

การเขียนโปรแกรม

Python

ขั้นพื้นฐาน

ด้านวิทยาศาสตร์สุขภาพ

พลายแก้ว ไชยเบญจวงศ์



สำนักพิมพ์มหาวิทยาลัยนเรศวร
Naresuan University Publishing House
www.nupress.grad.nu.ac.th



สำนักพิมพ์มหาวิทยาลัยนเรศวร Naresuan University Publishing House

บัณฑิตวิทยาลัย มหาวิทยาลัยนเรศวร 99 หมู่ 9 อาคารมหาธรรมราชา ชั้น 1 มหาวิทยาลัยนเรศวร
ตำบลท่าโพธิ์ อำเภอเมือง จังหวัดพิษณุโลก 65000 โทร. 0 5596 8833-8836 E-mail : nuph@nu.ac.th

www.nupress.grad.nu.ac.th สำนักพิมพ์มหาวิทยาลัยนเรศวร @nupress

สงวนลิขสิทธิ์ ตามพระราชบัญญัติลิขสิทธิ์ พ.ศ. 2537 โดยสำนักพิมพ์มหาวิทยาลัยนเรศวร ห้ามทำซ้ำ ดัดแปลง เผยแพร่ต่อสาธารณชนไม่ว่าส่วนใดส่วนหนึ่งของหนังสือเล่มนี้ ไม่ว่าในรูปแบบใด ๆ นอกจากจะได้รับอนุญาตเป็นลายลักษณ์อักษรจากสำนักพิมพ์มหาวิทยาลัยนเรศวร เท่านั้น

ข้อมูลทางบรรณานุกรมของหอสมุดแห่งชาติ

National Library of Thailand Cataloging in Publication Data

หลายแก้ว ไชยเบญจวงศ์.

การเขียนโปรแกรม Python ขั้นพื้นฐานด้านวิทยาศาสตร์สุขภาพ.-- พิษณุโลก : สำนักพิมพ์มหาวิทยาลัยนเรศวร, 2567.

346 หน้า.

1. ไพธอน (ภาษาคอมพิวเตอร์). 2. การเขียนโปรแกรมคอมพิวเตอร์. I. ชื่อเรื่อง

005.133

ISBN 978-616-426-333-8

ISBN (e-book) 978-616-426-334-5

สพน. 133

ราคา 450 บาท

พิมพ์ครั้งแรก กุมภาพันธ์ พ.ศ. 2567

จัดพิมพ์โดย สำนักพิมพ์มหาวิทยาลัยนเรศวร

วางจำหน่ายที่

1. ศูนย์หนังสือแห่งจุฬาลงกรณ์มหาวิทยาลัย
ถนนพญาไท แขวงวังใหม่ เขตปทุมวัน กรุงเทพฯ 10330 โทร. 0 2218 9812
2. ศูนย์หนังสือมหาวิทยาลัยเกษตรศาสตร์
ถนนงามวงศ์วาน แขวงลาดยาว เขตจตุจักร กรุงเทพฯ 10900 โทร. 0 2579 0113
3. ศูนย์หนังสือมหาวิทยาลัยธรรมศาสตร์
ถนนพระจันทร์ แขวงพระบรมมหาราชวัง เขตพระนคร กรุงเทพฯ 10200 โทร. 0 2613 3899
4. สำนักพิมพ์มหาวิทยาลัยนเรศวร
บัณฑิตวิทยาลัย มหาวิทยาลัยนเรศวร อาคารมหาธรรมราชา จังหวัดพิษณุโลก 65000 โทร. 0 5596 8833-8836

ประธานกองบรรณาธิการ รองศาสตราจารย์ ดร.กรองกาญจน์ ชูทิพย์ คณบดีบัณฑิตวิทยาลัย มหาวิทยาลัยนเรศวร

กองบรรณาธิการ รองศาสตราจารย์ ดร.สุชาติ แย้มเม่น • รองศาสตราจารย์สุทัศน์ เขียมวัฒนา • รองศาสตราจารย์ ดร.ศักดิ์ สมกุล •
รองศาสตราจารย์ ดร.เกตุจันทร์ จำปาไชยศรี • ศาสตราจารย์ ดร. พญ.สุรชาติพงษ์ พงษ์เจริญ •
ศาสตราจารย์ ดร. ภญ.กรรณก อังคนันท์ • รองศาสตราจารย์ ดร.นิหระ กิจธิระวัฒน์ • ศาสตราจารย์ ดร.สุธิสา ถ่าน้อย •
รองศาสตราจารย์ ดร.กิตติมา ขาววิชัย • รองศาสตราจารย์ ดร.รุ่งโรจน์ แก้วอุไร • รองศาสตราจารย์ นาวาโท ดร.วัฒน์ชัย หมั่นยิ่ง •
รองศาสตราจารย์ ดร.วิชรพล พุทธิรักษา • รองศาสตราจารย์ ดร.พงศ์พันธ์ กิจสนาโยธิน • ผู้ช่วยศาสตราจารย์ ดร.ยุวรงค์ จันทรวิจิตร •
ผู้ช่วยศาสตราจารย์จรรยาภรณ์ สุวพันธ์ • พัชรีย์ ท้วมใจดี • นวิพรรณ ต้นติพลาผล

ประสานงาน

ภักศิณี เท็ดสิทธิกุล

ฝ่ายขาย/การเงิน

พิมพ์ภรณ์ ดวงลาโรจน์ • วสันต์ มาสวัสดิ์

ออกแบบปก

สรญา แสงเย็นพันธ์

ออกแบบรูปเล่ม

สรญา แสงเย็นพันธ์ และสัณญา จันทา

พิมพ์ที่

บริษัท กู๊ดเอด พรินต์ติ้ง แอนด์ แพคเกจจิ้ง กรุ๊ป จำกัด 6/1 ซอยเสรีไทย 58 แขวงมีนบุรี เขตมีนบุรี กรุงเทพมหานคร 10510



สำนักพิมพ์นี้เป็นสมาชิกสมาคมผู้จัดพิมพ์
และผู้จำหน่ายหนังสือแห่งประเทศไทย
<https://pubat.or.th>



พิมพ์บน
กระดาษคุณภาพ เพื่อผลงานคุณภาพ
กระดาษจากเยื่อกระดาษรีไซเคิล



กรณีต้องการสั่งซื้อหนังสือปริมาณมาก หรือเข้าชั้นเรียนติดต่อฝ่ายจัดจำหน่ายสำนักพิมพ์มหาวิทยาลัยนเรศวร
โทร. 0 5596 8836 Email : nuph@nu.ac.th



คำนำ

ไพธอน จัดเป็นภาษาคอมพิวเตอร์ภาษาหนึ่งที่กำลังได้รับความนิยมอย่างรวดเร็ว ทั้งนี้เนื่องจากความง่ายของโครงสร้างภาษาที่ทำให้ผู้เริ่มเรียนรู้ภาษาคอมพิวเตอร์สามารถเข้าใจหลักการเขียนโปรแกรมและสามารถนำไปสร้างเป็นโปรแกรมอย่างง่ายสำหรับใช้งานเองได้ในเวลาอันสั้น ภาษาไพธอนสามารถนำไปสร้างเป็นโปรแกรมขั้นสูงได้ไม่ว่าจะเป็นโปรแกรมแสดงโครงสร้างโปรตีนแบบ 3D การค้นหาคำสำคัญหรือลำดับเบสของ DNA ที่ต้องการในฐานข้อมูลขนาดใหญ่ หรือแม้กระทั่งการจำลองการทำงานของควอนตัมคอมพิวเตอร์ที่จะมีใช้ในอนาคตอันใกล้ อีกทั้งยังเป็นภาษาที่ผู้พัฒนาโปรแกรมยักษ์ใหญ่อย่าง Google เลือกใช้ ซึ่งทำให้ผู้เริ่มต้นเรียนรู้สามารถนำไปต่อยอดเพื่อพัฒนาตัวเองให้เป็นนักเขียนโปรแกรมมืออาชีพได้ไม่ยาก ไพธอนเป็นภาษาคอมพิวเตอร์แบบ Open source ที่ทุกคนเข้าถึงได้โดยไม่เสียค่าใช้จ่าย มีกลุ่มผู้ใช้งานค่อนข้างใหญ่และมีการพัฒนาตัวเองอยู่ตลอดเวลา เป็นภาษาที่มีความหลากหลายและมีให้เลือกใช้ในเกือบจะทุก ๆ ระบบปฏิบัติการ ทำให้ไพธอนกลายเป็นภาษาคอมพิวเตอร์ที่มีผู้นำไปพัฒนาเป็นโปรแกรมใช้งานในด้านต่าง ๆ มากที่สุดภาษาหนึ่ง

หนังสือหรือเอกสารการเรียนรู้ภาษาไพธอนที่เป็นภาษาไทยส่วนใหญ่มักจะเน้นไปในแนวทางการใช้งานกับฐานข้อมูล การจัดการสินค้า คณิตศาสตร์ และการคำนวณอื่น ๆ ที่ไม่เกี่ยวข้องกับงานทางด้านวิทยาศาสตร์ชีวภาพ หนังสือเล่มนี้จัดทำขึ้นโดยมีวัตถุประสงค์เพื่อใช้เป็นเอกสารประกอบการเรียนการสอนในรายวิชา “การเขียนโปรแกรมไพธอนขั้นพื้นฐานด้านวิทยาศาสตร์สุขภาพ” มีเป้าหมายหลักเพื่อให้หนังสือที่ลงทะเลียนนำมาใช้เป็นเอกสารประกอบการเรียนเพื่อทำความเข้าใจกับบทเรียนให้ลึกซึ้งยิ่งขึ้น โดยมีตัวอย่างโปรแกรมที่เกี่ยวข้องกับการคำนวณด้านวิทยาศาสตร์สุขภาพ จุลชีววิทยา ตลอดจนการค้นหสำหรับงานทางชีวสารสนเทศเบื้องต้น ซึ่งองค์ความรู้ที่ได้รับจะเป็นพื้นฐานที่สำคัญในการพัฒนาไปสู่ความรู้ขั้นสูงทางด้านจุลชีววิทยา วิทยาศาสตร์สุขภาพ การแพทย์ เทคโนโลยีชีวภาพ และชีวสารสนเทศต่อไป อย่างไรก็ตามสำหรับนิสิตนักศึกษาที่ไม่ได้ลงทะเลียนเรียน ตลอดจนนักเรียนและผู้สนใจทั่วไปที่อยากจะเรียนรู้วิธีการเขียนโปรแกรมคอมพิวเตอร์ขั้นพื้นฐานสำหรับการสร้างโปรแกรมใช้งานอย่างง่าย ก็สามารถนำมาอ่านและศึกษาด้วยตนเองเพื่อพัฒนาความรู้ความเข้าใจในเรื่องการเขียนโปรแกรม หรือเพื่อสร้างโปรแกรมที่สามารถประยุกต์ใช้กับสิ่งที่ตนเองสนใจได้ สำหรับผู้อ่านที่สนใจและ



