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/
00000000000

เหมาะสำหรับนักเรียน นักศึกษา
และผู้ที่สนใจเขียนโปรแกรมภาษา Python

ผศ.สุดา เรียรมนตรี

IDC
PREMIER

มีเพียง “**ความรู้**” เท่านั้นที่มนุษย์ใช้พลิก “**โลก**”
และเปลี่ยน “**ชีวิต**” เราจึงสร้างสรรค์ และส่งมอบ “**ความรู้**”
ในรูปแบบที่ดีกว่า เพื่อให้คนไทย “**เรียนรู้**” ได้ตลอดชีวิต

Only “**Knowledge**” can help human
change “**The World**” and “**Their Lives**”.
With this truth, it drives us to deliver
“**Knowledge**” for Thai being able to
“**Learn**” better everyday.

Think Beyond
INFOPRESS
Read comics
Little HEART

คู่มือ Coding ภาษา PYTHON ฉบับสมบูรณ์ 3rd Edition

AUTHOR	ผศ.สุดา เธียรมนตรี thiansuda@gmail.com
EDITORIAL	กิตินันท์ พลสวัสดิ์ kitinan_p@idcpremier.com
EXECUTIVE EDITORS	สัจจะ จรัสรุ่งรวีวร, อรรณพ ขันธิกุล
PRODUCTION MANAGER	วรพล ณธิกุล
GRAPHIC DESIGNER	ชวนันท์ รัตนะ
PAGE LAYOUT	วุฒิพันธ์ สมพระเมฆ, สิริลักษณ์ วาระเลิศ
PROOFREADER	พรรณรัตน์ ขุราษี
PUBLISHING COORDINATORS	สุรีย์รัตน์ จิอู๋, ฉัตรดนัย ศรีสวัสดิ์
PRODUCT SPECIALIST	ระพีพัฒน์ โทจำปา

Python IDLE เป็นเครื่องหมายการค้าของบริษัท Python Software Foundation, PyCharm เป็นเครื่องหมายการค้าของบริษัท Jet Brains และเครื่องหมายการค้าอื่นๆ ที่อ้างถึงเป็นของบริษัทนั้นๆ

สงวนลิขสิทธิ์ตามพระราชบัญญัติลิขสิทธิ์ พ.ศ. 2537 โดยบริษัท ไอดีซี พรีเมียร์ จำกัด ห้ามลอกเลียนไม่ว่าส่วนใดส่วนหนึ่งของหนังสือเล่มนี้ ไม่ว่าในรูปแบบใดๆ นอกจากจะได้รับอนุญาตเป็นลายลักษณ์อักษรจากผู้จัดพิมพ์เท่านั้น

บริษัท ไอดีซี พรีเมียร์ จำกัด จัดตั้งขึ้นเพื่อเผยแพร่ความรู้ที่มีคุณภาพสู่ผู้อ่านชาวไทย เรายินดีรับงานเขียนของนักวิชาการและนักเขียนทุกท่าน ท่านผู้สนใจกรุณาติดต่อผ่านทางอีเมลที่ infopress@idcpremier.com หรือทางโทรศัพท์หมายเลข 0-2962-1081 (อัตโนมัติ 10 คู่สาย) โทรสาร 0-2962-1084

PUBLISHED AND DISTRIBUTED BY

IDC PREMIER

บริษัท ไอดีซี พรีเมียร์ จำกัด

200 หมู่ 4 ชั้น 19 ห้อง 1901 อาคารจัสมินอินเตอร์เนชั่นแนล

ทาวเวอร์ ถ.แจ้งวัฒนะ อ.ปากเกร็ด จ.นนทบุรี 11120

โทรศัพท์ 0-2962-1081 (อัตโนมัติ 10 คู่สาย)

โทรสาร 0-2962-1084

สำหรับร้านค้าและตัวแทนจำหน่าย

โทรศัพท์ 0-2962-1081-3

โทรสาร 0-2962-1084

พิมพ์ที่ บริษัท เฟิสท์ออฟเซท (1993) จำกัด

99/11 หมู่ 7 ต.ลำโพ อ.บางบัวทอง จ.นนทบุรี 11110

โทรศัพท์ 0-2191-7125-8

โทรสาร 0-2191-7129

INFOPRESS

ข้อมูลทางบรรณานุกรม

ผศ.สุดา เธียรมนตรี

คู่มือ Coding ภาษา PYTHON ฉบับสมบูรณ์

3rd Edition

นนทบุรี : ไอดีซีฯ, 2567

480 หน้า

1. ไพธอน

2. ภาษาที่ใช้ในการเขียนโปรแกรม

I ชื่อเรื่อง

005.13

Barcode 885-916-101-168-2

พิมพ์ครั้งที่ 1 พฤษภาคม 2567

E-Book มิถุนายน 2568

ราคา 395 บาท

คำนำ

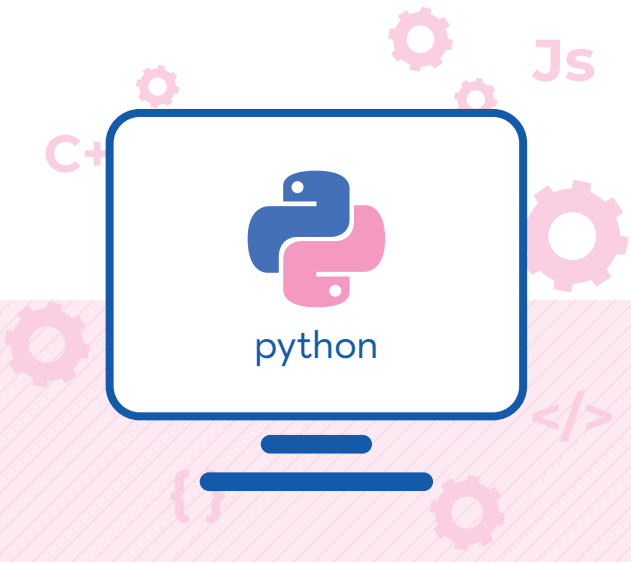
หนังสือคู่มือ Coding ภาษา PYTHON ฉบับสมบูรณ์ 3rd Edition เล่มนี้ได้นำเสนอเนื้อหาสำหรับผู้อ่านที่ต้องการเขียนโปรแกรมในระดับเริ่มต้นไปถึงการเขียนโปรแกรมเพื่อประยุกต์กับระบบงานจริง และยังตอบโจทย์กับการเปลี่ยนแปลงทางเทคโนโลยีดิจิทัลในปัจจุบัน ไม่ว่าจะเป็นเรื่องของวิทยาการข้อมูล (Data Science) ธุรกิจอัจฉริยะ (Business Intelligence) รวมถึงการใช้ปัญญาประดิษฐ์ด้านการเรียนรู้ของเครื่อง (Machine Learning) เพื่อการคาดการณ์ธุรกิจ โดยทั้งหมดที่กล่าวนั้น ภาษา Python สามารถรองรับการทำงานได้อย่างครอบคลุม ผู้เขียนได้นำเสนอความรู้และหลักการเขียนโปรแกรมอย่างเป็นลำดับขั้นตอน พร้อมทั้งอธิบายตัวอย่างโปรแกรมไว้อย่างครบถ้วน ด้วยคุณสมบัติของภาษา Python ที่มีโครงสร้างเรียบง่าย โปรแกรมที่เขียนจึงสั้นกระชับ ง่ายต่อการทำความเข้าใจ และรองรับการเขียนโปรแกรมที่หลากหลาย ผู้เขียนจึงได้นำเสนอและให้ตัวอย่างที่ใช้งานได้จริง ตั้งแต่การเขียนโปรแกรมเชิงโครงสร้าง โปรแกรมเชิงวัตถุ โปรแกรมจัดการส่วนติดต่อผู้ใช้แบบกราฟิก โปรแกรมติดต่อฐานข้อมูล โปรแกรมจัดทำรายงาน และการเขียนโปรแกรมประยุกต์กับงานด้านเว็บแอปพลิเคชัน การส่งข้อความผ่านเครือข่าย การจัดการข้อมูลทางธุรกิจ รวมถึงโปรแกรมสำหรับธุรกิจอัจฉริยะที่ช่วยตัดสินใจคาดการณ์ธุรกิจ โดยผู้เขียนได้นำเสนอเนื้อหาพร้อมตัวอย่างโปรแกรมที่สามารถนำไปประยุกต์เพื่อการต่อยอดได้

ผู้เขียนมีความตั้งใจที่จะให้หนังสือเล่มนี้เป็นจุดเริ่มต้นสำหรับผู้อ่านที่เริ่มต้นศึกษาการเขียนโปรแกรม และเป็นแนวทางสำหรับผู้อ่านที่ต้องการเขียนโปรแกรมประยุกต์กับการใช้งานจริงเพื่อการก้าวสู่นักเขียนโปรแกรมมืออาชีพต่อไป

ท้ายที่สุดนี้ ผู้เขียนขอขอบคุณบรรณาธิการ ทีมงาน DEV BOOK และผู้สนับสนุนทุกท่านที่มีส่วนร่วมในการจัดทำหนังสือเล่มนี้ให้สำเร็จได้เป็นอย่างดี

ผศ.สุดา เธียรมนตรี

thiansuda@gmail.com


Js
C+
python
</>
{ }

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
การใช้ตัวอักขระพิเศษ 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 : รับและแสดงข้อความ 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) และออบเจ็กต์ (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 กับการจัดการข้อมูลการรับสินค้าเข้าในคลังสินค้า 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

01 >>

รู้จักกับภาษา Python (Introduction to Python)

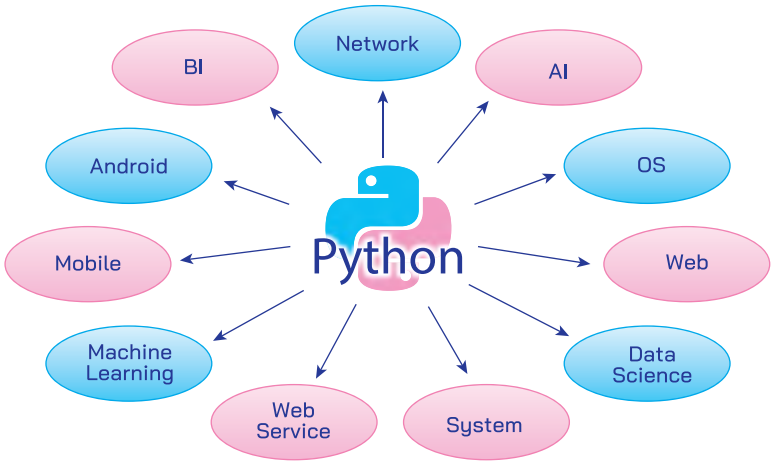
ภาษา Python เป็นภาษาระดับสูงภาษาหนึ่งที่ถูกนำมาใช้งานอย่างกว้างขวาง เนื่องจากเป็นภาษาที่เรียบง่าย สามารถเข้าใจง่ายและเรียนรู้ได้อย่างรวดเร็ว นอกจากนี้ Python ยังสามารถใช้งานได้ครอบคลุมลักษณะงานที่หลากหลาย รองรับการทำงานบนอุปกรณ์ได้หลายรูปแบบ ทำให้ Python เป็นภาษาโปรแกรมที่ได้รับความนิยมมากที่สุดภาษาหนึ่ง ซึ่งผู้อ่านจะได้รู้จักกับภาษา Python ในหนังสือเล่มนี้

แนะนำภาษา Python

Python เป็นภาษาโปรแกรมที่พัฒนาขึ้นโดย Guido van Rossum และทีมพัฒนา ซึ่งมีวัตถุประสงค์เพื่อพัฒนาภาษาที่ใช้สร้างโปรแกรมด้วยโครงสร้างที่เรียบง่าย และเป็นภาษาแบบเปิดเผยรหัส (Open Source) ที่มีการนำไปพัฒนาต่อๆ กัน ทำให้ Python มีความสามารถมากขึ้น มีคุณลักษณะที่หลากหลาย สามารถใช้พัฒนางานด้านต่างๆ ได้ เช่น งานด้านวิทยาศาสตร์ งานด้านฐานข้อมูล ปัญญาประดิษฐ์ เป็นต้น และสนับสนุนการทำงานบนทุกแพลตฟอร์ม โดยมีคุณสมบัติที่เด่นๆ ดังนี้

- ง่ายต่อการเรียนและเข้าใจ เนื่องจากโครงสร้างภาษาไม่มีความซับซ้อน
- มีการจัดการหน่วยความจำแบบอัตโนมัติ สามารถจัดการพื้นที่ในหน่วยความจำได้อย่างมีประสิทธิภาพ
- ทำงานได้ในทุกระบบ คือ มีคุณลักษณะของ Platform Independent

