



คู่มือ Coding ภาษา

PYTHON

ฉบับสมบูรณ์ 3rd Edition

- 🌈 อ่านเข้าใจง่าย มีตัวอย่างประกอบพร้อมคำอธิบายทุกหัวข้อ
- 🌈 แนะนำการเขียนโปรแกรมเกี่ยวกับ Data Science, Business Intelligence, AI และ Open Government Data เบื้องต้น
- 🌈 มีตัวอย่างประยุกต์การพัฒนาโปรแกรมที่หลากหลาย
- 🌈 เรียบเรียงจากอาจารย์ที่มีประสบการณ์สอนกว่า 25 ปี



ไฟล์ตัวอย่างภายในเล่ม
<https://serazu.com/000000000000>

เหมาะสำหรับนักเรียน นักศึกษา
และผู้ที่สนใจเขียนโปรแกรมภาษา Python



พศ.สุดา เรียบมนตรี

สารบัญ

บทที่ 1

รู้จักกับภาษา Python (Introduction to Python)	1
แนะนำภาษา Python.....	1
การทำงานของโปรแกรมภาษา Python.....	2
แนะนำเครื่องมือสำหรับเขียนโปรแกรมภาษา Python.....	3
การติดตั้งโปรแกรม Python.....	3
การเรียกใช้งานโปรแกรม Python	5
การติดตั้งและเรียกใช้งานโปรแกรม PyCharm.....	8
แบบฝึกหัด	17

บทที่ 2

เริ่มต้นเขียนโปรแกรมภาษา Python (Introduction to Python Programming).....	19
ขั้นตอนการเขียนโปรแกรมภาษา Python	19
ขั้นที่ 1 วิเคราะห์ปัญหาของโปรแกรมที่ต้องการพัฒนา	20
ขั้นที่ 2 ออกแบบการทำงานของโปรแกรม	20
ขั้นที่ 3 เริ่มต้นสร้างและเขียนโปรแกรม	21
ขั้นที่ 4 ทดสอบการทำงานของโปรแกรม	23
การเขียนคำอธิบายโปรแกรม (Comment)	24
แบบฝึกหัด	25

บทที่ 3

ตัวแปร ชนิดข้อมูล นิพจน์ และตัวดำเนินการ (Variable, Data Type, Expression and Operator)	27
รู้จักกับตัวแปร (Variable)	27
กฎการตั้งชื่อตัวแปรในภาษา Python	27
ชนิดของข้อมูล (Data Type)	28
ข้อมูลชนิดตัวเลข (Number)	28
ข้อมูลชนิดค่าความจริง (Boolean)	29
ข้อมูลชนิด None.....	29



ข้อมูลแบบเรียงลำดับ (Sequence).....	30
ข้อมูลชนิดเซต (Set)	31
ข้อมูลชนิดดิกชันนารี (Dictionary).....	32
การตรวจสอบชนิดของข้อมูล (Data Type Checking)	32
การแปลงชนิดของข้อมูล (Data Type Conversion).....	34
นิพจน์ (Expression).....	34
ตัวดำเนินการ (Operator).....	35
ตัวดำเนินการกำหนดค่า (Assignment Operator).....	35
ตัวดำเนินการทางคณิตศาสตร์ (Arithmetic Operator)	37
ตัวดำเนินการเปรียบเทียบ (Comparison Operator)	38
ตัวดำเนินการทางตรรกศาสตร์ (Logical Operator)	39
ตัวดำเนินการระดับบิต (Bitwise Operator)	39
ตัวดำเนินการแบบเป็นสมาชิก (Membership Operator).....	42
ตัวดำเนินการแบบแสดงเอกลักษณ์ (Identity Operator).....	43
ลำดับความสำคัญของตัวดำเนินการ (Operator of Precedence).....	44
แบบฝึกหัด	46

บทที่ 4

การแสดงผลและการรับข้อมูล (Data Output and Input)	47
การแสดงผลข้อมูลด้วยฟังก์ชัน print()	47
การใช้เครื่องหมาย + คั่นระหว่างข้อความ.....	47
การใช้เครื่องหมาย , คั่นระหว่างข้อความ.....	48
การใช้เครื่องหมาย * สำหรับแสดงผลข้อความที่ซ้ำกัน.....	49
การใช้ตัวอักษรพิเศษ.....	50
การใช้รหัสการแสดงผล.....	51
ฟังก์ชัน format().....	55
การรับข้อมูลด้วยฟังก์ชัน input()	57
การรับข้อมูล 1 ตัวแปร	57
การรับข้อมูลมากกว่า 1 ตัวแปร.....	58
แบบฝึกหัด	60

บทที่ 5

คำสั่งควบคุมทิศทางการทำงานของโปรแกรม (Control Statement).....61

คำสั่งควบคุมแบบตามลำดับ (Sequence Control Statement) 61

คำสั่งควบคุมแบบมีทางเลือก (Selection Control Statement) 62

คำสั่ง if : คำสั่งควบคุมให้โปรแกรมทำงานหรือไม่ทำงานในชุดคำสั่งที่กำหนด.....63

คำสั่ง if...else : คำสั่งควบคุมให้โปรแกรมเลือกทำงานในทางเลือกใดทางเลือกหนึ่งจาก 2 ทางเลือก 65

คำสั่ง if...elif...else : คำสั่งควบคุมให้โปรแกรมเลือกทำงานในทางเลือกใดทางเลือกหนึ่งจากหลายทางเลือก..... 69

คำสั่งควบคุมแบบทำซ้ำ (Iteration Control Statement)..... 72

คำสั่ง while : คำสั่งควบคุมแบบทำซ้ำด้วยจำนวนรอบที่ไม่แน่นอน 73

คำสั่ง while in range : คำสั่งควบคุมแบบทำซ้ำด้วยจำนวนรอบที่แน่นอน 74

คำสั่ง while True : คำสั่งควบคุมแบบทำซ้ำโดยทำอย่างน้อย 1 รอบ 76

คำสั่ง while else : คำสั่งควบคุมส่วนที่อยู่นอกเหนือการทำซ้ำ 78

คำสั่ง for : คำสั่งควบคุมแบบทำซ้ำด้วยจำนวนรอบที่แน่นอน..... 80

คำสั่ง break : คำสั่งออกจากการทำงานในลูปทันที..... 85

คำสั่ง continue : คำสั่งบังคับให้ข้ามไปทำงานรอบต่อไปทันที 87

คำสั่ง pass : คำสั่งบังคับให้ผ่านไปทำงานในชุดคำสั่งถัดไป 88

แบบฝึกหัด 90

บทที่ 6

ข้อมูลชนิดเรียงลำดับ เซต และดิกชันนารี

(Sequence, Set and Dictionary Data Type)..... 93

ข้อความ (String) 93

การเข้าถึงข้อมูลชนิด String 94

การเชื่อมต่อ และการทำซ้ำกับข้อมูลชนิด String..... 95

การดำเนินการกับข้อมูลชนิด String..... 95

ฟังก์ชันสำหรับข้อมูลชนิด String..... 96

ลิสต์และทูเปิล (List and Tuple) 99

การเข้าถึงข้อมูลชนิด List และ Tuple 100

ฟังก์ชันสำหรับข้อมูลชนิด List และ Tuple 102

เซต (Set) 106

ฟังก์ชันสำหรับข้อมูลชนิด Set..... 107



ข้อมูลชนิดดิกชันนารี (Dictionary)	110
การเข้าถึงข้อมูลชนิด Dictionary.....	110
ฟังก์ชันสำหรับข้อมูลชนิด Dictionary	111
แบบฝึกหัด	113

