

อรพิน ประวิติบริสุทธิ์
ผู้เขียนหนังสือติดอันดับขายดีด้าน Programming



PROVISION

คัมภีร์ Python

ฉบับสมบูรณ์

คู่มือภาษาไพธอนที่ละเอียดที่สุด พร้อมตัวอย่างใช้งานจริง

พื้นฐาน Python / ตัวแปร / โครงสร้างภาษา • วิธีใช้ IDE • ใช้งานข้อมูล MariaDB / MongoDB (NoSQL)
สร้างเว็บแอปด้วย Django • OOP • ดึงข้อมูลบนเว็บ (web scraping) • สร้าง API ด้วย Flask / FastAPI
ทำงาน Data Science ด้วยไลบรารี NumPy, Pandas • ทำ Data Visualization ด้วย Matplotlib



DOWNLOAD
[provision.co.th/
python_bible](https://provision.co.th/python_bible)

คัมภีร์ Python

ฉบับสมบูรณ์

ผู้แต่ง อรพิน ประวัติบริสุทธิ์
ออกแบบปก ธนกฤต เกียรติศักดิ์ากุล
จัดรูปเล่ม อภิขญา สุทธิประภา

ISBN : 978-616-204-793-0
978-616-204-804-3 (E-book)

ราคา : 395 บาท

ข้อมูลบรรณานุกรมของหอสมุดแห่งชาติ

อรพิน ประวัติบริสุทธิ์.

คัมภีร์ Python ฉบับสมบูรณ์.

--กรุงเทพฯ : โปรวิชั่น, 2564. 608 หน้า.

1. ไพธอน (ภาษาคอมพิวเตอร์). I. ชื่อเรื่อง.

005.133

ชื่อและเครื่องหมายการค้าอื่นๆ ที่อ้างถึงในหนังสือฉบับนี้ และในเนื้อหาหรือสื่อบันทึกข้อมูลอื่นๆ ที่เกี่ยวข้อง (ถ้ามี) เป็นสิทธิโดยชอบด้วยกฎหมายของเจ้าของแต่ละราย บริษัท โปรวิชั่น จำกัด มิได้อ้างความเป็นเจ้าของ ตัวแทน หรือมีส่วนเกี่ยวข้องกับแต่อย่างใด นอกจากนี้ทางบริษัทได้ใช้ความพยายามอย่างเต็มที่ในการตรวจสอบความถูกต้องของเนื้อหาเท่าที่จะทำได้ แต่มิได้ยอมรับหรือรับประกันความถูกต้องหรือทันสมัยของข้อมูลต่างๆ ในทุกกรณี ผู้อ่านพึงใช้วิจารณญาณของตนเอง หากพบข้อผิดพลาดหรือสงสัยใดๆ โปรดแจ้งที่อีเมล editor@provision.co.th เพื่อที่ทางบริษัทจะได้ตรวจสอบและแก้ไขตามความเหมาะสมในการพิมพ์ครั้งต่อไป รวมทั้งประกาศให้ทราบผ่านทางเว็บไซต์ www.provision.co.th/update ด้วย

สงวนสิทธิ์ตามพระราชบัญญัติลิขสิทธิ์ พ.ศ. 2537 โดย บริษัท โปรวิชั่น จำกัด ห้ามนำส่วนใดส่วนหนึ่งของหนังสือเล่มนี้ ไปทำซ้ำ ตัดแปลงหรือเผยแพร่ต่อสาธารณชนไม่ว่ารูปแบบใดๆ นอกจากนี้จะได้อนุญาตเป็นลายลักษณ์อักษรล่วงหน้าจากทางบริษัทเท่านั้น ข้อผลิตภัณฑ์และเครื่องหมายการค้าต่างๆ ที่อ้างถึงเป็นสิทธิโดยชอบด้วยกฎหมายของบริษัทนั้นๆ



DOWNLOAD
[provision.co.th/
Python_bible](http://provision.co.th/Python_bible)

PROVISION

จัดพิมพ์โดย : บริษัท โปรวิชั่น จำกัด
11/8 ซอยแจ้งวัฒนะ 14 แขวงทุ่งสองห้อง
เขตหลักสี่ กทม. 10210 โทร: 0-2077-4058

ผู้ก่อตั้ง/ประธานกรรมการ วติน เพิ่มทรัพย์
กรรมการบริหาร วงศ์ประชา จันทร์สมวงศ์
 อนิรุทธิ์ รัชตะวราห์
ประสานงานกอง บก. นนทพร ดันติเยี่ยมสกุล

ติดต่อสั่งซื้อหนังสือ

✉ sales@provision.co.th
โทร. 0-2077-4058 หรือ 08-1928-5299
💻 www.dplusshop.com f dplusshop

ติดต่อโฆษณาและการตลาด

✉ marketing@provision.co.th
สอบถามปัญหา/ติดต่อกอง บก.
✉ editor@provision.co.th

ติดตามข่าวสาร

💻 www.provision.co.th, www.dplusguide.com
f provision1991, dplusguide
t @Provision1991, @dplusguide
provision1991, dplusguide
@dplusguide

จัดจำหน่ายโดย :

บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน)
1858/87-90 ถ.เพชรดิน
แขวงบางนาใต้ เขตบางนา กรุงเทพฯ 10260
โทร. 0-2826-8000 FAX: 0-2826-8999

ที่ปรึกษากฎหมาย : คุณไพบุลย์ อมรภิญโญเกียรติ
สำนักงานกฎหมาย P&P LAW FIRM
โทร. 0-2651-2121

พิมพ์ที่ : บริษัท พิมพ์ดี จำกัด
นายเสริม พูนพนิช ผู้พิมพ์ผู้โฆษณา พ.ศ. 2564

INTRODUCTION

ปัจจุบันเทคโนโลยีสารสนเทศไม่ใช่เรื่องไกลตัวอีกต่อไป ชีวิตประจำวันของคนส่วนใหญ่ล้วนเกี่ยวข้องกับสัมพันธกับเทคโนโลยีสารสนเทศอยู่ตลอดเวลา จึงไม่ใช่เรื่องแปลกที่จะเห็นผู้ใหญ่และเด็กในยุคนี้หันมาศึกษาการเขียนโปรแกรมคอมพิวเตอร์กันอย่างจริงจัง เพื่อใช้สนับสนุนกิจการงานต่างๆ

กระทรวงศึกษาธิการได้เล็งเห็นถึงความสำคัญในจุดนี้ และคิดว่าเด็กๆ ควรจะต้องเรียนรู้เกี่ยวกับวิทยาการและเทคโนโลยีต่างๆ ดังนั้น ในเดือนพฤษภาคม 2561 กระทรวงศึกษาธิการจึงบรรจุวิชาใหม่ที่ชื่อว่า “วิทยาการคำนวณ” เข้าไว้ในหลักสูตรด้วย ซึ่งการเขียนโปรแกรมด้วยภาษา Python ก็เป็นเนื้อหาส่วนหนึ่งในวิชาวิทยาการคำนวณ

Python เป็นภาษาโปรแกรมที่คนส่วนใหญ่นิยมเลือกหยิบนำมาเขียนโปรแกรม เนื่องจาก Python สามารถทำงานได้บนระบบปฏิบัติการที่หลากหลาย และยังเป็น open source ที่สามารถดาวน์โหลดมาใช้งานได้ฟรีด้วย โครงสร้างของตัวภาษาก็เข้าใจง่าย ไม่ซับซ้อน เหมาะมากสำหรับผู้ต้องการเริ่มต้นศึกษาการเขียนโปรแกรม

ใน Part 1 ของหนังสือจะเริ่มสอนตั้งแต่พื้นฐานการเขียนโปรแกรม ใครไม่มีพื้นฐานมาก่อนหรือไม่รู้ว่าเขียนโปรแกรมต้องเริ่มจากตรงไหน หนังสือเล่มนี้ตอบใจหายมากๆ สามารถอ่านทำความเข้าใจได้โดยง่าย และเมื่อปูพื้นฐานเสร็จแล้ว ใน Part 2 และ Part 3 ก็จะอธิบายเนื้อหาที่ยากขึ้น โดยจะอธิบายถึงการพัฒนาเว็บแอปพลิเคชันด้วยภาษาไพธอน และการเขียนโปรแกรมภาษาไพธอนสำหรับงาน Data Science หนังสือเล่มนี้จึงเป็นหนังสือที่อธิบายเนื้อหาเกี่ยวกับ Python ไว้อย่างครบวงจรเลยทีเดียว

ท้ายสุดนี้ ผู้เขียนอยากขอพื้นที่วรรคสุดท้ายตรงหน้ากระต่ายนี้กล่าวคำขอบคุณ คุณประสิทธิ์ ประดิษฐ์สุทธิ และคุณวิภา ประดิษฐ์สุทธิ คุณพ่อคุณแม่ของผู้เขียน ผู้ซึ่งมอบกำลังใจดีๆ ให้กับผู้เขียนเสมอมา กำลังใจนี้เป็นส่วนสำคัญยิ่งในการผลักดันให้ผู้เขียนสร้างสรรค์ผลงานเขียนต่างๆ ออกมาสู่สายตาของผู้อ่าน

อรพิน ประดิษฐ์สุทธิ

AUTHOR

อรพิน ประดิษฐ์สุทธิ

- ☉ Sun Certified Java Programmer 1.4 (SCJP 1.4)
- ☉ Sun Certified Web Component Developer 1.4 (SCWCD 1.4)

ประวัติการศึกษา

จบการศึกษาระดับปริญญาตรี สาขาวิชาวิทยาการคอมพิวเตอร์ประยุกต์ จากคณะวิทยาศาสตร์ประยุกต์ สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ และจบการศึกษาปริญญาโท สาขาวิชาเทคโนโลยีสารสนเทศ จากคณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ประวัติการทำงาน

มีประสบการณ์ด้านการเป็นนักวิเคราะห์และออกแบบระบบให้กับบริษัทเอกชนที่มีชื่อเสียง และมีผลงานเขียนติดอันดับหนังสือขายดี เช่น คู่มือเขียนโปรแกรมด้วยภาษา C ฉบับสมบูรณ์, คู่มือเขียนโปรแกรมด้วยภาษา JAVA ฉบับสมบูรณ์, คู่มือสร้างเว็บไซต์ด้วย HTML5 CSS3 & JavaScript ฉบับสมบูรณ์, พัฒนาเว็บแอปบน Smartphone /Tablet ด้วย jQuery Mobile, เริ่มต้นเขียนโปรแกรมด้วยภาษา Scratch 3, One-stop Python เรียนรู้ภาษาไพธอนในเล่มเดียว, Python สำหรับงาน Data Science Data Visualization และ Machine Learning

CONTENT

1 BASIC PYTHON PART การเขียนโปรแกรม ภาษาไพธอนขั้นพื้นฐาน

CHAPTER 01 | ทำความรู้จักกับภาษา คอมพิวเตอร์ไพธอน

ภาษาคอมพิวเตอร์คืออะไร.....	16
ตัวแปลภาษาคืออะไร.....	17
ประเภทของภาษาคอมพิวเตอร์.....	18
ภาษาระดับต่ำ (Low level language).....	18
ภาษาระดับสูง (High level language).....	19
ความแตกต่างระหว่าง Interpreter และ Compiler.....	19
Interpreter	19
Compiler	20
ความเป็นมาของภาษาไพธอน	20
โปรแกรมภาษาไพธอนมีหน้าตาอย่างไร	21
ทำไมควรเลือกเขียนโปรแกรม ด้วยภาษาไพธอน.....	21

CHAPTER 02 | ปูพื้นฐานความรู้ ในการเขียนโปรแกรม

โปรแกรม คืออะไร.....	23
อยากเขียนโปรแกรม ... เริ่มอย่างไรดี?	23
เรียนรู้การเขียนซูโดโค้ด (Pseudocode)	25
เรียนรู้การเขียนผังงาน (Flowchart)	30

