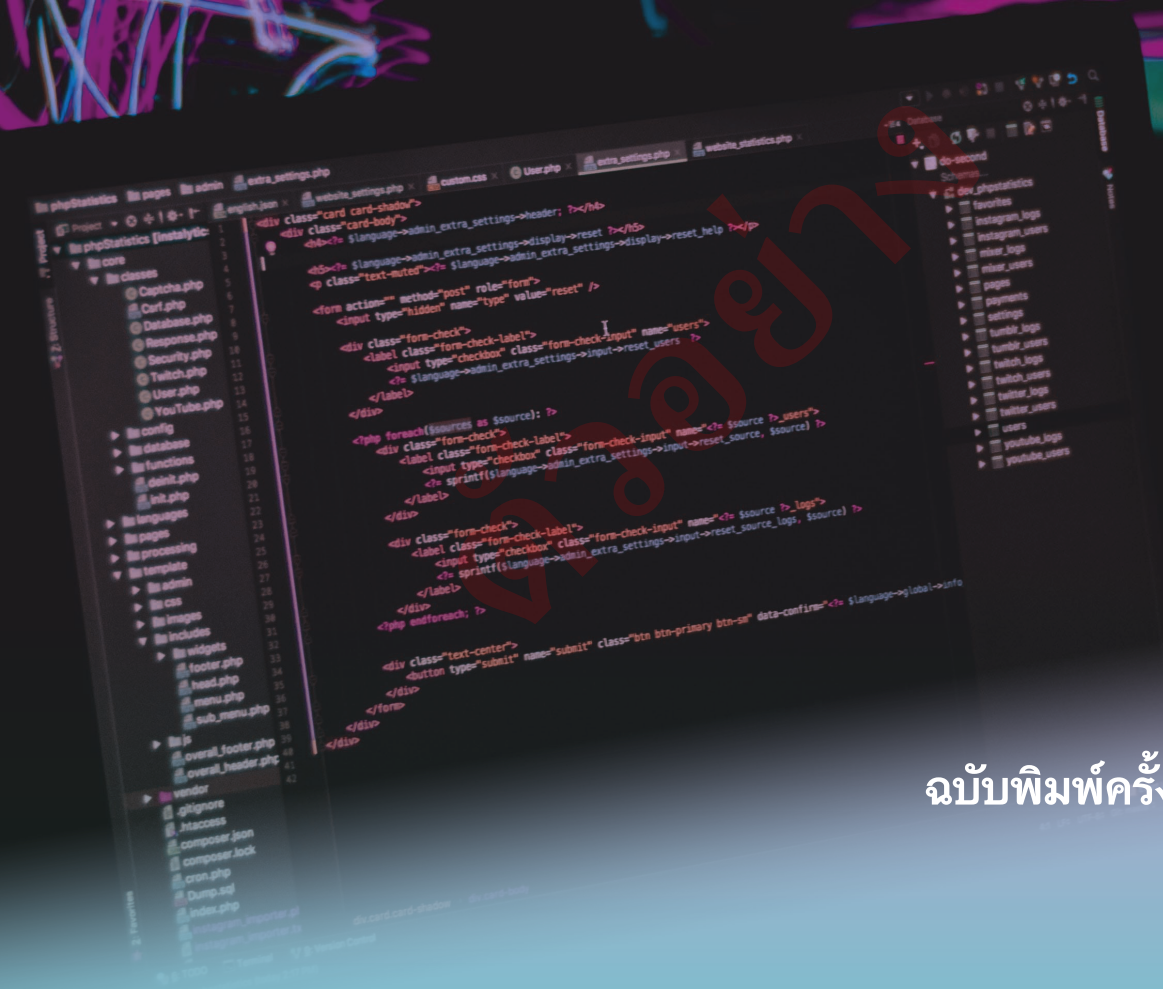
สำนักพิมพ์มหาวิทยาลัยธรรมศาสตร์

การเขียนโปรแกรมภาษา Python เบื้องต้น



ฉบับพิมพ์ครั้งที่ 2

ดร.ปิยะ เตชะธีราวัฒน์

สารบัญ

คำนำ **(9)**

บทที่ 1 แนะนำภาษาไพธอน **1**

1. เริ่มต้นการเขียนโปรแกรม 1
   - 1.1 IDLE (Python GUI) 1
   - 1.2 คำสั่งสคริปต์ 2
2. รูปแบบโปรแกรมภาษา Python 3
   - 2.1 Module 4
   - 2.2 Comment 4
   - 2.3 Statement 5
   - 2.4 Indentation and while space 5
   - 2.5 Literals 5
   - 2.6 Expressions 6
   - 2.7 Keywords 7
3. ตัวแปรและชนิดข้อมูลในภาษา Python 7
   - 3.1 ตัวแปร (Variable) 7
   - 3.2 Numbers 8
   - 3.3 Strings 9
   - 3.4 Lists 9
4. การรับค่าและการแสดงผล 11
   - 4.1 การแสดงผลด้วยฟังก์ชัน print() 11
   - 4.2 การรับค่าด้วยฟังก์ชัน input() 12