บทที่ 7

ฟังก์ชัน (Function)	115
ไลบรารีฟังก์ชัน (Library function)	115
Built in function	115
Module function	116
ฟังก์ชันในโมดูล datetime.....	118
ฟังก์ชันในโมดูล math	119
ฟังก์ชันที่เขียนขึ้นเอง (User defined function)	121
ตัวแปร global	122
ตัวแปร local	124
อาร์กิวเมนต์ (Argument) และพารามิเตอร์ (Parameter)	125
รูปแบบการเขียนฟังก์ชัน.....	128
กรณีที่ฟังก์ชันไม่มีการรับพารามิเตอร์และไม่มีการคืนค่า	128
กรณีที่ฟังก์ชันมีการรับพารามิเตอร์แต่ไม่มีการคืนค่า.....	130
กรณีที่ฟังก์ชันไม่มีการรับพารามิเตอร์แต่มีการคืนค่า.....	131
กรณีที่ฟังก์ชันมีการรับพารามิเตอร์และมีการคืนค่า.....	133
ฟังก์ชันโมดูล.....	138
ฟังก์ชันแลมด้า (Lambda function).....	140
แบบฝึกหัด	143

บทที่ 8

การจัดการข้อผิดพลาด (Exception Handling)	145
รู้จักกับ Exception	145
ประเภทของ Exception.....	146
การใช้งานคำสั่ง try ... except	146
การใช้ else ร่วมกับคำสั่ง try ... except.....	148
การใช้ finally ร่วมกับคำสั่ง try ... except.....	154
แบบฝึกหัด	158

บทที่ 9**Graphic User Interface (GUI) 159**

รู้จัก GUI Component	159
การสร้าง GUI.....	159
ประเภทของ GUI	160
Window : หน้าต่างสำหรับจัดวางคอมโพเนนต์	160
Frame : เฟรมหรือกรอบแสดงผล.....	161
Button : ปุ่ม และ PhotoImage : รูปภาพ.....	162
Label และ Message : แสดงผลข้อความ	164
Entry และ Text : รับและแสดงผลข้อความ.....	166
Radiobutton : ตัวเลือกที่เลือกได้หนึ่งตัวเลือก	167
Checkbox : ตัวเลือกที่เลือกได้มากกว่า 1 ตัวเลือก.....	169
Listbox : รายการข้อมูล และ Scrollbar.....	172
Combobox : รายการข้อมูลที่เลือกได้ 1 รายการ.....	173
แบบฝึกหัด	175

บทที่ 10**ตัวอย่างโปรแกรมประยุกต์ และ Event Handling 177**

ตัวอย่างโปรแกรมประยุกต์	177
Event Handling.....	186
การจัดการ command ด้วย Lambda function	193
แบบฝึกหัด	197

บทที่ 11**การเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming) 199**

รู้จักคลาส (Class) และออบเจกต์ (Object)	199
ประเภทของแอตทริบิวต์	201
ประเภทของเมธอด	203
ระดับการเข้าถึงข้อมูล (Access Modifier) Public, Private และ Protected	207
รู้จักและใช้งาน Encapsulation	210
รู้จักและใช้งาน Inheritance และ Polymorphism	216
แบบฝึกหัด	223



บทที่ 12	ทำงานกับไฟล์ (File I/O).....	225
	รู้จักกับ File.....	225
	การเขียนข้อมูลลงไฟล์.....	225
	การอ่านข้อมูลจากไฟล์.....	227
	ตัวอย่างการประยุกต์ใช้ไฟล์กับโปรแกรมจัดการยืม-คืนหนังสือ.....	229
	โครงสร้างข้อมูล.....	229
	โครงสร้างโปรแกรม.....	230
	แบบฝึกหัด.....	252
บทที่ 13	การเขียนโปรแกรมจัดการฐานข้อมูล (Database Programming).....	253
	การเชื่อมต่อกับฐานข้อมูล SQLite.....	253
	ตัวอย่างการประยุกต์ใช้ฐานข้อมูล SQLite กับการจัดการข้อมูลการโอนย้าย	
	สินค้าในคลังสินค้า.....	254
	เริ่มต้นเขียนโปรแกรมกับฐานข้อมูล SQLite.....	254
	การเชื่อมต่อกับฐานข้อมูล Microsoft Access.....	272
	การติดตั้งโมดูล pyodbc และโมดูล pypyodbc.....	272
	การประยุกต์ใช้ฐานข้อมูล Microsoft Access กับการจัดการสมัครสมาชิก.....	274
	เริ่มต้นเขียนโปรแกรมกับฐานข้อมูล Microsoft Access.....	275
	การเชื่อมต่อกับฐานข้อมูล MySQL.....	289
	การติดตั้งโปรแกรม XAMPP.....	289
	การประยุกต์ใช้ฐานข้อมูล MySQL กับการจัดการร้านเบเกอรี่.....	293
	เริ่มต้นเขียนโปรแกรมกับฐานข้อมูล MySQL.....	293
	แบบฝึกหัด.....	310
บทที่ 14	การสร้าง GUI Form ด้วย PyQt และ Qt Designer	311
	การติดตั้งโมดูล PyQt และ Qt Designer.....	311
	การใช้งาน Qt Designer.....	312
	การแปลง GUI Form เป็นไฟล์ภาษา Python.....	316
	การเขียนคำสั่งเพื่อใช้งานโมดูล PyQt6.....	320
	ตัวอย่างโปรแกรมประยุกต์.....	322

ตัวอย่างการประยุกต์ใช้ฐานข้อมูล SQLite กับการจัดการข้อมูลการรับสินค้า	
เข้าในคลังสินค้า.....	322
แบบฝึกหัด	367

บทที่ 15	การสร้างรายงานด้วย ReportLab	369
	การสร้างรายงานด้วยคลาส Paragraph.....	369
	การสร้างรายงานด้วยคลาส Canvas.....	372
	การสร้างรายงานด้วยคลาส Table	377
	การสร้าง Chart ในรายงาน	381
	แบบฝึกหัด	387

บทที่ 16	Python กับการประยุกต์ใช้ในงานด้านต่างๆ	389
	การเขียนเว็บแอปพลิเคชันด้วยโมดูล Django	389
	ขั้นตอนการสร้างโปรเจกต์ Django.....	390
	การเขียนโปรแกรมเว็บแอปพลิเคชัน.....	393
	การส่งข้อความผ่านเครือข่ายด้วยโมดูล socket	400
	ขั้นตอนการทำงานระหว่าง Client-Server Mode.....	400
	การเขียนโปรแกรมระหว่าง Client-Server Mode.....	400
	แบบฝึกหัด	405

บทที่ 17	Python กับงานด้าน Data Science	407
	ความหมายของ Data Science	407
	ขั้นตอนของ Data Science.....	408
	โมดูล pandas.....	410
	การเขียนโปรแกรมจัดการข้อมูลด้วยโมดูล pandas.....	410
	รู้จักกับความซ้ำกันของข้อมูล และการแทนที่ข้อมูลที่สูญหาย.....	415
	การเขียนโปรแกรมจัดการความซ้ำกันของข้อมูลและข้อมูลสูญหายด้วยโมดูล pandas	416
	การวิเคราะห์และนำเสนอข้อมูลในรูปแบบของกราฟแผนภูมิด้วยโมดูล matplotlib	420
	การเขียนโปรแกรมวิเคราะห์และนำเสนอข้อมูลด้วยโมดูล matplotlib	420



การจัดการข้อมูลด้วยโมดูล numpy	426
การจัดการข้อมูลในรูปแบบอาร์เรย์ด้วยโมดูล numpy	426
การเขียนโปรแกรมจัดการข้อมูลด้วยโมดูล numpy.....	427
ไฟล์ JSON	429
การเขียนโปรแกรมนำเสนอข้อมูลจากการนำเข้าไฟล์ JSON	430
การเขียนโปรแกรมนำเสนอข้อมูลไฟล์ JSON จากเว็บไซต์	432
แบบฝึกหัด	434