ต้องการศึกษาค้นคว้าเพิ่มเติมสามารถหาอ่านได้จากหนังสือการเรียนรู้ภาษาไพธอนภาคภาษาอังกฤษ “How to Think Like a Computer Scientist : Learning with Python 3 Documentation” และ “Bioinformatics Algorithms : Design and implementation in Python” ซึ่งเป็นเอกสารอ้างอิงหลักและใช้เป็นแนวทางสำหรับการพัฒนาหนังสือเล่มนี้

ผู้ช่วยศาสตราจารย์ ดร.พลายแก้ว ไชยเบญจวงศ์

ภาควิชาจุลชีววิทยาและปรสิตวิทยา

คณะวิทยาศาสตร์การแพทย์ มหาวิทยาลัยนเรศวร

ตัวอย่าง

สารบัญ

บทที่ 1 ภาษาไพธอนและโปรแกรม

ภาษาคอมพิวเตอร์

ภาษาไพธอน

โปรแกรม

การทำงานผิดพลาดของโปรแกรม

บทที่ 2 เริ่มต้นใช้งานไพธอน

โปรแกรมภาษาไพธอน

การติดตั้งโปรแกรมภาษาไพธอน

การใช้งานไพธอนในแบบตอบสนองทันที

ชนิดของข้อมูล

การทดลองป้อนข้อมูล

เครื่องหมายการคำนวณทางคณิตศาสตร์พื้นฐาน

การใช้ไพธอนเป็นเครื่องคิดเลข

การประยุกต์ใช้งาน # 1 : คำนวณช่วงเวลาที่ใช้ย้อมสีแบคทีเรีย

การประยุกต์ใช้งาน # 2 : คำนวณช่วงเวลาในการฆ่าเชื้อจุลินทรีย์

ตัวแปร

การประยุกต์ใช้งาน # 3 : คำนวณปริมาตรสารละลายตั้งต้น

การตั้งชื่อตัวแปร

การใช้เครื่องหมายทางคณิตศาสตร์กับสตริง

ฟังก์ชันการแปลงค่าชนิดตัวแปร

คำสั่งและการประมวลผล

การประยุกต์ใช้งาน # 4 : คำนวณจำนวนจุลินทรีย์ทั้งหมด

สรุปท้ายบท

แบบฝึกหัด

1

2

3

4

5

7

8

8

8

9

10

12

13

14

15

16

19

21

22

22

24

25

26

27



บทที่ 3 เล่นกับโปรแกรม	29
การเขียนโปรแกรม	30
การสร้างไฟล์โปรแกรมและการใช้งานโมดูล turtle	30
เรียนรู้และดัดแปลงโปรแกรม	33
การใช้งานคำสั่งวนซ้ำ คำสั่ง for loop	34
การสร้างรูปภาพชนิดด้วยคำสั่งวนซ้ำ	36
การประยุกต์ใช้งาน # 5 : วาดโครงสร้างกรดอะมิโน phenylalanine	37
การประยุกต์ใช้งาน # 6 : วาดโครงสร้างโมเลกุลฟีนอล	41
สรุปท้ายบท	44
แบบฝึกหัด	45
บทที่ 4 สตริง	47
สตริงคืออะไร	48
โครงสร้างสตริง	48
การระบุตัวอักษรด้วยดัชนี	49
ฟังก์ชัน len()	50
การใช้ไพธอนเป็นเครื่องถอดรหัสอย่างง่าย	51
การตัดคำด้วยเทคนิค slice	54
เทคนิคการก้าวและกระโดด	56
การใช้ไพธอนเป็นเครื่องมือค้นหาคำ	57
ฟังก์ชันอื่น ๆ ในเมทีอดสตริง	59
การประยุกต์ใช้งาน # 7 : หาลำดับเบสสาย complementary DNA	59
การประยุกต์ใช้งาน # 8 : คำนวณค่า GC content	63
การประยุกต์ใช้งาน # 9 : หาลำดับเบส DNA primers จาก 5' => 3'	65
การจัดรูปแบบการพิมพ์	68
สรุปท้ายบท	71
แบบฝึกหัด	72

บทที่ 5 ตัวแปรกลุ่ม Tuple / List / Dictionary	73
การจัดเก็บกลุ่มข้อมูล	74
ตัวแปร Tuple	74
ตัวแปร List	76
การใช้เครื่องหมายทางคณิตศาสตร์กับตัวแปร List	79
การแปลงค่าสตริงให้เป็น List	80
<i>การประยุกต์ใช้งาน # 10 : จัดการวัตถุแบบ FIFO ในระบบ GMP</i>	81
ตัวแปร Dictionary	82
<i>การประยุกต์ใช้งาน # 11 : ถอดรหัส DNA / แปลรหัส RNA</i>	86
สรุปท้ายบท	91
แบบฝึกหัด	92
บทที่ 6 การตรวจสอบเงื่อนไข	93
ข้อมูลประเภทบูลีน	95
เครื่องหมายเปรียบเทียบ	97
การกำหนดเงื่อนไข	98
เครื่องหมายทางตรรกศาสตร์	102
<i>การประยุกต์ใช้งาน # 12 : ประเมินผลตรวจอาหารตามเกณฑ์มาตรฐาน</i>	105
<i>การประยุกต์ใช้งาน # 13 : ตรวจสอบความใช้ได้ของข้อมูลลำดับเบส</i>	109
สรุปท้ายบท	113
แบบฝึกหัด	114
บทที่ 7 คำสั่งวนซ้ำ	117
คำสั่ง for loop กับการกำหนดค่าตัวแปรรูป	119
คำสั่ง for loop กับการสร้างตาราง	123
<i>การประยุกต์ใช้งาน # 14 : จัดการข้อมูลเชื้อที่ดื้อยาปฏิชีวนะหลายชนิด</i>	124
การใช้งานคำสั่งวนซ้ำ คำสั่ง while loop	127
คำสั่ง break และ continue	130
<i>การประยุกต์ใช้งาน # 15 : คำนวณน้ำหนักโมเลกุลของ polypeptide</i>	132
การใช้ไพธอนสร้างงานศิลปะ	134



สรุปท้ายบท	140
แบบฝึกหัด	141

บทที่ 8 ฟังก์ชัน 143

ฟังก์ชันคืออะไร	144
การสร้างและเรียกใช้ฟังก์ชัน	144
ตัวแปร local และ global	147
ฟังก์ชันภายในฟังก์ชัน	148
การประยุกต์ใช้งาน # 16 : สร้างเมนูสำหรับประมวลผล DNA/RNA	150
ฟังก์ชันในรูปแบบโมดูล	157
การประยุกต์ใช้งาน # 17 : หาค่าเฉลี่ยและค่ากลางเรขาคณิต	159
สรุปท้ายบท	163
แบบฝึกหัด	164

บทที่ 9 ไฟล์ 167

การสร้างไฟล์เก็บข้อมูล	168
การอ่านไฟล์	169
การประยุกต์ใช้งาน # 18 : จัดการไฟล์ลำดับเบสให้พร้อมวิเคราะห์	174
การสร้างไฟล์เพื่อทดสอบฟังก์ชัน	175
การประยุกต์ใช้งาน # 19 : สร้างไฟล์เพื่อทดสอบฟังก์ชันการคำนวณ	176
สรุปท้ายบท	178
แบบฝึกหัด	179

บทที่ 10 การจัดการข้อผิดพลาด 181

ข้อผิดพลาดของโปรแกรม	182
การรับมือกับข้อผิดพลาดด้วยคำสั่ง try และ except	184
คำสั่งปิดท้าย finally	186
การกระตุ้นให้เกิดข้อผิดพลาดเองด้วยคำสั่ง raise	188
สรุปท้ายบท	189
แบบฝึกหัด	190

บทที่ 11 โปรแกรมเชิงวัตถุ	191
การเขียนโปรแกรมเชิงวัตถุ	192
คลาสและตัวแปรวัตถุ	193
การสร้างเมทอด	196
คลาสหลายระดับ	200
เมทอดใช้เฉพาะภายในคลาส	201
การบันทึกข้อมูลแบบ pickle	203
<i>การประยุกต์ใช้งาน # 20 : ประมวลผลข้อมูล DNA/RNA แบบ OOP</i>	205
สรุปท้ายบท	213
แบบฝึกหัด	214
 บทที่ 12 วินโดว์และกราฟิก	 215
Graphic User Interface	216
โมดูล tkinter	217
<i>การประยุกต์ใช้งาน # 21 : สร้างโปรแกรมคำนวณดัชนีมวลกาย BMI</i>	220
การจัดวางวิดเจ็ตด้วยคำสั่ง grid()	223
การสร้างแบบฟอร์ม	226
Frame	226
Label และ Entry	228
Radiobutton	228
Listbox	229
Spinbox	230
Checkbutton	231
Scale	232
Text และ Scrollbar	233
Button	234
Menu	237
Toplevel	239
การใช้งาน Message Box	242
tkinter ในรูปแบบโปรแกรมเชิงวัตถุ	245