CHAPTER 03 | เครื่องมือเขียนโปรแกรม ภาษาไพธอน

IDE คืออะไร	36
Python IDLE	36
วิธีดาวน์โหลดและติดตั้งโปรแกรม Python for Windows.....	36
วิธีเรียกใช้งาน Python IDLE.....	38
เขียนโปรแกรมไพธอนโดยใช้ Python IDLE.....	39
วิธีที่ 1 : เขียนโปรแกรมลงบน Python Shell	39
วิธีที่ 2 : เขียนโปรแกรมในรูปแบบ source code	40
PyCharm	41
วิธีดาวน์โหลดและติดตั้ง โปรแกรม PyCharm	41
วิธีเรียกใช้งาน PyCharm.....	43
เขียนโปรแกรมไพธอนโดยใช้ PyCharm.....	44
Jupyter Notebook.....	47
วิธีดาวน์โหลดและติดตั้งโปรแกรม Anaconda	47
การสร้าง Virtual Environment สำหรับ Python 3.9 บน Anaconda	51
ขั้นตอนการติดตั้ง Python 3.9 บน Anaconda.....	52
การติดตั้ง Jupyter Notebook ของ Python 3.9 บน Anaconda.....	54
วิธีเรียกใช้งาน Jupyter Notebook.....	55
การสร้าง folder เพื่อจัดเก็บโค้ดโปรแกรม ...	57

เขียนโปรแกรมไพธอนโดยใช้	
Jupyter Notebook	58
การเพิ่มและลบเซลล์	59
การคัดลอก/ตัดเซลล์เดิมไปไว้ยังเซลล์ใหม่	61
การสลับที่เซลล์	64
การรวมเซลล์และแตกเซลล์	65
การบันทึกโค้ดโปรแกรมใน	
Jupyter Notebook	67
การเปิดไฟล์โค้ดโปรแกรมใน	
Jupyter Notebook	69

CHAPTER

04 เริ่มต้นเขียนโปรแกรมด้วยภาษาไพธอน

การย่อหน้าโค้ดโปรแกรม (Indentation)	71
การแตกโค้ดออกเป็นหลายบรรทัด (Continuation)	71
กฎการตั้งชื่อของภาษาไพธอน	72
Keywords ในภาษาไพธอน	72
การเขียนคอมเมนต์ในโปรแกรม	73
การรับและแสดงผลข้อมูล	73
รับข้อมูลด้วยฟังก์ชัน input()	73
แสดงผลข้อมูลด้วยฟังก์ชัน print()	74

CHAPTER

05 ตัวแปร

ตัวแปรคืออะไร	78
การกำหนดค่าให้กับตัวแปร	78
การกำหนดค่าให้กับตัวแปรหลายตัว	79
ใช้ฟังก์ชัน input() รับข้อมูลเก็บลงในตัวแปร	79
ใช้ฟังก์ชัน print() แสดงค่าของตัวแปร	
ออกทางจอภาพ	81
การเปลี่ยนแปลงค่าของตัวแปร	82

CHAPTER

06 ชนิดข้อมูล

ชนิดข้อมูลคืออะไร	85
ประเภทของชนิดข้อมูลพื้นฐาน	
ในภาษาไพธอน	85
ชนิดข้อมูลประเภทตัวเลข (Numeric)	85
จำนวนเต็ม (Integers)	85
จำนวนทศนิยม (Floating point numbers)	86
จำนวนเชิงซ้อน (Complex numbers)	86
ชนิดข้อมูลตรรกะ (Boolean)	88
ชนิดข้อมูลสายอักขระ (String)	88
การหาค่าความยาวสตริง	89
การค้นหาตัวอักษรในสตริง	89
การค้นหากลุ่มตัวอักษรในสตริง	90
Removeprefix และ Removesuffix	
ของสตริง	91
ชนิดข้อมูลรายการ (List)	92
การหาค่าความยาวลิสต์	93
การค้นหาข้อมูลในลิสต์	93
การค้นหากลุ่มข้อมูลในลิสต์	95
การแก้ไขข้อมูลในลิสต์	97
การเพิ่มข้อมูลในลิสต์ด้วย	
ฟังก์ชัน append()	98
การลบข้อมูลในลิสต์ด้วย	
ฟังก์ชัน remove() และ del()	100
การเชื่อมต่อข้อมูลของแต่ละลิสต์	
เข้าด้วยกัน	101
Tuple	101
การหาค่าความยาวทูเพิล	102
การค้นหาข้อมูลในทูเพิล	102
การค้นหากลุ่มข้อมูลในทูเพิล	102
Set	104
การหาค่าความยาวเซต	105
การค้นหาข้อมูลในเซต	105

CONTENT

การเพิ่มข้อมูลในเซตด้วย ฟังก์ชัน add() และ update()	105
การลบข้อมูลในเซตด้วย ฟังก์ชัน remove() และ discard()	106
การเชื่อมต่อข้อมูลของแต่ละเซต เข้าด้วยกัน	108
Dictionaries	108
การหาค่าความยาวดิกชันนารี	109
การค้นหาข้อมูลในดิกชันนารี	110
การแก้ไขข้อมูลในดิกชันนารี	111
การ Merge และ Update ข้อมูลในดิกชันนารี	111
การเพิ่มข้อมูลในดิกชันนารี	113
การลบข้อมูลในดิกชันนารี ด้วยฟังก์ชัน pop() และ del()	113
การตรวจสอบชนิดข้อมูล	115

CHAPTER

07 } นิพจน์และ ตัวดำเนินการต่างๆ

นิพจน์คืออะไร	117
ตัวดำเนินการ (Operator) ในภาษาไพธอน	118
ตัวดำเนินการทางคณิตศาสตร์ (Arithmetic operators)	118
ตัวดำเนินการทางตรรกศาสตร์ (Logical operators)	119
ตัวดำเนินการระดับบิต (Bitwise operators)	122
ตัวดำเนินการกำหนดค่า (Assignment operators)	126

ตัวดำเนินการเปรียบเทียบ (Comparison operators)	129
ตัวดำเนินการเอกลักษณ์ (Identity operators)	130
ตัวดำเนินการสมาชิก (Membership operators)	132
ลำดับความสำคัญของตัวดำเนินการ	132
การแปลงชนิดข้อมูล	137
Implicit type conversion	137
Explicit type conversion	138

CHAPTER

08 } คำสั่งเงื่อนไข

คำสั่งเงื่อนไข if	144
คำสั่งเงื่อนไข if-else	146
คำสั่งเงื่อนไข if-elif-else	148
คำสั่งเงื่อนไข nested-if	150

CHAPTER

09 } คำสั่งทำซ้ำ

คำสั่งทำซ้ำ while	155
การทำงานของคำสั่งทำซ้ำ while ร่วมกับคำสั่ง break	159
การทำงานของคำสั่งทำซ้ำ while ร่วมกับคำสั่ง continue	159
การทำงานของคำสั่งทำซ้ำ while ร่วมกับคำสั่ง else	161



คำสั่งทำซ้ำ for.....	161
การทำงานของคำสั่งทำซ้ำ for ร่วมกับฟังก์ชัน range()	163
การทำงานของคำสั่งทำซ้ำ for ร่วมกับคำสั่ง break.....	165
การทำงานของคำสั่งทำซ้ำ for ร่วมกับคำสั่ง continue.....	166
การทำงานของคำสั่งทำซ้ำ for ร่วมกับคำสั่ง else	167
การทำงานของคำสั่งทำซ้ำ for ร่วมกับคำสั่ง pass.....	168
การนำคำสั่งทำซ้ำ for มาซ้อนกัน (Nested for).....	168

CHAPTER

10

ฟังก์ชันและโมดูล

ฟังก์ชัน (Function)	171
User-define function	171
การสร้างฟังก์ชัน.....	172
การเรียกฟังก์ชันให้ทำงาน.....	172
การกำหนดพารามิเตอร์ของฟังก์ชัน	174
การกำหนดค่า default ของพารามิเตอร์ในฟังก์ชัน	174
Keyword arguments ส่งข้อมูลไป ฟังก์ชันแบบไม่เรียงลำดับพารามิเตอร์.....	176
การส่งข้อมูลไปยังฟังก์ชันแบบ ไม่จำกัดจำนวนข้อมูล.....	177
การส่งคืนค่าข้อมูลกลับออกไป จากฟังก์ชัน	178
Built-in functions	180

โมดูล (Module).....	181
วิธีสร้างโมดูล	181
วิธีเรียกใช้งานโมดูล.....	182
ใช้คำสั่ง import.....	182
ใช้คำสั่ง from ... import	183
วิธีเปลี่ยนชื่อโมดูล.....	183
Built-in modules.....	185
โมดูล calendar.....	185
โมดูล datetime	189
โมดูล zoneinfo.....	194

CHAPTER

11

การจัดการข้อผิดพลาด ของโปรแกรม (Exception Handling)

Exception คืออะไร	197
จะจัดการอย่างไรกับ Exception ที่เกิดขึ้น?.....	198
จัดการกับ Exception ด้วย try-except.....	198
จัดการกับ Exception ด้วย try-except-else.....	202
จัดการกับ Exception ด้วย try-except-finally.....	203
จัดการกับ Exception ที่เกิดขึ้นด้วย raise	204

CONTENT

CHAPTER

12

การจัดการไฟล์ (File Handling)

ไฟล์คืออะไร	207
การทำงานกับเท็กซ์ไฟล์ (Text File)	207
การเปิดและปิดเท็กซ์ไฟล์	207
เขียนข้อมูลลงในเท็กซ์ไฟล์ ด้วยเมธอด write(), writelines()	209
อ่านข้อมูลในเท็กซ์ไฟล์ด้วยเมธอด read(), readline(), readlines()	210
เปลี่ยนตำแหน่งปัจจุบันของไฟล์ ให้ชี้ไปที่ใหม่ด้วยเมธอด seek()	211
การทำงานกับไบนารีไฟล์ (Binary File)	213
จัดการไฟล์ด้วยคำสั่ง with	215
การจัดการกับไฟล์ประเภท CSV	216
วิธีเขียนข้อมูลลงไฟล์ CSV	216
วิธีอ่านข้อมูลในไฟล์ CSV	219
อ่านข้อมูลและแสดงผลในรูปแบบลิสต์	219
อ่านข้อมูลและแสดงผล ในรูปแบบดิกชันนารี	220
การจัดการกับไฟล์ประเภท Excel	221
การติดตั้งโมดูล Openpyxl	221
สำหรับ Python for Windows	221
สำหรับ PyCharm	222
สำหรับ Jupyter Notebook	224
วิธีเขียนข้อมูลลงไฟล์ Excel	224
วิธีอ่านข้อมูลในไฟล์ Excel	228
การจัดการกับไฟล์และไดเรกทอรี	231

CHAPTER

13

การเขียนโปรแกรม เชิงวัตถุ

ทำความเข้าใจกับคลาสและออบเจ็กต์	234
การสร้างคลาสและออบเจ็กต์	234
เรียนรู้การทำงานของ __init__()	235
การสร้างเมธอดของออบเจ็กต์	237
การห่อหุ้มข้อมูล (Encapsulation)	239
วิธีกำหนดระดับการเข้าถึงงาน private ให้ตัวแปรและเมธอด	241
Accessor Method และ Mutator Method	244
การสืบทอดคุณสมบัติ (Inheritance)	246
คลาสแม่และคลาสลูก คืออะไร?	247
ทำไมต้องสืบทอดคุณสมบัติ?	247
คลาสแม่สามารถถ่ายทอดอะไร ให้แก่คลาสลูกได้บ้าง?	247
วิธีการสร้างคลาสแม่และคลาสลูก	247
การสร้าง __init__() ในคลาสลูก	250
การสืบทอดคุณสมบัติแบบ Multilevel	251
Polymorphism (การพ้องรูป)	252
การทำงานของ Polymorphism ร่วมกับ Inheritance	253
การโอเวอร์โหลดตัวดำเนินการ (Operator Overloading)	255