บทที่ 18 Python กับงานด้าน Business Intelligence 437

องค์ประกอบของ Business Intelligence	438
Machine Learning.....	439
การติดตั้งโปรแกรม graphviz.....	439
แบบฝึกหัด	449

บทที่ 19 การใช้งาน Python บน Kaggle Platform 451

การใช้งาน Kaggle Platform	452
การสมัครสมาชิก.....	452
การจัดการชุดข้อมูล Datasets	453
การเขียนโค้ดโปรแกรม	454
การเขียนโปรแกรมบน Kaggle Platform	455
แบบฝึกหัด	460

บทที่

04 >>

การแสดงผล และการรับข้อมูล (Data Output and Input)



ในการเขียนโปรแกรม การรับและแสดงผลข้อมูลถือว่าเป็นพื้นฐานแรกๆ ที่ไม่ว่าจะใช้โปรแกรมภาษาใดๆ ก็ต้องให้ความสำคัญ เนื่องจากไม่ว่าโปรแกรมที่เขียนนั้นจะถูกต้องและมีประสิทธิภาพเพียงใด แต่หากไม่สามารถรับหรือแสดงผลข้อมูลได้ถูกต้อง โปรแกรมที่เขียนดีนั้นก็คงไม่สามารถใช้งานได้อยู่ดี

ในบทนี้ผู้อ่านจะได้เรียนรู้การรับและแสดงผลข้อมูล ซึ่งเป็นความรู้พื้นฐานที่สำคัญ

การแสดงผลข้อมูลด้วยฟังก์ชัน print()

ฟังก์ชัน print() เป็นเมธอดที่ใช้สำหรับแสดงผลข้อมูลทางจอภาพ มีรูปแบบการใช้งานดังนี้

การใช้เครื่องหมาย + คั่นระหว่างข้อความ

มีรูปแบบการใช้งานดังนี้

```
print(argument_1 + argument_2 + ... + argument_n)
```

โดยที่

argument_1, argument_2, argument_n	}	เป็นข้อมูลที่ต้องการแสดงผล อาจจะเป็นข้อความ ตัวแปร หรือนิพจน์ ซึ่งสามารถเขียนต่อกันได้ โดยใช้เครื่องหมาย "+"
---	---	--

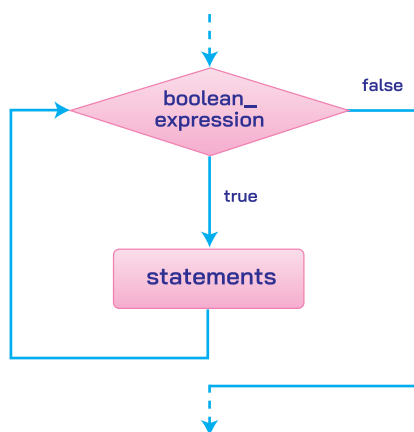
คำสั่ง while : คำสั่งควบคุมแบบทำซ้ำด้วยจำนวนรอบที่ไม่แน่นอน

คำสั่ง **while** เป็นคำสั่งควบคุมการทำงานของโปรแกรมแบบทำซ้ำ ที่มีการตรวจสอบเงื่อนไขที่เป็นนิพจน์ทางตรรกศาสตร์ก่อนจะทำงานที่ชุดคำสั่งภายในลูป โดยโปรแกรมจะทำงานที่ชุดคำสั่งภายในลูปก็ต่อเมื่อนิพจน์ที่กำหนดให้เป็นจริง และเมื่อโปรแกรมทำงานที่ชุดคำสั่งภายในลูปเสร็จสิ้น โปรแกรมจะตรวจสอบนิพจน์ใหม่อีกครั้ง และจะหยุดการทำซ้ำเมื่อนิพจน์ที่กำหนดให้เป็นเท็จ (โดยจำนวนรอบของการทำซ้ำจะไม่แน่นอน ขึ้นอยู่กับเงื่อนไขที่ใช้ในการตรวจสอบ) และโปรแกรมจะออกจากลูปไปทำงานที่คำสั่งถัดไปทันที ซึ่งมีรูปแบบการทำงานดังนี้

```
while boolean_expression:  
    statements
```

โดยที่ **boolean_expression** เป็นนิพจน์ที่เป็นเงื่อนไขทางตรรกศาสตร์ไว้ตรวจสอบเงื่อนไขว่าเป็นจริงหรือเท็จก่อนการทำงานเสมอ
statements เป็นชุดคำสั่งที่ต้องการให้มีการทำซ้ำ

จากลักษณะการทำงานของคำสั่ง สามารถอธิบายด้วย Flowchart ได้ดังนี้



ตัวอย่างโปรแกรมที่ 5-5 โปรแกรมการใช้คำสั่ง **while** เพื่อคำนวณผลรวมตั้งแต่ 1 ถึง 20
ตัวอย่างนี้เป็นโปรแกรมที่มีการทำงานซ้ำๆ โดยใช้คำสั่ง **while** ตรวจสอบเงื่อนไขของค่าตัวเลขที่ใช้ในการคำนวณผลรวมซ้ำๆ กัน เป็นผลรวมสะสมตั้งแต่ 1 ถึง 20

การทำงานของโปรแกรมจึงเริ่มต้นจากการคำนวณผลรวมสะสมเข้าไปเรื่อยๆ ตั้งแต่ค่าเริ่มต้นที่กำหนดคือ 1 จนถึงค่าสิ้นสุดที่ใช้ตรวจสอบในเงื่อนไขคือ 20 และแสดงผลลัพธ์ของโปรแกรกดังนี้



ฟังก์ชันสำหรับข้อมูลชนิด List และ Tuple

นอกจากการดำเนินการดังกล่าวข้างต้นแล้ว ภาษา Python ยังได้เตรียมฟังก์ชันสำหรับ List และ Tuple ที่น่าสนใจดังนี้

ฟังก์ชัน	คำอธิบาย
list1.count(x) หรือ tuple1.count(x)	ให้ค่าเป็นจำนวนข้อมูล x ที่พบใน list1 หรือ tuple1
len(list1) หรือ len(tuple1)	ให้ค่าเป็นจำนวนข้อมูลทั้งหมดของ list1 หรือ tuple1
list1.index(x) หรือ tuple1.index(x)	ให้ค่าเป็นลำดับที่ของข้อมูล x ที่พบใน list1 หรือ tuple1
list1.append(x)	ให้ค่าเป็น list1 ที่ต่อท้ายด้วยข้อมูล x
list1.remove(x)	ให้ค่าเป็น list1 ที่ลบข้อมูล x ออก
list1.pop(idx)	ให้ค่าเป็น list1 ที่ลบข้อมูลลำดับที่ idx ออก
list1.clear()	ให้ค่าเป็น list1 ที่ลบข้อมูลทั้งหมดออก

ตัวอย่างเช่น กำหนดให้ `list1 = ["Tiger", 20, "Smile", 20, "Snail", 30]` และ `tuple1 = ("Tiger", 20, "Smile", 20, "Snail", 30)`

ตัวอย่าง	ผลลัพธ์	คำอธิบาย
list1.count(20)	2	พบข้อมูล "20" ในข้อความ list1 จำนวน 2 ตัว
tuple1.count(30)	1	พบข้อมูล "30" ในข้อความ tuple1 จำนวน 1 ตัว
len(list1)	6	จำนวนข้อมูลใน list1 รวมเท่ากับ 6 ตัว
len(tuple1)	6	จำนวนข้อมูลใน tuple1 รวมเท่ากับ 6 ตัว
list1.index(20)	1	ลำดับที่ของข้อมูลที่มีค่า "20" ที่พบในข้อความ list1 เท่ากับ 1
tuple1.index(30)	5	ลำดับที่ของข้อมูลที่มีค่า "30" ที่พบในข้อความ tuple1 เท่ากับ 5
list1.append("Coconut") list1.append(22)	["Tiger", 20, "Smile", 20, "Snail", 30, "Coconut", 22]	ข้อความ list1 เดิม ที่ต่อท้ายด้วย "Coconut" และ 22