Canvas	248
การสร้างภาพและการกำหนดตำแหน่ง	248
การสร้างกราฟด้วย Canvas	251
การประยุกต์ใช้งาน # 22 : สร้างกราฟแสดงการเจริญของแบคทีเรีย	259
สรุปท้ายบท	273
แบบฝึกหัด	276

บทที่ 13 โมดูล Regular Expressions 279

Regular Expressions คืออะไร	280
ไวยากรณ์ RE ในไพธอน	281
การสร้างรูปแบบการค้นหาและการใช้งานเมทอด RE	282
การประยุกต์ใช้งาน # 23 : หาลำดับเบสของ degenerate primers	289
สรุปท้ายบท	296
แบบฝึกหัด	296

บรรณานุกรม 298

เฉลยแบบฝึกหัด 302

บทที่ 2	302
บทที่ 3	303
บทที่ 4	304
บทที่ 5	305
บทที่ 6	307
บทที่ 7	309
บทที่ 8	311
บทที่ 9	313
บทที่ 10	315
บทที่ 11	316
บทที่ 12	319
บทที่ 13	325

ดัชนี 328

CHAPTER

01

ภาษาไพธอน
และโปรแกรม



1.1 ภาษาคอมพิวเตอร์

คอมพิวเตอร์ มีบทบาทอย่างมากกับชีวิตในปัจจุบันของมนุษย์ยุคใหม่ แทบจะเรียกได้ว่าเป็นปัจจัยหนึ่งของชีวิตเลยทีเดียว นอกเหนือจากคอมพิวเตอร์ตั้งโต๊ะ โน้ตบุ๊ก และอุปกรณ์ทางวิทยาศาสตร์ที่สังเกตเห็นได้ง่ายแล้ว คอมพิวเตอร์ยังแฝงมาในรูปของ โทรศัพท์มือถือ รถยนต์ โทรทัศน์ กล้องถ่ายรูป เครื่องซักผ้า เตาไมโครเวฟ และเครื่องมืออุปกรณ์ต่าง ๆ รอบตัว ส่วนสำคัญที่มีผลต่อการทำงานของอุปกรณ์ที่มีคอมพิวเตอร์นั้น คือ ซอฟต์แวร์ (software) ซอฟต์แวร์ถูกใช้เป็นตัวกลางในการติดต่อระหว่างความต้องการใช้งานอุปกรณ์ของมนุษย์กับการทำงานของตัวอุปกรณ์หรือที่เรียกว่า ฮาร์ดแวร์ (hardware) ให้สามารถทำงานในทิศทางตามที่ต้องการ โดยทั่วไปการสื่อสารกับฮาร์ดแวร์ต่าง ๆ จะเป็นไปในลักษณะที่เป็นสัญญาณไฟฟ้าในรูปแบบเดียวกับเลขฐานสอง ดังนั้นเลข 0 และ 1 จึงถือว่าเป็นภาษาที่ใกล้เคียงกับภาษาที่เครื่องเข้าใจมากที่สุด หรืออาจเรียกได้ว่าเป็นภาษาเครื่อง (machine language) ตัวอย่างที่ใกล้เคียงกับภาษาเครื่องอีกภาษาคือ ภาษาแอสเซมบลี (assembly) ซึ่งใช้เลขฐาน 16 เป็นหลัก ทั้งสองภาษาลือว่าเป็นภาษาระดับต่ำ (low-level language) ภาษาไพธอนจัดเป็นภาษาคอมพิวเตอร์ระดับสูง (high-level language) เพราะสามารถสื่อสารกับผู้เขียนโปรแกรมได้ง่ายกว่า เนื่องจากมีความใกล้เคียงกับรูปแบบของภาษามนุษย์ แต่ในการทำงานจริงภาษาระดับสูงจะต้องถูกแปลงเป็นภาษาเครื่องก่อนเสมอ ตัวอย่างภาษาคอมพิวเตอร์ระดับสูงอื่น ๆ เช่น ภาษาซี (C++) ปาสคาล (Pascal) จาวา (Java) เบสิก (Basic) เป็นต้น

ภาษาคอมพิวเตอร์จัดเป็นภาษาที่มีรูปแบบแน่นอน (formal languages) ถูกพัฒนาและออกแบบเพื่อการใช้งานตามวัตถุประสงค์ที่กำหนด เช่นเดียวกับสมการทางคณิตศาสตร์ สูตรเคมี ฯลฯ มีลักษณะตรงกันข้ามกับภาษาที่ใช้สื่อสารกันทั่วไป เช่น ภาษาไทย ภาษาอังกฤษ ภาษาจีน ฯลฯ ซึ่งจัดเป็นภาษาที่เกิดและมีวิวัฒนาการเองตามธรรมชาติ (natural language) ภาษาที่มีรูปแบบแน่นอนจะมีรูปแบบและกฎที่ตายตัว ยกตัวอย่างเช่น $2 \times 4 = 8$ มีความหมาย แต่ $2 = x \times 4$ ไม่มีความหมายใด ๆ H_2O หมายถึงน้ำ แต่ $_2HO$ ไม่มีความหมายแต่อย่างใด ในขณะที่ภาษาธรรมชาติ เช่น ภาษาพูดที่ว่า “วันนี้ฝนตกหนัก” กับ “ตกหนักฝนวันนี้” ประโยคแรกมีโครงสร้างทางภาษาไทยที่ถูกต้องและมีความหมาย ประโยคหลังผิดรูปแบบโครงสร้างทางภาษาแต่ก็ยังสามารถสื่อความหมายออกมาได้ ภาษาคอมพิวเตอร์มีลักษณะที่ไม่กำกวม ไม่เอนเอียง ตีความตามตัวอักษรที่เขียนเป็นคำสั่ง ในขณะที่ภาษาพูดและภาษาเขียนของเรามีลักษณะที่กำกวมหรือชัดเจนก็ได้ เอนเอียงหรือกะทัดรัดก็ได้ และอาจมีความหมายตรงตัวหรือเป็นแค่เชิงอุปมาอุปไมย ตัวอย่างที่เห็นได้ชัดคือบทกวีที่มีลักษณะแทบจะตรงกันข้ามกับภาษาคอมพิวเตอร์ทุกประการ

CHAPTER

02

ง่าย
เริ่มต้นใช้งาน
ไพธอน

2.1 โปรแกรมภาษาไพธอน

โปรแกรมภาษาไพธอนเป็น Open Source ที่ได้รับการดูแลจากหน่วยงานที่ไม่แสวงหากำไร ในนามของ Python Software Foundation ซึ่งทุกคนสามารถเข้าไปดาวน์โหลดได้ฟรีที่เว็บไซต์ <https://www.python.org> ตัวโปรแกรมมีให้ดาวน์โหลดได้ในหลายระบบปฏิบัติการ ทั้ง Windows, Linux/UNIX, MacOS และสำหรับระบบปฏิบัติการอื่น ๆ มี 2 เวอร์ชันหลัก คือ Python 2.7.x และ Python 3.x.x อย่างไรก็ตาม Python 2.7.x จะหยุดพัฒนาในอีกไม่นาน ในการติดตั้งภาษาไพธอนสำหรับการเรียนรู้ไปพร้อมกับหนังสือเล่มนี้ให้ติดตั้งไพธอนเวอร์ชันล่าสุด โดยดำเนินการติดตั้งตามคำแนะนำของผู้ดูแลเว็บไซต์

2.2 การติดตั้งโปรแกรมภาษาไพธอน

เข้าสู่เว็บไซต์ Python ทางการที่ <https://www.python.org/downloads> และทำการดาวน์โหลดโปรแกรมภาษาไพธอนที่สอดคล้องกับระบบปฏิบัติการของตัวเอง แนะนำให้ดาวน์โหลดไพธอนเวอร์ชันล่าสุดที่เป็น Python 3.x.x เพื่อให้สอดคล้องกับเนื้อหาในบทเรียนนี้ หลังจากการติดตั้งเสร็จสิ้นสามารถตรวจสอบว่า Python ได้รับการติดตั้งเสร็จเรียบร้อยแล้วหรือไม่ โดยไปที่โฟลเดอร์ Python 3.x ตรวจสอบไฟล์ชื่อ IDE ถ้าพบแสดงว่าโปรแกรมภาษาไพธอนติดตั้งเรียบร้อยแล้ว

2.3 การใช้งานไพธอนในแบบตอบสนองทันที

เมื่อเริ่มต้นใช้งานโปรแกรมภาษาไพธอนเป็นครั้งแรก ไพธอนจะแสดงวินโดว์ Python Shell โดยมีเครื่องหมาย “>>>” เรียกว่า Python prompt อยู่หน้าเคอร์เซอร์ที่กำลังกะพริบ ซึ่งแสดงถึงความ “พร้อม” ในการรอรับคำสั่งจากผู้ใช้งาน ผู้ที่เคยใช้งาน Command Prompt บนระบบปฏิบัติการ Windows จะรู้สึกคุ้นเคยกับรูปแบบที่ปรากฏเพราะมีลักษณะในเบื้องต้นที่คล้ายคลึงกัน เมื่อทดลองทำการคำนวณเบื้องต้นโดยพิมพ์ $10 + 7$ แล้วกดคีย์ enter (สำหรับ PC) หรือ return (สำหรับ Mac) เครื่องก็จะประมวลผลและให้คำตอบ 17 ทันทีในบรรทัดถัดไป และขึ้นเครื่องหมายพร้อม “>>>” เพื่อรอรับคำสั่งต่อไป แต่เมื่อทดลองพิมพ์ตัวอักษร เช่น python ลงไปตรง ๆ ไพธอนจะรายงานข้อผิดพลาดให้เห็นทันที