คำถามท้ายบท 14

บทที่ 2 ตัวดำเนินการในภาษาไพธอน **15**

1. Assignment operator 15
2. Arithmetic operators 16
3. Comparison operators 17

ตัวอย่าง

4. Logical operators 20
5. Bitwise Operators 21
6. Sequence Operators 23
7. Truth Value Testing 25
คำถามท้ายบท 26

บทที่ 3 คำสั่งควบคุม 27

1. คำสั่ง if 27
   1.1 รูปแบบการทำงานของคำสั่ง if 28
   1.2 รูปแบบของการเขียนคำสั่ง if 28
2. คำสั่ง if-else 29
   2.1 รูปแบบการทำงานของคำสั่ง if else 29
   2.2 รูปแบบของการเขียนคำสั่ง if else 30
3. คำสั่ง if-elif 32
   3.1 รูปแบบการทำงานของคำสั่ง if elif 32
   3.2 รูปแบบของการเขียนคำสั่ง if elif 33
4. คำสั่งวนซ้ำ Loop 36
   4.1 คำสั่งวนซ้ำแบบ While Loop 36
   4.2 คำสั่งวนซ้ำแบบ For Loop 39
5. คำสั่ง Break 43
6. คำสั่ง continue 46

คำถามท้ายบท 48

บทที่ 4 Python Function 49

1. ฟังก์ชัน (function)/โปรแกรมย่อย (subroutine) 50
2. การเรียกใช้ฟังก์ชัน 51
3. พารามิเตอร์ของฟังก์ชัน (parameter) 52
   3.1 Varvaras parameters 54
   3.2 Keyword Argument 55
4. การคืนค่าจากฟังก์ชัน 56

ตัวอย่าง

5. ขอบเขตของตัวแปร (scope of variable) 58
5.1 ตัวแปรเฉพาะที่ (local variable) 58
5.2 ตัวแปรส่วนกลาง (global variable) 59
6. มอดูลและเนมสเปซ 60
6.1 มอดูล (module) 60
6.2 เนมสเปซ (namespace) 61
7. ฟังก์ชันทางคณิตศาสตร์ (math functions) 62
8. แรนดอมนัมเบอร์ (Random number) 63
9. Docstrings 64
คำถามท้ายบท 66

บทที่ 5 Array List 69

1. การประกาศและใช้งาน List 69
2. การเข้าถึงข้อมูลภายใน List 71
2.1 List methods 73
2.2 การใช้งานฟังก์ชันกับ List 77
2.3 การอ่านค่าใน List ด้วยคำสั่ง For loop 78
3. List slicing 80
4. การใช้งานคำสั่ง del เพื่อลบข้อมูลใน List 81
5. การประกาศตัวแปร String 82
5.1 String concatenation 84
5.2 Charterers of string 86
5.3 String methods 90
5.4 Find and search 91
5.5 Text and sentence 92
5.6 Formatting string 93
5.7 Type checking 96
5.8 List and sequence methods 97
6. อาร์เรย์ 1 มิติ 99
7. อาร์เรย์ 2 มิติ 99