ประเภทของ Exception

ในหัวข้อนี้ผู้เขียนจะอธิบายประเภทของ Exception ซึ่งเป็น Built in Exception ในภาษา Python ที่ผู้อ่านมีโอกาสพบเจอบ่อยๆ เพื่อให้ผู้อ่านสามารถจัดการกับข้อผิดพลาดที่อาจเกิดขึ้นได้อย่างถูกต้อง โดยสามารถแสดงประเภทของ Exception ที่น่าสนใจพร้อมคำอธิบายได้ดังนี้

Exception	คำอธิบาย
IndexError	ความผิดพลาดที่เกิดจากการเข้าถึงข้อมูลเกินขอบเขตข้อมูลที่กำหนดไว้
KeyError	ความผิดพลาดที่เกิดจากการเข้าถึง key ที่ไม่มีอยู่ในข้อมูล Dictionary
NameError	ความผิดพลาดที่เกิดจากการเข้าถึงตัวแปรที่ไม่มีอยู่ในขอบเขตของ local หรือ global
TypeError	ความผิดพลาดที่เกิดจากการใช้ฟังก์ชันที่ไม่สอดคล้องกับชนิดของข้อมูล
ValueError	ความผิดพลาดที่เกิดจากการเข้าถึงตัวแปรที่ค่าของข้อมูลไม่สอดคล้องกับชนิดของข้อมูล
ZeroDivisionError	ความผิดพลาดที่เกิดจากการหารด้วยศูนย์

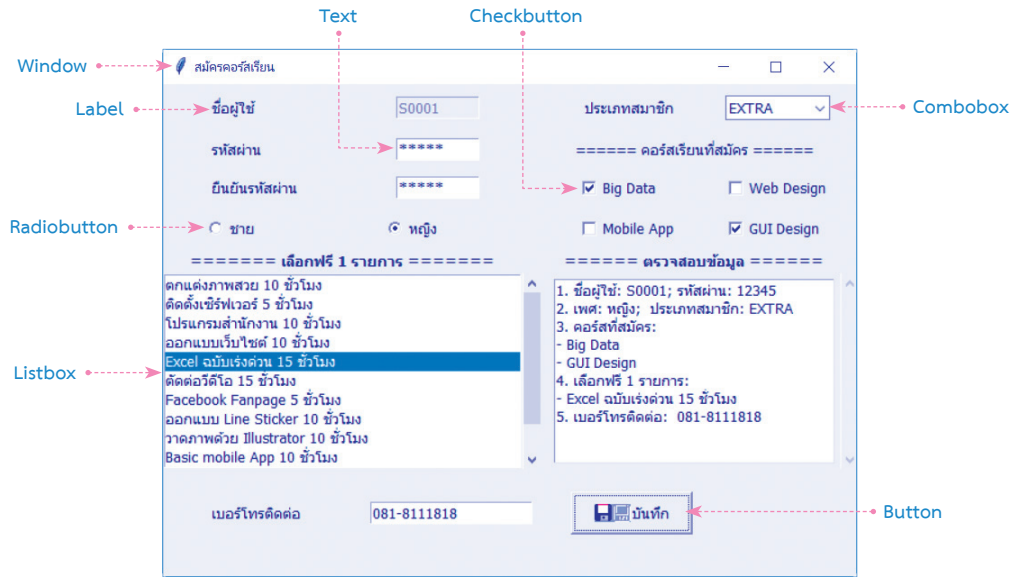
นอกจาก Exception ที่กล่าวข้างต้นแล้วนั้น ผู้อ่านยังสามารถศึกษาประเภทของ Exception เพิ่มเติมได้ที่ <https://docs.python.org/3/library/exceptions.html#Exception>

การใช้งานคำสั่ง try ... except

หลักการทำงานของคำสั่ง try ... except จะคล้ายกับคำสั่ง if คือ ถ้าเกิดข้อผิดพลาดในคำสั่งที่ตามหลังคำสั่ง try โปรแกรมจะตรวจสอบที่คำสั่ง except ว่าเป็นข้อผิดพลาดประเภทใด และทำงานที่ชุดคำสั่งหลังคำสั่ง except ที่ตรงกับข้อผิดพลาด ซึ่งมีรูปแบบการใช้งานดังนี้

```
try:
    statements_1
except TheException as e:
    statements_2
```

โดยที่	<code>statements_1</code>	เป็นชุดคำสั่งที่ต้องการจะตรวจจับข้อผิดพลาดในกรณีที่โปรแกรมทำงาน
	<code>TheException</code>	เป็นประเภทของ Exception ที่ต้องการตรวจจับ
	<code>e</code>	เป็นตัวแปรที่เก็บค่าประเภทของ Exception
	<code>statements_2</code>	เป็นชุดคำสั่งที่กำหนดให้ทำงานเมื่อเกิดข้อผิดพลาดตรงตาม Exception



คอมโพเนนต์ต่างๆ

ประเภทของ GUI

ในหัวข้อนี้ผู้อ่านจะได้รู้จัก GUI ประเภทต่างๆ ที่ใช้สำหรับสร้างส่วนติดต่อกับผู้ใช้งานแบบกราฟิก ดังนี้

Window : หน้าต่างสำหรับจัดวางคอมโพเนนต์

Window เป็นคอมโพเนนต์ประเภท Container ซึ่งเป็นองค์ประกอบแรกของ GUI ที่ผู้อ่านจะต้องสร้างขึ้นมาเพื่อเป็นหน้าต่างสำหรับใช้จัดวางคอมโพเนนต์อื่นๆ ผู้อ่านสามารถสร้างหน้าต่างจากคลาส Tk และจะต้องกำหนดคุณสมบัติหลักๆ ให้กับหน้าต่างนี้ด้วย เพื่อความเข้าใจ ผู้อ่านสามารถศึกษาการใช้งานจากตัวอย่างโปรแกรมต่อไปนี้

ตัวอย่างโปรแกรมที่ 9-1 โปรแกรมการสร้าง Window มีโค้ดดังนี้

1	from tkinter import *
2	
3	root = Tk()
4	root.title("GUI Example")
5	root.geometry("300x250")
6	root.option_add("*font", 'tahoma 10 bold')
7	root.mainloop()

ระดับการเข้าถึงข้อมูล (Access Modifier) Public, Private และ Protected

Access Modifier คือ คีย์เวิร์ดที่ใช้กำหนดระดับการเข้าถึงข้อมูล เป็นกลไกของการกำหนดระดับการเข้าใช้งานสมาชิกของคลาส เพื่อรักษาความปลอดภัยและป้องกันการเปลี่ยนแปลงข้อมูลภายในคลาส ซึ่งอาจเกิดขึ้นได้ในขณะที่มีการนำคลาสดังกล่าวไปใช้งานซ้ำ โดยภาษา Python มีการแบ่งระดับของ Access Modifier เป็น 3 ระดับคือ

1. Public : มีขอบเขตการเข้าใช้งานได้จากทุกคลาส

Public เป็นการกำหนดระดับ Access Modifier ให้กับแอตทริบิวต์หรือเมธอดที่ต้องการให้คลาสใดๆ เรียกใช้งานก็ได้ แต่ในบางข้อมูล เช่น ข้อมูลเงินเดือนอาจจะถูกเปลี่ยนแปลงค่าได้ง่าย หากกำหนดการเข้าถึงเป็นแบบ Public ก็จะทำให้เกิดความไม่ปลอดภัยกับข้อมูล ดังนั้น การกำหนดระดับ Access Modifier ให้ข้อมูลเงินเดือน จึงควรเป็นแบบ Private เพื่อเป็นการซ่อนข้อมูลไว้ไม่ให้คลาสอื่นสามารถเปลี่ยนแปลงแก้ไขค่าได้โดยตรง