CHAPTER

03

เล่นกับโปรแกรม

3.1 การเขียนโปรแกรม

เราได้เรียนรู้การใช้งานไพธอนในแบบตอบสนองทันที (immediate mode) หรือที่เราเรียกว่า โหมด Python prompt ไปแล้วในบทที่ 2 ในบทนี้เราจะใช้งานไพธอนในแบบโปรแกรม (program mode) ซึ่งก็คือการเขียนโปรแกรมหรือส่วนของโปรแกรม การเขียนโปรแกรมคือการนำเอาคำสั่งและเงื่อนไขต่าง ๆ มาจัดเรียงกันเป็นลำดับเพื่อให้คอมพิวเตอร์ดำเนินการประมวลผล โดยสามารถใช้งานได้ทันทีหรือบันทึกเป็นไฟล์เก็บไว้ใช้งานในภายหลัง แต่ก่อนที่จะก้าวเข้าสู่การเขียนโปรแกรมใช้งานทางวิทยาศาสตร์สุขภาพอย่างเป็นทางการ ในบทเรียนนี้จะเริ่มต้นด้วยการทำความเข้าใจกับลักษณะโครงสร้างและการทำงานของสิ่งที่เรียกว่า “โปรแกรม” ก่อน

ในหนังสือเรื่อง “How to Think Like a Computer Scientist : Learning with Python 3 Documentation, Release 3rd Edition” ผู้เขียนแนะนำการใช้งานโมดูล “turtle” สำหรับผู้เริ่มต้นเรียนรู้การเขียนโปรแกรม ทั้งนี้เพื่อให้ผู้เรียนสามารถเข้าใจการทำงานของตัวโปรแกรมและหัดคิดอย่างเป็นระบบเหมือนนักวิทยาศาสตร์ที่ทำงานด้านคอมพิวเตอร์ “โมดูล” จัดเป็นองค์ประกอบที่จะเข้ามาขยายศักยภาพการทำงานของตัวโปรแกรมหลัก ในตัวโปรแกรมภาษาไพธอนที่ดาวน์โหลดมาก็มีโมดูลแบบ build-in อยู่มากมายให้เลือกใช้ โมดูล turtle หรือโมดูลเต่า เป็นโมดูลเกี่ยวกับการวาดภาพ ใช้งานง่าย จัดเป็นโมดูลที่เหมาะสมกับการเรียนรู้หลักการทำงานของตัวโปรแกรมที่เขียน ในบทนี้จึงขอให้น้องๆ เป็นผู้ช่วยสอนการเขียนโปรแกรมตามแนวคิดของผู้เขียนหนังสือดังกล่าว

3.2 การสร้างไฟล์โปรแกรมและการใช้งานโมดูล turtle

ในวินโดว์ Python Shell ตรงบริเวณเมนูด้านบน เมื่อเลือกคำว่า “File” จะปรากฏเมนูย่อยให้เลือกตัวเลือกสำหรับสร้างไฟล์ใหม่ (New File) ซึ่งจะปรากฏวินโดว์ใหม่ขึ้นมา ให้ทำการ save as โดยตั้งชื่อไฟล์และเลือกโฟลเดอร์ที่จัดเก็บ ไฟล์ที่บันทึกไว้จะมีนามสกุล .py โดยอัตโนมัติ เสร็จแล้วให้พิมพ์ข้อความดังต่อไปนี้โดยไม่ต้องพิมพ์หมายเลขบรรทัด

```
1 import turtle
2
3 area = turtle.Screen()
4 tao = turtle.Turtle()
5
```

CHAPTER

04

အိတ်
အိတ်

အိတ်



4.1 สตริงคืออะไร

“สตริง (string)” ในภาษาอังกฤษ หมายถึงลักษณะของสิ่งยาว ๆ ที่มีลักษณะแบบเดียวกับเส้นเชือก ลวด ที่มักจะใช้มัดหรือผูกได้ ในภาษาคอมพิวเตอร์จะมองภาพตัวหนังสือ ตัวเลข สัญลักษณ์ใด ๆ หรือแม้แต่ช่องว่าง ที่ถูกนำมาเรียงต่อ ๆ กันเป็นแถว แนว หรือบรรทัด ว่าเป็นเส้นของตัวอักษร จึงเป็นที่มาของคำว่า สตริง

จากบทที่ 2 เราได้เรียนรู้ว่าข้อมูลประเภทสตริงจัดอยู่ใน class ‘str’ เป็นข้อมูลที่ประกอบไปด้วยตัวอักษร (ในที่นี้หมายถึง ตัวหนังสือ ตัวเลข ช่องว่าง หรือสัญลักษณ์ใด ๆ) ภายในเครื่องหมายคำพูด โดยเราสามารถประยุกต์ใช้ฟังก์ชันการคำนวณบางตัว เช่น บวก คูณ ซึ่งสามารถให้ผลลัพธ์ที่มีลักษณะจำเพาะที่ไม่ใช่ตัวเลข แต่เป็นแบบลักษณะของสตริงเอง ฟังก์ชันอื่น ๆ เช่น type() print() int() และ float() ก็สามารถประยุกต์ใช้ได้กับข้อมูลแบบสตริงตามความเหมาะสมของสถานการณ์ class ‘str’ เป็นโมดูลเช่นเดียวกับ turtle ต่างกันที่ class ‘str’ เป็น build-in module หรือโมดูลมาตรฐานที่ไพธอนมีอยู่แล้ว จึงไม่จำเป็นต้องใช้คำสั่ง import ผนวกโมดูลเข้ามาอีก ในบทเรียนนี้เราจะเรียนรู้ฟังก์ชันที่เกี่ยวข้องกับข้อมูลหรือตัวแปรสตริงเพิ่มเติม ฟังก์ชันใด ๆ ที่ออกแบบมาให้สามารถใช้งานเฉพาะกับสตริงโดยตรงในภาษาไพธอนจะเรียกว่า เมทอดสตริง (string method)

4.2 โครงสร้างสตริง

ข้อมูลประเภทสตริงเมื่อจัดเก็บในตัวแปรใด ๆ จะมีโครงสร้างที่สามารถชี้บ่งบอกตำแหน่งตัวอักษรแต่ละตัวได้อย่างชัดเจนเรียกว่า ดัชนี (index) โดยดัชนีของสตริงแต่ละตัวจะเป็นตัวเลขที่แสดงตำแหน่งของตัวอักษรเรียงตามลำดับจากซ้ายไปขวาและเริ่มต้นที่ 0 เสมอ เช่น x = “Python” มีโครงสร้างดัชนี ดังรูปที่ 4.1

x	=	“	P	y	t	h	o	n	”
หมายเลขดัชนี			0	1	2	3	4	5	

รูปที่ 4.1 แสดงค่าดัชนีของสตริง “Python”

หมายเลขดัชนีสามารถมีค่าเป็นลบได้ แต่จะเป็นการระบุตำแหน่งสตริงในทิศทางตรงข้าม คือจากขวาไปซ้าย โดยมีค่าเริ่มเป็น -1 ดังรูปที่ 4.2

CHAPTER

05

ตัวแปรกลุ่ม

Tuple / List / Dictionary



5.1 การจัดเก็บกลุ่มข้อมูล

ข้อมูลที่หลากหลายสามารถรวมให้เป็นกลุ่มและนำมาจัดเก็บในที่เดียวกันเป็นกลุ่มข้อมูลได้ โดยข้อมูลย่อยแต่ละชนิดแต่ละชุดสามารถจำแนกออกจากกันด้วยการใช้เครื่องหมายคอมม่าคั่น โครงสร้างข้อมูลแบบกลุ่มเป็นรูปแบบข้อมูลที่จัดเก็บโดยมีวัตถุประสงค์เพื่อการจัดแบ่งข้อมูลให้เป็นหมวดหมู่ หรือเพื่อให้เกิดความสะดวกต่อการจัดการข้อมูล ไพธอนมีโครงสร้างการจัดเก็บข้อมูลแบบกลุ่มอยู่หลายประเภท ในบทที่แล้วเราได้รู้จักการจัดเก็บหรือกำหนดค่าข้อมูลแบบกลุ่มให้กับตัวแปรประเภท sequence 3 ชนิด คือ Tuple () List [] และ Dictionary { } ในบทนี้เราจะมาทำความเข้าใจในรายละเอียดของการจัดเก็บข้อมูลในตัวแปร sequence แต่ละชนิด

5.2 ตัวแปร Tuple

ตัวแปร Tuple เป็นตัวแปรประเภท sequence ที่ใช้จัดเก็บกลุ่มข้อมูลที่รวบรวมอยู่ในวงเล็บธรรมดา การกำหนดค่าข้อมูลให้กับตัวแปร Tuple สามารถทำได้ดังนี้

```
>>> record1 = ("Ann", 1960, "US")
>>> record2 = ("Tom", 1967, "UK")
>>> record3 = ("Li", 1969, "China")
>>> friends = (record1, record2, record3)
```

จะเห็นว่าโครงสร้างการจัดเก็บข้อมูลโดยใช้ตัวแปร Tuple ทำให้เราสามารถรวบรวมข้อมูลที่เกี่ยวข้องกันเป็นกลุ่มก้อน รูปแบบการจัดเก็บข้อมูลในตัวอย่างจะคล้ายกับการบันทึกประวัติบุคคล ซึ่งสามารถนำมาจัดเก็บซ้อนเข้าไปเป็นชั้น ๆ ได้ ทีนี้ลองสังพิมพ์ข้อมูลในตัวแปร friends ดู

```
>>> print (friends)
(('Ann', 1960, 'US'), ('Tom', 1967, 'UK'), ('Li', 1969, 'China'))
```

การนับจำนวนสมาชิกหรือจำนวนข้อมูลในตัวแปร friends ก็เหมือนกับการนับจำนวน record ที่บันทึกไว้ สามารถทำได้โดยใช้ฟังก์ชัน len ()

CHAPTER

06

การตรวจสอบ
เงื่อนไข



ในการใช้ชีวิตประจำวันของมนุษย์ไม่ว่าจะเป็นชนชาติใดภาษาใดในโลกย่อมต้องเผชิญหน้ากับสภาวะที่ต้องตัดสินใจอยู่ตลอดเวลา การตัดสินใจมักจะลงเอยด้วยการเลือกทางใดทางหนึ่งจาก 2 ทางเลือกหรือมากกว่า เช่น ใช่/ไม่ใช่ ชาย/ขวา ไป/ไม่ไป เลือกซื้อ 3 จากตัวเลือก 4 ตัวเลือก เป็นต้น กระบวนการตัดสินใจประกอบไปด้วย การกำหนดเงื่อนไข การประเมินข้อเท็จจริงตามเงื่อนไขและการเลือกตัวเลือกที่ถูกกำหนดโดยเงื่อนไข เช่น