CHAPTER

14

โฟรอนกับระบบจัดการฐานข้อมูล MariaDB

ฐานข้อมูลคืออะไร	259
DBMS คืออะไร.....	259
รู้จักกับระบบจัดการฐานข้อมูล MariaDB	259
การดาวน์โหลดและติดตั้ง MariaDB	260
การเรียกใช้ phpMyAdmin.....	265
การสร้างฐานข้อมูลด้วย phpMyAdmin.....	267
การติดตั้งโมดูล pymysql	
เพื่อเชื่อมต่อ MariaDB กับโฟรอน	270
การติดตั้งโมดูล pymysql	
บน Python for Windows.....	270
การติดตั้งโมดูล pymysql	
บน PyCharm.....	271
การติดตั้งโมดูล pymysql	
บน Jupyter Notebook.....	273
เขียนโปรแกรมโฟรอนเชื่อมต่อกับฐานข้อมูล MariaDB	277
การเชื่อมต่อกับฐานข้อมูลของ MariaDB	277
การเพิ่มข้อมูลลงในตารางข้อมูล (INSERT).....	277
การเรียกดูข้อมูลลงในตารางข้อมูล (SELECT).....	279
fetchall()	280
fetchone().....	281
fetchmany().....	282
การปรับปรุงข้อมูลในตารางข้อมูล (UPDATE)	285
การลบข้อมูลในตารางข้อมูล (DELETE).....	286

CHAPTER

15

โฟรอนกับระบบจัดการฐานข้อมูล MongoDB

ทำความเข้าใจกับระบบจัดการฐานข้อมูล MongoDB.....	289
การดาวน์โหลดและติดตั้ง MongoDB.....	291
การสั่ง Start MongoDB Server.....	295
การเชื่อมต่อฐานข้อมูลด้วย MongoDB Compass	297
การสร้างฐานข้อมูลด้วย MongoDB Compass	299
การติดตั้งโมดูล pymongo เพื่อเชื่อมต่อ MongoDB กับโฟรอน	305
การติดตั้งโมดูล pymongo บน Python for Windows	305
การติดตั้งโมดูล pymongo บน PyCharm	306
การติดตั้งโมดูล pymongo บน Jupyter Notebook.....	308
เขียนโปรแกรมโฟรอนเชื่อมต่อกับฐานข้อมูล MongoDB	312
การเชื่อมต่อกับฐานข้อมูลของ MongoDB	312
การเรียกดูข้อมูล (SELECT) ใน Collection.....	313
เรียกดูข้อมูลด้วยเมธอด find().....	313
เรียกดูข้อมูลด้วยเมธอด find_one()	315
โอเปอเรเตอร์ที่ใช้ในการกำหนดเงื่อนไขเรียกดูข้อมูล	316
การเพิ่มข้อมูล (INSERT DOCUMENT) ลงใน Collection	318
การปรับปรุงข้อมูล (UPDATE) ใน Collection	322
การลบข้อมูล (DELETE DOCUMENT) ใน Collection	326

2 PART PYTHON WEB APPLICATION

การพัฒนาเว็บแอปพลิเคชันด้วยภาษาไพธอน

CHAPTER

16

พัฒนา Web Application ด้วย Python Django

Django คืออะไร.....	332
รู้จักกับ Django MVT Architecture.....	332
การติดตั้ง Django.....	333
โครงสร้างของ Django Project	334
การสร้าง Django Project	335
สั่งรันเซิร์ฟเวอร์ด้วย runserver	336
การสร้าง Apps ของ Django Project.....	338
การสร้าง View ของ Apps	339
การสร้าง Template ของ Apps.....	343
Django Template Language.....	348
Template Variables.....	348
Template Tag.....	348
Template Filters	349
สร้าง Template ด้วย Django Template Language.....	349
การสร้าง Model ของ Apps	
เชื่อมต่อกับ MariaDB.....	354
ทำความเข้าใจกับ Django ORM	361
การสร้าง Model ของ Apps	
เชื่อมต่อกับ MongoDB	363
เรียนรู้การทำงานของ Django Admin Site	367
Workshop : ระบบจัดการข้อมูลลูกค้า.....	372
การสร้าง link เชื่อมโยง	
หน้าเว็บเพจใน Django	380

CHAPTER

17

Web scraping ดึงข้อมูลจากเว็บไซต์ด้วยไพธอน

Web scraping คืออะไร.....	408
การทำ Web scraping ผิดกฎหมายหรือไม่.....	409
วิเคราะห์ไฟล์ robots.txt	
ก่อนทำ Web scraping.....	410
กฎของไฟล์ robots.txt	411
การกำหนดเนื้อหาในไฟล์ robots.txt.....	412
โปรโตคอล HTTP	
กับการทำ Web scraping	414
HTTP Status Code.....	415
การใช้เครื่องมือ Inspect	
สำรวจเนื้อหาเว็บไซต์	418
ทบทวนพื้นฐาน HTML	
ก่อนเริ่มทำ Web scraping.....	420
คอมเมนต์ใน HTML	421
โครงสร้างของภาษา HTML.....	421
HEAD : ส่วนหัวของเอกสาร HTML	422
BODY : ส่วนเนื้อหาของเอกสาร HTML	422
เรียนรู้ DOM Tree	423
เครื่องมือสำหรับการทำ Web scraping	425
Scrapy	425
BeautifulSoup	425
Selenium.....	426
การติดตั้งโมดูล Requests และ BeautifulSoup.....	427
การติดตั้งโมดูล Requests และ BeautifulSoup บน Python	
for Windows	427
การติดตั้งโมดูล Requests และ BeautifulSoup บน PyCharm.....	428



การติดตั้งโมดูล Requests และ
BeautifulSoup บน Jupyter Notebook..... 431

**การติดตั้งโมดูล Selenium
และ WebDriver..... 432**

การติดตั้งโมดูล Selenium
บน Python for Windows 432

การติดตั้งโมดูล Selenium
บน PyCharm 433

การติดตั้งโมดูล Selenium
บน Jupyter Notebook..... 435

การติดตั้ง Chrome Driver
บน Windows 436

**Workshop 1 : ดึงข้อมูลจำนวนประชากร
(Population) จาก Wikipedia..... 440**

เริ่มต้นใช้งาน BeautifulSoup..... 442

แนวคิดในการดึงข้อมูลจากแท็ก HTML..... 442

การนำข้อมูลจาก Web scraping
มาบันทึกในรูปแบบไฟล์ csv..... 444

**Workshop 2 : ดึงข้อมูลราคาสินค้า
จากเว็บไซต์ 445**

การทำงานของ
Single Page Application 445

แนวคิดการดึงข้อมูลจากผลลัพธ์
การค้นหาบนเว็บไซต์ 446

การใช้งาน Selenium
เพื่อดึงข้อมูลราคาสินค้า..... 449

ตัวอย่างการดึงราคาสินค้า
จากเว็บไซต์ Tops Online 449

ตัวอย่างการดึงราคาสินค้า
จากเว็บไซต์ Big C Online 453

ตัวอย่างการดึงราคาสินค้า
จากเว็บไซต์ Tesco Lotus Online..... 455

CHAPTER

18

รู้จักกับ API และ Microservices

API คืออะไร 458

REST API (RESTful web services)..... 459

การกำหนด URL ของ API..... 459

คุณสมบัติของ REST API..... 460

JSON : รูปแบบข้อมูลที่รับส่งผ่าน API..... 460

API กับ Microservices Architecture..... 462

ข้อดีของการออกแบบแอปพลิเคชัน
เป็น Microservices Architecture 463

HTTP Method กับการทำงาน
ของ REST API..... 464

HTTP Status Code..... 465

มาตรฐาน HTTP Status Code 465

HTTP Status Code ที่ใช้กับ REST API..... 465

API Specification (Swagger)
และ API Documentation..... 467

โครงสร้างไฟล์ API Specification
(Swagger file) 468

openapi 468

info 469

servers 469

paths 469

API Versioning..... 470

หลักการเปลี่ยนเวอร์ชัน API 471

CONTENT

CHAPTER

19

เรียนรู้การสร้าง API ด้วย flask

ทำความเข้าใจกับ flask	473
Virtual Environment และการติดตั้ง virtualenv	473
ทำไมต้องสร้าง Virtual Environment	474
การติดตั้งโมดูล virtualenv	475
การสร้างโปรเจกต์สำหรับพัฒนาเว็บแอปพลิเคชันด้วย flask	475
การติดตั้งโมดูล flask	477
การพัฒนา API ด้วย flask	479
สร้าง Hello World API ด้วย flask	479
FLASK_APP และ FLASK_ENV	480
การรันโปรแกรม flask	481
การรันโปรแกรมแบบ flask run	481
การรันโปรแกรมแบบ app.run()	483
การจัดการกับ Response ของ API	484
การคืนผลลัพธ์ (Response) แบบ JSON	484
การระบุ HTTP status code และ Content-Type	487
การสร้าง API และรับพารามิเตอร์แบบ HTTP GET method	491
การรับพารามิเตอร์แบบ Path Info	491
การรับพารามิเตอร์แบบ Path Info มากกว่า 1 ตัว	492
การรับพารามิเตอร์แบบ Query String	493
การสร้าง API และรับพารามิเตอร์แบบ HTTP POST method	499

CHAPTER

20

สร้าง API ด้วย FastAPI

FastAPI คืออะไร	503
การติดตั้ง FastAPI	503
ASGI และ Uvicorn	504
การติดตั้ง Uvicorn	505
การพัฒนา API ด้วย FastAPI	506
การประกาศ Path และ HTTP Method ของ API	506
การรันโปรแกรม FastAPI บน Uvicorn	507
การระบุ HTTP status code	508
การสร้าง API และรับพารามิเตอร์แบบ HTTP GET method	511
การรับพารามิเตอร์แบบ Path Info	511
การรับพารามิเตอร์แบบ Query String	514
การกำหนด Query String แบบ Optional	515
การสร้าง API และรับพารามิเตอร์แบบ HTTP POST method	517
FastAPI กับ API Documentation (Swagger)	520
/docs	520
/redoc	521
การดาวน์โหลดไฟล์ swagger	524
การดาวน์โหลดไฟล์ swagger ของ /docs	524
การดาวน์โหลดไฟล์ swagger ของ /redoc	525

3

PART

การเขียนโปรแกรม ภาษาไพธอนสำหรับงาน Data Science

CHAPTER

21

ทำความเข้าใจกับ Data Science

Data Science คืออะไร	528
Data Science กับ Python	528
ไลบรารีพื้นฐานสำหรับการทำ Data Science ของ Python	529
Pandas	529
NumPy	530
Matplotlib	530

CHAPTER

22

การทำงานของไลบรารี NumPy

NumPy คืออะไร	532
การติดตั้ง NumPy	532
การสร้าง ndarray	532
การสร้าง ndarray จากข้อมูลชนิดอื่น ของ Python	532
การสร้าง ndarray จาก arange()	534
การสร้าง ndarray จากเลขสุ่ม ด้วยโมดูล Random	536
การเรียกใช้โมดูล Random	536
rand()	537
randint()	539
การสลับตำแหน่งสมาชิกแบบสุ่มใน ndarray (Shuffling & Premutation)	541