2. Private : มีขอบเขตการเข้าใช้งานข้อมูลภายในคลาสเดียวกันเท่านั้น

Private เป็นการกำหนดระดับ Access Modifier ให้กับแอตทริบิวต์หรือเมธอดที่ต้องการใช้งานภายในคลาสเท่านั้น ไม่อนุญาตให้คลาสอื่นเรียกใช้งาน ผู้อ่านสามารถศึกษาการกำหนดระดับ Access Modifier เป็น Private เพิ่มเติมได้จากตัวอย่างโปรแกรมในหัวข้อ Encapsulation

3. Protected : มีขอบเขตการเข้าใช้งานข้อมูลภายในคลาสเดียวกันและคลาสที่สืบทอดกัน

Protected เป็นการกำหนดระดับ Access Modifier ให้แอตทริบิวต์หรือเมธอดที่อนุญาตให้คลาสที่มีความสัมพันธ์จากคลาสแม่สู่คลาสลูกใช้งานเท่านั้น ไม่อนุญาตให้คลาสอื่นเรียกใช้ได้ โดยผู้อ่านสามารถศึกษาเพิ่มเติมได้ในหัวข้อ Inheritance

< TIP >



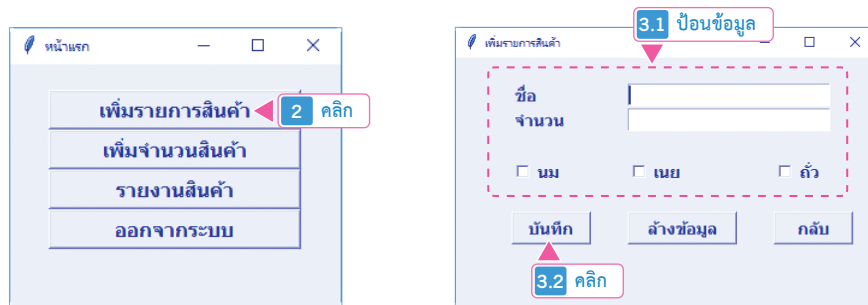
สรุปหลักการ การกำหนดระดับของ Access Modifier ให้กับข้อมูลใดๆ มีแนวคิดในการพิจารณาดังนี้

- **กรณีที่ข้อมูลทำหน้าที่เป็นตัวแปรสาธารณะ**คือ อนุญาตให้คลาสใดๆ เรียกใช้ได้โดยตรง ให้กำหนดเป็นแบบ Public
- **กรณีที่ข้อมูลมีค่าเป็นส่วนตัว**คือ ไม่อนุญาตให้คลาสอื่นมาเปลี่ยนแปลงค่าได้ ให้กำหนดเป็นแบบ Private โดยการกำหนดชื่อข้อมูลให้มีเครื่องหมาย `_` นำหน้าชื่อ
- **กรณีที่ข้อมูลมีความสัมพันธ์ในลักษณะที่มีการสืบทอดจากคลาสแม่สู่คลาสลูก** ให้กำหนดเป็นแบบ Protected โดยการกำหนดชื่อข้อมูลให้มีเครื่องหมาย `_` นำหน้าชื่อ

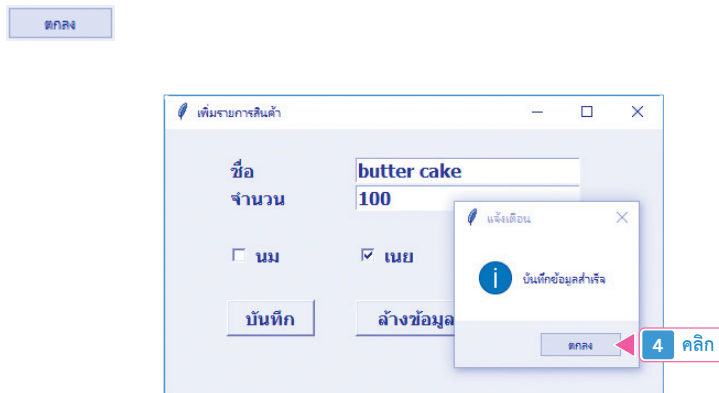
3. หน้าจอ เพิ่มรายการสินค้า

3.1 ป้อนข้อมูลรายละเอียดของรายการสินค้า

3.2 คลิกปุ่ม **บันทึก**

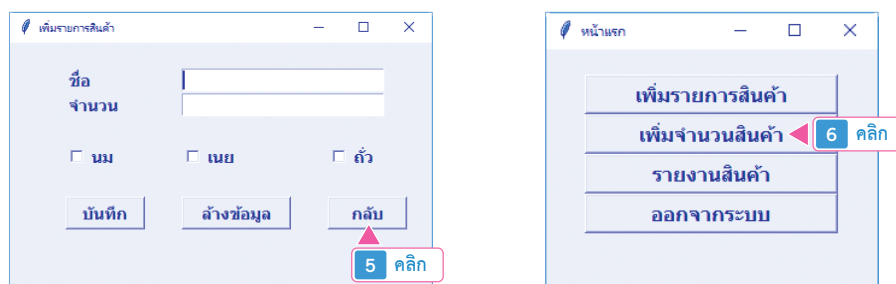


4. โปรแกรมจะบันทึกข้อมูลลงตาราง product และแสดงข้อความแจ้งเตือน แล้วคลิกปุ่ม

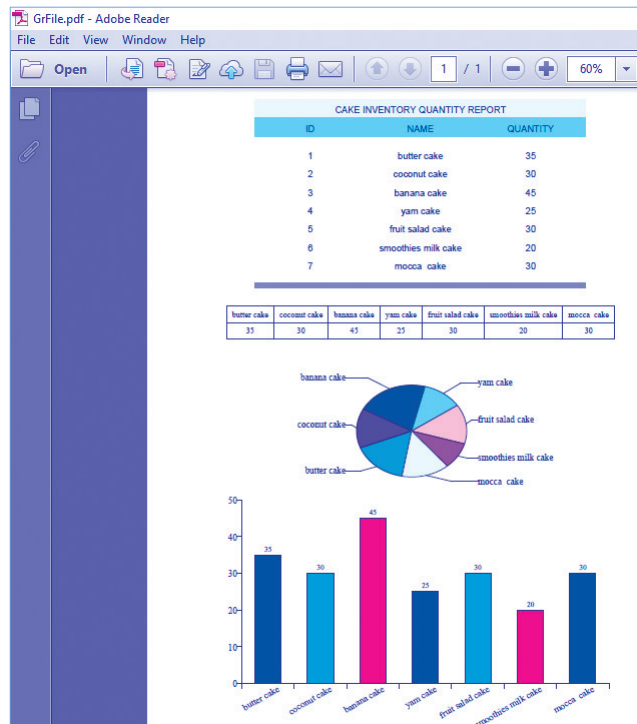


5. โปรแกรมจะเคลียร์ข้อมูลบนหน้าจอ และเมื่อคลิกปุ่ม **กลับ** โปรแกรมจะแสดงหน้าจอ หน้าแรก