# กำหนดเงื่อนไข	“ถ้าวันนี้ฝนตกจะไม่ไปทำงาน”
ตัวแปร	ฝนตก / ไม่ตก
ทางเลือก	ไปทำงาน / ไม่ไปทำงาน
# ประเมินข้อเท็จจริง	วันนี้ฝนไม่ตก
# เลือกตัวเลือก	ไปทำงาน

ในโปรแกรมคอมพิวเตอร์เราเรียกกระบวนการเหล่านี้รวม ๆ ว่า การตรวจสอบเงื่อนไข (conditionals) การกำหนดเงื่อนไขเกิดขึ้นตอนออกแบบและเขียนโปรแกรม การประเมินข้อเท็จจริงและการเลือกตัวเลือกตามเงื่อนไขเกิดขึ้นตอนโปรแกรมทำงาน โดยโปรแกรมจะทำการตรวจสอบเหตุการณ์แต่ละเหตุการณ์ที่เกิดขึ้นแล้วนำมาประเมินหาข้อเท็จจริงโดยใช้หลักการเปรียบเทียบข้อมูล ซึ่งคำตอบที่ได้ก็จะเป็นอย่างใดอย่างหนึ่งคือไม่จริงก็เท็จ ถ้าเป็นจริงก็จะดำเนินการตามตัวเลือกที่วางแผนไว้ ถ้าไม่จริงอาจจะไม่มีการดำเนินการใด ๆ หรืออาจดำเนินการตามทางเลือกสำรองอีกทางหนึ่ง

การตรวจสอบเงื่อนไขถือเป็น 1 ใน 5 ชุดคำสั่งพื้นฐานที่ขาดไม่ได้สำหรับผู้เรียนรู้การเขียนโปรแกรมคอมพิวเตอร์ทุกภาษา จากตัวอย่างโปรแกรมในบทก่อนหน้านี้เราจะพัฒนาต่อยอดโปรแกรมได้อย่างง่ายดายถ้าเราสามารถตรวจสอบเงื่อนไขบางประการที่เกิดขึ้นได้

```
seq = "UGAGUGAUCGUAGUCGAUGCGCGCGCUAGUCGAUUGAUGAUGAC"

# Search program I
print ('Search program I : "AUG" position : ')
print (seq.find("AUG"))
```

CHAPTER

07

คำอธิบาย
คำสั่งวนซ้ำ



การทำงานในโรงงานอุตสาหกรรมส่วนใหญ่ในปัจจุบันมักจะใช้เครื่องจักรขนาดใหญ่ในการผลิตสินค้า โดยมีคนงานจำนวนหนึ่งคอยตรวจสอบการทำงานของเครื่องจักร ตรวจสอบภาพสินค้าแต่ละชิ้นที่ถูกกลิ้งผ่านสายพานการผลิต หรือนำสินค้าบรรจุลงกล่อง ซึ่งลักษณะรูปแบบการทำงานจะเป็นลักษณะซ้ำ ๆ กันตลอดเวลา แน่นนอนความผิดพลาดย่อมมีโอกาสเกิดขึ้นได้ตลอดเวลาอันเนื่องมาจากความอ่อนล้าของร่างกายมนุษย์ ในทางตรงกันข้ามโปรแกรมคอมพิวเตอร์สามารถทำงานในรูปแบบซ้ำ ๆ กันได้อย่างดีเยี่ยม ผลการทำงานครั้งแรกเป็นอย่างไครั้งสุดท้ายก็เป็นอย่างนั้นไม่เปลี่ยนแปลงไม่ว่าจะต้องทำงานซ้ำ ๆ กี่รอบก็ตาม การทำงานซ้ำ ๆ แบบวนลูปจึงเป็นคำสั่งพื้นฐานที่ทุกภาษาคอมพิวเตอร์ขาดไม่ได้ เรียกว่า iteration ในไพธอนมีคำสั่งวนซ้ำหรือวนลูปอยู่ 2 ชนิด คือ for loop และ while loop เราได้เรียนรู้ความหมายและการประยุกต์ใช้งานคำสั่ง for loop มาบ้างแล้วในบทที่ 3 บทที่ 4 และจากการประยุกต์ใช้งาน # 11 ในบทที่ 5 ซึ่งเป็นการใช้งานคำสั่ง for loop ในรูปแบบต่าง ๆ กัน เช่น

for ตัวแปรลูป in range (จำนวนเต็ม) หรือ (start, stop, step)

for ตัวแปรลูป in ตัวแปรสตริง / Tuple / List / Dictionary

แบบแรกอาจใช้เป็นเพียงการนับรอบหรือวนรอบการทำงานตามจำนวนเต็มที่กำหนดใน in range() หรืออาจจะเป็นการดึงตัวเลขที่เก็บในตัวแปรลูปแต่ละรอบมาใช้งาน ซึ่งตัวเลขจะถูกดึงออกมาตามลำดับหรือกระโดดเป็นช่วง ๆ จะถูกกำหนดโดยเงื่อนไขที่เพิ่มขึ้นมาภายใน in range (start, stop, step) ในกรณีที่ใส่ตัวเลขจำนวนเต็มเพียงตัวเดียวจะมีความหมายถึง stop แบบที่สองจะเป็นรูปแบบการนำค่าที่เก็บในตัวแปรแต่ละชนิดมาใช้งานตามลำดับ โดยการวนซ้ำจะสิ้นสุดเมื่อค่าภายในตัวแปรถูกดึงมาใช้งานจนครบทุกค่า การใช้งานคำสั่งวนซ้ำช่วยให้การเขียนโปรแกรมกระชับ สั้น และเข้าใจง่าย อย่างไรก็ตามคำสั่ง for loop มีข้อจำกัดในการวนซ้ำบางประการ เช่น ไม่สามารถนำมาใช้กับการทำงานที่ไม่ทราบจำนวนรอบ หรือการทำงานในแบบวนรอบต่อเนื่องไม่หยุด การแก้ปัญหาเหล่านี้จำเป็นต้องใช้คำสั่งวนซ้ำอีกชนิด คือ while loop เข้ามาช่วย แต่ก่อนที่จะเข้าเรื่องการใช้งาน while loop เรามาเรียนรู้การใช้งานคำสั่ง for loop เพิ่มเติมอีกนิด

CHAPTER

08

ฟังก์ชัน

คำอธิบาย



8.1 ฟังก์ชันคืออะไร

ฟังก์ชันจัดเป็นกระบวนการทำงานของคอมพิวเตอร์ที่ถูกรวบรวมเป็นชุด ๆ โดยในแต่ละชุดจะมีการประมวลผลที่เป็นไปในทิศทางที่สอดคล้องกัน ในทุก ๆ ภาษาคอมพิวเตอร์ในโลกจะต้องมีฟังก์ชันสำเร็จรูปในลักษณะที่เรียกว่า build-in เพื่ออำนวยความสะดวกให้นักเขียนโปรแกรมทำงานและเลือกใช้ภาษาระดับสูงในการเขียนโปรแกรม แทนที่จะต้องมาจดจำหรือเรียนรู้การคำนวณหรือประมวลผลเลขฐานสองซึ่งเป็นภาษาเดียวที่คอมพิวเตอร์เข้าใจ แทบจะเรียกได้ว่าชุดคำสั่งพื้นฐาน 5 รูปแบบซึ่งประกอบไปด้วย การรับข้อมูล การแสดงผลลัพธ์ การคำนวณ การตรวจสอบเงื่อนไข การทำซ้ำ ล้วนมีกระบวนการทำงานในรูปแบบ build-in function ที่ถูกเขียนขึ้นมาทั้งสิ้น ลองมาดูตัวอย่างการใช้ฟังก์ชันที่เคยผ่านกันมาบ้างแล้ว

```
>>> 3+7
10
>>> str(10)
'10'
>>> x = '10'
>>> print ("Ten")
Ten
>>>
```

ฟังก์ชันที่เรียกใช้งานข้างต้น คือ การบวก การแปลงค่าเป็นสตริง การกำหนดค่าตัวแปร และการพิมพ์ ทั้งหมดจัดเป็นฟังก์ชันพื้นฐานที่มีในภาษาคอมพิวเตอร์ทุกภาษา ในไพธอนมีวิธีการสร้างฟังก์ชันอีกรูปแบบที่ผู้ใช้สามารถกำหนดทิศทางการประมวลผลและเรียกใช้งานเองได้

8.2 การสร้างและเรียกใช้ฟังก์ชัน

ลองมาดูตัวอย่างการสร้าง และเรียกใช้งานฟังก์ชันสำหรับการคำนวณหาค่ารากที่สองแบบง่าย ๆ

```
>>> def sqr(num):
    result = num**(1.0/2)
    return result
>>> x = sqr(2)
>>> print (x)
1.41421356237
>>>
```

