1 Hello world1
2 Hello world2
3 Hello world3 Hello world Hello world Hello world Hello worldHello
4 Hello world4
5 Hello world5
6 |
```

การทำงานร่วมกับไฟล์เราต้องเปิดไฟล์ก่อนทุกครั้ง โดนการเปิดไฟล์เราสามารถระบุโหมดของการเปิดได้

- เปิดเพื่ออ่านไฟล์

```
f = open('input.txt', 'r')
```

- เปิดเพื่อเขียนไฟล์

```
f = open('output.txt', 'w')
```

- เปิดเพื่อเขียนต่อไฟล์ (ถ้าไม่มีไฟล์นั้นอยู่จะทำการสร้างใหม่)

```
f = open('output.txt', 'a')
```

หลังจากจัดการกับไฟล์เสร็จแล้วอย่าลืมปิดไฟล์ด้วยทุกครั้ง

```
f.close()
```



read() : อ่านข้อมูลทั้งไฟล์

```
f = open('input.txt', 'r')  
print(f.read())  
f.close()
```

```
Hello world1  
Hello world2  
Hello world3 Hello world Hello world Hello world Hello worldHello  
worldHello worldHello worldHello worldHello worldHello worldHello  
worldHello worldHello worldHello worldHello worldHello worldHello  
worldHello worldHello worldHello worldHello worldHello worldHello  
worldHello worldHello worldHello worldHello worldHello worldHello  
worldHello worldHello world  
Hello world4  
Hello world5
```



`readline()` : อ่านข้อมูลทั้งไฟล์ทีละแถวแล้วจะเลื่อน cursor ไปยังบรรทัดถัดไป ถ้าต้องการอ่านบรรทัดแรกให้ต้องปิดแล้วเปิดไฟล์ใหม่

```
f = open('input.txt', 'r')
print(f'1:{f.readline()}')
print(f'2:{f.readline()}')
print(f'3:{f.readline()}')
print(f'4:{f.readline()}')
print(f'5:{f.readline()}')
print(f'6:{f.readline()}')
print(f'7:{f.readline()}')
f.close()
```

```
1:Hello world1

2:Hello world2

3:Hello world3 Hello world Hello world Hello world Hello
worldHello worldHello worldHello worldHello worldHello
worldHello worldHello worldHello worldHello worldHello
worldHello worldHello worldHello worldHello worldHello
worldHello worldHello worldHello worldHello worldHello
worldHello worldHello world

4:Hello world4

5:Hello world5

6:
7:
```



`readline()` : อ่านข้อมูลทั้งไฟล์ทีละแถวแล้วจะเลื่อน cursor ไปยังบรรทัดถัดไป ถ้าต้องการอ่านบรรทัดแรกให้ต้องปิดแล้วเปิดไฟล์ใหม่

```
f = open('input.txt', 'r')
print(f'1:{f.readline().strip()}')
print(f'2:{f.readline().strip()}')
print(f'3:{f.readline().strip()}')
print(f'4:{f.readline().strip()}')
print(f'5:{f.readline().strip()}')
print(f'6:{f.readline().strip()}')
print(f'7:{f.readline().strip()}')
f.close()
```

```
1:Hello world1
2:Hello world2
3:Hello world3 Hello world Hello world Hello
world Hello worldHello worldHello worldHello
worldHello worldHello worldHello worldHello
worldHello worldHello worldHello worldHello
worldHello worldHello worldHello worldHello
worldHello worldHello worldHello worldHello
worldHello worldHello worldHello worldHello
world
4:Hello world4
5:Hello world5
6:
7:
```





`readlines()` : อ่านข้อมูลทั้งไฟล์โดยแบ่งเป็น **list** ของบรรทัด

```
f = open('input.txt', 'r')
lines = f.readlines()
print(lines)
f.close()
```

```
['Hello world1\n', 'Hello world2\n', 'Hello world3 Hello
world Hello world Hello world Hello worldHello worldHello
worldHello worldHello worldHello worldHello worldHello
worldHello worldHello worldHello worldHello worldHello
worldHello worldHello worldHello worldHello worldHello
worldHello worldHello worldHello worldHello worldHello
world\n', 'Hello world4\n', 'Hello world5\n']
```



`readlines()` : อ่านข้อมูลทั้งไฟล์โดยแบ่งเป็น **list** ของบรรทัด

```
f = open('input.txt', 'r')
lines = f.readlines()
for l in lines:
    print(l)
f.close()
```

```
Hello world1

Hello world2

Hello world3 Hello world Hello world Hello world Hello
worldHello worldHello worldHello worldHello worldHello
worldHello worldHello worldHello worldHello worldHello
worldHello worldHello worldHello worldHello worldHello
worldHello worldHello worldHello worldHello worldHello
worldHello worldHello world

Hello world4

Hello world5
```



`write()` : เขียน **string** ลงไปบนไฟล์ (ต้องขึ้นบรรทัดใหม่เองด้วย `'\n'`)