โอเปอเรชันกับ ndarray

(Vectorization & Broadcasting) 543

Vectorization 543

Broadcasting 544

กฎการทำโอเปอเรชันของ ndarray 545

 ndarray 2 ตัว มีจำนวนมิติเท่ากัน
 แต่จำนวนสมาชิกไม่เท่ากันในบางมิติ 546

 ndarray 2 ตัว มีจำนวนมิติไม่เท่ากัน
 แต่จำนวนสมาชิกเท่ากันในมิติหนึ่ง
 หรือหลายมิติ 548

 ndarray 2 ตัว มีจำนวนมิติเท่ากัน
 และจำนวนสมาชิกในต่างมิติเท่ากัน 552

 ndarray 2 ตัว ซึ่งตัวใดตัวหนึ่ง
 มีจำนวนสมาชิกเป็น 1 554

 ndarray 2 ตัว ที่ไม่สามารถนำมาทำ
 โอเปอเรชันกันได้ 554

การเข้าถึงสมาชิกใน ndarray

(Array Indexing) 555

 การเข้าถึงสมาชิกใน ndarray
 มากกว่า 1 มิติ 555

Index Slicing 556

CHAPTER

23

โครงสร้างข้อมูลของ Pandas

Pandas คืออะไร 561

รู้จักกับโครงสร้างข้อมูลของ Pandas 561

 Series 561

 DataFrame 562

 Panel 563

เรียนรู้การสร้างข้อมูลแบบ Series 564

 การสร้างข้อมูลแบบ Series จาก List 565

การสร้างข้อมูลแบบ Series โดยกำหนด index.....	566	markedgecolor : กำหนดสีขอบ ของตัวเน้นจุดพิกัด.....	582
การสร้างข้อมูลแบบ Series โดยกำหนด dtype	567	markerfacecolor : กำหนดสีด้านใน ของตัวเน้นจุดพิกัด.....	582
การสร้างข้อมูลแบบ Series จาก Dictionary.....	568	color : การกำหนดสีของกราฟ.....	583
การสร้างข้อมูลแบบ Series จาก Tuple.....	568	title : กำหนดหัวเรื่องกราฟ	583
การสร้างข้อมูลแบบ Series จาก ndarray ของ NumPy	569	xlabel และ ylabel : กำหนดชื่อเลเบล ของแกน x,y.....	583
เรียนรู้การสร้างข้อมูลแบบ DataFrame.....	570	xticks และ yticks : กำหนดค่า ลำดับข้อมูลของแกน x,y	584
การสร้างข้อมูลแบบ DataFrame จาก Series.....	571	figure : ปรับแต่ง container สำหรับการพล็อตกราฟ.....	585
การสร้าง DataFrame จากข้อมูล Series ชุดเดียว.....	571	การปรับขนาดรูปภาพกราฟ.....	586
การสร้าง DataFrame จากข้อมูล Series หลายชุด	572	legend : กำหนดคำอธิบายสัญลักษณ์ ในกราฟ	590
การสร้างข้อมูลแบบ DataFrame จาก List	573	การสร้าง Bar charts.....	593
การสร้างข้อมูลแบบ DataFrame จาก List ของ Dictionary	574	การสร้าง Horizontal bar charts.....	595
การสร้างข้อมูลแบบ DataFrame จาก ndarray ของ NumPy	576	การสร้าง line graphs.....	596
		การสร้าง Area plots.....	597
		การสร้าง Pie charts	598

CHAPTER

24

การทำ Data Visualization ด้วย Matplotlib

รู้จักกับไลบรารี Matplotlib..... 578

เรียนรู้การทำงานของฟังก์ชัน plot()
ใน Matplotlib..... 580

linestyle : กำหนดรูปแบบเส้นกราฟ.....580

linewidth : กำหนดความหนา
ของเส้นกราฟ.....580

marker : กำหนดตัวเน้นจุดพิกัด.....581

markersize : กำหนดขนาดของ
ตัวเน้นจุดพิกัด
 581 |

APPENDIX

สรุปแท็กของภาษา HTML

แท็กสำหรับจัดการแสดงผล.....602

แท็กของฟอร์ม.....603

แท็กของรูปภาพ.....604

แท็กของตาราง.....605

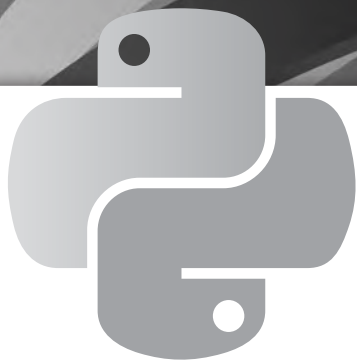
แท็กของการจัดกลุ่มข้อความ.....607

แท็กของการแสดงรายการ.....607



PART 1 Basic Python

การเขียนโปรแกรม
ภาษาไพธอนขั้นพื้นฐาน



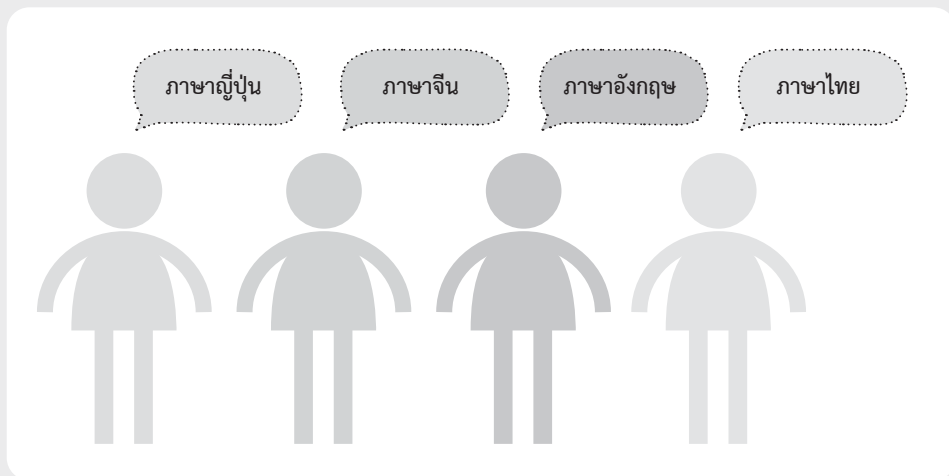
CHAPTER

01

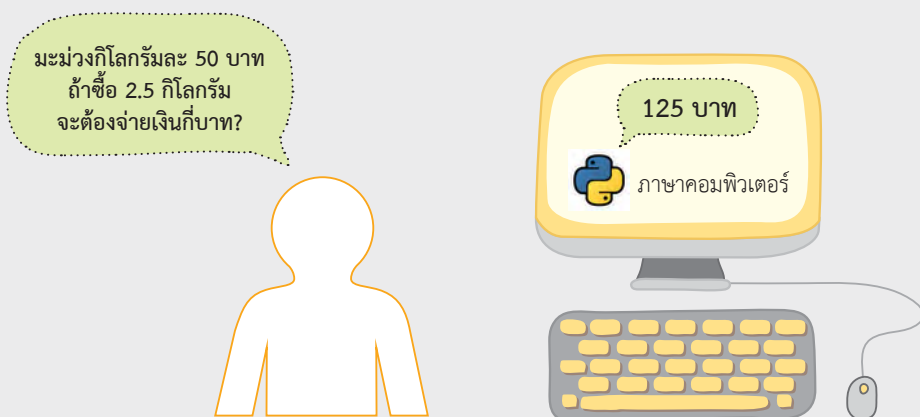
ทำความรู้จักกับภาษา คอมพิวเตอร์ไพธอน

ภาษาคอมพิวเตอร์คืออะไร?

มนุษย์ทุกคนบนโลกสื่อสารเข้าใจกันด้วย “ภาษามนุษย์”

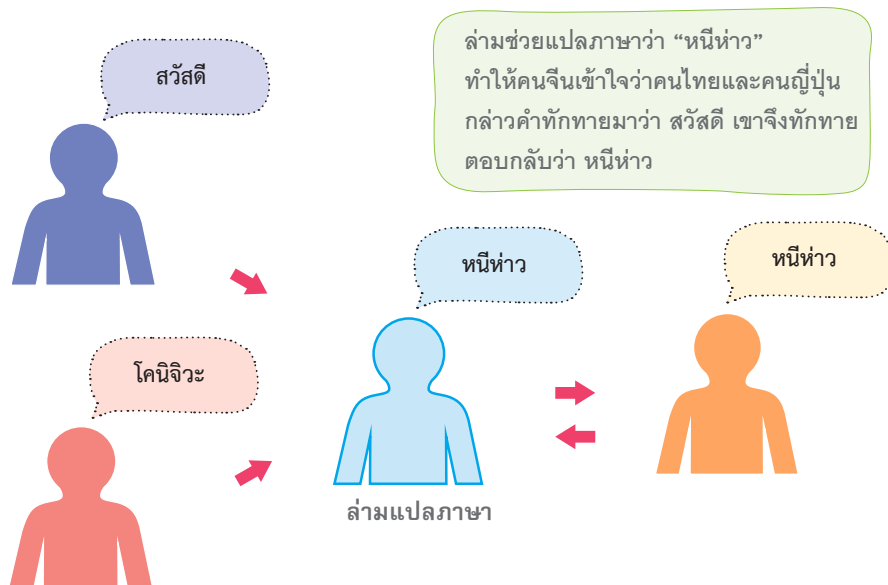


แต่คอมพิวเตอร์ไม่เข้าใจ “ภาษามนุษย์” การที่มนุษย์จะสื่อสารกับคอมพิวเตอร์ให้เข้าใจต้องใช้ “ภาษาคอมพิวเตอร์” เท่านั้น โดยไพธอน (Python) ก็จัดเป็นภาษาคอมพิวเตอร์ประเภทหนึ่ง



ตัวแปลภาษาคืออะไร

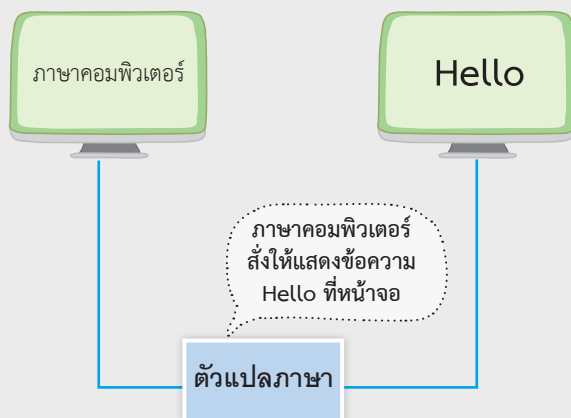
ภาษามนุษย์แบ่งออกเป็นหลากหลายภาษา เช่น ภาษาไทย ภาษาจีน ภาษาฝรั่งเศส ภาษาญี่ปุ่น เป็นต้น ในบางครั้งหากไม่เข้าใจในภาษานั้นๆ ก็ต้องอาศัย “ล่าม” เป็นผู้ช่วยแปลภาษา เพื่อแปลผลให้เข้าใจกัน



ภาษาคอมพิวเตอร์ก็เช่นเดียวกับภาษามนุษย์ คือมีหลายภาษาที่สามารถใช้สื่อสารกับคอมพิวเตอร์ได้ บางภาษาสื่อสารแล้วคอมพิวเตอร์ก็เข้าใจได้เลย แต่บางภาษาที่ไม่ใช่ภาษาที่คอมพิวเตอร์เข้าใจ

ดังนั้น ภาษาคอมพิวเตอร์บางตัวจึงต้องใช้ “ตัวแปลภาษา” เป็นล่ามในการทำหน้าที่แปลจากภาษาคอมพิวเตอร์ไปเป็นภาษาที่เครื่องคอมพิวเตอร์เข้าใจ ว่าต้องการให้คอมพิวเตอร์ต้องทำงานอะไร

ภาษาคอมพิวเตอร์ไพธอนก็ต้องใช้ตัวแปลภาษาเช่นกัน



ประเภทของภาษาคอมพิวเตอร์

ภาษาคอมพิวเตอร์ แบ่งออกเป็น 2 ประเภท คือ ภาษาระดับต่ำ และภาษาระดับสูง ไพธอนจัดว่าเป็นภาษาระดับสูง