CHAPTER

09

ไฟล์

การเขียนโปรแกรม



9.1 การสร้างไฟล์เก็บข้อมูล

ในฐานะที่เป็นโปรแกรมเมอร์ เมื่อมีความต้องการที่จะทดสอบข้อมูลชุดหนึ่ง โปรแกรมเมอร์สามารถเขียนข้อมูลที่ต้องการทดสอบใส่ลงในเนื้อหาของโปรแกรมได้โดยตรง ซึ่งมีผลทำให้ข้อมูลที่จะใช้ทดสอบกลายเป็นส่วนหนึ่งของโปรแกรมที่ใช้เขียน ในการเขียนโปรแกรมให้ผู้อื่นใช้งาน โปรแกรมเมอร์อาจเขียนส่วนโปรแกรมให้ผู้ใช้สามารถป้อนข้อมูลที่จะทดสอบทีละตัวผ่านการใช้งานคำสั่ง `input()` เมื่อทำการรันโปรแกรม ผลลัพธ์ก็จะแสดงให้เห็นทางหน้าจอหรือเก็บเอาไว้ในตัวแปรเพื่อรอคำสั่งพิมพ์ผลลัพธ์ ซึ่งการนำผลลัพธ์ไปใช้งานต้องดำเนินการก่อนที่จะปิดตัวโปรแกรมหรือปิดเครื่องคอมพิวเตอร์ เนื่องจากข้อมูลที่เป็นผลลัพธ์จะถูกเก็บในหน่วยความจำชั่วคราวของคอมพิวเตอร์ หรือที่เรียกว่า RAM (Random Access Memory) และจะสูญสลายหายไปทันทีที่โปรแกรมถูกปิด การใช้วิธี copy & paste ย้ายข้อมูลโดยอาศัยฟังก์ชันที่มีอยู่แล้วในระบบปฏิบัติการของคอมพิวเตอร์แต่ละเครื่องใส่ลงไฟล์ประเภท Word หรือ Text เพื่อบันทึกนับเป็นวิธีที่ง่ายที่สุด อย่างไรก็ตามไม่ว่าจะมีฟังก์ชันสำหรับการสร้างและบันทึกไฟล์ข้อมูลประเภท Text โดยเฉพาะ และสามารถบันทึกใส่สื่อต่าง ๆ ได้ เช่น hard drive, USB drive หรือแม้กระทั่งแผ่น CD-RW เช่นเดียวกับไฟล์ข้อมูลทั่วไป ที่นี้มาทดลองสร้างไฟล์แรกใน Python prompt กัน

```
>>> f = open('first.txt', 'w')
>>> f.write ('Hello, How are you ?\n')
>>> f.close()
```

ใน Python Shell เลือก open และลองหาไฟล์ที่มีชื่อ 'first' ทั้งนี้จะต้องเลือกชนิดไฟล์เป็น Text files หรือ All files เมื่อพบแล้วลองดับเบิลคลิกดู จะพบข้อความทักทายที่เขียนเอาไว้

เนื่องจากข้อมูลที่บันทึกเป็นประเภท Text file ดังนั้นโปรแกรมอื่น ๆ ในเครื่องคอมพิวเตอร์ที่อ่าน Text file ได้ก็สามารถอ่านไฟล์นี้ได้เช่นกัน มาเขียนโปรแกรมต่อกันอีกนิด

```
>>> h = open('first.txt', 'a')
>>> h.write ('Fine, thank you ! ')
>>> h.write ('And you ? \n')
>>> h.close()
```


CHAPTER

10

ปัจจัย
สำคัญ

การจัดการ
ข้อผิดพลาด



10.1 ข้อผิดพลาดของโปรแกรม

ลองนึกภาพผู้ใช้โปรแกรมคนหนึ่งกำลังใช้งานโปรแกรมที่เราเขียนให้ แล้วอยู่ดี ๆ โปรแกรมก็แสดงข้อผิดพลาด หยุดชะงัก แล้วกระเด็นหลุดออกจากตัวโปรแกรม แน่นอนว่าไม่มีโปรแกรมเมอร์คนไหนอยากจะทำให้เหตุการณ์แบบนี้เกิดขึ้น โปรแกรมที่ดีจะต้องแสดงข้อผิดพลาดให้ผู้ใช้โปรแกรมรับรู้และพร้อมที่จะรับคำสั่งที่ถูกดัดแปลงใหม่โดยไม่มีการหยุดชะงักหรือหลุดออกจากตัวโปรแกรม ข้อผิดพลาดที่ได้เรียนรู้มาแล้วในบทที่ 1 มี 3 ประเภท คือ ข้อผิดพลาดทางไวยากรณ์ ข้อผิดพลาดขณะประมวลผล และข้อผิดพลาดของผลลัพธ์ ข้อผิดพลาดทางไวยากรณ์จะต้องได้รับการแก้ไขทันที ไม่เช่นนั้นโปรแกรมก็ไม่สามารถทำงานได้ ข้อผิดพลาดขณะประมวลผลและข้อผิดพลาดของผลลัพธ์สามารถตรวจสอบได้โดยใช้ไฟล์ทดสอบที่มีข้อมูลทุกรูปแบบ แล้วทำการแก้ไขก่อนจะนำมาใช้งานจริง กรณีที่นำออกมาใช้งานแล้วพบข้อผิดพลาดที่หลังจากจะต้องแก้ไขด้วยการออกโปรแกรมเวอร์ชันใหม่ ๆ หรือเวอร์ชันอัปเดต สิ่งที่แตกต่างกันระหว่างข้อผิดพลาดขณะประมวลผลกับข้อผิดพลาดของผลลัพธ์ระหว่างการใช้งานคือ ความผิดพลาดขณะประมวลผลจะทำให้โปรแกรมหยุดชะงักและหลุดออกมา ในขณะที่ความผิดพลาดของผลลัพธ์เพียงแต่แสดงผลที่ไม่ตรงกับค่าที่เป็นจริงหรือค่าที่คาดหวังเอาไว้โดยโปรแกรมยังคงทำงานต่อไปตามปกติ การจัดการข้อผิดพลาดของโปรแกรมในบทนี้ จะเกี่ยวข้องกับการรับมือและแก้ไขข้อผิดพลาดขณะประมวลผลหรือที่เรียกว่า exceptions ลองมาดูตัวอย่างข้อผิดพลาดที่เกิดขึ้นจากการประมวลผลโปรแกรมในโหมด Python prompt

```
>>> float('w')           # ไม่สามารถแปลงค่าสตริงให้เป็นตัวเลขที่มีทศนิยมได้
Traceback (most recent call last):
  File "<pyshell#14>", line 1, in <module>
    float('w')
ValueError: could not convert string to float: w
```

```
>>> 5/0                   # หารด้วยเลข 0 ซึ่งไม่สามารถนำมาใช้เป็นตัวหารได้
Traceback (most recent call last):
  File "<pyshell#2>", line 1, in <module>
    5/0
ZeroDivisionError: integer division or modulo by zero
```

CHAPTER

11

โปรแกรมเชิงวัตถุ



11.1 การเขียนโปรแกรมเชิงวัตถุ

โปรแกรมเชิงวัตถุ (Object-Oriented Programming) หรือ OOP เป็นรูปแบบการเขียนโปรแกรมชนิดหนึ่งที่เกิดขึ้นมาจากสถาบัน MIT (Massachusetts Institute of Technology) ในช่วงปลายทศวรรษที่ 50 ต่อเนื่องถึงต้นทศวรรษที่ 60 ปัจจุบันได้กลายมาเป็นรูปแบบการเขียนโปรแกรมที่ได้รับความนิยมในการนำมาใช้สรรค์สร้างซอฟต์แวร์ใหม่ ๆ ออกมา การเขียนโปรแกรมเชิงวัตถุได้รับการพัฒนาขึ้นมาเพื่อรองรับการขยายตัวและพัฒนาอย่างรวดเร็วของซอฟต์แวร์ทั้งทางด้านขนาดและความซับซ้อน ทำให้สามารถพัฒนาต่อยอดโปรแกรมที่ใหญ่และซับซ้อนขึ้นเรื่อย ๆ OOP เป็นรูปแบบโปรแกรมที่มีใช้ในเกือบจะทุกภาษาคอมพิวเตอร์ที่ได้รับความนิยมมากในปัจจุบัน เช่น C++, Java, Python ฯลฯ

การเขียนโปรแกรมเชิงวัตถุอาศัยแนวคิดในการมองตัวโปรแกรมเป็นวัตถุ (object) ที่ประกอบไปด้วยข้อมูลและวิธีการใช้งาน ซึ่งการใช้โปรแกรมสร้างวัตถุแต่ละชนิดมักจะกำหนดให้มีความสอดคล้องกับตัววัตถุหรือแนวคิดในโลกความจริง เช่น โทรศัพท์มือถือ เป็นวัตถุที่มีข้อมูล (ยี่ห้อ รุ่น ฯลฯ) และฟังก์ชันที่ใช้ในการโทรติดต่อเบ็ดเสร็จอยู่ใน การสร้างโปรแกรมคอมพิวเตอร์ที่สามารถใช้โทรติดต่อได้ก็จะมองส่วนของโปรแกรมที่เขียนเป็นเหมือนตัวโทรศัพท์ ซึ่งในส่วนของโปรแกรมก็จะประกอบไปด้วยวิธีการต่าง ๆ ที่โทรศัพท์ทั่วไปมี เช่น วิธีต่อสาย วิธีส่งสัญญาณเมื่อมีสายเข้า วิธีวางสาย เป็นต้น การเขียนโปรแกรมเชิงวัตถุมีแนวคิดและรูปแบบที่แตกต่างอย่างสิ้นเชิงจากวิธีการเขียนโปรแกรมเชิงวิธีการ (procedural programming) หรือการเขียนโปรแกรมเชิงฟังก์ชัน ซึ่งเป็นวิธีที่เน้นการสร้างฟังก์ชันเพื่อใช้งานโดยตรง รูปแบบการเขียนโปรแกรมที่ได้เรียนรู้มาแล้วในบทก่อนหน้านี้ส่วนมากจะเป็นแบบ procedural programming ที่ต้องเขียนวิธีการหรือฟังก์ชันเพื่อดำเนินการอย่างใดอย่างหนึ่งกับข้อมูลที่ป้อนเข้าไป เช่น ฟังก์ชัน C2F(c) ในหัวข้อ 8.5 เป็นการเขียนโปรแกรมเชิงฟังก์ชันที่ใช้สำหรับการแปลงค่าอุณหภูมิจากหน่วยองศาเซลเซียสไปเป็นองศาฟาเรนไฮต์ โดยพารามิเตอร์ c คืออุณหภูมิในหน่วยองศาเซลเซียสที่ต้องการจะแปลง วิธีการใช้งานฟังก์ชันใน Python prompt ก็โดยการพิมพ์ C2F (37) ลงไปตรง ๆ ตามด้วยการกดปุ่ม enter หรือ return ผลการแปลงอุณหภูมิเป็นฟาเรนไฮต์ก็จะปรากฏขึ้น

