

บทที่ 5

การโปรแกรมกับสายอักขระใน Python

Chapter 5 String



Asst.Prof.Dr.Nutthapat Kaewrattanapat
Suan Sunandha Rajabhat University

Course: Computer Programming and Developing Applications for Education
Department of Digital Technology for Education
Faculty of Education, Suan Sunandha Rajabhat University

อาจารย์บรรยาย



ผู้ช่วยศาสตราจารย์ ดร.ณัฐภัทร แก้วรัตนภัทร

การศึกษา: มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

- 2565 ปริญญาเอก ปรัชญาดุสิตบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศและการสื่อสารเพื่อการศึกษา (GPA. 4.00)
- 2551 ปริญญาโท วิทยาศาสตรมหาบัณฑิต สาขาวิชาระบบสารสนเทศเพื่อการจัดการ (GPA. 3.58)
- 2549 ปริญญาตรี วิทยาศาสตรบัณฑิต เกียรตินิยมอันดับ 1 สาขาวิชาวิทยาการคอมพิวเตอร์ (GPA. 3.66)

คุณวุฒิวิชาชีพและประกาศนียบัตร

- คุณวุฒิวิชาชีพ จาก สถาบันคุณวุฒิวิชาชีพ (องค์การมหาชน) สาขาวิชาชีพอุตสาหกรรมดิจิทัล สาขาธุรกิจดิจิทัลและพาณิชย์อิเล็กทรอนิกส์ อาชีพนักจัดการเทคโนโลยีสารสนเทศสำหรับธุรกิจ ระดับ 6 เลขที่หนังสือรับรอง PQCN-ICT-ECM-0-251100-B-64/000029
- วิทยาศาสตรข้อมูลด้วยภาษาไพทอน, มหาวิทยาลัยธรรมศาสตร์
- การโปรแกรมสำหรับนักภาษาศาสตร์, มหาวิทยาลัยมหิดล
- การโปรแกรมภาษาไพทอน, มหาวิทยาลัยเพนซิลวาเนีย, สหรัฐอเมริกา
- การโปรแกรมสำหรับทุกคน, มหาวิทยาลัยมิชิแกน, สหรัฐอเมริกา

ติดต่อ: nutthapat.ke@ssru.ac.th

Course Description

หลักการ ทฤษฎี ที่เกี่ยวข้องกับการเขียนโปรแกรมและพัฒนาแอปพลิเคชัน หลักการ พัฒนาโปรแกรมคอมพิวเตอร์ คุณสมบัติของโปรแกรมภาษาชนิดต่าง ๆ หลักการ เบื้องต้นเกี่ยวกับองค์ประกอบ ลักษณะคำสั่ง การเขียนโปรแกรม ขั้นตอนวิธี การ วิเคราะห์ การออกแบบ แอปพลิเคชันเพื่อการศึกษา การประเมินซอฟต์แวร์ สามารถพัฒนาแอปพลิเคชันเพื่อการศึกษา

Principles, theories associated with computer programming and development applications, computer programming principles, computer language, elements of computer language, syntax, computer programming, algorithms, analysis and design application for education, software evaluation, candidate teachers able to developing applications for education.

Algorithm Design

Python Programming Language

System Development Life Cycle: SDLC

Reference: <https://edu.ssru.ac.th/useruploads/files/20230724/1772131ed638786bc8d19918b37249af72c36be4.pdf>

Course Outline

- Chapter 1 – Computer Programming
- Chapter 2 – Introduction to Python Programming
- Chapter 3 – Conditional Statement
- Chapter 4 – Iteration Statement
- **Chapter 5 – Strings**
- Chapter 6 – Lists, Tuples, Sets, Dictionaries
- Chapter 7 – Functions
- Chapter 8 – Object-Oriented Programming: OOP
- Chapter 9 – Python and Database Management System
- Chapter 10 – System Development Life Cycle: SDLC

Measurement and Evaluation

การวัดและประเมินผล

1. ระหว่างการจัดการเรียนรู้

- สอบ Pre-test
- การมอบหมายงาน
- สอบ Post-test
- การมีส่วนร่วมในชั้นเรียน

0%
24%
12%
4%

2. โครงการประจำภาคเรียน (Term Project)