- รองรับการเขียนโปรแกรมแบบ Multi-paradigm Programming สนับสนุนการเขียนโปรแกรมด้วยแนวคิดที่หลากหลาย เช่น การเขียนโปรแกรมเชิงโครงสร้าง (Structured Programming) การเขียนโปรแกรมเชิงฟังก์ชัน (Functional Programming) การเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming) และการเขียนโปรแกรมจัดการฐานข้อมูล (Database Programming) ซึ่งผู้อ่านจะได้ศึกษาและเรียนรู้ต่อไป



Python สามารถทำอะไรได้บ้าง

การทำงานของโปรแกรมภาษา Python

ภาษา Python มีการทำงานด้วยตัวแปลภาษาแบบอินเตอร์พรีเตอร์ โดยผู้อ่านสามารถพิมพ์คำสั่งเป็นลักษณะภาษาสคริปต์ให้คอมพิวเตอร์ทำงานโต้ตอบทีละคำสั่ง และแสดงผลลัพธ์ทันทีหากมีข้อผิดพลาด โดยโปรแกรมก็จะมีการแจ้งเตือน

นอกจากนั้นผู้อ่านยังสามารถเขียนโปรแกรมเรียงต่อกันเป็นไฟล์ได้ โดยไฟล์ Source Code จะมีนามสกุลเป็น .py เมื่อมีการรันโปรแกรม Source Code ดังกล่าวจะถูกนำไปสร้างเป็น **Python Byte Code** ที่นำไปใช้งานได้บนหลายแพลตฟอร์ม

โดย **Python Byte Code** จะถูกคอมไพล์ด้วย Python Virtual Machine (PVM) ให้เป็นภาษาเฉพาะของแพลตฟอร์มนั้น เพื่อให้โปรแกรมสามารถทำงานได้

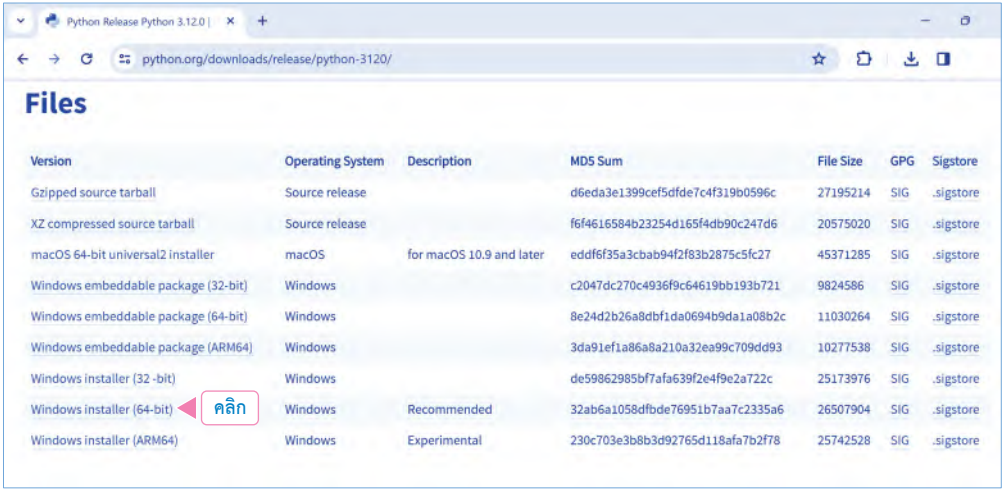
แนะนำเครื่องมือสำหรับเขียนโปรแกรมภาษา Python

ในการเขียนโปรแกรมด้วยภาษา Python จะมีเครื่องมือที่รองรับอยู่มากมาย ซึ่งมีคุณลักษณะที่แตกต่างกันไป มีทั้งการใช้งานผ่านเครือข่ายอินเทอร์เน็ต หรือการดาวน์โหลดเพื่อติดตั้งใช้งานในเครื่องคอมพิวเตอร์ส่วนบุคคล เช่น

- **Google Collaboratory หรือ Google Colab** เป็นเครื่องมือที่รองรับการเขียนโปรแกรมด้วย Python เป็นบริการคลาวด์ที่สนับสนุนการทำงานโดยผู้ใช้ไม่จำเป็นต้องดาวน์โหลดติดตั้งโปรแกรมแต่อย่างใด สามารถใช้งานผ่านเครือข่ายอินเทอร์เน็ต
- **Kaggle Platform** เป็นเว็บไซต์ชุมชนที่เป็นแพลตฟอร์มสำหรับการแบ่งปันความรู้ รวมถึงการแข่งขันด้านการวิเคราะห์ข้อมูล รองรับการเขียนโปรแกรมภาษา Python สามารถใช้งานผ่านเครือข่ายอินเทอร์เน็ต
- **Visual Studio Code หรือ VSCode** เป็นเครื่องมือที่ใช้สำหรับพัฒนาแอปพลิเคชันด้วยภาษาต่างๆ หนึ่งในนั้นก็คือ ภาษา Python โดยได้รับการสนับสนุนจากบริษัท Microsoft สามารถใช้งานได้ฟรี
- **Python Integrated DeveLopment Environment หรือ Python IDLE** เป็นเครื่องมือที่รวมอยู่ในชุดอินเตอร์พรีเตอร์ของโปรแกรมภาษา Python อยู่แล้ว
- **PyCharm Community Edition** เป็นชุดเครื่องมือของบริษัท JetBrains ซึ่งเป็นบริษัทที่พัฒนาเครื่องมือสำหรับเขียนโปรแกรม โดยโปรแกรมในเวอร์ชันที่เป็น Community Edition สามารถใช้งานได้ฟรี

การติดตั้งโปรแกรม Python

ผู้อ่านสามารถดาวน์โหลดโปรแกรม Python เวอร์ชัน 3.12.0 ได้จากเว็บไซต์ https://www.python.org/downloads/release/python-3120/ ตามหน้าต่างโปรแกรมที่แสดงนี้ ให้คลิกที่ **Download Windows installer (64-bit)** เมื่อการดาวน์โหลดเสร็จสมบูรณ์ก็จะได้ไฟล์ python-3.12.0-amd64.exe

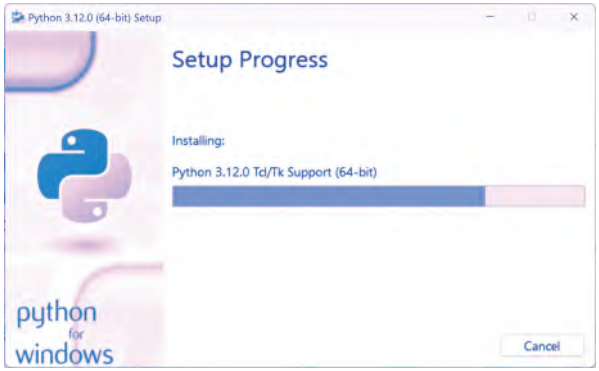


ในการติดตั้งโปรแกรมที่ได้จากการดาวน์โหลด จะทำการติดตั้งและใช้งานบนระบบปฏิบัติการ Windows 11 โดยผู้อ่านสามารถดับเบิลคลิกที่ชอร์ตคัตของโปรแกรม

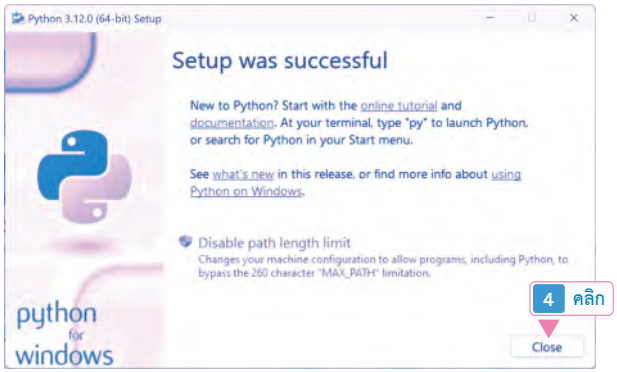
แล้วจะปรากฏหน้าต่างสำหรับติดตั้งโปรแกรม ให้ดำเนินการตามขั้นตอนดังนี้

1. คลิก ☑ ที่ Use admin privileges when installing py.exe
   คลิก ☑ ที่ Add python.exe to PATH
2. คลิกที่ Install Now

3. จะปรากฏหน้าต่างแสดงการติดตั้งโปรแกรม



4. จะปรากฏหน้าต่างแสดงการติดตั้งเสร็จสมบูรณ์ ให้คลิกปุ่ม Close

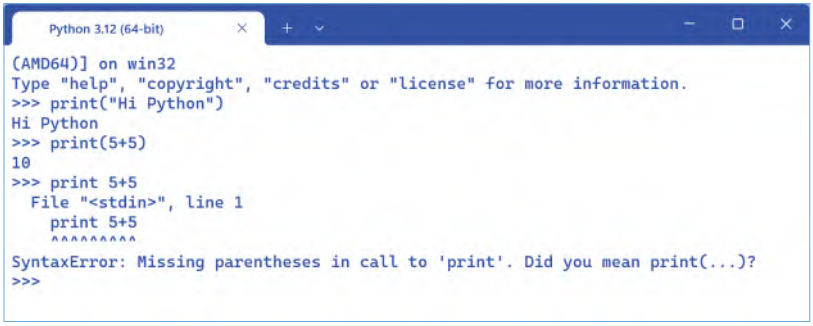


การเรียกใช้งานโปรแกรม Python

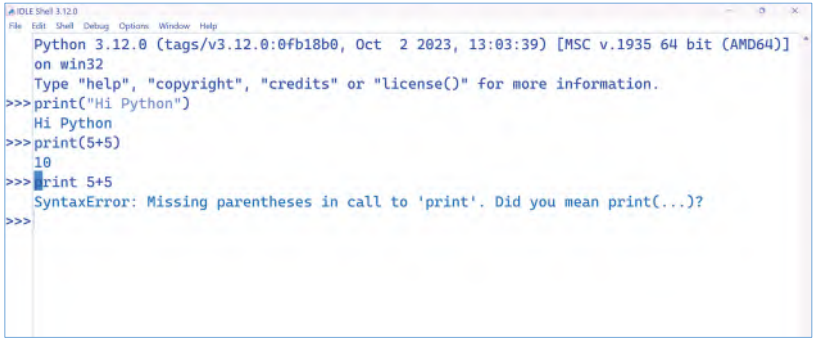
ในการเรียกใช้งานโปรแกรม Python มีขั้นตอนดังนี้

1. คลิกที่ ซึ่งเป็นปุ่ม Start ของระบบปฏิบัติการ Windows 11 จะปรากฏรายชื่อชอร์ตคัต ซึ่งเป็นทางเข้าใช้งานของโปรแกรม ผู้อ่านสามารถเรียกใช้งานได้ 2 รูปแบบ ดังนี้
   - เรียกใช้งานผ่าน Python โดยคลิกที่ Python 3.12 (64-bit) หรือ
   - เรียกใช้งานผ่าน IDLE โดยคลิกที่ IDLE (Python 3.12 64-bit)

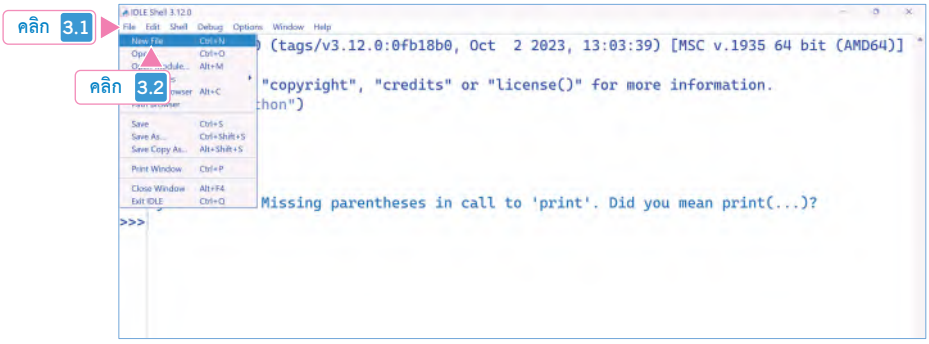
2. ในกรณีที่ผู้อ่านเรียกใช้ผ่าน Python 3.12 (64-bit) จะปรากฏหน้าต่าง ซึ่งผู้อ่านสามารถพิมพ์คำสั่งเพื่อให้คอมพิวเตอร์ทำงานโต้ตอบกับเราได้ทีละคำสั่ง โดยสามารถทดสอบการทำงานและมีการแสดงผลลัพธ์ได้ทันทีหลังจากพิมพ์คำสั่งหลังเครื่องหมาย >>> และกดปุ่ม Enter เพื่อรันคำสั่ง หากมีข้อผิดพลาดในการใช้คำสั่ง ก็จะมีการแจ้งเตือนทันที