6. ที่หน้าจอ หน้าแรก ให้คลิกปุ่ม **เพิ่มจำนวนสินค้า**



3. ข้อมูลถูกนำไปสร้างเป็นไฟล์ GrFile.pdf ซึ่งแสดงผลดังนี้



สรุปท้ายบท

ในบทนี้ผู้อ่านได้เรียนรู้การสร้างรายงานด้วยโมดูล ReportLab ที่สามารถติดตั้งได้โดยง่าย และสามารถสร้างรายงานเพื่อนำเสนอข้อมูลได้จากหลายแหล่ง ทั้งที่เป็นข้อมูลจากฐานข้อมูล ไฟล์ หรือข้อมูลบนหน้าจอ รูปแบบการสร้างรายงานก็ทำได้ทั้งแบบ Paragraph ที่นำข้อมูลไปจัดวางด้วยค่ารูปแบบต่างๆ แบบ Canvas ที่เป็นการวาดวัตถุอ้างอิงพิกัดตำแหน่ง x, y แบบ Table ที่นำเสนอข้อมูลในรูปแบบตาราง และยังสามารถนำเสนอด้วย Chart ต่างๆ ได้ โดยผู้เขียนได้อธิบายด้วยโปรแกรมในรูปแบบต่างๆ ซึ่งผู้อ่านสามารถนำไปประยุกต์ใช้เพื่อการสร้างรายงานได้ตามต้องการ

แบบฝึกหัด

1. จงสร้างรายงานเพื่อแสดงผลข้อมูลรหัสนักศึกษา ชื่อนักศึกษา และคะแนนสอบ โดยใช้ข้อมูลจากตาราง Student ในฐานข้อมูลชื่อ mydata.db ในบทที่ 13
2. จงเขียนโปรแกรมสร้างรายงานรูปแบบของกราฟแท่ง เพื่อแสดงผลคะแนนของนักศึกษาแต่ละคน โดยใช้แหล่งข้อมูลเดียวกับข้อ 1.



การส่งข้อความผ่านเครือข่ายด้วยโมดูล socket

การเขียนโปรแกรมผ่านเครือข่ายแบบเครื่องลูกข่าย-แม่ข่าย (Client-Server) จะประกอบด้วยสถาปัตยกรรม 2 ส่วนคือ

- 1) คอมพิวเตอร์ที่ทำหน้าที่เป็น Server ซึ่งจะทำงานให้บริการอยู่ตลอดเวลา
- 2) คอมพิวเตอร์ที่ทำหน้าที่เป็น Client ที่สามารถร้องขอการเชื่อมต่อเพื่อใช้บริการจาก Server ได้

ขั้นตอนการทำงานระหว่าง Client-Server Mode

ในการทำงานระหว่าง Client-Server Mode นั้น คอมพิวเตอร์ที่ทำหน้าที่เป็น Server จะมีหมายเลข IP Address ประจำเครื่อง และจะเปิดการเชื่อมต่อ (Connection) รอไว้ที่หมายเลข Port ใดๆ เพื่อรอการร้องขอการเชื่อมต่อจากคอมพิวเตอร์ที่ทำหน้าที่เป็น Client ซึ่ง Port ในที่นี้ก็คือช่องทางการเชื่อมต่อระหว่าง Server กับ Client โดยจะถูกกำหนดด้วยเลขจำนวนเต็มระหว่าง 0 - 65535 แต่ผู้อ่านไม่ควรนำหมายเลข Port ตั้งแต่ 0-1024 มาใช้ เนื่องจากหมายเลข Port ดังกล่าวถูกจองไว้เพื่อการใช้งานต่างๆ เช่น หมายเลข 80 ใช้เพื่องานบริการ Web Server, หมายเลข 21 ใช้เพื่องานบริการ FTP Server เป็นต้น ผู้เขียนจึงขอแนะนำให้ใช้หมายเลข Port ตั้งแต่ 1024 ขึ้นไป

การเขียนโปรแกรมเพื่อเชื่อมต่อการทำงานระหว่าง Server กับ Client ในภาษา Python จะใช้โมดูล socket ซึ่งจะทำหน้าที่สร้างลิงค์เพื่อเปิด socket หรือเรียกว่าเป็นการเปิด Port และมีการนำ Buffer มาใช้จัดเรียงข้อมูลไว้ในพื้นที่ที่ได้จัดเตรียมไว้ก่อน เพื่อแก้ปัญหาของการรับส่งข้อมูลเนื่องมาจากความเร็วในการสื่อสารระหว่าง Server กับ Client ที่แตกต่างกัน

การเขียนโปรแกรมระหว่าง Client-Server Mode

เนื่องจากคอมพิวเตอร์ที่ทำหน้าที่เป็น Server จะทำงานให้บริการอยู่ตลอดเวลา การเขียนโปรแกรมฝั่ง Server Mode จึงเริ่มต้นจากการสร้างและผูก socket เข้ากับหมายเลข IP Address ประจำเครื่อง เพื่อเปิดการเชื่อมต่อรอไว้ที่หมายเลข Port ที่กำหนด จนกระทั่งมีคอมพิวเตอร์ที่ทำหน้าที่เป็น Client ร้องขอการเชื่อมต่อผ่านหมายเลข IP Address และหมายเลข Port ที่ตรงกัน การสื่อสารระหว่างกันก็จะเกิดขึ้นและสามารถส่งข้อความถึงกันได้

เพื่อความเข้าใจ ผู้อ่านสามารถศึกษาการเขียนโปรแกรมระหว่าง Client-Server Mode ได้จากตัวอย่างโปรแกรมต่อไปนี้

Python กับงานด้าน Data Science



จากที่ได้กล่าวไว้ในบทที่ 16 ด้วยเรื่องคุณสมบัติของภาษา Python ที่รองรับการพัฒนางานที่หลากหลาย ซึ่งความโดดเด่นอันหนึ่งที่เป็นที่ยอมรับกันโดยทั่วไปก็คือ งานด้านวิทยาศาสตร์ข้อมูล (Data Science) ดังนั้น ในบทนี้ผู้อ่านจะได้เรียนรู้การประยุกต์ใช้ Python กับงานด้าน Data Science

ความหมายของ Data Science

Data Science ภาษาไทยเรียกว่า วิทยาศาสตร์ข้อมูล หมายถึง การนำข้อมูลมาใช้ประโยชน์ เพื่อค้นหาสิ่งที่ยังไม่รู้หรือยังไม่ค้นพบ (Insights) ทำให้เกิดคุณค่าในตัวข้อมูล และนำมาใช้ประโยชน์เพื่อการตัดสินใจได้ โดยมีขั้นตอนในภาพรวมดังนี้

1. การจัดเก็บบันทึกข้อมูล (Data Collection)
2. การจัดการและจัดเตรียมข้อมูล (Data Preparation and Management)
3. การวิเคราะห์ข้อมูล (Data Analysis)
4. การนำข้อมูลมาใช้ประโยชน์ (Data Utilization)



โมดูล pandas

pandas เป็นโมดูลหรือไลบรารีที่ทำให้เราสามารถจัดการกับข้อมูลได้ โดยการอ่านไฟล์ข้อมูลรูปแบบของไฟล์ CSV แล้วเราจะได้ข้อมูลในรูปแบบของ DataFrame ที่ข้อมูลแสดงตามแถวและคอลัมน์ ที่สามารถเข้าถึงได้ตามความต้องการใช้งาน

ในการทำงานของโมดูล pandas เวอร์ชันปัจจุบัน ได้ใช้ PyArrow (The Arrow Python bindings) ทำให้ Pandas ทำงานเร็วขึ้นและประหยัดหน่วยความจำมากขึ้น แทนที่แต่เดิมที่ใช้โครงสร้างข้อมูลของ NumPy เพื่อการจัดการหน่วยความจำ ผู้อ่านจึงต้องติดตั้งโมดูล Pyarrow ด้วย