- โครงการและการนำเสนอ

20%

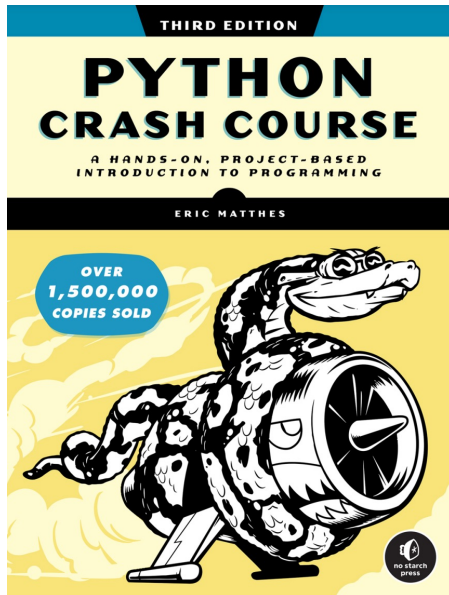
3. การสอบประมวลความรู้ (Comprehensive Examination)

- ปรนัย 70 ข้อ (35 คะแนน) อัตนัย 1 ข้อ (5 คะแนน)

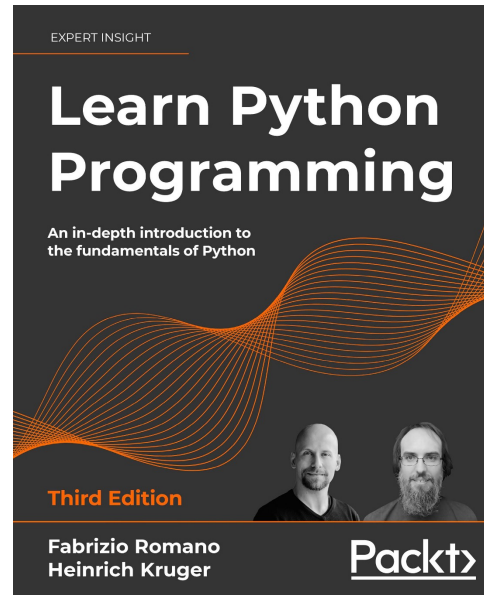
40%

ร้อยละ	ระดับผลการเรียน	ความหมาย
86 – 100	A	ดีเยี่ยม
82 – 85	A-	ดีเยี่ยม
78 – 81	B+	ดีมาก
74 – 77	B	ดี
70 – 73	B-	ค่อนข้างดี
66 – 69	C+	ปานกลางค่อนข้างดี
62 – 65	C	ปานกลาง
58 – 61	C-	ปานกลางค่อนข้างอ่อน
54 – 57	D+	ค่อนข้างอ่อน
50 – 53	D	อ่อน
46 – 49	D-	อ่อนมาก
0 – 45	F	ตก

Learning Materials – References



Python Crash Course: A Hands-On, Project-Based Introduction to Programming [3 ed.]



Learn Python Programming
An in-depth introduction to the fundamentals of Python [3 ed.]

Website

- <https://www.w3schools.com/python/>
- <https://realpython.com/>
- <https://docs.python.org/3/tutorial/index.html>
- <https://pythontutor.com/>

การโปรแกรมสายอักขระ (String)

การโปรแกรมสายอักขระ (String) ในภาษา Python คือการทำงานกับข้อความหรือตัวอักษรที่เรียงต่อกันในเครื่องหมายคำพูด

ภาษา Python สามารถใช้เครื่องหมาย Single Quotation (') หรือเครื่องหมาย Double Quotation (") ในการนิยามสายอักขระได้ ตัวอย่างสายอักขระในภาษา Python เช่น

```
1. name = 'Nutthapat'
```

```
2. message = "Hello, world!"
```

การโปรแกรมสายอักขระ (String)

```
name = "nutthapat Kaewrattanapat"  
print(name.title())
```

A

nutthapat kaewrattanapat

B

Nutthapat Kaewrattanapat

การโปรแกรมสายอักขระ (String)

```
name = "nutthapat Kaewrattanapat"  
print(name.upper())
```

A

nutthapat kaewrattanapat

B

NUTTHAPAT KAEWRATTANAPAT

การโปรแกรมสายอักขระ (String)

```
name = "nutthapat Kaewrattanapat"  
print(name.lower())
```

A

nutthapat kaewrattanapat

B

NUTTHAPAT KAEWRATTANAPAT

การโปรแกรมสายอักขระ (String)

```
name = "nutthapat Kaewrattanapat"  
print("\t", name)
```

A
nutthapat kaewrattanapat

B
\t nutthapat kaewrattanapat

การโปรแกรมสายอักขระ (String)

```
name = "Nutthapat\nKaewrattanapat"  
print(name)
```

A
nutthapat
kaewrattanapat

B
nutthapat
kaewrattanapat

การโปรแกรมสายอักขระ (String)

```
name = "Nutthapat\nKaewrattanapat"  
print(name.title())
```