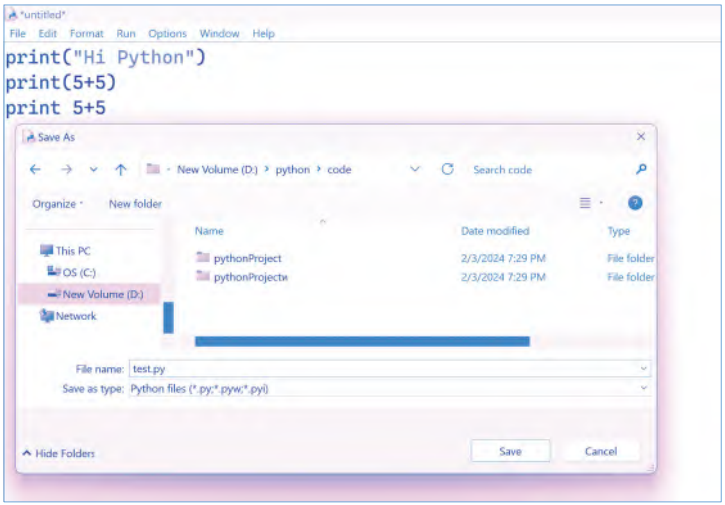
3. ในกรณีที่ผู้อ่านเรียกใช้ผ่าน IDLE (Python 3.12 64-bit) จะปรากฏหน้าต่าง Python Shell ซึ่งทำงานเหมือนกับการเรียกใช้ผ่าน Python 3.12 (64-bit)



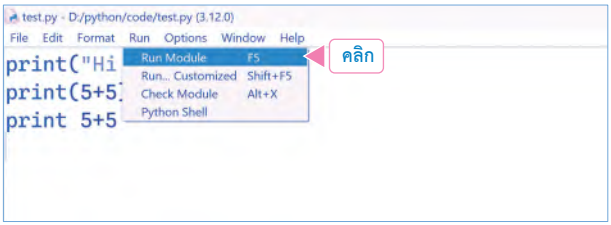
แต่ถ้าผู้อ่านต้องการเขียนคำสั่งที่เรียงลำดับการทำงานต่อกันในลักษณะของโปรแกรม ก็สามารถทำผ่านการเรียกใช้แบบนี้ได้ โดยคลิกที่เมนู File แล้วคลิกเลือก New File



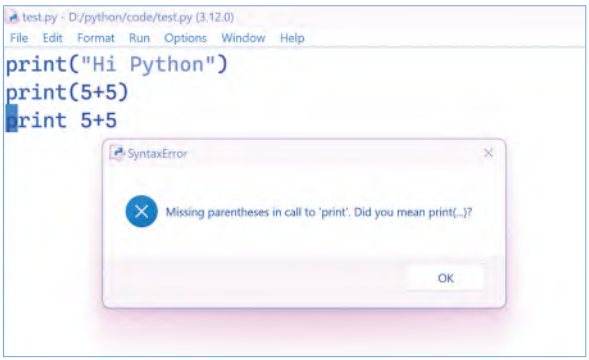
จากนั้นจะปรากฏหน้าต่าง ซึ่งผู้อ่านสามารถพิมพ์คำสั่งเรียงต่อกัน และบันทึกเป็นไฟล์ซึ่งมีนามสกุลเป็น .py



ขั้นตอนต่อไป ผู้อ่านสามารถทดสอบการทำงานของโปรแกรม โดยคลิกที่เมนู Run เลือก Run Module หรือกดปุ่ม F5



หากมีข้อผิดพลาดในการใช้คำสั่ง ก็จะมีการแจ้งเตือนดังรูป



หากไม่พบข้อผิดพลาดในการใช้คำสั่ง โปรแกรมก็จะแสดงผลลัพธ์ตามชุดคำสั่งที่ผู้อ่านได้เขียนไว้

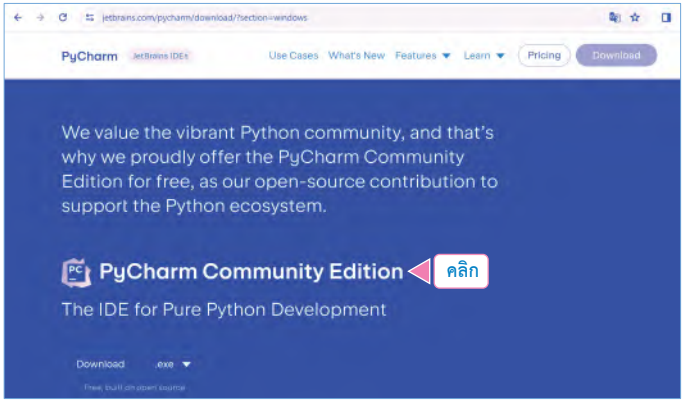
นอกจากเครื่องมือดังกล่าวข้างต้นแล้ว ยังมีอีกเครื่องมือที่ได้รับความนิยมเป็นอย่างมาก นั่นก็คือ PyCharm Community Edition เนื่องจาก PyCharm มีคุณสมบัติเด่นๆ มากมาย ได้แก่

- มีเวอร์ชันที่ใช้งานฟรี โดยเวอร์ชันที่ใช้ในหนังสือเล่มนี้ คือ เวอร์ชัน 2023.3.2
- มีสภาพแวดล้อมที่ช่วยอำนวยความสะดวกในการพัฒนาโปรแกรม (IDE : Integrated Development Enviroment)
- มีส่วนที่ติดต่อผู้ใช้ (User Interface) ที่มีความสวยงาม น่าใช้งาน

ในส่วนของขั้นตอนการดาวน์โหลดและติดตั้ง PyCharm ผู้เขียนจะขออธิบายในหัวข้อต่อไป

การติดตั้งและเรียกใช้งานโปรแกรม PyCharm

โปรแกรม PyCharm ที่ใช้อธิบายในหนังสือเล่มนี้คือ เวอร์ชัน Community Edition 2023.3.2 เป็น Python เวอร์ชัน 3.12.0 โดยผู้อ่านสามารถดาวน์โหลดได้จากเว็บไซต์ https://www.jetbrains.com/pycharm/download/#section=windows



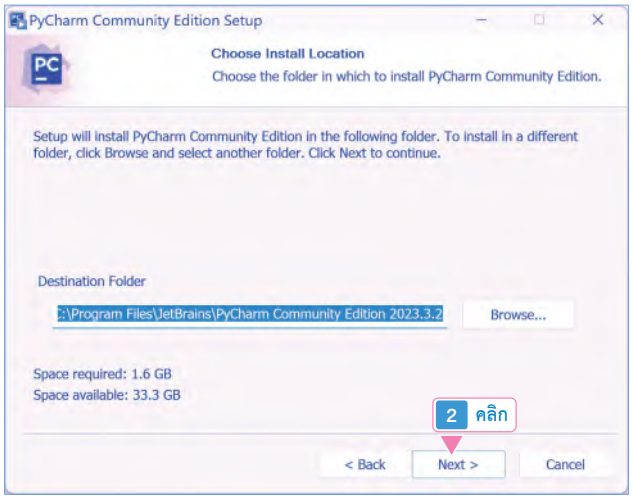
ตามหน้าต่างโปรแกรมที่แสดงนี้ ให้คลิกที่ Download และเมื่อการดาวน์โหลดเสร็จสมบูรณ์ ก็จะได้ไฟล์ pycharm-community-2023.3.2.exe

ในการติดตั้งโปรแกรมที่ได้จากการดาวน์โหลด ผู้อ่านสามารถดับเบิลคลิกที่ชอร์ตคัตของโปรแกรม

และจะปรากฏหน้าต่างสำหรับติดตั้งโปรแกรม ให้ดำเนินการตามขั้นตอนดังนี้

1. คลิกปุ่ม Next >

2. จะปรากฏหน้าต่างให้เลือกตำแหน่งที่ต้องการติดตั้งโปรแกรม ในที่นี้เลือกเป็นดีฟอลต์ และคลิกปุ่ม Next >



3. จะปรากฏหน้าต่างให้เลือกค่าการติดตั้งโปรแกรม ให้ดำเนินการดังนี้

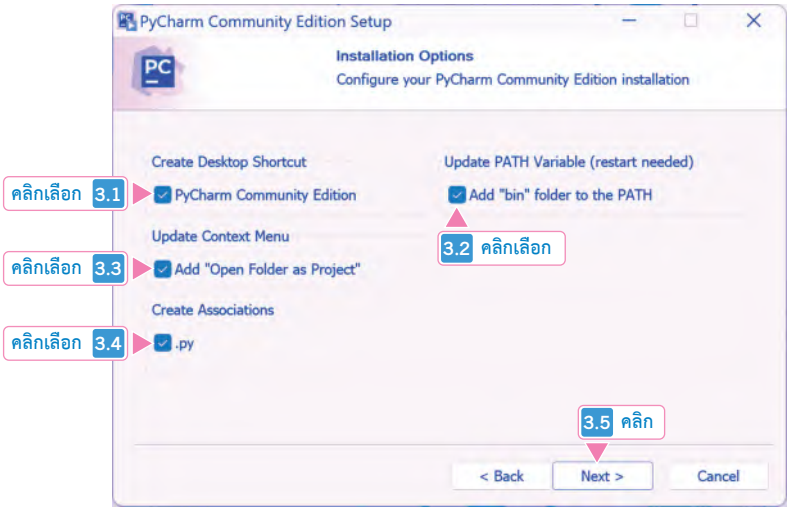
3.1 คลิกเลือก ☑ ที่ PyCharm Community Edition

3.2 คลิกเลือก ☑ ที่ Add “bin” folder to the PATH

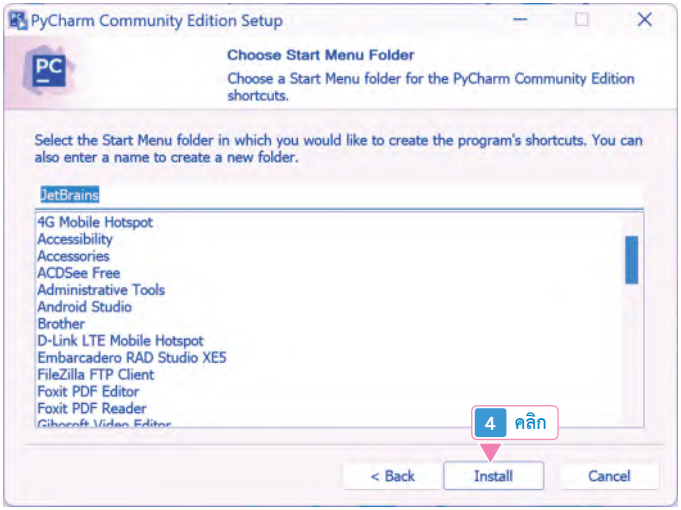
3.3 คลิกเลือก ☑ ที่ Add “Open Folder as Project”

3.4 คลิกเลือก ☑ ที่ .py

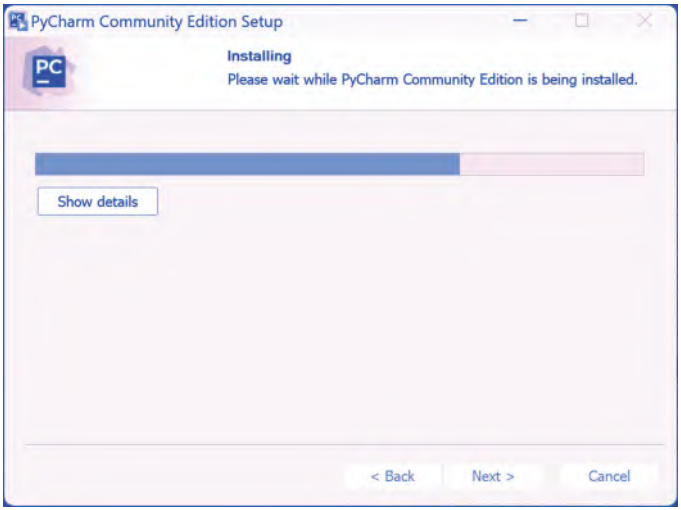
3.5 คลิกปุ่ม Next >



4. จะปรากฏหน้าต่างให้เลือกตำแหน่งโฟลเดอร์ที่ต้องการติดตั้งซอร์ตคัต ในที่นี้เลือกเป็นดีฟอลต์ และคลิกปุ่ม Install



5. จะปรากฏหน้าต่างแสดงการติดตั้งโปรแกรม

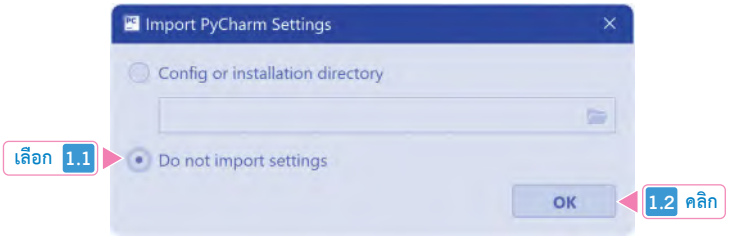


6. จะปรากฏหน้าต่างแสดงการติดตั้งเสร็จสมบูรณ์ ให้คลิกเลือก Reboot now และคลิกปุ่ม Finish