การเขียนโปรแกรมจัดการข้อมูลด้วยโมดูล pandas

ก่อนทำการเขียนโปรแกรมผู้อ่านจะต้องเตรียมไฟล์ข้อมูล CSV เพื่อใช้งาน ในที่นี้ผู้เขียนขออธิบายโดยใช้ไฟล์ data.csv ซึ่งเป็นข้อมูลยอดขายและค่าคอมมิชชั่นของพนักงานขายจำนวน 20 รายการ ดังนี้

```
data.csv - Notepad
File Edit Format View Help
no,id,sale,percomm,com
1,100128,49800.00 ,3%,996.00
2,200129,320050.00 ,5%,16002.50
3,270130,128750.00 ,3%,5150.00
4,288131,530250.00 ,7%,37117.50
5,315132,96850.50 ,3%,2905.50
6,336133,320000.00 ,5%,16000.00
7,358634,245000.00 ,5%,9800.00
8,381135,105000.00 ,3%,4200.00
9,403636,350000.00 ,7%,14000.00
10,426137,175000.00 ,7%,7000.00
11,448638,315000.00 ,3%,12600.00
12,471139,455000.00 ,5%,9100.00
13,493640,595000.00 ,3%,17850.00
14,516141,735000.00 ,7%,29400.00
15,538642,875000.00 ,3%,26250.00
16,561143,915000.00 ,5%,36600.00
17,583644,1155000.00 ,7%,23100.00
18,606145,295000.00 ,7%,20650.00
19,628646,435000.00 ,3%,57400.00
20,651147,575000.00 ,5%,94150.00
```

โครงสร้างข้อมูล ประกอบด้วย

1. ลำดับที่ (no)
2. รหัสพนักงาน (id)
3. ยอดขาย (sale)
4. เปอร์เซ็นต์ค่าคอมมิชชั่น (percomm)
5. ค่าคอมมิชชั่น (com)



การวิเคราะห์และนำเสนอข้อมูลในรูปแบบของกราฟ แผนภูมิด้วยโมดูล matplotlib

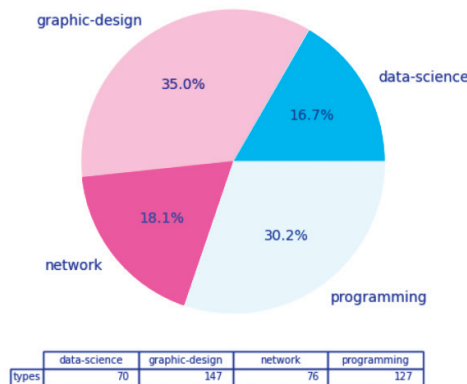
matplotlib เป็นโมดูลหรือไลบรารีที่ใช้สำหรับการนำเสนอข้อมูล ในรูปแบบของกราฟแผนภูมิ สามารถใช้งานร่วมกับข้อมูลในรูปแบบของ DataFrame ที่ได้จากการใช้โมดูล pandas ได้ โดยสามารถนำข้อมูลจาก DataFrame มาจัดทำเป็นแผนภูมิด้วยเมธอด plot ได้ เช่น แผนภูมิวงกลม แผนภูมิเส้น แผนภูมิแท่ง เป็นต้น

การเขียนโปรแกรมวิเคราะห์และนำเสนอข้อมูลด้วยโมดูล matplotlib

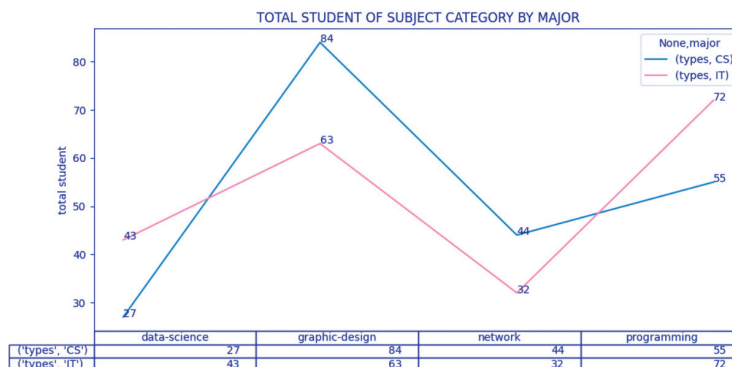
ในส่วนของการวิเคราะห์และนำเสนอข้อมูล ผู้เขียนได้เลือกประเด็นวิเคราะห์จากข้อมูลการลงทะเบียนเรียนรายวิชาเลือกของนักศึกษาใน 3 ประเด็น และแสดงเป็นแผนภูมิดังนี้

1. สัดส่วนเปอร์เซ็นต์ของจำนวนนักศึกษาที่ลงทะเบียนเรียนในแต่ละกลุ่มรายวิชา

PERCENT OF TOTAL STUDENT FOR EACH SUBJECT CATEGORY



2. จำนวนนักศึกษาที่ลงทะเบียนเรียนในแต่ละกลุ่มรายวิชาแยกตามวิชาเอก



ในการวิเคราะห์และนำเสนอข้อมูลนั้น ข้อมูลที่ใช้ส่วนใหญ่มาจากแหล่งข้อมูลที่เกิดจากรายการ transaction ในแต่ละวันของหน่วยงาน เช่น ข้อมูลการลงทะเบียนเรียนวิชาของหลักสูตรในสถานศึกษา หรือข้อมูลปริมาณน้ำฝนที่ตกในแต่ละเดือน แต่ในความเป็นจริงแล้วเราสามารถค้นหาและนำเข้าสู่ข้อมูลได้จากแหล่งอื่นโดยเฉพาะ เช่น ข้อมูลจากศูนย์กลางข้อมูลเปิดภาครัฐ (Open Government Data) ภายใต้ชื่อ “data.go.th” ซึ่งภาครัฐได้จัดเตรียมชุดข้อมูลไว้มากมายและมีความหลากหลาย โดยผู้อ่านสามารถเข้าถึงแหล่งข้อมูลดังกล่าวได้ที่เว็บไซต์ <https://data.go.th/>

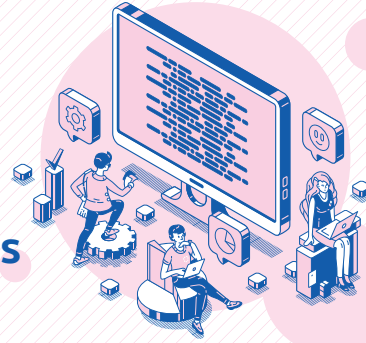


ในหัวข้อต่อไป ผู้เขียนจะได้อธิบายถึงการเขียนโปรแกรมภาษา Python เพื่อดึงข้อมูลจากเว็บไซต์ศูนย์กลางข้อมูลเปิดภาครัฐ ซึ่งได้จัดเตรียมไว้ในรูปแบบของ API เป็นไฟล์ JSON โดยจะอธิบายใน 2 ลักษณะ คือ โหลดไฟล์มาไว้ในเครื่อง และโหลดไฟล์จากการระบุเว็บไซต์ที่อยู่ของชุดข้อมูล

ไฟล์ JSON

ไฟล์ JSON (JavaScript Object Notation) เป็นรูปแบบข้อมูลที่ใช้สำหรับการแลกเปลี่ยนแบ่งปันกัน จะอยู่ในรูปข้อความธรรมดา (Plain Text) ที่เป็นสากล สามารถอ่านเข้าใจได้ ภายใน JSON ประกอบด้วยข้อมูลที่มีรูปแบบการจัดเก็บข้อมูลแบบ key, value โดยการเขียนข้อมูลชนิด JSON มีชื่อฟิลด์ครอบด้วยเครื่องหมาย double quote, เครื่องหมาย : (colon), value แล้วครอบทั้งหมดด้วยเครื่องหมายปีกกา