ตัวอย่าง

8. อาร์เรย์ 3 มิติ 100
คำถามท้ายบท 102

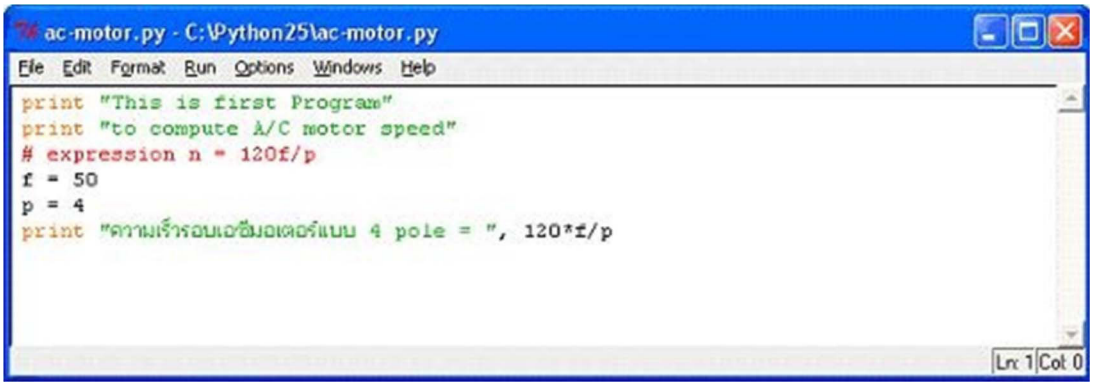
บทที่ 6 การจัดการแฟ้มข้อมูล 105

1. การเปิดแฟ้ม (open file) 106
   • การเปิดไฟล์โดยใช้ฟังก์ชัน open() 106
2. การปิดแฟ้ม 111
3. การใช้ with open() เพื่อทำการเปิดไฟล์ 114
4. การอ่านข้อมูลจากไฟล์ 114
   4.1 การอ่านจากแฟ้มทั้งหมด 115
   4.2 การอ่านข้อมูลจากไฟล์โดยใช้ฟังก์ชัน read() 117
   4.3 การกำหนดตำแหน่งที่เริ่มอ่านข้อมูลจากไฟล์โดยใช้ฟังก์ชัน seek() 118
   4.4 การอ่านข้อมูลออกจากไฟล์บรรทัดต่อบรรทัดโดยใช้ for...loop 120
   4.5 การอ่านแฟ้มแต่ละบรรทัดโดยใช้ฟังก์ชัน readline() 121
   4.6 การอ่านข้อมูลออกจากไฟล์เป็นลิสต์ของข้อมูลแต่ละบรรทัด โดยใช้ฟังก์ชัน readlines() 122
5. การเข้าถึงข้อมูลแต่ละคำ 123
6. การเขียนแฟ้ม 124
7. การเปลี่ยนชื่อแฟ้ม 126
8. การแสดงผลสารบบและแฟ้ม 127
9. การลบแฟ้มและสารบบ 128
10. การค้นหาแฟ้ม 129
11. ฟังก์ชันที่สามารถใช้งานกับไฟล์ 130

คำถามท้ายบท 131

บรรณานุกรม 132

ดัชนี 133



รูปที่ 3 หน้าต่างการเขียนโปรแกรมกับ GUI

2. รูปแบบโปรแกรมภาษา Python

ในการเขียนโปรแกรมไพธอน ผู้เขียนโปรแกรมจะต้องศึกษารูปแบบพื้นฐานและไวยากรณ์ของภาษานั้นๆ โดยรูปแบบพื้นฐานของภาษาจะมีการแบ่งออกเป็นส่วนย่อยๆ ที่เรียกว่าฟังก์ชัน แสดงในตัวอย่างดังนี้

ตัวอย่าง ในการรันไพธอน ใน IDE

ในการสร้างไฟล์ใหม่ผู้เขียนโปรแกรมสามารถทำได้โดยไปที่ File -> New File จะปรากฏกล่อง Text editor ของภาษา Python ขึ้นมา เพื่อรันโปรแกรม Run -> Run Module หรือกด F5 โปรแกรมจะเปลี่ยนกลับไปยัง Python shell และเริ่มต้นทำงาน ดังรูปที่ 4

Partone.py	ผลของการทำงาน
name = input (‘What is your nickname? \n’)	What is your nickname?
print (‘Hi %s.’ % name)	Dear
print (‘Enjoy yourself with python’)	Hi Dear
	Enjoy yourself with python

รูปที่ 4 การเขียนโปรแกรมและผลลัพธ์ของโปรแกรม

ตัวอย่าง

1. ฟังก์ชัน (Function)/โปรแกรมย่อย (Subroutine)

คือกลุ่มคำสั่งที่มีการนิยามชื่อที่ใช้สำหรับอ้างอิงแทนกลุ่มคำสั่งเหล่านั้นเพื่อให้สามารถเรียกทำงานได้ตามต้องการ กลุ่มของคำสั่งที่ประกอบขึ้นเป็นฟังก์ชันจะทำหน้าที่ใดอย่างหนึ่งโดยเฉพาะแม้ว่าจะมีการประเมินคำสั่ง def แต่โค้ดในบล็อกที่ซ้อนกันจะไม่ถูกเรียกใช้งานดังนั้นข้อผิดพลาดจำนวนมากอาจไม่ถูกตรวจพบจนกว่าแต่ละบรรทัดของโค้ดนั้นได้ถูกตรวจสอบ