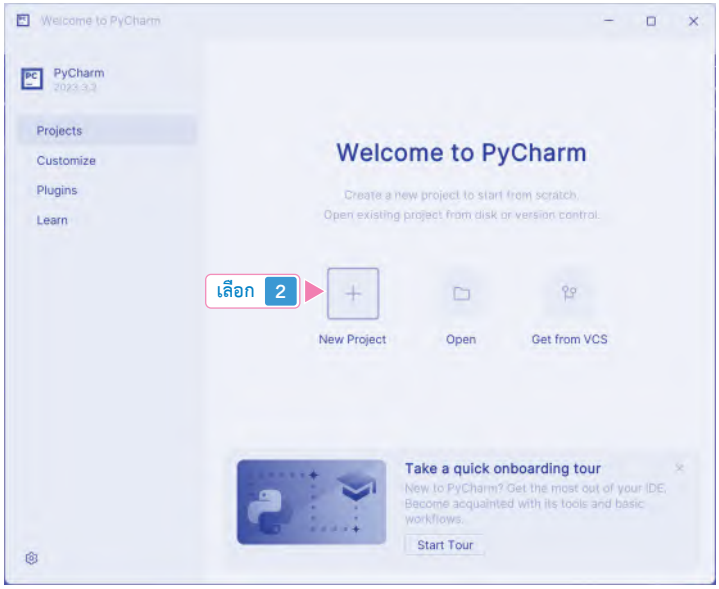
ในการเรียกใช้งานโปรแกรม PyCharm มีขั้นตอนดังนี้

1. ดับเบิลคลิกที่

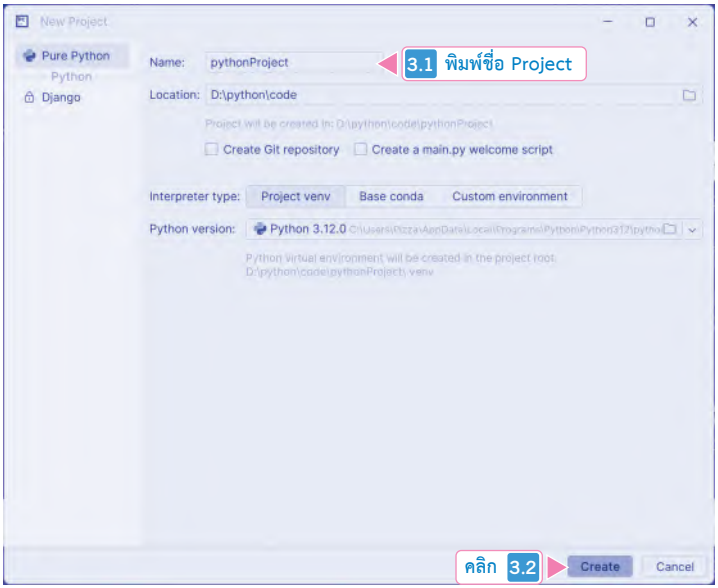
ซึ่งเป็นชอร์ตคัตของทางเข้าใช้งานของโปรแกรม จะปรากฏหน้าต่าง ให้กำหนดการนำเข้าการตั้งค่า ให้คลิกเลือก Do not import settings และคลิกปุ่ม OK



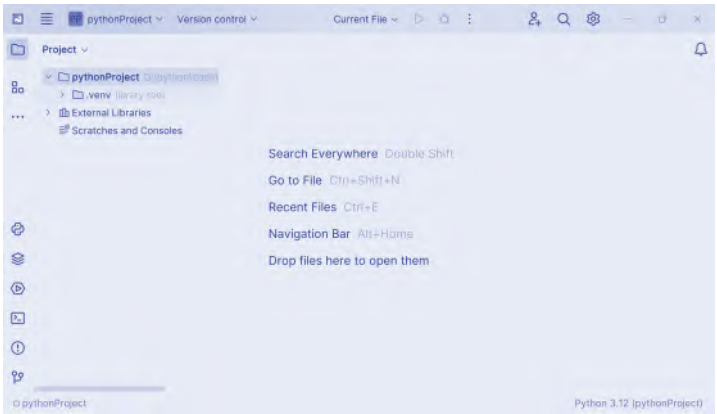
2. จะปรากฏหน้าต่างต้อนรับเข้าสู่โปรแกรม PyCharm ให้คลิกที่ + เพื่อสร้าง Project ใหม่



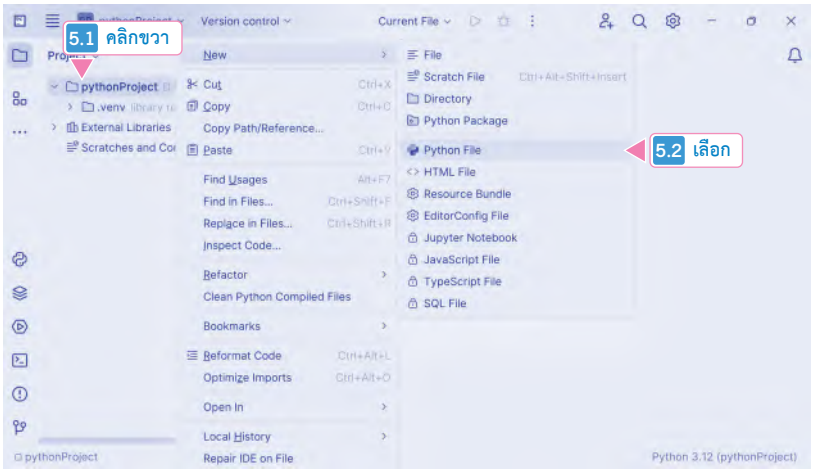
3. จะปรากฏหน้าต่างให้เลือกตำแหน่งที่ต้องการจัดเก็บ Project โปรแกรม ในที่นี้เลือกตำแหน่งโฟลเดอร์ที่ต้องการ คือ D:\python\code พิมพ์ชื่อ Project ตรงช่อง Name ว่า pythonProject และคลิกปุ่ม **Create**



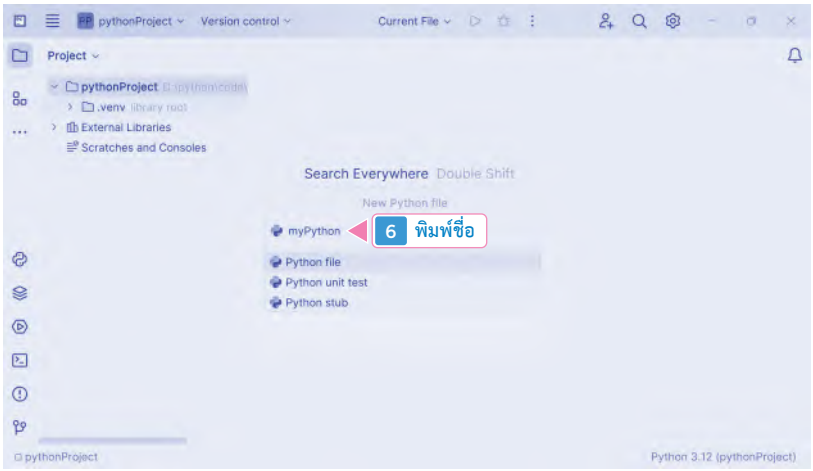
4. จะปรากฏหน้าต่างโปรแกรม PyCharm ที่มีการสร้างโฟลเดอร์ pythonProject ไว้ในโฟลเดอร์ code โดยการสร้างไฟล์เพื่อการเขียนโค้ดโปรแกรมจะอยู่ภายใต้โฟลเดอร์นี้ คือ D:\python\code\ pythonProject



5. ให้คลิกขวาที่โฟลเดอร์ pythonProject เลือก New > Python File

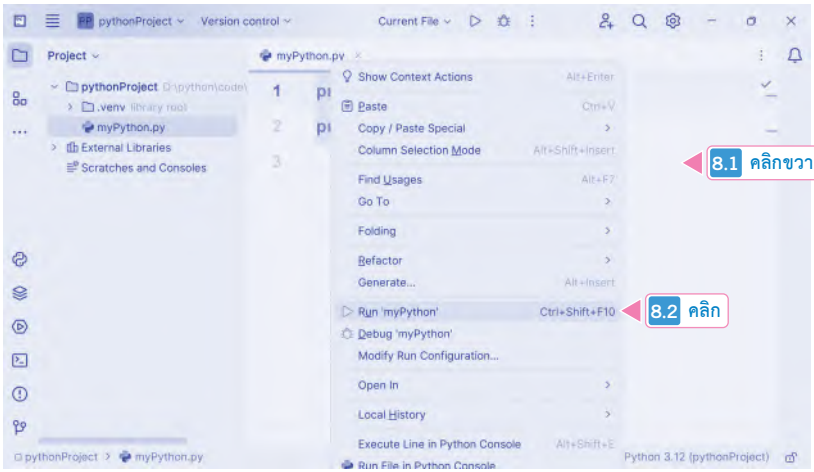


6. จะปรากฏหน้าต่างให้ตั้งชื่อไฟล์ ในที่นี้ใช้ชื่อไฟล์ว่า myPython และกดปุ่ม Enter



7. จะปรากฏหน้าต่างให้เขียนโปรแกรมชุดคำสั่งที่ต้องการ ให้เขียนชุดคำสั่งดังรูป

8. เมื่อต้องการรันโปรแกรม ให้คลิกขวาตรงพื้นที่ว่าง และเลือก Run ‘myPython’



9. โปรแกรมก็จะแสดงผลการทำงาน

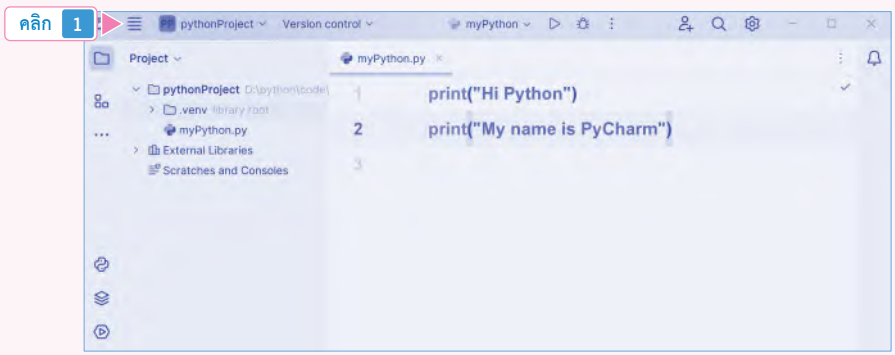
```
Hi Python
My name is PyCharm

Process finished with exit code 0
```

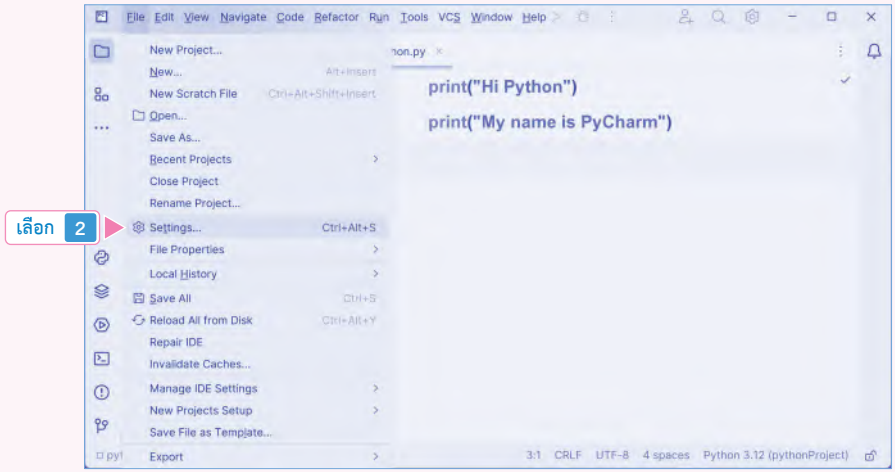
< NOTE >

จากการทำงานของโปรแกรม ผู้อ่านจะพบว่าพื้นหลังการทำงานเป็นการใช้ Apperence แบบพื้นหลังสีดำ ซึ่งถูกกำหนดมาตั้งแต่ติดตั้งโปรแกรม PyCharm เปลี่ยน Apperence หากต้องการเป็นแบบพื้นหลังสีขาว เราสามารถตั้งค่าได้ดังนี้