Python กับงานด้าน Business Intelligence



จากที่กล่าวไว้ในบทที่ 17 เรื่องการประยุกต์ใช้ Python กับงานด้าน Data Science นั้น จะเป็นการศึกษาวิเคราะห์ข้อมูลที่มีมากมายหลากหลายประเภท ใช้การเขียนโค้ดโปรแกรมเพื่อการวิเคราะห์ข้อมูลในมุมมองต่างๆ ที่ไม่ได้เน้นเฉพาะข้อมูลเชิงธุรกิจ ในบทนี้ผู้เขียนจะได้นำเสนอถึงการใช้เครื่องมือและเทคโนโลยีในการวิเคราะห์ข้อมูลธุรกิจเพื่อให้ได้ Insights มาสนับสนุนการพัฒนาธุรกิจให้เป็นธุรกิจอัจฉริยะ (Business Intelligence)

ความหมายของ Business Intelligence

Business Intelligence หรือธุรกิจอัจฉริยะ หมายถึงการใช้เครื่องมือจัดการกับการวิเคราะห์ข้อมูลด้วยกลยุทธ์และเทคโนโลยีต่าง ๆ ที่ช่วยสนับสนุนข้อมูลให้องค์กรสามารถตัดสินใจในกิจกรรมทางธุรกิจได้ โดยมีองค์ประกอบในภาพรวม ดังนี้

1. คลังข้อมูล (Data warehouse)
2. การวิเคราะห์ธุรกิจ (Business Analytics)
3. การจัดการผลการดำเนินงานขององค์กร (Business Performance Management: BPM)
4. ส่วนติดต่อผู้ใช้เพื่อการสื่อสารกับผู้ใช้ระบบ (User Interface)

Machine Learning

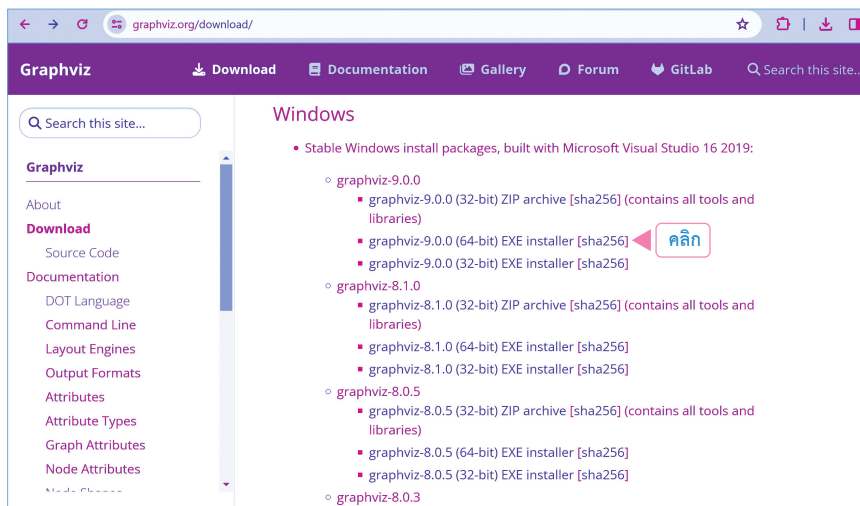
เป็นการทำให้คอมพิวเตอร์สามารถเรียนรู้ได้จากข้อมูลที่ป้อนเข้าไปจำนวนมากๆ หลากหลายรูปแบบ จนกระทั่งคอมพิวเตอร์มีความฉลาดมากพอที่จะเรียนรู้ด้วยตัวเอง และแยกแยะเพื่อการประมวลผลและทำงานจนได้ผลลัพธ์ออกมา โดยรูปแบบของ Machine Learning ที่จะนำมาใช้กับ Predictive Analytics จึงเป็นรูปแบบของ Supervised Model ที่เป็นการสร้างโมเดลเพื่อแปลงข้อมูลป้อนเข้าให้คอมพิวเตอร์ได้เรียนรู้

ในตัวอย่างโปรแกรมที่ผู้เขียนจะอธิบายต่อไป เป็นการทำ Machine Learning ด้วยการ classification และสร้างเป็น Decision Tree โดยใช้โมดูล scikit-learn ในภาษา Python ที่มีฟังก์ชันรองรับการสร้างโมเดล รวมถึงฟังก์ชันการคำนวณค่าความแม่นยำอีกด้วย

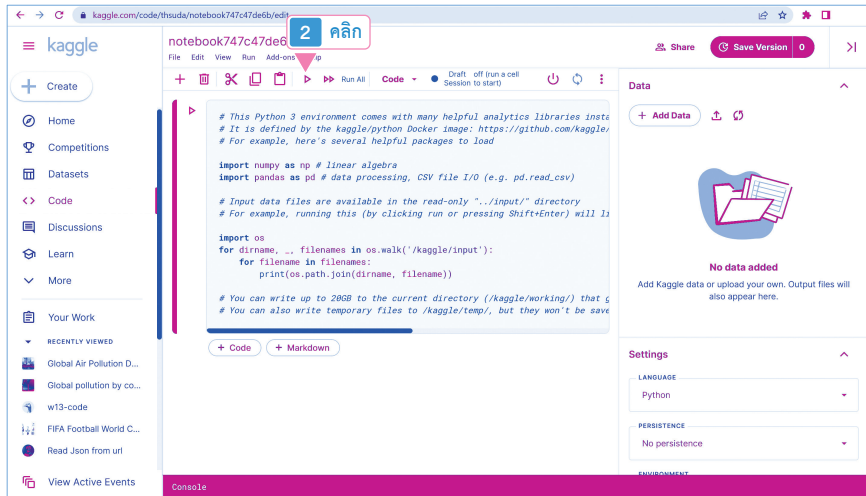
ก่อนทำการเขียนโปรแกรม ให้ผู้อ่านทำการติดตั้งโมดูล scikit-learn และโมดูล graphviz โดยใช้ขั้นตอนเดียวกับการติดตั้งโมดูล pyodbc ซึ่งผู้เขียนได้อธิบายไว้ในบทที่ 13 หลังจากติดตั้งได้สำเร็จแล้ว ผู้อ่านจะต้องติดตั้งโปรแกรม graphviz ซึ่งเป็นเครื่องมือที่ใช้ในการวาดแผนภาพ Decision Tree และสามารถใช้งานได้ฟรี

การติดตั้งโปรแกรม graphviz

ผู้อ่านสามารถดาวน์โหลดโปรแกรม graphviz เวอร์ชัน 9.0.0 ได้จากเว็บไซต์ <https://graphviz.org/download/> ตามหน้าต่างโปรแกรมที่แสดงนี้ ให้คลิกที่ **graphviz-9.0.0 (64-bit) EXE installer [sha256]** เมื่อการดาวน์โหลดเสร็จสมบูรณ์ ก็จะได้ไฟล์ windows_10_cmake_Release_graphviz-install-9.0.0-win64.exe



2. จะปรากฏหน้าต่างแสดงพื้นที่ให้เราสามารถเขียนโค้ดโปรแกรมและรันโปรแกรม โดยคลิกปุ่ม ▶



การเขียนโปรแกรมบน Kaggle Platform

เนื่องจาก Kaggle Platform เป็นแพลตฟอร์มชุมชนที่มีการแบ่งปันความรู้ ชุดข้อมูลต่างๆ ทางด้าน Data Analytics เป็นหลัก ผู้เขียนจึงขอแนะนำตัวอย่างโปรแกรมที่นำเข้าสู่ชุดข้อมูลที่มีการแบ่งปันกันอยู่ในแพลตฟอร์ม โดยใช้ชุดข้อมูล global air pollution dataset.csv ซึ่งเป็นข้อมูลแสดงภาวะมลพิษของแต่ละประเทศในโลก โดยจะนำมาวิเคราะห์และนำเสนอ 10 อันดับประเทศที่มีภาวะมลพิษทางอากาศมากที่สุด ซึ่งมีขั้นตอนดังนี้

1. เข้าสู่ระบบที่เว็บไซต์ [https:// www.kaggle.com/](https://www.kaggle.com/) จะปรากฏหน้าต่างการทำงาน ดังนี้