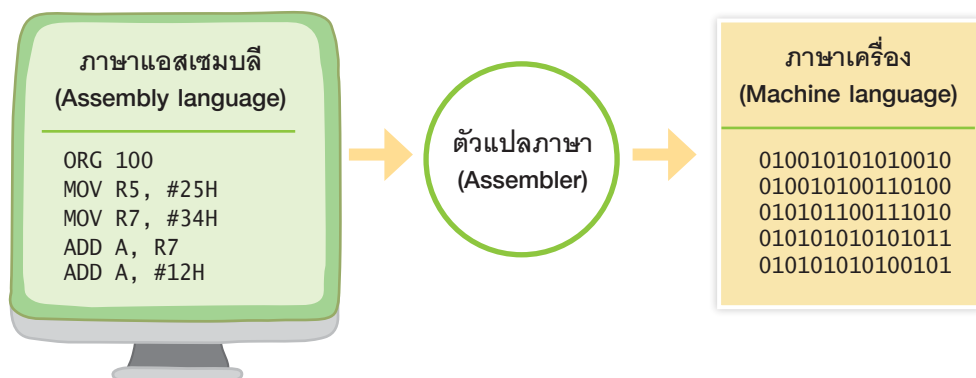
ภาษาระดับต่ำ (Low level language)

แบ่งออกเป็น 2 ภาษา คือ

- **ภาษาเครื่อง (Machine language)**
เป็นภาษาที่ต้องเขียนคำสั่งโปรแกรมด้วยเลขฐานสองที่มีแค่ 0 กับ 1 เท่านั้น การเขียนโปรแกรมด้วยภาษาเครื่องจึงมีความยาก เนื่องจากผู้เขียนโปรแกรมต้องศึกษาสถาปัตยกรรมและฮาร์ดแวร์ของเครื่องคอมพิวเตอร์ให้ละเอียดก่อน จึงจะสามารถเขียนโปรแกรมได้

```
0000030 0000 0000 0000 0010 0000 0000 0000 0204
0000040 0004 8384 0084 c7c8 00c8 4748 0048 e8e9
0000050 00e9 6a69 0069 a8a9 00a9 2828 0028 fdfc
0000060 00fc 1819 0019 9898 0098 d9d8 00d8 5857
0000070 0057 7b7a 007a bab9 00b9 3a3c 003c 8888
0000080 8888 8888 8888 8888 288e be88 8888 8888
0000090 3b83 5788 8888 8888 7667 778e 8828 8888
00000a0 d61f 7abd 8818 8888 467c 585f 8814 8188
00000b0 8b06 e8f7 88aa 8388 8b3b 88f3 88bd e988
00000c0 8a18 880c e841 c988 b328 6871 688e 958b
00000d0 a948 5862 5884 7e81 3788 1ab4 5a84 3eec
00000e0 3d86 dc88 5cbb 8888 8888 8888 8888 8888
00000f0 8888 8888 8888 8888 8888 8888 8888 0000
0000100 0000 0000 0000 0000 0000 0000 0000 0000
*
0000130 0000 0000 0000 0000 0000 0000 0000
000013e
```

- **ภาษาแอสเซมบลี (Assembly language)** เป็นภาษาที่เขียนคำสั่งโปรแกรม (source code) โดยใช้ตัวอักษรแทนเลขฐานสอง เมื่อเขียนคำสั่งเสร็จต้องใช้ตัวแปลภาษาที่เรียกว่า “แอสเซมเบลเลอร์” แปลเป็นภาษาเครื่องอีกทีหนึ่ง การเขียนโปรแกรมด้วยภาษาแอสเซมบลียังนับว่ามีความยุ่งยาก เนื่องจากผู้เขียนต้องเข้าใจโครงสร้างของซีพียูหรือไม่ไครโปรเซสเซอร์ เพราะคำสั่งต่าง ๆ จะอิงกับรุ่นของไมโครโปรเซสเซอร์



ภาษาระดับสูง (High level language)

เนื่องจากภาษาระดับต่ำมีความยากในการเขียนคำสั่งโปรแกรม ภาษาระดับสูงจึงถูกพัฒนาขึ้น โดยภาษาระดับสูงจะมีรูปแบบคำสั่งคล้ายกับประโยคภาษาอังกฤษที่มนุษย์เข้าใจได้ การเขียนคำสั่งโปรแกรมจึงทำได้ง่าย ภาษาระดับสูงมีมากมายหลายภาษา เช่น Python, Ruby, C, JAVA เป็นต้น

เมื่อเขียนคำสั่งโปรแกรมด้วยภาษาระดับสูงเสร็จแล้ว จะต้องใช้ตัวแปลภาษาในการแปลเป็น ภาษาเครื่องอีกทีหนึ่ง โดยตัวแปลภาษาของภาษาระดับสูง แบ่งออกเป็น 2 ประเภท คือ Interpreter และ Compiler โดยไพธอนจะใช้ตัวแปลภาษาแบบ Interpreter

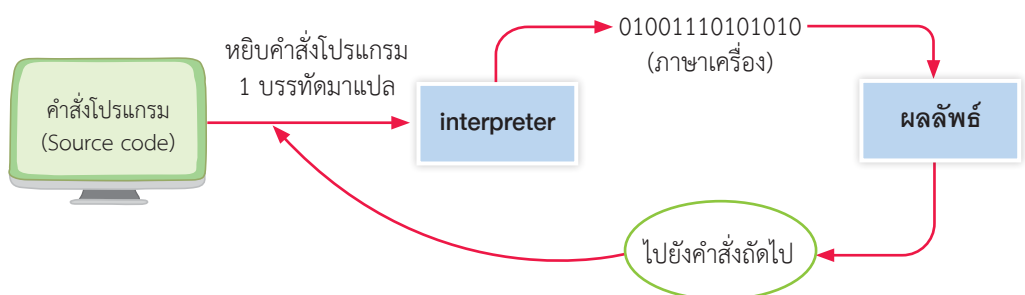
ความแตกต่างระหว่าง Interpreter และ Compiler

Interpreter

Interpreter จะแปลคำสั่งโปรแกรมทีละบรรทัดไปเป็นภาษาเครื่อง แล้วทำงานทันที จึงใช้เวลา ไม่นานตั้งแต่สั่งงานจนได้ผลลัพธ์ แต่เมื่อแปลเสร็จแล้ว หากมีการเรียกใช้งานโปรแกรมเดิมซ้ำจะ ทำได้ช้ากว่าโปรแกรมที่แปลด้วย Compiler เนื่องจากต้องแปลเป็นภาษาเครื่องใหม่ทุกครั้ง ซึ่งถ้า โปรแกรมไม่มีการแก้ไข ก็เสียเวลาแปลโดยไม่จำเป็น

แต่มีข้อดีคือหาจุดผิดพลาดของโปรแกรมได้ง่าย โดยถ้าคำสั่งไม่ถูกต้องก็จะหยุดการแปลและ แจ้งจุดผิดพลาดให้ผู้เขียนโปรแกรมทราบเพื่อดำเนินการแก้ไขทันที

ภาษาคอมพิวเตอร์ที่ใช้การแปลแบบ Interpreter ได้แก่ Python, Ruby เป็นต้น

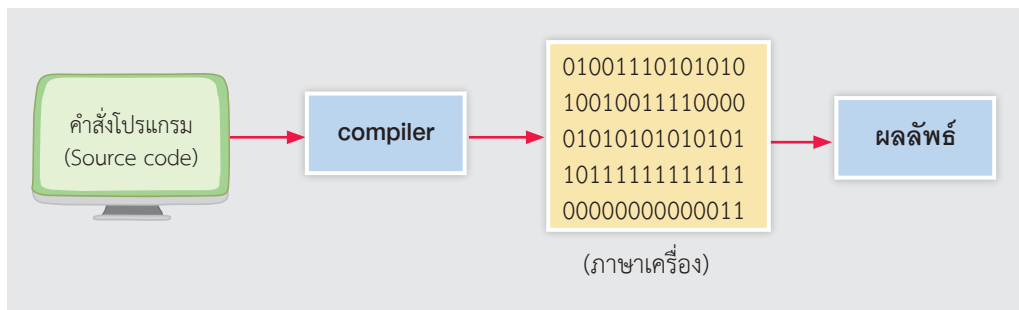


Compiler

Compiler จะแปลทุกคำสั่งของโปรแกรมเป็นภาษาเครื่องในครั้งเดียวและเก็บไฟล์ภาษาเครื่องที่ได้เอาไว้ จากนั้นจึงทำงานตามคำสั่งในโปรแกรมนั้น ดังนั้นจึงใช้เวลาในการแปลนาน แต่เมื่อแปลเสร็จแล้ว การเรียกโปรแกรมให้แสดงผลหรือทำงานจะทำได้เร็วกว่า Interpreter และเมื่อต้องการเรียกใช้โปรแกรมซ้ำโดยโปรแกรมไม่มีการแก้ไขใดๆ ก็ไม่จำเป็นต้องแปลใหม่อีก เพียงแค่เรียกจากไฟล์โปรแกรมภาษาเครื่องมาใช้ได้เลย

แต่มีข้อเสียคือการหาจุดผิดพลาดของโปรแกรมจะยากกว่า Interpreter เพราะทำการแปลผลทีเดียวทั้งโปรแกรม

ภาษาคอมพิวเตอร์ที่ใช้การแปลผลแบบ Compiler ได้แก่ C, C++ เป็นต้น



ความเป็นมาของภาษาไพธอน

- ค.ศ. 1991 ▶ **Python 1.0** Guido van Rossum เป็นผู้สร้างภาษา Python ขึ้น เขาเป็นแฟนคลับของคณะตลกจากอังกฤษที่ชื่อ Monty Python จึงตั้งชื่อภาษาโปรแกรมนี้นว่า “Python”
- ค.ศ. 2000 ▶ **Python 2.0** เพิ่มความสามารถต่าง ๆ ของภาษาโปรแกรมสมัยใหม่
- ค.ศ. 2008 ▶ **Python 3.0** ความสามารถหลัก ๆ ของ Python 3.0 ได้ถูกนำไปใส่ไว้ใน Python 2.6.x และ Python 2.7.x ด้วย เพื่อให้ผู้พัฒนาโปรแกรมนที่ยังคงใช้ Python 2.0 สามารถพัฒนาโปรแกรมโดยใช้เวอร์ชันเดิมได้
- ค.ศ. 2020 ▶ เลิกเวอร์ชันก่อน Python 3.6 กลุ่มผู้พัฒนาตัดสินใจยุติการแก้ไขข้อผิดพลาดของเวอร์ชันเก่าคือ เวอร์ชัน 2.x ทั้งหมด และ 3.x ที่ต่ำกว่า 3.6 เพื่อเน้นไปที่การพัฒนาเวอร์ชันใหม่ๆ ในอนาคต

โปรแกรมภาษาไพธอนมีหน้าตาอย่างไร

โปรแกรมภาษาไพธอนมีหน้าตาคล้ายกับข้อความภาษาอังกฤษเลยล่ะ โปรแกรมอ่านง่าย สั้น กระชับ ไม่ซับซ้อน มาดูตัวอย่างโปรแกรมไพธอนกัน

ตัวอย่างที่ 1.1

โปรแกรมแสดงข้อความ Good Morning

```
print("Good Morning")
```

ตัวอย่างที่ 1.2

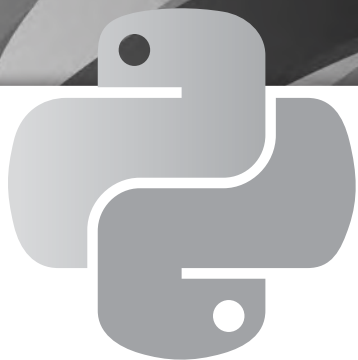
โปรแกรมหาค่าผลบวก/ลบ/คูณ/หาร
ของตัวเลข 2 จำนวน

```
a = 20
b = 10
print(a+b)
print(a-b)
print(a*b)
print(a/b)
```