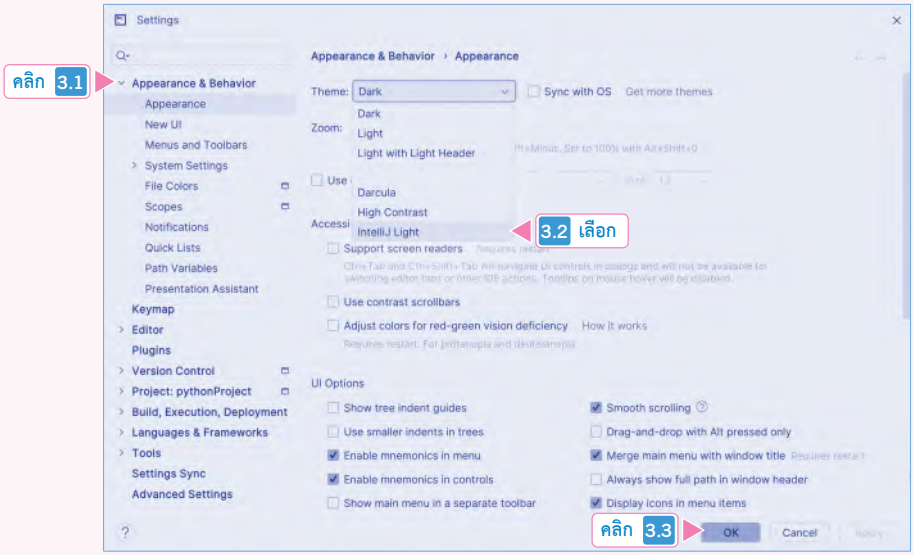
1. จากหน้าต่างโปรแกรม PyCharm คลิกที่ ☰



2. จะปรากฏหน้าต่างที่แสดงชุดเมนู คลิกที่เมนู File เลือก Settings



3. จะปรากฏหน้าต่าง คลิกที่ Apperance > Theme คลิกเลือก IntelliJ Light และคลิกปุ่ม **OK**



4. จะปรากฏหน้าต่างโปรแกรมที่พื้นหลังเป็นสีขาว

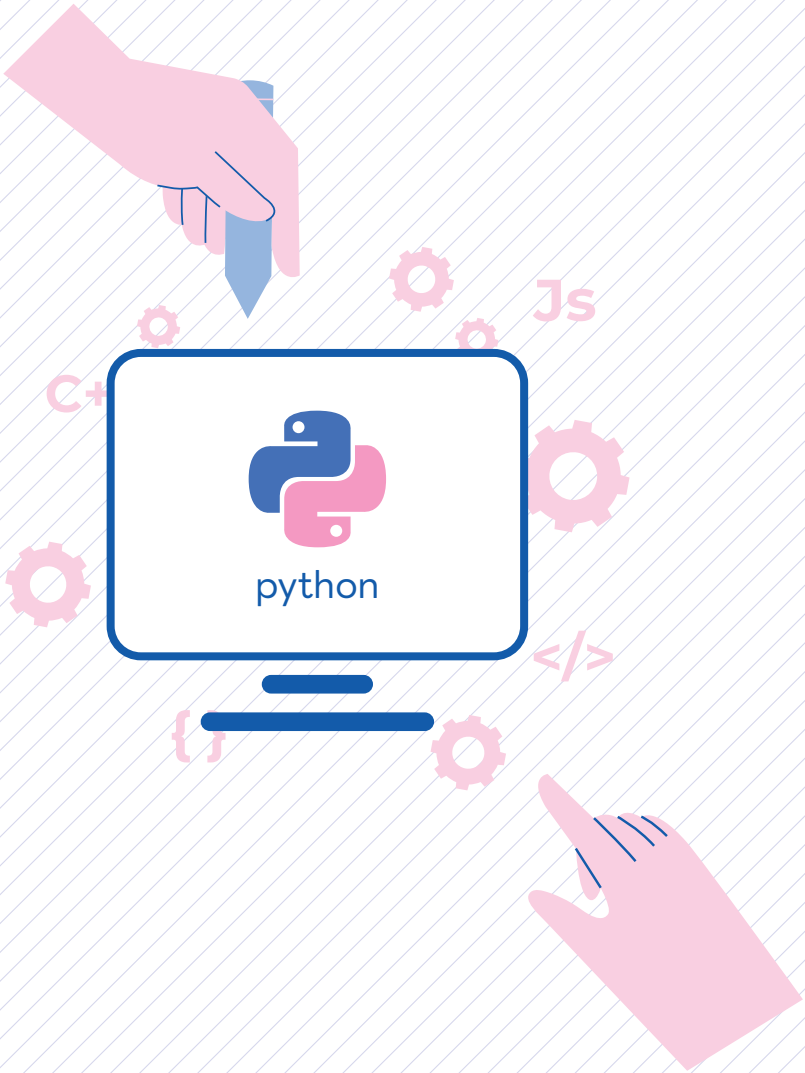
ในส่วนรายละเอียดการใช้งานต่างๆ ของโปรแกรม PyCharm Community Edition นั้น ผู้เขียนจะได้อธิบายในบทต่อไป

สรุปท้ายบท

ในบทนี้ผู้อ่านได้รู้จักภาษา Python ในเบื้องต้น ผู้เขียนได้แนะนำเครื่องมือสำหรับเขียนโปรแกรมภาษา Python การดาวน์โหลด การติดตั้งและเรียกใช้งานโปรแกรม Python และ PyCharm Community Edition ซึ่งเป็นเครื่องมือที่นิยมนำมาใช้ในการเขียนโปรแกรมภาษา Python โดยผู้อ่านได้เรียนรู้การใช้โปรแกรม Python เบื้องต้นผ่าน Python IDLE และ PyCharm Community Edition ในส่วนรายละเอียดการใช้งานต่างๆ ของโปรแกรม PyCharm Community Edition ผู้อ่านจะได้ศึกษาในบทต่อไป

แบบฝึกหัด

1. จงอธิบายคุณสมบัติของโปรแกรมภาษา Python
2. จงอธิบายการทำงานของตัวแปลภาษาแบบอินเตอร์พรีเตอร์
3. จงอธิบายขั้นตอนการทำงานของโปรแกรมภาษา Python


Js
C+
python
</>
{ }

บทที่

02 >>

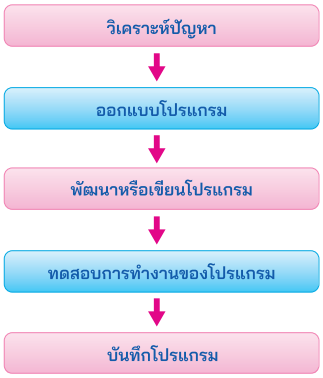
เริ่มต้นเขียนโปรแกรมภาษา Python (Introduction to Python Programming)

ในบทนี้ผู้อ่านจะได้เริ่มเขียนโปรแกรมด้วยภาษา Python และโปรแกรม PyCharm กัน ซึ่งผู้อ่านจะได้ศึกษาขั้นตอนการเขียนโปรแกรม รวมไปถึงได้เรียนรู้การใช้เครื่องมือและการทำงานของ PyCharm ด้วย

ขั้นตอนการเขียนโปรแกรมภาษา Python

ก่อนที่ผู้อ่านจะได้เรียนรู้การเขียนโปรแกรม ควรจะต้องทราบถึงขั้นตอนการพัฒนาโปรแกรมก่อน เพื่อให้เห็นภาพรวมการทำงานของโปรแกรม และเพิ่มความเข้าใจในการเรียนรู้มากยิ่งขึ้น

ในการพัฒนาโปรแกรมใดๆ สิ่งแรกที่ควรทำก็คือ วิเคราะห์ปัญหาเพื่อออกแบบขั้นตอนการทำงานของโปรแกรม จากนั้นจึงเขียนโปรแกรมและทดสอบการทำงานของโปรแกรม สุดท้ายคือการบันทึกโปรแกรมที่ผู้อ่านได้เขียนไว้



ขั้นตอนการเขียนโปรแกรม

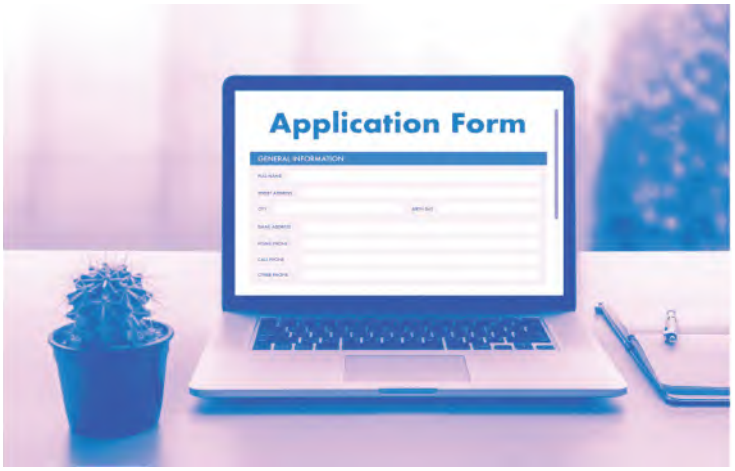
ขั้นที่ 1 วิเคราะห์ปัญหาของโปรแกรมที่ต้องการพัฒนา

ขั้นตอนแรกของการพัฒนาโปรแกรมที่ผู้อ่านทุกท่านควรทำเสมอก็คือ การวิเคราะห์ปัญหาของโปรแกรมที่ต้องการพัฒนา ในบางครั้งอาจรวมไปถึงการออกแบบโปรแกรมในภาพรวมด้วยก็ได้

เราวิเคราะห์ปัญหาของโปรแกรมที่ต้องการก่อนที่จะลงมือพัฒนาโปรแกรม เพื่อต้องการให้ผู้พัฒนาทราบถึงความต้องการของโปรแกรม รวมทั้งการรู้ถึงทรัพยากรที่ต้องใช้งาน เพื่อให้โปรแกรมที่พัฒนาทำงานได้ตรงตามความต้องการ

ขั้นที่ 2 ออกแบบการทำงานของโปรแกรม

ในบางครั้งการออกแบบการทำงานของโปรแกรมนั้น จะเกี่ยวข้องกับหลายๆ ส่วน ไม่ว่าจะเป็นการออกแบบหน้าตาโปรแกรม การออกแบบการทำงานของโปรแกรม การออกแบบฐานข้อมูลที่ใช้สำหรับโปรแกรม เป็นต้น



ตัวอย่างการออกแบบหน้าตาโปรแกรม

ในการออกแบบการทำงานของโปรแกรม ผู้อ่านควรร่างการออกแบบในกระดาษก่อนออกแบบจริง โดยเฉพาะโปรแกรมที่มีขนาดใหญ่และมีความซับซ้อน เพื่อไม่ให้เกิดความสับสนในการเขียนโปรแกรม ซึ่งปัจจุบันอาจออกแบบโดยใช้โปรแกรมเฉพาะทาง

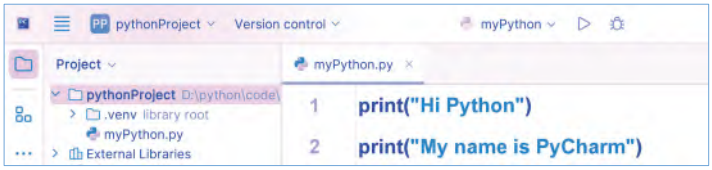
ตัวอย่างการออกแบบโปรแกรม

ขั้นที่ 3 เริ่มต้นสร้างและเขียนโปรแกรม

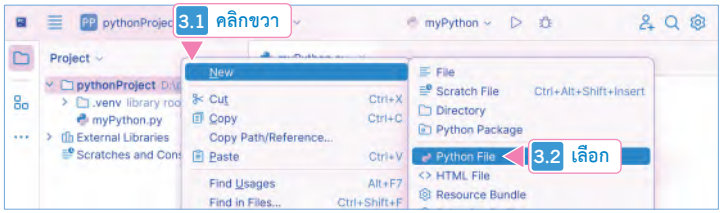
เมื่อได้หน้าตาโปรแกรมที่ออกแบบแล้ว ต่อไปผู้อ่านจะเริ่มเขียนโปรแกรมโดยใช้งานผ่านโปรแกรม PyCharm ดังนี้

1. เรียกใช้โปรแกรม PyCharm โดยดับเบิลคลิกที่ชอร์ตคัตของโปรแกรมซึ่งอยู่บนเดสก์ทอป

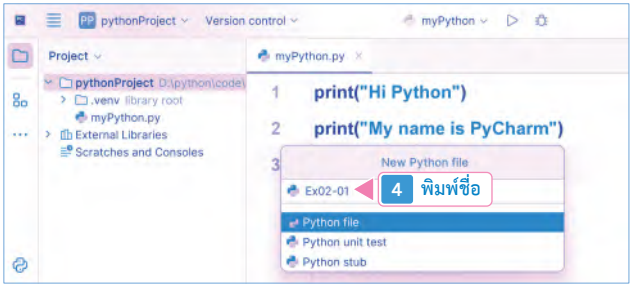
2. จะปรากฏหน้าต่างโปรแกรม PyCharm โดยโปรแกรมจะเปิดไฟล์ myPython.py ที่ผู้อ่านสร้างในบทที่ 1 ค้างไว้