A
Nutthapat
Kaewrattanapat

B
nutthapat
kaewrattanapat

การเชื่อมต่อคำและตัวแปร (String Concatenation)

ในภาษา Python, การรวม (Concatenate) สตริง (String) มีหลายวิธี

วิธีที่ 1

```
1. str1 = "Hello, "  
2. str2 = "world!"  
3. result = str1 + str2  
4. print(result)    # ผลลัพธ์คือ "Hello, world!"
```

วิธีที่ 2

```
1. str1 = "Hello, "  
2. str2 = "world!"  
3. result = f"{str1}{str2}"  
4. print(result)    # ผลลัพธ์คือ "Hello, world!"
```

การเชื่อมต่อคำและตัวแปร (String Concatenation)

ในภาษา Python, การรวม (Concatenate) สตริง (String) มีหลายวิธี

วิธีที่ 3

```
1. strings_list = ["Hello", "world", "!"]  
2. result = " ".join(strings_list)  
3. print(result)    # ผลลัพธ์คือ "Hello world !"
```

วิธีที่ 4

```
1. str1 = "Hello, "  
2. str2 = "world!"  
3. print("%s %d" %(str1, str2))
```

การเชื่อมต่อคำและตัวแปร (String Concatenation)

ในภาษา Python, การรวม (Concatenate) สตริง (String) มีหลายวิธี

วิธีที่ 5

```
1. name = "John"  
2. age = 30  
3. message = "My name is {} and I am {} years old.".format(name, age)
```

การแปลงประเภทข้อมูลอื่นเป็น String (Casting)

```
1. number = 42
```

```
2. string_number = str(number)
```

การหลบหลีกเพื่อใช้อักขระพิเศษ (Escape Characters)

```
1. my_string = "She said, \"Hello!\""
2. print(my_string)
```

การโปรแกรมสายอักขระ (String)

สายอักขระในภาษา Python เป็นรายการลำดับของอักขระ โดยสามารถเข้าถึงอักขระทีละตัวได้โดยใช้ตัวดำเนินการวงเล็บสี่เหลี่ยม (bracket) เช่น

Strings: สายอักขระ

```
myText = "I am Thai"
```

Arrays: แถวข้อมูล

[0]
I

[1]

[2]
a

[3]
m

[4]

[5]
T

[6]
h

[7]
a

[8]
i

การโปรแกรมสายอักขระ (String)

สายอักขระในภาษา Python เป็นรายการลำดับของอักขระ โดยสามารถเข้าถึงอักขระทีละตัวได้โดยใช้ตัวดำเนินการวงเล็บสี่เหลี่ยม (bracket) เช่น

```
1. university = 'Suan Sunandha'
2. 1_text = university[0]
3. print(1_text)
```

S

การโปรแกรมสายอักขระ (String)

```
myText = "เพราะฉันนั้นต้องการมี เลี้ยวหน้าที่ที่ยิ่งใหญ่"  
print (myText[6])
```

A
u

B
ด

วนลูปเพื่อแสดงทุกๆ อักขระออกมา

```
myText = "เพราะฉันนั้นต้องการมี เลี้ยวหน้าที่ที่ยิ่งใหญ่"
```

```
for character in myText:  
    print(character)
```

วนลูปเพื่อแสดงทุกๆ อักขระออกมา

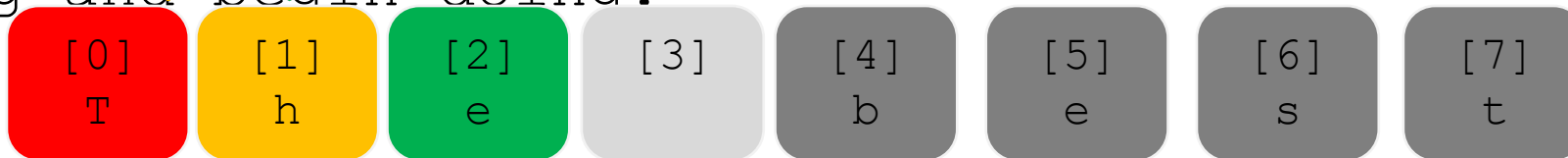
```
myText = "เพราะฉันนั้นต้องการมี เลี้ยวหน้าที่ที่ยิ่งใหญ่"
```

```
for character in myText:  
    if character == "น"  
        print(character)
```

การโปรแกรมสายอักขระ (String)