```
f = open('output.txt', 'w')
messages = ['m1', 'm2', 'm3', 'm4']
for m in messages:
    f.write(m)
f.close()
```

output.txt

m1m2m3m4



`write()` : เขียน **string** ลงไปบนไฟล์ (ต้องขึ้นบรรทัดใหม่เองด้วย `'\n'`)

```
f = open('output.txt', 'w')
messages = ['m1', 'm2', 'm3', 'm4']
for m in messages:
    f.write(m)
f.close()
```

output.txt

m1m2m3m4

```
f = open('output.txt', 'w')
messages = ['m1', 'm2', 'm3', 'm4']
for m in messages:
    f.write(f'{m}\n')
f.close()
```

output.txt

m1  
m2  
m3  
m4



`write()` : เขียน **string** ลงไปบนไฟล์ (ต้องขึ้นบรรทัดใหม่เองด้วย `'\n'`)

```
f = open('output.txt', 'w')
messages = ['m1', 'm2', 'm3', 'm4']
for m in messages:
    f.write(m)
f.close()
```

output.txt

m1m2m3m4

```
f = open('output.txt', 'w')
messages = ['m1', 'm2', 'm3', 'm4']
for m in messages:
    f.write(f'{m}\n')
f.close()
```

โหมดนี้จะเขียนทับไฟล์เดิม

output.txt

m1  
m2  
m3  
m4



`write()` : เขียน **string** ลงไปบนไฟล์ (ต้องขึ้นบรรทัดใหม่เองด้วย `'\n'`)

```
f = open('output.txt', 'w')
messages = ['m1', 'm2', 'm3', 'm4']
for m in messages:
    f.write(m)
f.close()
```

output.txt

m1m2m3m4

```
f = open('output.txt', 'a')
messages = ['m1', 'm2', 'm3', 'm4']
for m in messages:
    f.write(f'{m}\n')
f.close()
```

โหมดนี้จะเขียนต่อจากบรรทัดสุดท้ายในไฟล์เดิม (จะสร้างใหม่ถ้าไม่เคยมีไฟล์อยู่)

output.txt

m1m2m3m4m1  
m2  
m3  
m4



ถ้าต้องทำงานกับการเขียนด้วยภาษาอื่น ๆ เช่น ภาษาไทย ต้องใส่ **comment** ด้านล่างไว้  
บรรทัดแรกของ **code**

```
# -*- coding: utf-8 -*-
```



```
# -*- coding: utf-8 -*-  
f = open('output.txt', 'w')  
f.write('สร้างไฟล์ใหม่\n')  
f.close()  
  
f = open('output.txt', 'r')  
print(f.read())  
f.close()  
  
f = open('output.txt', 'a')  
f.write('ระวังเวลาใช้โหมด w เพราะจะเขียนทับไฟล์ที่มีอยู่แล้ว\n')  
f.close()  
  
f = open('output.txt', 'r')  
print(f.read())  
f.close()
```

output.txt

สร้างไฟล์ใหม่  
ระวังเวลาใช้โหมด w เพราะจะเขียนทับไฟล์ที่มีอยู่แล้ว





```
# -*- coding: utf-8 -*-  
with open('output.txt', 'w') as f:  
    f.write('สร้างไฟล์ใหม่\n')  
  
with open('output.txt', 'r') as f:  
    print(f.read())  
  
with open('output.txt', 'a') as f:  
    f.write('ระวังเวลาใช้โหมด w เพราะจะเขียนทับไฟล์ที่มีอยู่แล้ว\n')  
  
with open('output.txt', 'r') as f:  
    print(f.read())
```

output.txt

สร้างไฟล์ใหม่  
ระวังเวลาใช้โหมด w เพราะจะเขียนทับไฟล์ที่มีอยู่แล้ว

with จะทำการปิดไฟล์ให้เองโดยอัตโนมัติ  
นิยมใช้ถ้ากลัวลืมใส่ `f.close()`



```
import os  
os.remove('output.txt')
```



```
import os  
os.remove('output.txt')
```

ถ้าไม่มีไฟล์เวลานรัน **code** จะรันไม่ได้ → `FileNotFoundError`

```
Traceback (most recent call last):  
  File "/mnt/d/Workspace/itds120/main.py", line 3, in <module>  
    os.remove('output.txt')  
FileNotFoundError: [Errno 2] No such file or directory:  
'output.txt'
```



```
import os                ตรวจสอบว่ามีไฟล์อยู่หรือเปลื่อก่อนที่จะลบทิ้ง
if os.path.exists('output.txt'):
    os.remove('output.txt')
else:
    print('The file does not exist')
```



Mahidol University  
*Wisdom of the Land*

อ่านเพิ่มเติมเกี่ยวกับไฟล์



[https://www.tutorialspoint.com/python/python\\_files\\_io.htm](https://www.tutorialspoint.com/python/python_files_io.htm)