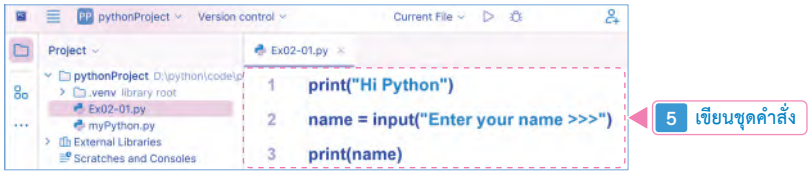
3. คลิกขวาที่โฟลเดอร์ pythonProject แล้วเลือกคำสั่ง New > Python File



4. จะปรากฏหน้าต่างให้ตั้งชื่อไฟล์ ในที่นี้ใช้ชื่อไฟล์ว่า Ex02-01 และกดปุ่ม Enter



5. จะปรากฏหน้าต่างให้เขียนโปรแกรมชุดคำสั่ง ให้เขียนชุดคำสั่งดังรูป และชุดคำสั่งจะถูกบันทึกโดยอัตโนมัติ



จากโปรแกรม Ex02-01.py อธิบายการทำงานของโปรแกรมได้ดังนี้

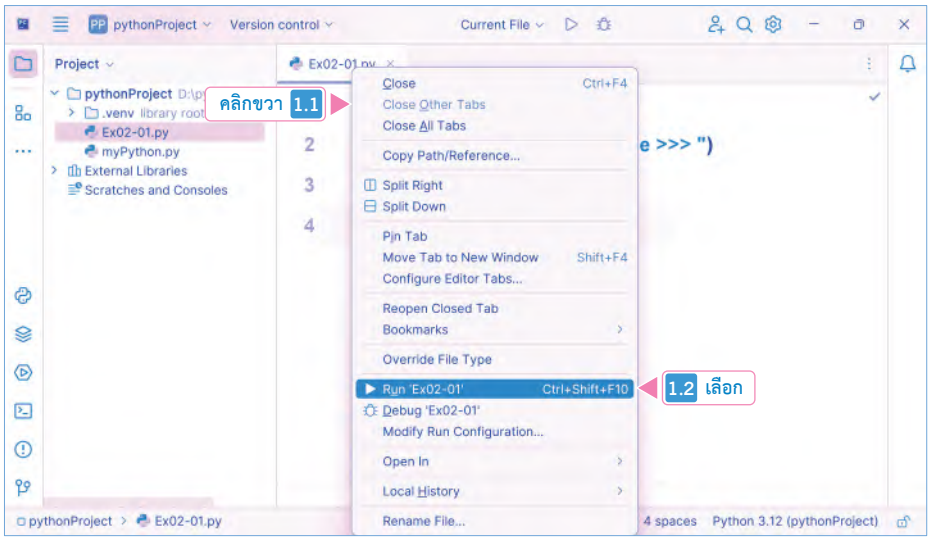
บรรทัดที่ 1 คำสั่ง print ให้แสดงผลข้อความ Hi Python

บรรทัดที่ 2 คำสั่ง input ให้รับข้อมูลหลังจากข้อความ Enter your name >>> และเก็บค่าไว้ในตัวแปร name (ผู้อ่านสามารถเรียนรู้การใช้ตัวแปรได้ในบทที่ 3)

บรรทัดที่ 3 คำสั่ง print ให้แสดงผลข้อความที่เก็บในตัวแปร name

ขั้นที่ 4 ทดสอบการทำงานของโปรแกรม

1. ในการทดสอบการทำงานของโปรแกรม ผู้อ่านสามารถทำได้โดยคลิกขวาตรงพื้นที่ว่าง แล้วเลือกคำสั่ง Run 'Ex02-01'

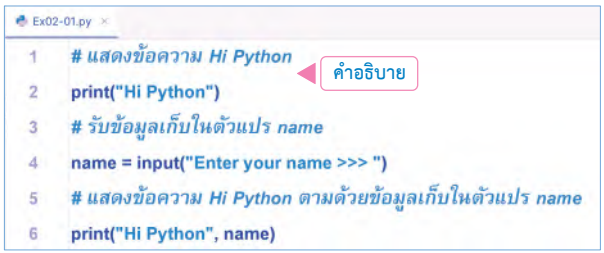


2. โปรแกรมก็จะแสดงผลการทำงาน

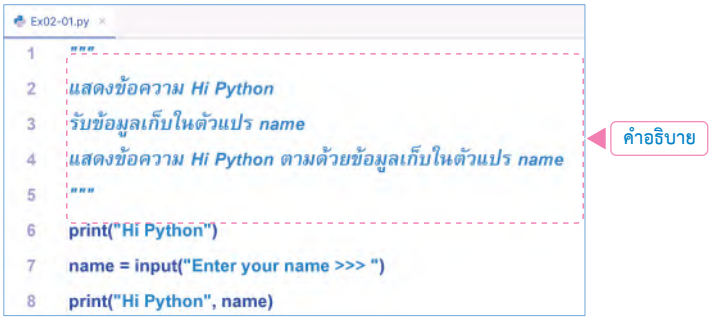
การเขียนคำอธิบายโปรแกรม (Comment)

เพื่อให้ง่ายต่อการทำความเข้าใจ และง่ายสำหรับการนำกลับมาแก้ไขหรือพัฒนาโปรแกรมใหม่ในภายหลัง ผู้อ่านควรจะเขียนคำอธิบายโปรแกรมในแต่ละส่วนการทำงาน เพื่อช่วยอธิบายให้ผู้อ่านเข้าใจการทำงานของโปรแกรม ซึ่งจะช่วยให้ผู้อ่านสามารถแก้ไขหรือพัฒนาโปรแกรมนั้นๆ ต่อได้โดยง่าย

ในการเขียนคำอธิบายโปรแกรมในภาษา Python ทำได้โดยใช้เครื่องหมาย # แล้วพิมพ์ข้อความต่อด้านหลังดังนี้



จากรูป เป็นการเขียนคำอธิบายในกรณี 1 บรรทัด เท่านั้น แต่ในกรณีที่มีคำอธิบายหลายบรรทัด ให้ใช้เครื่องหมาย """ ... """ ดังนี้



< NOTE >

ข้อพึงสังเกตในการศึกษาโค้ดโปรแกรม

1. การเขียนโค้ดโปรแกรมภาษา Python ข้อมูลที่เป็นชนิดข้อความ สามารถใช้เครื่องหมาย double quote ("...") หรือ single quote ('...') ครอบข้อความที่ต้องการก็ได้
2. ในหนังสือเล่มนี้ โค้ดโปรแกรมบางบรรทัดที่มีความยาวมากเกินไป ในหนังสืออาจไม่สามารถแสดงโค้ดภายในบรรทัดเดียวกันได้ หนังสือเล่มนี้จะขึ้นบรรทัดใหม่โดยไม่มีเครื่องหมาย \ ต่อท้ายบรรทัดและในการเขียนโค้ดโปรแกรมของผู้อ่านในคอมพิวเตอร์ ให้ผู้อ่านพิมพ์โค้ดโปรแกรมต่อไปในบรรทัดเดียวกันโดยไม่ต้องขึ้นบรรทัดใหม่

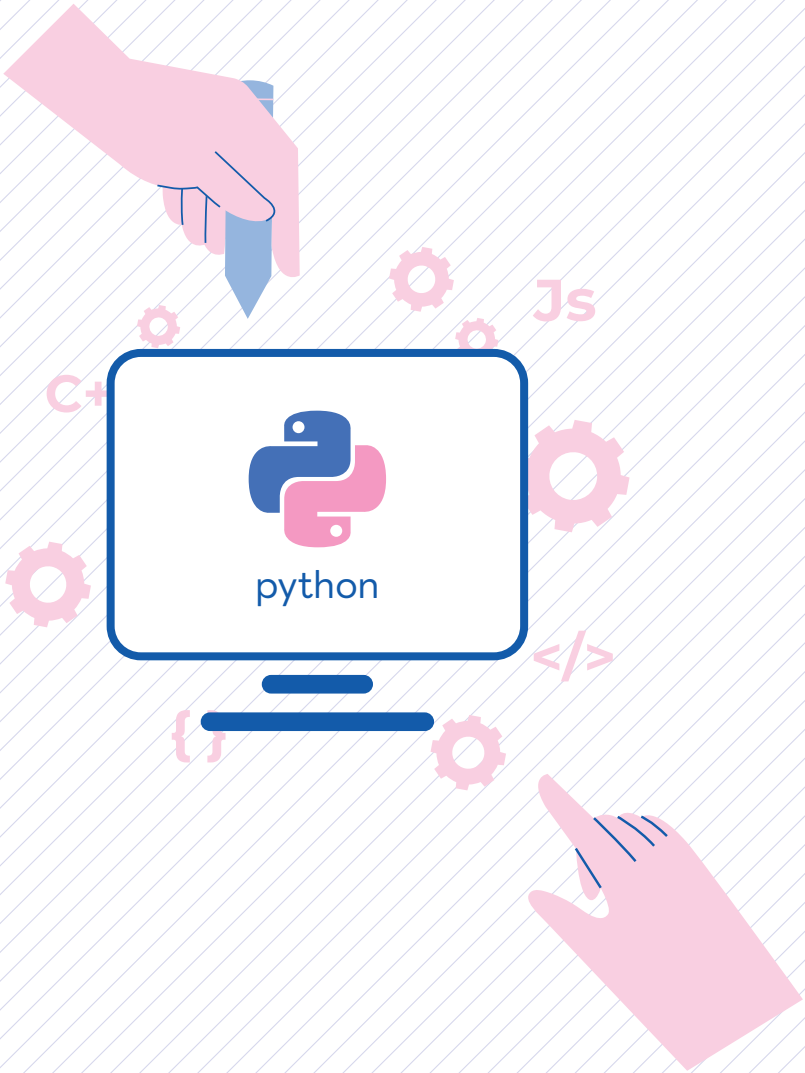
สรุปท้ายบท

ในบทนี้ผู้อ่านได้เรียนรู้ขั้นตอนการเขียนโปรแกรมภาษา Python ซึ่งมีการสร้างโปรเจ็กต์ด้วยโปรแกรม PyCharm การใช้คำสั่ง print() สำหรับการแสดงผล และการใช้คำสั่ง input() สำหรับการรับข้อมูล

ส่วนการเขียนคำอธิบายโปรแกรมจะทำได้ 2 กรณีคือ แบบที่มีคำอธิบาย 1 บรรทัด ใช้เครื่องหมาย # และแบบที่มีคำอธิบายหลายบรรทัดให้เขียนแทรกระหว่างเครื่องหมาย """..."""