การสร้างฟังก์ชัน

```
def ชื่อฟังก์ชั้น ():
        กลุ่มของคำสั่ง
```

กลุ่มคำสั่งหรือบล็อกเริ่มต้นด้วยใช้เครื่องหมาย ‘:’ รวมกับการใช้ระยะเยื้องของบล็อก และสิ้นสุดบล็อกเมื่อยุติการใช้ระยะเยื้องนั้น โดยมีข้อแม้ว่าบล็อกเดียวกันต้องมีระยะเยื้องที่เท่ากัน

ตัวอย่าง การสร้างฟังก์ชันแนะนำตัว

```
def show info():
        print(“My name is Somchai Duj”)
        print(“ID 591075xxxx”)
```

บรรทัด 1 สร้างชื่อ ฟังก์ชัน show_info() คือชื่อที่จะเอาใช้ในการเรียกฟังก์ชัน

บรรทัด 2 สร้างข้อความ “My name is Somchai Duj” เมื่อเรียกใช้ฟังก์ชันข้อความนี้จะขึ้นเป็นตัวแรก

บรรทัด 2 สร้างข้อความ “ID 591075xxxx” ข้อความนี้จะขึ้นต่อจากข้อความแรก

บรรทัดแรกของนิยามฟังก์ชันเรียกว่า header ส่วนที่เหลือเรียกว่าร่างกายส่วนหัวจะต้องลงท้ายด้วยโคลอน (:) และ body จะต้องเยื้องโดยการเยื้องเป็นสี่ช่องว่างเสมอ body สามารถมีจำนวนเท่าไรก็ได้

ตัวอย่าง

```
y = 30
area = cal_area(x,y)
print("Area of Rectangle is ",area)
```

6.
```
def printName():
    Name1 = "nut"
    Name2 = "nick"
    Name3 = "nook"
        print("Name : ",name1,name2,name3)
    Name1 = "art"
    Name3 = "gun"
        printName()
        print("Name : ",name1,name3)
```
เมื่อรันโปรแกรม ในบรรทัดที่ 9 ตัวแปร name1 จะมีค่าว่าอะไร

7.
```
def printN(n):
    ...
    Return n
    Def calN(w,h)
    ...
    Return area
    Name input("Enter name : ")
    Number1 = input(input("Enter number : "))
    Number2 = input(input("Enter number : "))
```
จงบอกวิธีการเรียกใช้ฟังก์ชัน calN()

8. จงเขียนโปรแกรมที่สามารถหาค่าที่มากที่สุดใน 3 ตัวเลขที่ใส่ค่าไป เช่น ใส่ค่า 3,6,5 คำตอบที่ได้ต้อง คำตอบ 6

ตัวอย่าง

3. List Slicing

สามารถตัดข้อมูลจาก List หนึ่งแล้วนำไปสร้างเป็น List ใหม่ได้ โดยวิธีดังกล่าวนั้นเรียกว่า slicing ในการตัดข้อมูลใน List นั้นจะทำในรูปแบบ [a:b] เมื่อ a เป็น Index เริ่มต้นและ b เป็น Index ก่อนสมาชิกตัวสุดท้ายที่ต้องการตัด

ตัวอย่าง

```
ch = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h']
a = ch[0:4] # a - d
b = ch[4:9] # e - h
c = ch[:3] # a - c
d = ch[3:] # c - h
e = ch[:] # copy all list, or equivalent to e = ch
f = ch[0:2] + ch[6:8] # a - b and g - h

print(a)
print(b)
print(c)
print(d)
print(e)
print(f)
```

ตัวอย่าง

มีตัวแปร ch ซึ่งมีสมาชิกเป็นตัวอักษรในภาษาอังกฤษจาก a ถึง h หลังจากนั้นได้ทำการ slice ข้อมูลจาก List ดังกล่าว ในตัวแปร a นั้นเป็นการตัดเอาข้อมูลจากตำแหน่งที่ 0 ถึง 3 มา ในตัวแปร b นั้นตัดเอาตำแหน่งที่ 4 ถึง 8 ถัดมาเป็นตัวแปร c d และ e เป็นการเว้นว่างตำแหน่งข้างหน้าและข้างหลัง ซึ่งถ้าตำแหน่งข้างหน้าถูกเว้นว่างไว้ เป็นการตัดเอาสมาชิกตั้งแต่ตำแหน่งเริ่มต้นของ List และถ้าตำแหน่งสิ้นสุดถูกเว้นว่างไว้ เป็นการตัดเอาจนถึงสมาชิกตัวสุดท้ายของ List และในตัวแปร f เป็นการตัดเอาสองส่วนของ List มาต่อกัน ในการนำสอง List มาต่อกันนั้นจะใช้ตัวดำเนินการ +