ทำไมควรเลือกเขียนโปรแกรมด้วยภาษาไพธอน

ไพธอนเป็นภาษาคอมพิวเตอร์ภาษาหนึ่งที่คนนิยมเลือกนำมาเขียนโปรแกรม ด้วยเหตุผลต่าง ๆ ดังนี้

- 1 ทำงานได้หลายระบบปฏิบัติการ**
โปรแกรมภาษาไพธอนสามารถทำงานบนระบบปฏิบัติการต่าง ๆ ได้ เช่น Windows, Mac OS X, Linux, Unix เป็นต้น ไม่จำเป็นต้องเขียนโปรแกรมไพธอนบน Windows แล้วจะต้องใช้งานบน Windows เท่านั้น
- 2 Open-source**
ไพธอนเป็น Open source ที่ใครก็ได้สามารถดาวน์โหลดไปใช้งานได้ฟรี หรือแก้ไขเพิ่มเติมได้โดยไม่มีค่าใช้จ่าย
- 3 เข้าใจง่าย**
โครงสร้างของโปรแกรมภาษาไพธอนไม่ซับซ้อน คำสั่งเข้าใจง่าย เหมาะกับผู้ที่ต้องการเริ่มเขียนโปรแกรม
- 4 Object Oriented**
ไพธอนสนับสนุนการพัฒนาโปรแกรมเชิงวัตถุ (Object Oriented Programming)
- 5 Library**
ไพธอนมีไลบรารีจำนวนมากให้เรียกใช้งานได้ เช่น ไลบรารีเกี่ยวกับการทำงานด้านคณิตศาสตร์ ด้านกราฟฟิก ปัญญาประดิษฐ์ เป็นต้น
- 6 ใช้ได้หลาย Database**
ไพธอนมีฟังก์ชันที่สนับสนุนการทำงานกับฐานข้อมูลประเภทต่าง ๆ เช่น MySQL, Oracle, Informix เป็นต้น



CHAPTER

02

ปูพื้นฐานความรู้ ในการเขียนโปรแกรม

โปรแกรม คืออะไร?

โปรแกรม คือ ชุดของคำสั่งที่กำหนดขึ้นมาเพื่อสั่งให้คอมพิวเตอร์ทำงานใดงานหนึ่ง เป็นการนำคำสั่งต่าง ๆ มาเขียนเรียงต่อกันเป็นขั้นตอนจนเกิดเป็นชุดคำสั่ง หรือที่เรียกว่าโปรแกรมนั่นเอง โดยชุดคำสั่งนี้ต้องถูกสร้างขึ้นจากภาษาคอมพิวเตอร์ ซึ่งในปัจจุบันภาษาไพธอน (Python) เป็นภาษาที่ได้รับความนิยมมากที่สุดภาษาหนึ่ง

อยากเขียนโปรแกรม ... เริ่มอย่างไรดี?

- 1** คิดก่อนว่าโจทย์หรือปัญหาคืออะไร อยากให้คอมพิวเตอร์ช่วยทำงานอะไรให้เรา



อยากให้คอมพิวเตอร์
ถามชื่อเพื่อนและ
กล่าวคำทักทาย

- 2** เขียนอธิบายออกมาให้ชัดเจนเป็นขั้นตอนว่าทำอะไร คอมพิวเตอร์ถึงจะแก้โจทย์ปัญหานี้ได้ เพื่อให้ได้งานใดงานหนึ่งออกมา ขั้นตอนนี้ เรียกว่า “การเขียนอัลกอริทึม”

- อัลกอริทึม คือ กระบวนการแก้ปัญหาที่อธิบายออกมาเป็นขั้นตอนตามลำดับ
- เมื่อได้อัลกอริทึมแล้วมักจะนำไปเขียนเป็นชุดโค้ด (Pseudocode) หรือ ผังงาน (Flowchart) เพื่อให้อ่านเข้าใจง่ายขึ้น รายละเอียดอธิบายไว้ในหัวข้อถัดไป



1. ถามชื่อด้วยข้อความ
“คุณชื่ออะไร”
2. กล่าวคำทักทายด้วยข้อความ
“สวัสดีค่ะ คุณ ...”

- 3** ภาษาคอมพิวเตอร์มีหลายประเภท ต้องเลือกว่าจะใช้ภาษาไหนเขียนโปรแกรม



อยากเขียนโปรแกรม
ด้วยภาษาไพธอนค่ะ

4 ศึกษาคำสั่งต่างๆ ในภาษาคอมพิวเตอร์ ที่เลือก ว่ามีคำสั่งอะไรให้ใช้งานบ้าง และแต่ละคำสั่งทำงานอย่างไร รวมถึงหลักการเขียนโปรแกรมของภาษาคอมพิวเตอร์นั้นว่าเป็นอย่างไ

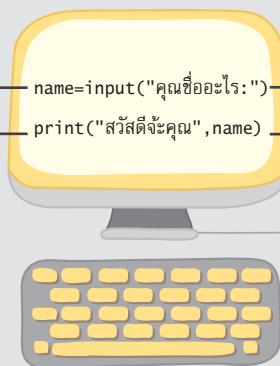
คำสั่ง input ในไพธอน ใช้รับข้อมูลเข้ามา ในโปรแกรม

คำสั่ง print ในไพธอน ใช้แสดงข้อความออก ทางหน้าจอคอมพิวเตอร์



5 เริ่มเขียนโปรแกรมกันเลย หยิบคำสั่งมาเขียนเรียงเป็น ลำดับขั้นตอนตามอัลกอริทึม ที่กำหนดไว้

1 คำสั่ง
1 คำสั่ง



2 คำสั่งนี้รวมกัน ก็จะเป็นชุดคำสั่ง (โปรแกรม) นั่นเอง

ก่อนเริ่มเขียนโปรแกรมต้องติดตั้งเครื่องมือที่จะใช้เขียนโปรแกรมลงบนเครื่องคอมพิวเตอร์ก่อน (วิธีการติดตั้งเครื่องมือเขียนโปรแกรมภาษาไพธอนจะอธิบายในบทที่ 3)

6 เขียนโปรแกรมเสร็จแล้ว ต้องไม่ลืม “ทดสอบ” ว่าโปรแกรมทำงานถูกต้องหรือไม่

ถ้าโปรแกรมทำงานไม่ถูกต้อง ต้องกลับไปแก้ไขโปรแกรมและทดสอบซ้ำจนกว่าโปรแกรมจะทำงานได้ถูกต้อง



เรียนรู้การเขียนชุดโค๊ด (Pseudocode)

ชุดโค๊ดจะช่วยให้ผู้พัฒนาโปรแกรมสามารถลำดับขั้นตอนต่างๆ ในการเขียนเป็นโปรแกรมได้ง่ายขึ้น

- ชุดโค๊ด (เรียกอีกอย่างว่า รหัสเทียม) เป็นคำสั่งที่เขียนอยู่ในรูปแบบประโยคภาษาอังกฤษหรือภาษาไทยที่เข้าใจง่าย
- การเขียนชุดโค๊ดไม่มีหลักการตายตัว เขียนอย่างไรก็ได้ เพื่อสื่อให้ผู้อ่านเข้าใจง่าย
- ชุดโค๊ด ไม่ใช่ภาษาคอมพิวเตอร์ แต่เป็นแค่ตัวแทนอธิบายอัลกอริทึมเท่านั้น
- ชุดโค๊ดมีจุดเริ่มต้น จุดสิ้นสุด และลำดับขั้นตอนที่ชัดเจน โดย 1 บรรทัด คือ 1 คำสั่ง และควรใช้ย่อหน้าจัดระเบียบให้อ่านง่าย



ตัวอย่างที่ 2.1

แสดงการเขียนชุดโค๊ดหาผลบวกของตัวเลข 2 จำนวน

อัลกอริทึม

1. รับค่าตัวเลข 2 จำนวนเข้ามา
2. นำเลข 2 จำนวนนั้นมาบวกกัน
3. แสดงค่าผลบวกของตัวเลข

ชุดโค๊ดภาษาไทย

เริ่มต้น

รับค่า ตัวเลข1,ตัวเลข2

ผลลัพธ์ = ตัวเลข1+ตัวเลข2

แสดงผล ผลลัพธ์

จบ

ชุดโค๊ดภาษาอังกฤษ

BEGIN

INPUT NUM1,NUM2

SUM = NUM1+NUM2

DISPLAY SUM

END



ตัวอย่างที่ 2.2

แสดงการเขียนชุดโค๊ดหาค่าพื้นที่วงกลม

อัลกอริทึม

1. กำหนดค่า $\pi = 3.14$
2. รับค่ารัศมีวงกลม
3. คำนวณพื้นที่วงกลมจากสูตร พื้นที่วงกลม = πr^2
4. แสดงค่าพื้นที่วงกลม

ชุดโค้ดภาษาไทย	ชุดโค้ดภาษาอังกฤษ
<pre> เริ่มต้น กำหนด ไพน์ เป็น 3.14 รับค่า รัศมีวงกลม พื้นที่วงกลม=ไพน์*(รัศมีวงกลม)² แสดงผล พื้นที่วงกลม จบ </pre>	<pre> BEGIN SET PI TO 3.14 INPUT RADIUS AREA = PI*(RADIUS)² DISPLAY AREA END </pre>



ตัวอย่างที่ 2.3

แสดงการเขียนชุดโค้ดคำนวณเกรดจากคะแนน

อัลกอริทึม
1. รับค่าคะแนน 2. ตรวจสอบคะแนน ดังนี้ - คะแนนมากกว่าหรือเท่ากับ 80 เกรด A - คะแนนอยู่ระหว่าง 70-79 เกรด B - คะแนนอยู่ระหว่าง 60-69 เกรด C - คะแนนอยู่ระหว่าง 50-59 เกรด D - คะแนนน้อยกว่า 50 เกรด F 3. แสดงค่าคะแนนและเกรดที่ได้รับ

ชุดโค้ดภาษาไทย	ชุดโค้ดภาษาอังกฤษ
<pre> เริ่มต้น รับค่า คะแนน ถ้า คะแนน ≥ 80 แล้ว เกรด = "A" แต่ถ้า คะแนน < 80 และ คะแนน ≥ 70 แล้ว เกรด = "B" แต่ถ้า คะแนน < 70 และ คะแนน ≥ 60 แล้ว เกรด = "C" แต่ถ้า คะแนน < 60 และ คะแนน ≥ 50 แล้ว เกรด = "D" </pre>	<pre> BEGIN INPUT SCORE IF SCORE ≥ 80 THEN GRADE = "A" ELSE IF SCORE < 80 AND SCORE ≥ 70 THEN GRADE = "B" ELSE IF SCORE < 70 AND SCORE ≥ 60 THEN GRADE = "C" ELSE IF SCORE < 60 AND SCORE ≥ 50 THEN GRADE = "D" END IF </pre>

ชุดโค้ดภาษาไทย	ชุดโค้ดภาษาอังกฤษ
มีฉนั้น เกรด = "F" จบการตรวจสอบคะแนน แสดงผล คะแนน,เกรด จบ	ELSE GRADE = "F" END IF DISPLAY SCORE, GRADE END

 ตัวอย่างที่ 2.4 แสดงการเขียนชุดโค้ดหาค่าเฉลี่ยของตัวเลข 3 ตัว

อัลกอริทึม
1. รับค่าตัวเลขเข้ามาทีละตัว รวม 3 ตัว 2. นำค่าตัวเลขทั้งหมดมาบวกกันแล้วหารด้วย 3 จะได้ค่าเฉลี่ยออกมา 3. แสดงผลค่าเฉลี่ยของตัวเลข