ในมุมมองของการเขียนโปรแกรมแบบ OOP จะมองอุณหภูมิเป็นวัตถุชนิดหนึ่ง ซึ่งสอดคล้องพอดีกับชื่ออุปกรณ์ “เทอร์โมมิเตอร์” ที่ใช้สำหรับวัดอุณหภูมิ เทอร์โมมิเตอร์ที่ใช้กันโดยทั่วไปมี 2 ชนิด คือ แบบวัดเป็นหน่วยองศาเซลเซียสกับแบบวัดเป็นหน่วยองศาฟาเรนไฮต์ ในขณะที่ระบบการแสดงผลอุณหภูมิมีทั้งที่เป็นแบบปรอทและแบบดิจิทัล โดยที่แบบดิจิทัลอาจมีฟังก์ชันพิเศษในการแปลงอุณหภูมิกลับไปมาได้ ในการสร้างวัตถุที่เป็นเทอร์โมมิเตอร์ขึ้นมาใช้งานจำเป็นต้องระบุก่อนว่าเป็นเทอร์โมมิเตอร์

CHAPTER

12

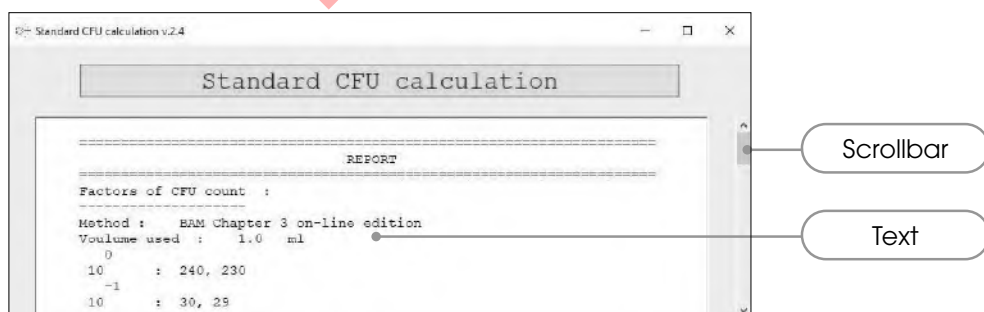
อินทรีย์และกราฟิก



12.1 Graphic User Interface

เนื่องจากภาษาไพธอนเป็นภาษาคอมพิวเตอร์ที่มีลักษณะใกล้เคียงกับภาษาอังกฤษ ผู้เขียนโปรแกรมมือใหม่จึงสามารถจดจำศัพท์เทคนิคและรูปแบบของภาษาได้ไวและสามารถนำมาประยุกต์ใช้ในการแก้ปัญหาเฉพาะด้านในงานของตนเองได้ ในกรณีการเขียนโปรแกรมเพื่อใช้งานเอง ผู้ใช้งานมักจะไม่ค่อยมีปัญหากับรูปแบบการรันโปรแกรม การป้อนข้อมูล และการแสดงผลบนหน้าจอในลักษณะที่เป็นบรรทัดต่อบรรทัด แต่ในกรณีการเขียนโปรแกรมให้ผู้อื่นใช้งาน ทั้งผู้เขียนและผู้ใช้โปรแกรมอาจจะเกิดความรู้สึกไม่ประทับใจเมื่อเทียบกับการใช้งานโปรแกรมในลักษณะที่เป็นหน้าต่างหรือวินโดว์ ลักษณะโปรแกรมคอมพิวเตอร์หรือส่วนของโปรแกรมที่แสดงภาพบนจอในรูปแบบภาพกราฟิกของวินโดว์และทำหน้าที่เป็นตัวกลางระหว่างผู้ใช้งานกับตัวโปรแกรมหลัก โดยอำนวยความสะดวกให้ผู้ใช้งานโปรแกรมที่เขียนขึ้นมาได้โดยง่าย เรียกว่า Graphic User Interface (GUI) ในภาษาไพธอนเองก็มีโมดูลมาตรฐานที่ช่วยให้ผู้เขียนโปรแกรมสามารถนำมาใช้พัฒนาการแสดงผลบนจอแบบ GUI ของตัวเองได้ เรียกว่า tkinter (tk interface)

โมดูล tkinter ประกอบไปด้วยเครื่องมือต่าง ๆ ที่ผู้เขียนโปรแกรมสามารถเลือกนำมาใช้ในการสร้างภาพกราฟิกเพื่อติดต่อกับผู้ใช้โปรแกรม เช่น เมนู ปุ่มกด ส่วนของข้อความ ช่องสำหรับกรอกข้อความ ฯลฯ เป็นต้น เครื่องมือและส่วนประกอบต่าง ๆ ข้างต้นเรียกรวม ๆ ว่า วิดเจ็ต (widget) ตัวอย่างในรูปที่ 12.1 เป็นส่วนหนึ่งของวินโดว์จากโปรแกรมสำหรับการคำนวณหาปริมาณจุลินทรีย์ทั้งหมดแบบวิธีมาตรฐาน (Standard CFU calculation v2.4) ที่สร้างขึ้นมาจากโมดูล tkinter ของไพธอน ภายในวินโดว์จะแสดงให้เห็นถึงลักษณะวิดเจ็ตชนิดต่าง ๆ 6 ชนิด คือ Scrollbar, Text, Frame, Label, Spinbox และ Button



CHAPTER

13

လူနာ Regular
Expressions



ในการใช้งานภาษาเขียนเกือบทุกภาษาในโลก บางครั้งจะพบเห็นรูปแบบพิเศษซึ่งไม่สามารถสื่อสารเป็นภาษาพูดที่ชัดเจนได้ ยกตัวอย่างเครื่องหมายที่นิยมใช้ในการเขียนภาษาไทย เช่น ร้อนมาก ๆ กรุงเทพฯ ฯลฯ เป็นต้น การใช้เครื่องหมายหรือสัญลักษณ์ ๆ ๆ ฯลฯ เหล่านี้ ช่วยสื่อความหมายในการเน้นย้ำ ย่อความ ยกตัวอย่าง เพื่อให้ภาษาเขียนมีความชัดเจน กระชับ และสมบูรณ์แบบมากยิ่งขึ้น ในภาษาคอมพิวเตอร์ก็มีรูปแบบที่บางครั้งสามารถใช้อักษร เครื่องหมาย หรือสัญลักษณ์ ช่วยในการขยายความหมายที่กว้างขึ้นกว่าเดิม เช่น การใช้คำสั่งแสดงชื่อไฟล์ตามด้วยสัญลักษณ์ *.py ใน command line ของ Microsoft Windows จะหมายถึงให้คอมพิวเตอร์ค้นหาและแสดงรายชื่อไฟล์ภาษาไพธอนทุกไฟล์ในโฟลเดอร์ที่มีนามสกุล py (โดยที่อาจไม่เคยเห็นหรือรู้จักไฟล์ดังกล่าวมาก่อน) ในไพธอนเมื่อใช้คำสั่ง `from tkinter import *` จะหมายถึงให้นำทุกเมทอดที่มีอยู่ในโมดูล tkinter (ทั้งที่รู้จักและไม่รู้จัก) มาผนวกเข้ากับตัวโปรแกรมหลักของไพธอน เป็นต้น จะเห็นว่าการใช้สัญลักษณ์ * ทำให้ไม่จำเป็นต้องพิมพ์ชื่อเมทอดทุกตัวใน tkinter และยังใช้ค้นหาไฟล์ทุกไฟล์ที่มีนามสกุล py ใน Windows ได้

วิธีการค้นหาคำหรือข้อความที่เคยเรียนมาแล้วในบทที่ 4 “สตริง” เป็นวิธีที่ต้องกำหนดตัวอักษรแบบตายตัวเพื่อค้นหาสตริงที่ตรงกันกับชุดตัวอักษรที่ใช้ค้นหา วิธีนี้สามารถค้นหาสตริงที่ต้องการจากข้อความขนาดใหญ่ได้ดี แต่ไม่สามารถใช้ค้นหาคำหรือชุดสตริงที่มีลักษณะใกล้เคียงหรือมีรูปแบบที่จำเพาะแต่ไม่ตายตัว อย่างไรก็ตาม ในภาษาไพธอนมี build-in module ที่สนับสนุนการขยายขีดความสามารถในการค้นหา ด้วยการประยุกต์ใช้เครื่องหมายและสัญลักษณ์ต่าง ๆ ร่วมกับตัวอักษรมาสร้างรูปแบบการค้นหาที่ยืดหยุ่นมากขึ้น เรียกว่า Regular Expressions

13.1 Regular Expressions คืออะไร

Regular Expressions (RE) คือ ลำดับของตัวอักษรที่กำหนดรูปแบบการค้นหา ซึ่งมักจะนำมาประยุกต์ใช้กับอัลกอริทึมเพื่อช่วยค้นหาตัวอักษร คำ ประโยค หรือสตริงที่ต้องการในลักษณะที่ยืดหยุ่นกว่าการค้นหาแบบปกติ บางครั้งอาจพบเห็นเป็นคำย่อ เช่น RE หรือ REGEX โมดูล RE จัดเป็นโมดูลที่มีใช้ในเกือบทุกภาษาคอมพิวเตอร์ในยุคหลัง ๆ แต่ละภาษาก็จะมีรายละเอียดทางไวยากรณ์ที่แตกต่างกันไป โดยรูปแบบการใช้งาน RE ในไพธอนกับในภาษาเพิร์ล (Perl) จะมีความใกล้เคียงกัน RE ถูกนำมาประยุกต์ใช้กับงานค้นหาที่หลากหลาย การประยุกต์ใช้ RE ทางวิทยาศาสตร์ชีวภาพสามารถพบเห็นได้บ่อยในงานชีววิทยาโมเลกุล เช่น การค้นหาส่วนของโปรตีนที่ทำหน้าที่เฉพาะ (protein domain) การค้นหาตำแหน่งการตัดของเอนไซม์ตัดจำเพาะบนสาย DNA (restriction site หรือ recognition site) หรือ การออกแบบ