แบบฝึกหัด

1. จงอธิบายขั้นตอนการเขียนโปรแกรมภาษา Python
2. จงอธิบายวิธีการเขียนคำอธิบายโปรแกรม (Comment) ในภาษา Python


Js
C+
python
</>
{ }

บทที่

03 >>

ตัวแปร ชนิดข้อมูล นิพจน์ และตัวดำเนินการ (Variable, Data Type, Expression and Operator)

ในการเขียนโปรแกรมให้สามารถทำงานได้อย่างถูกต้อง ควรเริ่มต้นจากการทำความเข้าใจเกี่ยวกับตัวแปรและชนิดของข้อมูล รวมถึงนิพจน์และตัวดำเนินการระหว่างข้อมูลในแต่ละชนิด ซึ่งในบทนี้ผู้อ่านจะได้เรียนรู้สิ่งเหล่านี้อย่างละเอียด

รู้จักกับตัวแปร (Variable)

ในการเขียนโปรแกรม ผู้อ่านจะต้องใช้ข้อมูลชนิดต่างๆ มาประมวลผลในโปรแกรม เช่น บวก ลบ คูณ หาร เพื่อให้ได้ผลการทำงานตามที่ต้องการ จึงมีการจัดเก็บข้อมูลไว้ในพื้นที่ของหน่วยความจำ ซึ่งจะเรียกแทนค่าข้อมูลในหน่วยความจำว่า "ตัวแปร"

ดังนั้น **ตัวแปร (Variable)** จึงหมายถึง ชื่อหรือสัญลักษณ์ที่กำหนดขึ้นเพื่ออ้างอิงถึงตำแหน่งหน่วยความจำที่ใช้เก็บข้อมูล ซึ่งสามารถเก็บข้อมูลที่มีชนิดของข้อมูลแตกต่างกันได้ และสามารถเปลี่ยนแปลงค่าข้อมูลได้ตลอดเวลา

กฎการตั้งชื่อตัวแปรในภาษา Python

ในการใช้งานตัวแปรทุกครั้ง ผู้อ่านจะต้องกำหนดชื่อที่จะใช้อ้างถึงตัวแปรนั้นๆ ซึ่งการตั้งชื่อตัวแปรในภาษา Python จะมีกฎง่ายๆ ดังนี้

- ชื่อตัวแปรต้องขึ้นต้นด้วย a-z, A-Z หรือเครื่องหมาย _ (Underscore)
- ภายในชื่อตัวแปรประกอบด้วย a-z, A-Z, 0-9 หรือเครื่องหมาย _ (Underscore)
- ห้ามเป็นช่องว่างหรือเครื่องหมายอื่นๆ
- การใช้ตัวอักษรพิมพ์เล็กและพิมพ์ใหญ่จะแตกต่างกัน เรียกว่าเป็น **Case Sensitive**

- ห้ามใช้ชื่อซ้ำกับ**คำสงวน (Reserved Word)** ในภาษา Python ดังนี้

and	as	assert	break	class
continue	def	del	elif	else
except	exec	finally	for	from
global	if	import	in	is
lambda	nonlocal	not	or	pass
raise	return	try	while	with
yield	True	False	None	

หมายเหตุ การใช้ชื่อตัวแปรควรให้สื่อความหมาย จะทำให้ผู้อ่านเข้าใจและสามารถปรับปรุงแก้ไขได้ง่าย

การใช้งานตัวแปรในภาษา Python นั้น ไม่ต้องมีการประกาศหรือกำหนดตัวแปรขึ้นมาก่อน อีกทั้งไม่ต้องกำหนดชนิดของข้อมูล ผู้อ่านสามารถกำหนดค่าข้อมูลหรือการคำนวณใดๆ ที่ต้องการแล้วเก็บค่าผลลัพธ์ในตัวแปรได้เลย Python จะรับรู้ชนิดข้อมูลจากบริบทของคำสั่ง ซึ่งหมายความว่าชนิดข้อมูลจะเป็นไปตามค่าของข้อมูลที่จัดเก็บนั่นเอง

ตัวอย่างเช่น

```
name = "I am Python"   #ตัวแปร name มีค่าเท่ากับ I am Python มีชนิดเป็นข้อความ
score = 86             #ตัวแปร score มีค่าเท่ากับ 86 มีชนิดเป็นจำนวนเต็ม
grade = 4.0            #ตัวแปร grade มีค่าเท่ากับ 4.0 มีชนิดเป็นจำนวนทศนิยม
```

ชนิดของข้อมูล (Data Type)

ในการเขียนโปรแกรมจะมีการใช้งานข้อมูลอยู่ตลอดเวลา ซึ่งข้อมูลแต่ละชนิดก็จะมีรูปแบบและขอบเขตที่แตกต่างกัน ดังนั้น ก่อนผู้อ่านจะเริ่มเขียนโปรแกรม ควรรู้จักกับข้อมูลชนิดต่างๆ ก่อน

ข้อมูลชนิดตัวเลข (Number)

ข้อมูลชนิดตัวเลข (Number) จะประกอบด้วยเลขจำนวนเต็ม, เลขจำนวนทศนิยม และเลขจำนวนเชิงซ้อน โดยมีรายละเอียดดังนี้

1. **ข้อมูลเลขจำนวนเต็ม (Integer Number)** ประกอบด้วยเลขจำนวนเต็มที่เป็นทั้งค่าบวก ค่าลบ และยังสามารถใช้กับเลขฐานได้ดังนี้
   - เลขฐานสอง ใช้ตัวอักษร 0b นำหน้า
   - เลขฐานแปด ใช้ตัวอักษร 0o นำหน้า
   - เลขฐานสิบหก ใช้ตัวอักษร 0x นำหน้า

2. **ข้อมูลเลขจำนวนทศนิยม (Floating Point Number)** ประกอบด้วยเลขจำนวนที่เป็นทศนิยม มีเครื่องหมาย . คั่นระหว่างตัวเลข และสามารถเขียนในรูปแบบของเลขยกกำลังสิบ (Exponential Form) ได้อีกด้วย
3. **ข้อมูลเลขจำนวนเชิงซ้อน (Complex Number)** ประกอบด้วยเลขจำนวนเชิงซ้อน ซึ่งจะถูกใช้ในงานทางวิทยาศาสตร์และวิศวกรรม โดยเขียนอยู่ในรูปแบบของคู่ x+yi โดยเรียก x ว่า จำนวนจริง และเรียก y ว่า ส่วนจินตภาพ และจะใช้ j แทน i

Python จะรับรู้ชนิดข้อมูลแต่ละประเภทจากการที่ผู้อ่านกำหนดค่าลงในชุดคำสั่ง ดังนั้น ผู้อ่านจะต้องพึงระวังในการกำหนดค่าข้อมูลกับตัวแปร ให้ตรงตามชนิดข้อมูลที่ต้องการใช้งาน

ตัวอย่างเช่น

```
A = 10          #ตัวแปร A มีค่าเท่ากับ 10 มีชนิดเป็นจำนวนเต็ม
B = 0b1010      #ตัวแปร B เก็บข้อมูลเลขฐานสอง มีค่าเท่ากับ 10 มีชนิดเป็นจำนวนเต็ม
C = 0o12        #ตัวแปร C เก็บข้อมูลเลขฐานแปด มีค่าเท่ากับ 10 มีชนิดเป็นจำนวนเต็ม
D = 0xA         #ตัวแปร D เก็บข้อมูลเลขฐานสิบหก มีค่าเท่ากับ 10 มีชนิดเป็นจำนวนเต็ม
E = 10.0        #ตัวแปร E มีค่าเท่ากับ 10.0 มีชนิดเป็นจำนวนทศนิยม
F = 10x+2j      #ตัวแปร F มีค่าเท่ากับ 10x+2j มีชนิดเป็นจำนวนเชิงซ้อน
G = 15e-2       #ตัวแปร G มีค่าเท่ากับ 0.15 มีชนิดเป็นจำนวนทศนิยม
H = 35e+2       #ตัวแปร H มีค่าเท่ากับ 3500.0 มีชนิดเป็นจำนวนทศนิยม
```

ในส่วนของการดำเนินการกับข้อมูลชนิดตัวเลขในแต่ละประเภท ผู้อ่านจะได้ศึกษาและเรียนรู้ในหัวข้อตัวดำเนินการต่อไป

ข้อมูลชนิดค่าความจริง (Boolean)

ข้อมูลชนิดค่าความจริง (Boolean) เป็นข้อมูลที่มีค่าความจริงเป็นจริงหรือเป็นเท็จเท่านั้น ซึ่งจะแทนค่าความจริงที่เป็นจริงด้วย True และจะแทนค่าความจริงที่เป็นเท็จด้วย False

ตัวอย่างเช่น

```
A = 9>5         #ตัวแปร A มีค่าเท่ากับ True
B = 9-5>9+5     #ตัวแปร B มีค่าเท่ากับ False
C = True        #ตัวแปร C มีค่าเท่ากับ True
D = False       #ตัวแปร D มีค่าเท่ากับ False
```

ข้อมูลชนิด None

ข้อมูลชนิด None เป็นข้อมูลที่ไม่มีค่าข้อมูล หรือยังไม่ได้รับการกำหนดค่า ซึ่งไม่ใช่ข้อมูลที่เก็บค่าศูนย์ หรือข้อมูลที่เป็นค่าว่างแต่อย่างใด

ผู้อ่านสามารถนำ None มาใช้ตรวจสอบสถานะของการดำเนินการใดๆ ที่ไม่สามารถให้ค่าผลลัพธ์ได้ โดยใช้รูปแบบเหมือนกับการตรวจสอบข้อมูลชนิด Boolean

ตัวอย่างเช่น

```
ans = None          #ตัวแปร ans มีค่าเท่ากับ None
X = 3               #ตัวแปร X มีค่าเท่ากับ 3
Y = 0               #ตัวแปร Y มีค่าเท่ากับ 0
If Y != 0
    ans = X/Y

#ผลที่ได้คือ ตัวแปร ans มีค่าเท่ากับ None
```

ข้อมูลแบบเรียงลำดับ (Sequence)

ข้อมูลแบบเรียงลำดับ (Sequence) จะประกอบด้วยข้อมูลที่จัดเก็บเรียงลำดับต่อกัน ได้แก่ ข้อความ (String), ลิสต์ (List) และทูเปิล (Tuple) โดยมีรายละเอียดดังนี้

1. **ข้อมูลชนิดข้อความ (String)** เป็นข้อมูลที่ประกอบด้วยข้อความ ตัวอักษร ตัวเลขที่ไม่สามารถคำนวณได้ จำนวนตั้งแต่ 1 ตัว มาเรียงต่อกันภายในเครื่องหมาย double quote ("...") หรือเครื่องหมาย single quote ('...')

   โดยการกำหนดค่าตัวแปรชนิดนี้ ผู้อ่านจะต้องเขียนให้อยู่ในบรรทัดเดียวกัน แต่สามารถใช้เครื่องหมาย backslash (\) แทรก หากต้องการแยกเป็นบรรทัดใหม่

   ข้อมูลในกลุ่มนี้ ได้แก่ ตัวอักษร (A ถึง Z, a ถึง z) ตัวเลข (0 ถึง 9) และสัญลักษณ์พิเศษต่างๆ เช่น + - = $ * เป็นต้น

   ในกรณีที่มีข้อมูลเพียง 1 ตัว Python จะรับรู้ชนิดของข้อมูลนี้ว่าเป็น**ชนิดตัวอักขระ (Character)** ซึ่งจะมีตัวเลขแทนแต่ละตัวอักขระไว้สำหรับใช้ในการดำเนินการ เรียกตัวเลขเหล่านั้นว่า **รหัสแอสกี้ (Ascii Code)**

ตัวอย่างเช่น

```
strA = "3020158"        #ตัวแปร strA มีค่าเท่ากับ 3020158
strB = '3020158'        #ตัวแปร strB มีค่าเท่ากับ 3020158
strC = '"Monday"'       #ตัวแปร strC มีค่าเท่ากับ "Monday"
strD = "'Jan'"          #ตัวแปร strD มีค่าเท่ากับ 'Jan'
strE = "A & " \
  "B"                   #ตัวแปร strE มีค่าเท่ากับ A & B
strF = "A"              #ตัวแปร strF มีค่าเท่ากับ A
noF = ord(strF)         #ord เป็นฟังก์ชันแปลงตัวอักขระ "A" เป็นค่าตัวเลข ดังนั้น ตัวแปร noF มีค่าเท่ากับ 65
```

2. **ข้อมูลชนิดลิสต์ (List)** เป็นข้อมูลที่ประกอบด้วยรายการข้อมูลชนิดใดๆ เรียงต่อกันภายในเครื่องหมายวงเล็บก้ามปู [...] และคั่นด้วยเครื่องหมายคอมม่า (,) ข้อมูลในเครื่องหมายวงเล็บก้ามปู [...] สามารถเป็นค่าว่างได้ ซึ่งจะเรียกว่า **ลิสต์ว่าง (Empty List)**

ตัวอย่างเช่น

```
listA = ["Anna", 20, "Bangkok", "Aree", 22, "Chiangmai"]
#ตัวแปร listA มีค่าเท่ากับ ["Anna", 20, "Bangkok", "Aree", 22, "Chiangmai"]

listA.append("Adam")
#ตัวแปร listA มีค่าเท่ากับ ["Anna", 20, "Bangkok", "Aree", 22, "Chiangmai", "Adam"]
#ตัวแปร listA[0] มีค่าเท่ากับ Anna
#ตัวแปร listA[1] มีค่าเท่ากับ 20
#ตัวแปร listA[2] มีค่าเท่ากับ Bangkok
#ตัวแปร listA[3] มีค่าเท่ากับ Aree
#ตัวแปร listA[4] มีค่าเท่ากับ 22
#ตัวแปร listA[5] มีค่าเท่ากับ Chiangmai
#ตัวแปร listA[6] มีค่าเท่ากับ Adam
```

3. **ข้อมูลชนิดทูเปิล (Tuple)** เป็นข้อมูลที่ประกอบด้วยรายการข้อมูลชนิดใดๆ เรียงต่อกัน คั่นด้วยเครื่องหมายคอมม่า (,) และสามารถเป็นค่าว่างได้เหมือนกับข้อมูลชนิดลิสต์ แต่จะเรียงต่อกันภายในเครื่องหมายวงเล็บ (...) หรือไม่ต้องมีเครื่องหมายวงเล็บ (...) ครอบก็ได้ และไม่สามารถเพิ่ม เปลี่ยนแปลง หรือลบข้อมูลในทูเปิลได้