ชุดโค้ดภาษาไทย	ชุดโค้ดภาษาอังกฤษ
เริ่มต้น กำหนด ผลรวมทั้งหมด เป็น 0 กำหนด จำนวนครั้ง เป็น 0 ทำซ้ำ รับค่า ตัวเลข ผลรวมทั้งหมด = ผลรวมทั้งหมด (ก่อนหน้านี้) + ตัวเลข จำนวนครั้ง = จำนวนครั้ง + 1 ทำซ้ำจนกระทั่ง จำนวนครั้ง = 3 ค่าเฉลี่ย = ผลรวมทั้งหมด / 3 แสดงผล ค่าเฉลี่ย จบ	BEGIN SET TOTAL_NUM TO 0 SET TIMES TO 0 REPEAT INPUT NUMBER TOTAL_NUM = TOTAL_NUM + NUMBER TIMES = TIMES + 1 UNTIL TIMES = 3 AVG_NUM = TOTAL_NUM/3 DISPLAY AVG_NUM END

จำนวนครั้ง = จำนวนครั้ง + 1

หรือ

TIMES = TIMES + 1

หมายถึง ให้บวกเพิ่ม “จำนวนครั้ง” ขึ้น 1 ค่า
ถ้าจำนวนครั้ง = 3 แปลว่า รับค่าตัวเลขครบ 3 ค่าแล้ว

TOTAL_NUM
0

SET TOTAL_NUM TO 0
SET TIMES TO 0

TIMES
0

REPEAT
INPUT NUMBER
TOTAL_NUM = TOTAL_NUM + NUMBER
TIMES = TIMES + 1
UNTIL TIMES = 3

ครั้งที่ 1

TOTAL_NUM = 0 + 20 = 20 , TIME = 0 + 1 = 1

ครั้งที่ 2

TOTAL_NUM = 20 + 30 = 50 , TIME = 1 + 1 = 2

ครั้งที่ 3

TOTAL_NUM = 50 + 40 = 90 , TIME = 2 + 1 = 3

TIMES = 3 สิ้นสุดการทำซ้ำ

ตัวอย่างที่ 2.5 แสดงการเขียนชุดโค้ดตรวจสอบการล็อกอินเข้าสู่ระบบ

อัลกอริทึม

1. รับค่าชื่อผู้ใช้งานและรหัสผ่าน
2. ตรวจสอบว่าชื่อผู้ใช้งานและรหัสผ่านถูกต้องหรือไม่
 - ถ้าถูกต้อง แสดงข้อความ “ยินดีต้อนรับสู่ระบบ”
 - ถ้าไม่ถูกต้อง ให้รับค่าชื่อผู้ใช้งานและรหัสผ่านมาใหม่

ชุดโค้ดภาษาไทย

เริ่มต้น

กำหนด สถานะการล็อกอิน = “ไม่ถูกต้อง”

ขณะที่ สถานะการล็อกอิน = “ไม่ถูกต้อง”

รับค่า ชื่อผู้ใช้งาน, รหัสผ่าน

ถ้า (ชื่อผู้ใช้งาน = “admin” และ รหัสผ่าน = “admin”) แล้ว

สถานะการล็อกอิน = “ถูกต้อง”

แสดงข้อความ “ยินดีต้อนรับสู่ระบบ”

จบการตรวจสอบชื่อผู้ใช้งานและรหัสผ่าน

จบ

ชุดโค้ดภาษาอังกฤษ

BEGIN

SET LOGIN_FLAG = “INCORRECT”

WHILE LOGIN_FLAG = “INCORRECT”

INPUT USERNAME, PASSWORD

IF (USERNAME = “admin” AND PASSWORD = “admin”) THEN

LOGIN_FLAG = “CORRECT”

DISPLAY “WELCOME TO SYSTEM”

END IF

END

ถ้าสถานะการล็อกอินมีค่าเป็น “INCORRECT” (ไม่ถูกต้อง)
แล้วกลุ่มคำสั่งนี้จะถูกทำงานไปเรื่อยๆ

เรียนรู้การเขียนผังงาน (Flowchart)

- ผังงาน เรียกอีกอย่างว่า โฟลว์ชาร์ต (Flowchart) เป็นตัวแทนอธิบายอัลกอริทึมที่ชัดเจนเพียงกับชุดใดก็ได้ แตกต่างกับชุดใดก็ได้ตรงที่ผังงานใช้สัญลักษณ์รูปภาพเป็นตัวสื่อความหมาย
- สัญลักษณ์รูปภาพแต่ละรูปในผังงานมีความหมายเฉพาะตัว
- ผังงาน จะใช้ลูกศรกำหนดทิศทางการทำงานในแต่ละขั้นตอน

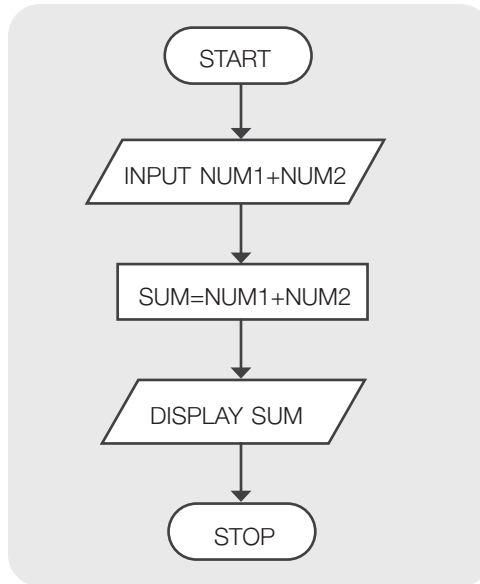
สัญลักษณ์รูปภาพของโฟลว์ชาร์ต

	จุดเริ่มต้น (START) หรือ จุดสิ้นสุด (STOP) ของผังงาน
	รับข้อมูลเข้า (INPUT) หรือ แสดงผลข้อมูลออก (OUTPUT)
	การประมวลผล หรือ กระบวนการปฏิบัติงาน (PROCESS) ใดๆ
	การตัดสินใจ (DECISION) เพื่อเลือกปฏิบัติหรือการเปรียบเทียบ (COMPARE)
	ทิศทาง (FLOW) การดำเนินงาน
	จุดเชื่อมต่อ (CONNECTION) สำหรับเชื่อมต่อผังงานในหน้ากระดาษเดียวกัน
	จุดเชื่อมต่อ (CONNECTION) สำหรับเชื่อมต่อผังงานที่อยู่คนละหน้ากระดาษ
	การทำงานย่อย (SUBPROGRAM) สำหรับเรียกการทำงานย่อยต่างๆ ให้ทำงาน