หนังสือแนะนำ



400
บาท

ระบาดวิทยาและสถิติ

Epidemiology and Statistics

ผู้แต่ง : ผศ. ดร. นพ.ศักดิ์ชัย ไชยมหาพฤกษ์

ระบาดวิทยาและสถิติ เป็นศาสตร์ที่มีความสัมพันธ์กันมากจะต้องเรียนรู้ และประยุกต์ใช้ด้วยกันในการแพทย์ การสาธารณสุข หนังสือแบ่งออกเป็น 3 ตอน ตอนที่ 1 ระบาดวิทยา ประกอบด้วยบทนำสู่ระบาดวิทยา การวัดทาง ระบาด วิทยา การวัดความสัมพันธ์ การออกแบบวิจัย การวิจัยแบบสังเกต การวิจัยทาง คลินิกเชิงทดลอง อคติ ตัวแปรกวน การทดสอบวินิจฉัยโรค และการวัด ตอนที่ 2 สถิติประกอบด้วยสถิติเชิงพรรณนา สถิติเชิงอ้างอิง การทดสอบทางสถิติ โปรแกรม คอมพิวเตอร์สถิติ PSPP การเปรียบเทียบตัวแปรเชิงตัวเลข การวิเคราะห์ความ สัมพันธ์ การวิเคราะห์ถดถอย และการวิเคราะห์การรอดชีพ ตอนที่ 3 การประยุกต์ใช้ ระบาดวิทยาและสถิติ ประกอบด้วย การสาธารณสุข การป้องกันควบคุมโรคติดต่อ และเวชศาสตร์เชิงประจักษ์

หนังสือเล่มนี้เป็นหนังสือความรู้เบื้องต้นด้านระบาดวิทยาและสถิติ ครอบคลุมเนื้อหาสำหรับนิสิตแพทย์ตามเกณฑ์ความรู้ความสามารถของ แพทยสภา แต่ในขณะเดียวกันก็ครอบคลุมเนื้อหาที่เป็นประโยชน์ต่อผู้สนใจทั่วไปในการเรียนรู้ ได้ด้วยตนเอง



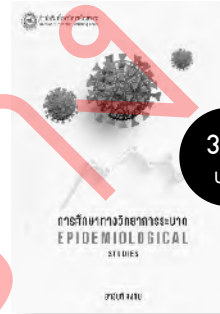
400
บาท

เทคนิคเหมืองข้อมูล

Data Mining Techniques

ผู้แต่ง : ผศ. ดร.จรัสศรี รุ่งรัตนอุบล

เทคนิคเหมืองข้อมูลเป็นหนึ่งในความรู้สมัยใหม่ที่หลายองค์กรนำไปประยุกต์ใช้ เพื่อสนับสนุนการตัดสินใจ การวางแผนกลยุทธ์ การปรับปรุงการให้บริการและการดำเนินงาน ต่าง ๆ เหมืองข้อมูลเป็นศาสตร์ที่ผสมผสานระหว่างสถิติ ปัญญาประดิษฐ์และการรู้จำรูปแบบ และการจัดการข้อมูลหรือ ฐานข้อมูลเข้าไว้ด้วยกัน หนังสือเล่มนี้อธิบายหลักการทำงาน เหมืองข้อมูลและเทคนิคเหมืองข้อมูล ต่าง ๆ จึงเหมาะกับบุคคลทั่วไปที่สนใจศึกษาการทำงานของเหมืองข้อมูลเพื่อเป็นทักษะเพิ่มเติมหรือ เพื่อต้องการนำเหมืองข้อมูลไปประยุกต์ใช้กับงานวิจัย หรือเพื่อสกัดสารสนเทศที่สำคัญจากข้อมูล ขนาดใหญ่และนำสารสนเทศที่ได้ไปประยุกต์ใช้ให้เกิดประโยชน์ โดยผู้อ่านจะได้เรียนรู้การทำงานของเทคนิคเหมืองข้อมูลที่สำคัญ เช่น การค้นหาเพื่อนบ้านใกล้ที่สุด k ตัว เทคนิคต้นไม้ตัดสินใจ เทคนิค การสร้างกฎเพื่อการจำแนกตัวจำแนกเบย์อย่างง่าย โครงข่ายประสาทเทียม การวิเคราะห์การจัดกลุ่มแบบลำดับขั้นและ แบบเคมีน การวิเคราะห์กฎความสัมพันธ์ด้วยขั้นตอนอะพริออรี การวิเคราะห์การถดถอย เป็นต้น โดยผู้อ่านจะเข้าใจการทำงานของแต่ละเทคนิคอย่างเป็นขั้นตอนซึ่งจะเป็น พื้นฐานในการเรียนรู้เทคนิคเหมืองข้อมูลที่มีความซับซ้อนมากยิ่งขึ้นต่อไปได้นอกจากนี้ผู้อ่านยังสามารถ ใช้โปรแกรมเหมืองข้อมูลเวกา (Weka) ซึ่งเป็นโปรแกรมที่ใช้งานสำหรับศึกษาการทำงานของเทคนิคเหมืองข้อมูลและการประยุกต์ใช้เพื่องานวิจัย



300
บาท

การศึกษาทางวิทยาการระบาด

Epidemiological Studies

ผู้แต่ง : รศ. ดร.อาจันต์ สงทับ

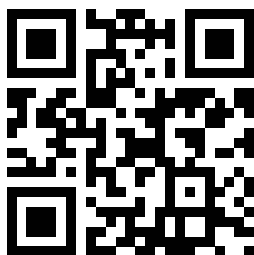
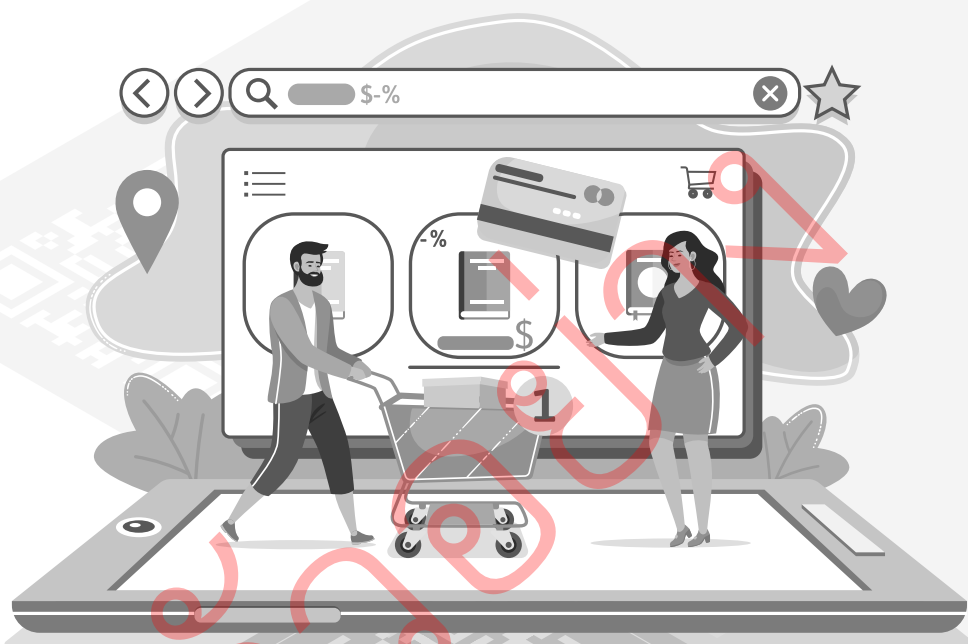
ความรู้ด้านวิทยาการระบาดหรือระบาดวิทยา (Epidemiology) ได้ก้าวหน้าไปมากตั้งแต่อดีตจนถึงปัจจุบัน จนกระทั่งถือว่าวิทยาการระบาดเป็นแขนงหนึ่งที่สำคัญยิ่งของสาขาด้านวิทยาศาสตร์สุขภาพ บุคลากรด้านสุขภาพทุกสาขาที่เกี่ยวข้องจำเป็นต้องศึกษาวิทยาการระบาดสำหรับนำมาประยุกต์ใช้ในงานของตนเพื่อให้สามารถแก้ไขปัญหาสุขภาพของประชาชนได้อย่างมีประสิทธิภาพในการดำเนินงานด้านสุขภาพ การนำหลักการศึกษาวินิจฉัยวิทยาการระบาดมาประยุกต์ใช้นั้นต้องอาศัยเครื่องมืออาจเป็นหนังสือหรือตัวบุคคลที่มีประสบการณ์ในการชี้แนะแนวทางให้การดำเนินการวิจัย ประสบผลสำเร็จ ซึ่งหนังสือเล่มนี้มุ่งเน้นเรื่องการศึกษาทางวิทยาการระบาดแต่ละชนิดที่บุคลากรด้านสุขภาพสามารถนำไปใช้เป็นแนวทางดำเนินการวิจัยเพื่อให้เกิดองค์ความรู้ใหม่อันนำมาแก้ไขปัญหาสุขภาพ ซึ่งจะช่วยในการพัฒนาคุณภาพชีวิตของมวลมนุษยชาติ



สำนักพิมพ์
มหาวิทยาลัยนเรศวร

สั่งซื้อหนังสือออนไลน์

จัดส่งถึงบ้านสะดวกรวดเร็ว



สั่งซื้อทันที

กรณีต้องการสั่งซื้อหนังสือปริมาณมาก หรือเข้าชั้นเรียนติดต่อได้ที่
ฝ่ายจัดจำหน่ายสำนักพิมพ์มหาวิทยาลัยนเรศวร

✉ nuph@nu.ac.th

📘 สำนักพิมพ์มหาวิทยาลัยนเรศวร

☎ 0 5596 8833-8836

🐦 [nu_publishing](https://www.facebook.com/nu_publishing)



NUPH
online store

www.nupress.grad.nu.ac.th