3. การใช้ with open( ) เพื่อทำการเปิดไฟล์

โดยเมื่อใช้งานเสร็จจะมีการปิดไฟล์ให้อัตโนมัติโดยไม่ต้องเขียนคำสั่งเพิ่มเติม สำหรับการเปิดไฟล์เพื่อใช้งานใน python เรายังสามารถใช้คีย์เวิร์ด with ร่วมด้วย ในการเปิดใช้งานไฟล์ด้วยฟังก์ชัน open() ที่ได้พูดถึงไปข้างต้นแล้วนั้น โดยการกำกับด้วยคีย์เวิร์ด with นั้นเพื่อจะให้มั่นใจว่าทุกครั้งที่เปิดไฟล์ด้วยคำสั่ง open() หลังจากโปรแกรมสิ้นสุดการดำเนินการของชุดคำสั่งที่อยู่ภายในบล็อกดังกล่าวแล้ว จะทำการปิดไฟล์ให้เพื่อคืนทรัพยากรด้วยทันที โดยไม่ต้องเขียนคำสั่งเพื่อเรียกใช้ฟังก์ชัน close() เพิ่มเติม

ตัวอย่าง การใช้งาน with open()

```
# -*- coding: utf-8 -*-
# ใช้งานคีย์เวิร์ด with ร่วมกับฟังก์ชัน open() เพื่อทำการเปิดใช้งานไฟล์ที่ระบุ
# ที่อยู่ใน Folder ปัจจุบันและทำการระบุขอบเขตการใช้งานไฟล์
# เป็น 'r' ซึ่งจะทำให้เพียงแค่สามารถอ่านข้อมูลจากไฟล์ที่ระบุได้
# และทำการระบุ encoding ให้ข้อมูลที่อ่านมาใช้งานในรูปแบบ utf-8
# โดยฟังก์ชันดังกล่าวจะคืนค่ามาเป็น object ชนิดไฟล์และถูกเก็บไว้
# ในตัวแปร "source_file" สำหรับการอ่านข้อมูล ในลำดับต่อไป
# สำหรับการเขียนหรืออ่าน ในลำดับต่อไป
with open("data.csv", 'r', encoding='utf-8') as source_file:     # ทำการอ่านข้อมูลจากไฟล์โดยระบุการ encoding โดยใช้ utf-8
data = source_file.readline()
print(data, end="")                                              # ทำการ print ค่าข้อมูลเพื่อทำการแสดงผล โดยระบุให้ไม่ต้องใส่ อักขระพิเศษสำหรับขึ้นบรรทัดใหม่ "\n" เพิ่มเติมอีก
```

ผลลัพธ์

```
"Location ","Date(yyyy/mm/dd)","Region","Maximum  Temperature(°C) ","Minimum Temperature(°C)"
```

4. การอ่านข้อมูลจากไฟล์

หากเปิดไฟล์เพื่อต้องการเพียงแค่อ่านข้อมูล เราก็ควรระบุขอบเขตสำหรับการอ่านเท่านั้น ‘r’ การกำหนดขอบเขตดังกล่าวช่วยป้องกันการผิดพลาดที่อาจเกิดขึ้นจากการเขียนคำสั่งผิดพลาดได้ด้วย

ตัวอย่าง

การเขียนโปรแกรมภาษา Python เบื้องต้น

หนังสือเล่มนี้อธิบายการเขียนโปรแกรม Python ที่เน้นให้ง่ายและรวดเร็วต่อการเรียนรู้สำหรับผู้ที่มีพื้นฐานในการเขียนโปรแกรมเบื้องต้นและผู้ที่ไม่มีพื้นฐานมาก่อน โดยเน้นการอธิบายและใช้ตัวอย่างที่สามารถเข้าใจได้ง่ายประกอบกับตัวอย่างการเขียนโปรแกรมอย่างกระชับ ทั้งนี้หนังสือเล่มนี้อธิบายและเตรียมพื้นฐานในการประยุกต์ใช้ภาษา Python กับงานด้านต่างๆ เช่น Data Science, Artificial Intelligence และเครือข่ายคอมพิวเตอร์อีกด้วย

ตัวอย่าง

ดร.ปิยะ เตชะธีราวัฒน์

- Ph.D. (Computer Engineering) RMIT University, Australia.
- Bachelor of Engineering (Honours) UNSW University, Australia.

ปัจจุบันดำรงตำแหน่งผู้ช่วยศาสตราจารย์ประจำภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยธรรมศาสตร์

ISBN 978-616-602-071-7
9 786166 020717
ราคา 200 บาท
หมวดคอมพิวเตอร์

http://thammasatpress.com