สามารถเข้าถึงช่วงของอักขระในสายอักขระได้โดยใช้ตัวดำเนินการช่วง (slice operator)

```
myText = "The best way to get started is to quit  
talking and begin doing."
```



```
print (myText[0:3])
```

A
The

B
The b

การโปรแกรมสายอักขระ (String)

สามารถเข้าถึงช่วงของอักขระในสายอักขระได้โดยใช้ตัวดำเนินการช่วง (slice operator) เช่น

```
1. university = 'Suan Sunandha'
```

```
2. text = university[0:4]
```

```
3. print(text)
```

Suan

การโปรแกรมสายอักขระ (String)

สถานการณ์: ต้องการเข้าถึงคำว่า Sunandha ต้อง Slicing อย่างไร?

```
1. university = 'Suan Sunandha'
```

```
2. text = university[_____]
```

```
3. print(text)
```

Sunandha

การโปรแกรมสายอักขระ (String)

สถานการณ์: ต้องการเข้าถึงคำว่า Sunandha ต้อง Slicing อย่างไร?

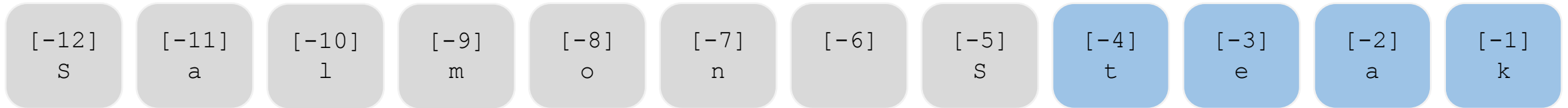
```
1. university = 'Suan Sunandha'
```

```
2. text = university[5:]
```

```
3. print(text)
```

Sunandha

การโปรแกรมสายอักขระ (String)



```
myText = "Salmon Steak"  
print(myText[-4:])
```



teak

การโปรแกรมสายอักขระ (String)



```
myText = "Salmon Steak"  
print(myText[:-4])
```

→ ?

ตรวจสอบการมีของคำในสายอักขระ

```
1. myText = "เพราะฉันนั้นต้องการมี เลี้ยวหน้าที่ที่ยิ่งใหญ่"  
2. print("ฉัน" in myText)
```

A
False

B
True

การค้นหาและแทนที่ข้อความ (find และ replace)

- `str.find()` ใช้หาตำแหน่งแรกที่พบข้อความในสายอักขระ
- `str.replace()` ใช้แทนที่ข้อความในสายอักขระด้วยข้อความใหม่

```
1. my_string = "Hello, World!"  
2. position = my_string.find("World")  
3. # ผลลัพธ์คือ 7  
4. new_string = my_string.replace("World", "Python")  
5. # ผลลัพธ์คือ "Hello, Python!"
```

การเปรียบเทียบสายอักขระ (String Comparison)

```
1. str1 = "apple"  
2. str2 = "banana"  
3. result = str1 < str2  
4. # ผลลัพธ์คือ True (เปรียบเทียบตามลำดับอักขระ)
```

การกำหนดความยาวของสายอักขระ (String Padding)

สามารถใช้ `str.ljust()`, `str.rjust()`, หรือ `str.center()` เพื่อจัดวางสายอักขระให้อยู่ในตำแหน่งที่ต้องการในรูปแบบที่กำหนด

```
1. text = "Python"
2. padded_left = text.ljust(10, "-")
3. # ผลลัพธ์คือ "Python----"

4. padded_right = text.rjust(10, "*")
5. # ผลลัพธ์คือ "****Python"

6. centered = text.center(10, "~")
7. # ผลลัพธ์คือ "~Python~~~"
```

ให้สร้างตัวแปร text เพื่อเก็บข้อความยาว 200 คำ และสร้างตัวแปร words เพื่อรับค่าจากผู้ใช้ จากนั้นให้ค้นหาว่ามีคำ words นั้น ในข้อความที่เก็บไว้ใน text หรือไม่?

```
text = "hello, world..."  
words = input("ป้อนคำที่ต้องการ: ") # สมมติป้อน hello
```

แสดงผล "มีคำว่า hello ในข้อความ"

ให้สร้างตัวแปร text เพื่อเก็บข้อความยาว 200 คำ และสร้างตัวแปร words เพื่อรับค่าจากผู้ใช้ จากนั้นให้แทนที่ words นั้นๆ ทุกตัวที่มีใน text ด้วย "*****"

```
text = "hello, world..."  
words = input("ป้อนคำที่ต้องการ: ") # สมมติป้อน hello  
  
แสดงผล "*****, world..."
```