ตัวอย่างเช่น

```
tupleA = ("Anna", 20, "Bangkok", "Aree", 22, "Chiangmai")
#ตัวแปร strA มีค่าเท่ากับ ("Anna", 20, "Bangkok", "Aree", 22, "Chiangmai")

tupleB = "Anna", 20, "Bangkok", "Aree", 22, "Chiangmai"
#ตัวแปร strB มีค่าเท่ากับ ("Anna", 20, "Bangkok", "Aree", 22, "Chiangmai")

#ตัวแปร tupleA[0] มีค่าเท่ากับ Anna
#ตัวแปร tupleA[1] มีค่าเท่ากับ 20
#ตัวแปร tupleA[2] มีค่าเท่ากับ Bangkok
#ตัวแปร tupleB[3] มีค่าเท่ากับ Aree
#ตัวแปร tupleB[4] มีค่าเท่ากับ 22
#ตัวแปร tupleB[5] มีค่าเท่ากับ Chiangmai
**ตัวแปร tupleA และ tupleB ไม่สามารถใช้ฟังก์ชัน append() เนื่องจากเป็นข้อมูลชนิด tuple**
```

ข้อมูลชนิดเซต (Set)

ข้อมูลชนิดเซต (Set) จะประกอบด้วยรายการข้อมูลชนิดใดๆ เรียงต่อกัน คั่นด้วยเครื่องหมายคอมม่า (,) สามารถเป็นค่าว่างได้ และสามารถเพิ่ม เปลี่ยนแปลง หรือลบข้อมูลในเซตได้เหมือนกับข้อมูลชนิดลิสต์ แต่มีข้อแตกต่างกันคือ ข้อมูลชนิดเซตจะเรียงต่อกันภายในเครื่องหมายวงเล็บปีกกา {...} การจัดเก็บข้อมูลจะไม่มีลำดับ กล่าวคือ ไม่สามารถเข้าถึงข้อมูลด้วยเลขตำแหน่งข้อมูล และจะไม่เก็บข้อมูลที่ซ้ำกัน

ตัวอย่างเช่น

```
setA = {"Anna", 20, "Bangkok", "Aree", 22, "Chiangmai", 22}
#ตัวแปร setA มีค่าเท่ากับ {"Chiangmai", "Aree", "Anna", 20, 22, "Bangkok"}

setA.add("Aree")
#ตัวแปร setA มีค่าเท่ากับ {20, 22, "Chiangmai", "Aree", "Anna", "Bangkok"}
**เมื่อแสดงผลข้อมูลใน setA ข้อมูลที่ได้จะไม่เรียงลำดับ และข้อมูลที่ซ้ำกันจะแสดงออกมาเพียงค่าเดียวเท่านั้น**
```

ข้อมูลชนิดดิกชันนารี (Dictionary)

ข้อมูลชนิดดิกชันนารี (Dictionary) เป็นข้อมูลที่ประกอบด้วยรายการข้อมูลชนิดใดๆ เรียงต่อกันและคั่นด้วยเครื่องหมายคอมม่า (,) เหมือนกับข้อมูลชนิดลิสต์ แต่จะเรียงต่อกันภายในเครื่องหมายวงเล็บปีกกา {...} และอยู่ในรูปแบบที่ประกอบด้วย 2 ส่วนคือ key: value โดยที่ key เป็นข้อมูลที่เป็นตัวชี้ และ value เป็นค่าข้อมูลที่สามารถอ้างอิงมาจาก key ได้

ตัวอย่างเช่น

```
scores = {'Anna': 20.6, 'Meena': 22.1, 'Wanna': 23.0}
scores['Weena'] = 21.7

#ตัวแปร scores['Weena'] มีค่าเท่ากับ 21.7
#ตัวแปร scores['Meena'] มีค่าเท่ากับ 22.1
#ตัวแปร scores['Wanna'] มีค่าเท่ากับ 23.0
#ตัวแปร scores['Anna'] มีค่าเท่ากับ 20.6
```

ส่วนของการใช้งานในรายละเอียด และฟังก์ชันสำหรับการดำเนินการกับข้อมูลแบบเรียงลำดับ ข้อมูลชนิดเซต และข้อมูลชนิดดิกชันนารี ผู้อ่านจะได้ศึกษาและเรียนรู้ในบทที่ 6 ต่อไป

การตรวจสอบชนิดของข้อมูล (Data Type Checking)

เนื่องจากในภาษา Python ผู้อ่านไม่ต้องประกาศชนิดข้อมูลของตัวแปร Python เราสามารถรับรู้ชนิดข้อมูลจากค่าที่กำหนดในชุดคำสั่ง แต่ในบางครั้งก็มีความจำเป็นต้องทราบชนิดข้อมูลของตัวแปรก่อนที่จะประมวลผลอย่างใดอย่างหนึ่ง ดังนั้น ผู้อ่านสามารถใช้ฟังก์ชัน **type()** ตรวจสอบชนิดของข้อมูลได้ ซึ่งมีรูปแบบการทำงานดังนี้

```
type(variable)
```

โดยที่ **variable** เป็นตัวแปรที่ต้องการตรวจสอบชนิดข้อมูล

ตัวอย่างเช่น

```
a = 5
print(type(a))
#ชนิดข้อมูลของตัวแปร a คือ <class 'int'>
b = 5.0
print(type(b))
#ชนิดข้อมูลของตัวแปร b คือ <class 'float'>
c = [1, 2, 3]
print(type(c))
#ชนิดข้อมูลของตัวแปร c คือ <class 'list'>
d = (1, 2, 3)
print(type(d))
#ชนิดข้อมูลของตัวแปร d คือ <class 'tuple'>
e = {1, 2, 3}
print(type(e))
#ชนิดข้อมูลของตัวแปร e คือ <class 'set'>
f = {1: "one", 2: "two", 3: "three"}
print(type(f))
#ชนิดข้อมูลของตัวแปร f คือ <class 'dict'>
g = float
print(type(g))
#ชนิดข้อมูลของตัวแปร g คือ <class 'type'>
h = len
print(type(h))
#ชนิดข้อมูลของตัวแปร h คือ <class 'builtin_function_or_method'>
```

จากตัวอย่างข้างต้น นอกจากชนิดข้อมูลที่ใช้แล้ว ภาษา Python ยังสามารถตรวจสอบชื่อที่ใช้เป็นชนิดข้อมูลหรือชื่อที่ใช้เป็นฟังก์ชันในภาษา Python ได้ เช่น

- float ตรวจสอบได้เป็น <class 'type'> ซึ่งหมายถึง ชนิดข้อมูล
- len ตรวจสอบได้เป็น <class 'builtin_function_or_method'> ซึ่งหมายถึง ฟังก์ชัน

เมื่อผู้อ่านทราบชนิดข้อมูลโดยการใช้ฟังก์ชัน type() แล้ว หากต้องการจะแปลงชนิดข้อมูลก่อนจะประมวลผลใดๆ ก็สามารถทำได้ ซึ่งผู้อ่านจะได้ศึกษาและเรียนรู้ในหัวข้อถัดไป

การแปลงชนิดของข้อมูล (Data Type Conversion)

การแปลงชนิดข้อมูลในภาษา Python ผู้อ่านสามารถใช้ Built-in Function ที่มีอยู่แล้วใน Python ได้เลย โดยฟังก์ชันเหล่านั้นจะมีชื่อเหมือนกับชนิดข้อมูลนั้นๆ ซึ่งก็คือ ชื่อของคลาสนั่นเอง

ตัวอย่างเช่น

- ฟังก์ชัน int() ใช้แปลงข้อมูลประเภทใดๆ ให้เป็นชนิดตัวเลขจำนวนเต็ม
- ฟังก์ชัน str() ใช้แปลงข้อมูลประเภทใดๆ ให้เป็นชนิดตัวอักษร

ตารางแสดงตัวอย่างของฟังก์ชันสำหรับการแปลงข้อมูลในภาษา Python

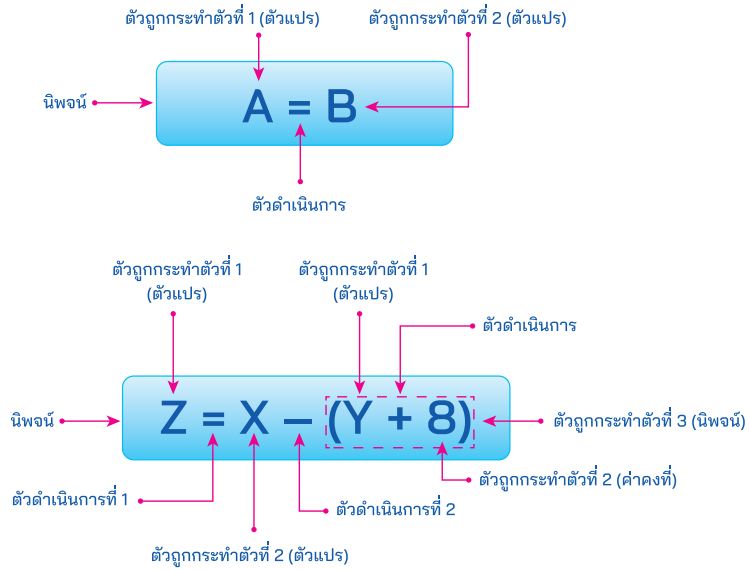
ชื่อฟังก์ชัน	คำอธิบาย
int(x [,base])	แปลงข้อมูล x จากฐานที่กำหนด base ให้เป็นเลขจำนวนเต็ม
float(x)	แปลงข้อมูล x ให้เป็นเลขจำนวนทศนิยม
complex(real [,im])	สร้างตัวเลขจำนวนเชิงซ้อนจากค่า real และค่า imagine
str(x)	แปลงข้อมูล x ให้เป็นตัวอักษร
tuple(s)	แปลงข้อมูลแบบเรียงลำดับ s ให้เป็นข้อมูลทูเปิล
list(s)	แปลงข้อมูลแบบเรียงลำดับ s ให้เป็นข้อมูลลิสต์
set(s)	แปลงข้อมูลแบบเรียงลำดับ s ให้เป็นข้อมูลเซต
dict(d)	แปลงข้อมูล d ให้เป็นข้อมูลดิกชันนารี
chr(x)	แปลงข้อมูลเลขจำนวนเต็ม x ให้เป็นตัวอักขระ
ord(x)	แปลงข้อมูลตัวอักขระ x ให้เป็นค่าเลขจำนวนเต็ม
hex(x)	แปลงข้อมูลตัวเลข x ให้เป็นเลขฐานสิบหก ชนิดตัวอักษร
oct(x)	แปลงข้อมูลตัวเลข x ให้เป็นเลขฐานแปด ชนิดตัวอักษร
bin(x)	แปลงข้อมูลตัวเลข x ให้เป็นเลขฐานสอง ชนิดตัวอักษร

นิพจน์ (Expression)

นิพจน์ (Expression) หมายถึง ข้อความหรือประโยคที่เขียนอยู่ในรูปสัญลักษณ์ โดยการนำข้อมูล ตัวแปร ฟังก์ชัน หรือค่าคงที่มาสัมพันธ์กับตัวดำเนินการ (Operator) อย่างใดอย่างหนึ่ง ซึ่งผู้เขียนจะได้อธิบายในหัวข้อถัดไป

ในการสร้างนิพจน์หนึ่งๆ นั้น นิพจน์จะต้องมีตัวถูกกระทำ (Operand) อย่างน้อยหนึ่งตัว และตัวดำเนินการ (Operator) อย่างน้อยหนึ่งตัวเสมอ

ตัวอย่างเช่น



นิพจน์ข้อความ (String Expression) หมายถึง ข้อความที่มีการเชื่อมต่อกันโดยใช้เครื่องหมายบวก (+) หรือเครื่องหมายคอมม่า (,) ซึ่งมีความหมายดังนี้

เครื่องหมาย	คำอธิบาย
บวก (+)	ข้อความที่มีการเชื่อมต่อกันจะถูกนำมาต่อกัน
คอมม่า (,)	ข้อความที่มีการเชื่อมต่อกันจะถูกนำมาต่อกัน โดยมีช่องว่างระหว่างข้อความ (blank) 1 ช่อง

ตัวอย่างเช่น

```
print("A"+"&"+"B")          #จะแสดงข้อความ A&B
print("A","&","B")          #จะแสดงข้อความ A & B
```

ตัวดำเนินการ (Operator)

ตัวดำเนินการในภาษา Python สามารถจัดกลุ่มตามการทำงานได้ดังนี้

ตัวดำเนินการกำหนดค่า (Assignment Operator)

ตัวดำเนินการประเภทนี้ ใช้สำหรับกำหนดค่าให้กับตัวแปรทางด้านซ้ายของตัวดำเนินการ =