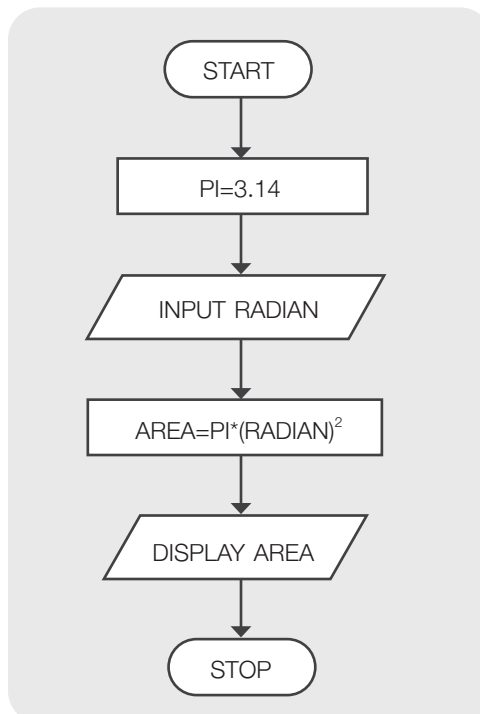
ตัวอย่างที่ 2.6

แสดงการเขียนโฟลว์ชาร์ตหาผลบวกของตัวเลข 2 จำนวน



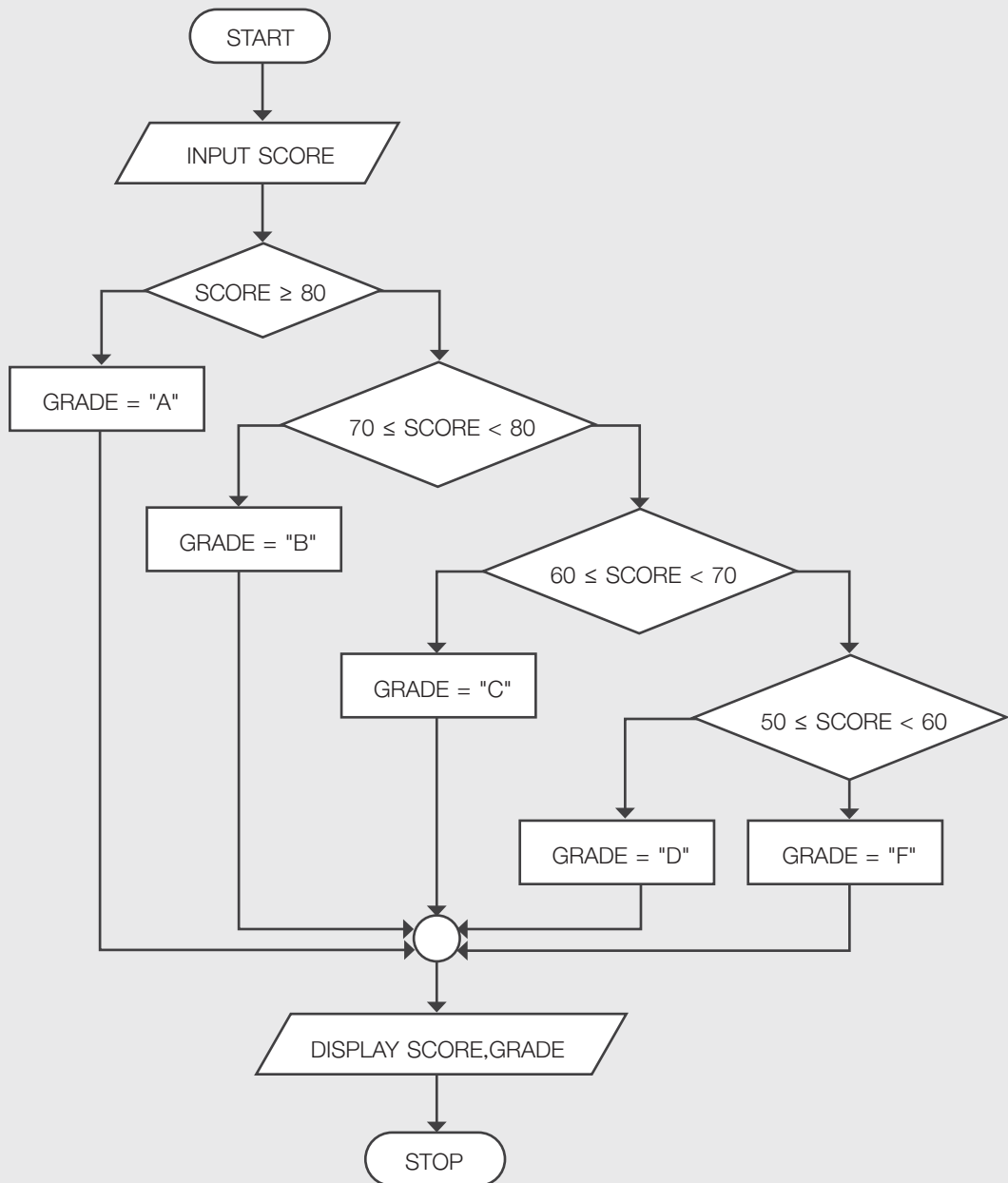
ตัวอย่างที่ 2.7

แสดงการเขียนโฟลว์ชาร์ตหาพื้นที่วงกลม





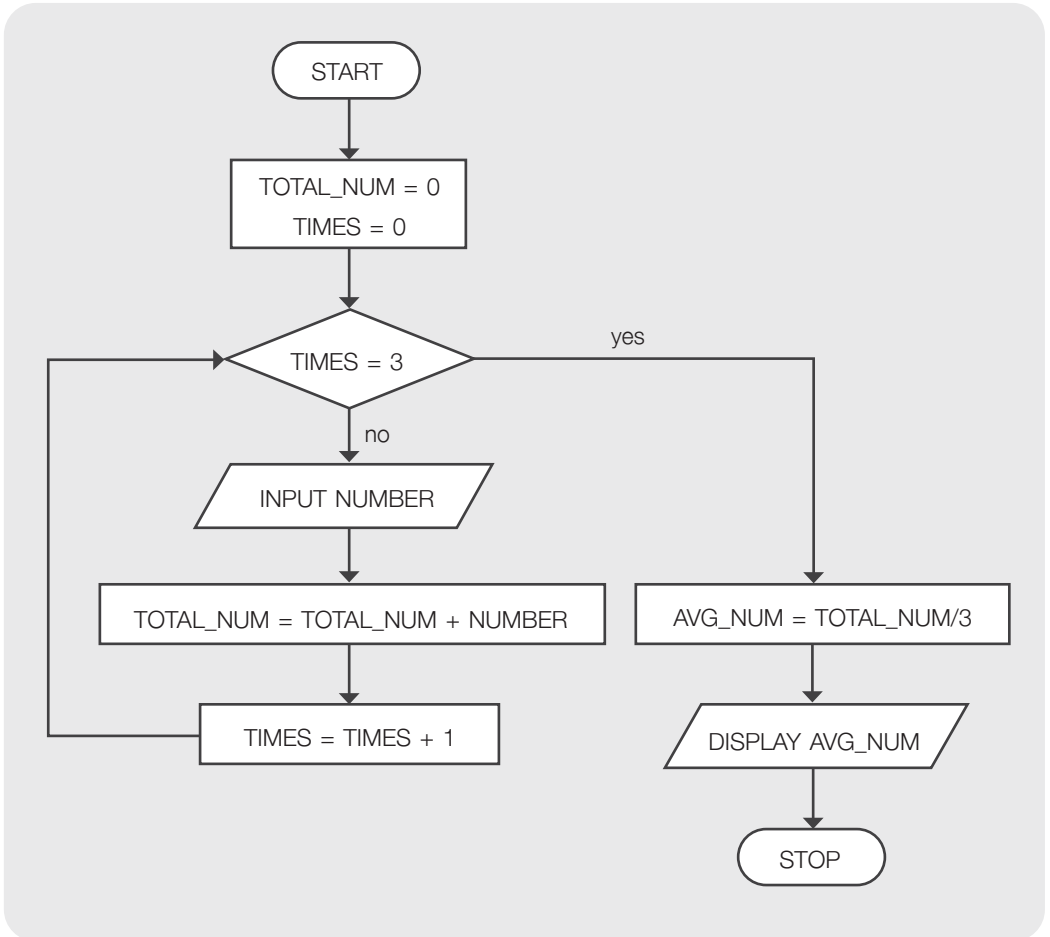
ตัวอย่างที่ 2.8 แสดงการเขียนโปรแกรมคำนวณเกรดจากคะแนน





ตัวอย่างที่ 2.9

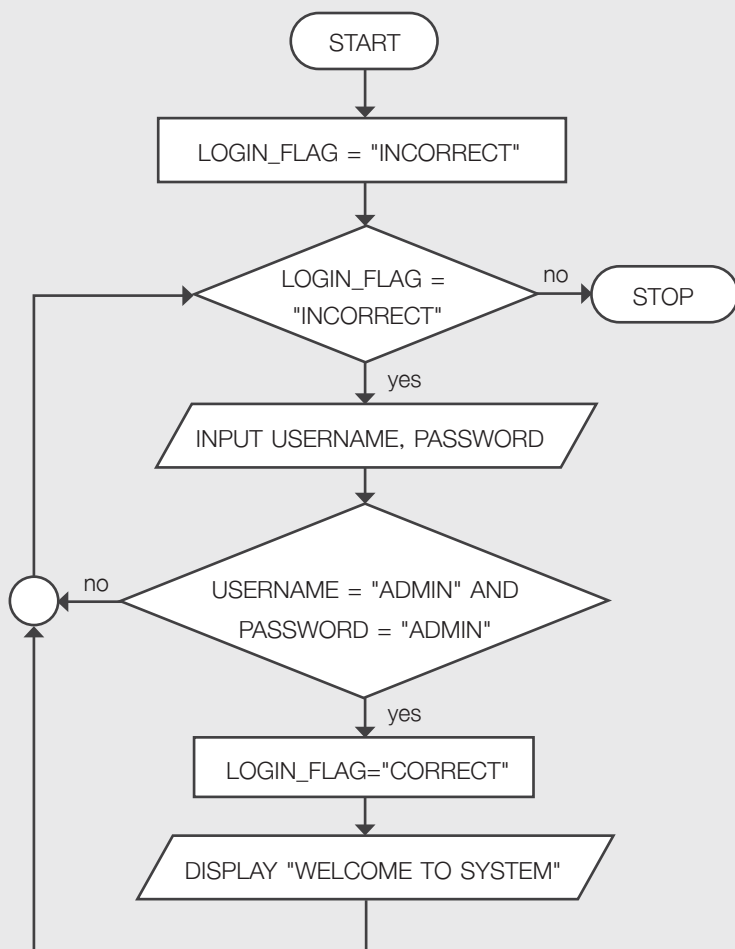
แสดงการเขียนโฟลว์ชาร์ตหาค่าเฉลี่ยของตัวเลข 3 ตัว

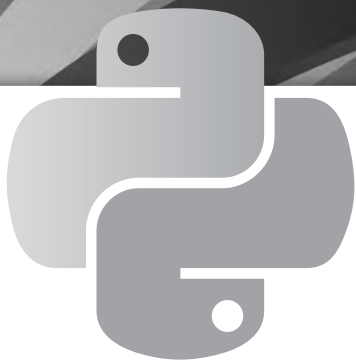




ตัวอย่างที่ 2.10

แสดงการเขียนโฟลว์ชาร์ตตรวจสอบการล็อกอินเข้าใช้ระบบ





CHAPTER

03

เครื่องมือเขียนโปรแกรม ภาษาไพธอน

IDE คืออะไร

การเขียนโปรแกรมในปัจจุบันส่วนใหญ่เรามักจะเขียนโปรแกรมผ่าน IDE (Integrated Development Environment) ต่างๆ มากกว่าการเขียนโปรแกรมด้วย text editor ธรรมดา เนื่องจาก IDE เป็นเครื่องมือพัฒนาโปรแกรมที่ช่วยให้ผู้เขียนโปรแกรมสามารถเขียนโปรแกรมได้สะดวกและง่ายขึ้น

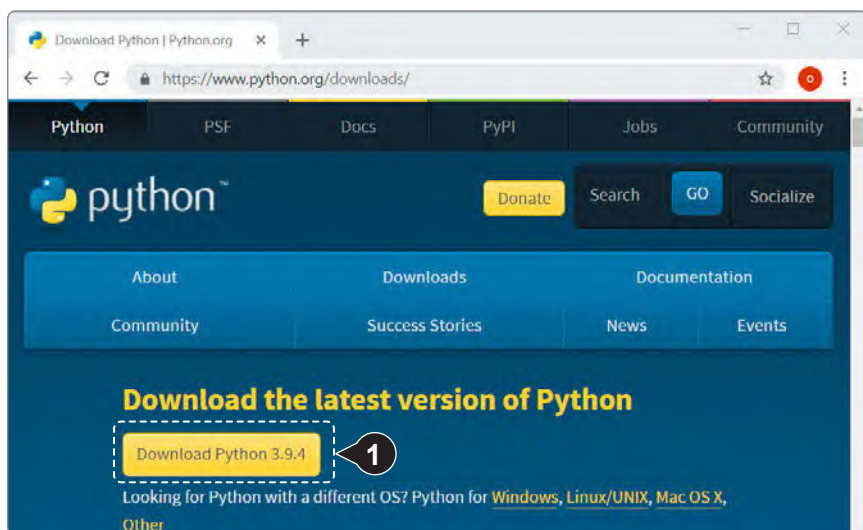
IDE เป็นหน้าต่าง editor สำหรับใช้เขียนโค้ด ทดสอบและรันโปรแกรม นอกจาก IDE จะประกอบด้วย source code editor สำหรับเขียนโปรแกรมแล้ว มันยังรวมตัวแปลภาษา และ debugger ที่ช่วยตรวจสอบข้อผิดพลาดของโปรแกรมไว้ในตัวมันด้วย อีกทั้ง IDE บางตัวยังประกอบด้วยไลบรารีที่จำเป็นสำหรับการเขียนโปรแกรมภาษาต่างๆ ด้วย การเขียนโปรแกรมในแต่ละภาษาจึงจำเป็นต้องเลือก IDE ที่เหมาะสมกับการเขียนโปรแกรมภาษานั้นๆ ก่อนเขียนโปรแกรมภาษาไพธอนจึงต้องติดตั้ง IDE ก่อน

IDE สำหรับเขียนโปรแกรมภาษาไพธอนมีหลายประเภท ในหนังสือเล่มนี้จะอธิบายเครื่องมือ 3 ตัวที่นิยมใช้งาน คือ Python IDLE, PyCharm และ Jupyter Notebook

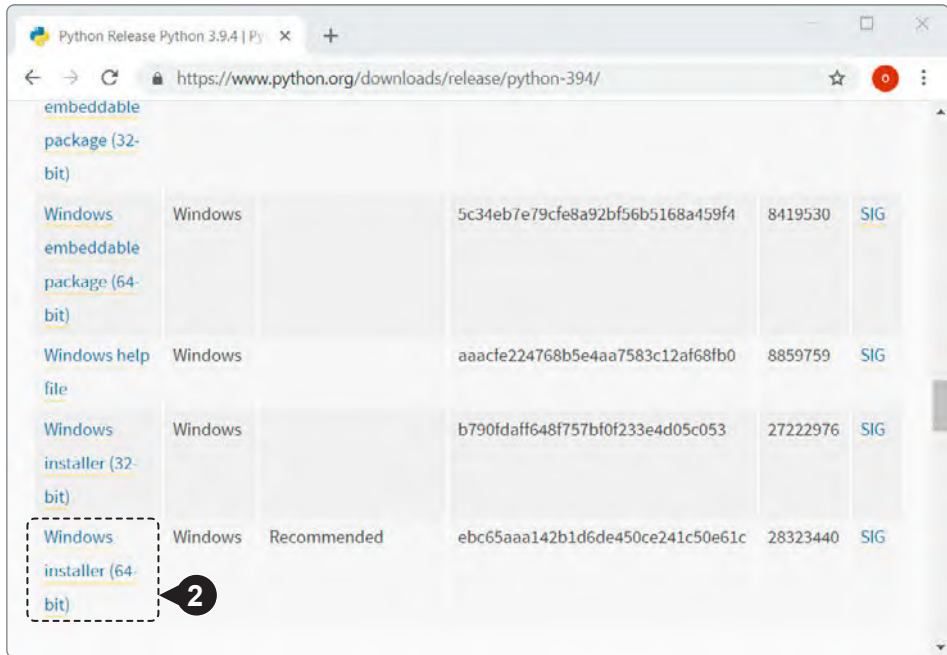
Python IDLE

วิธีดาวน์โหลดและติดตั้งโปรแกรม Python for Windows

1. ไปที่ URL <https://www.python.org/downloads> แล้วคลิกที่ **Download Python 3.9.4** ดังรูป เพื่อเริ่มต้นดาวน์โหลดโปรแกรม



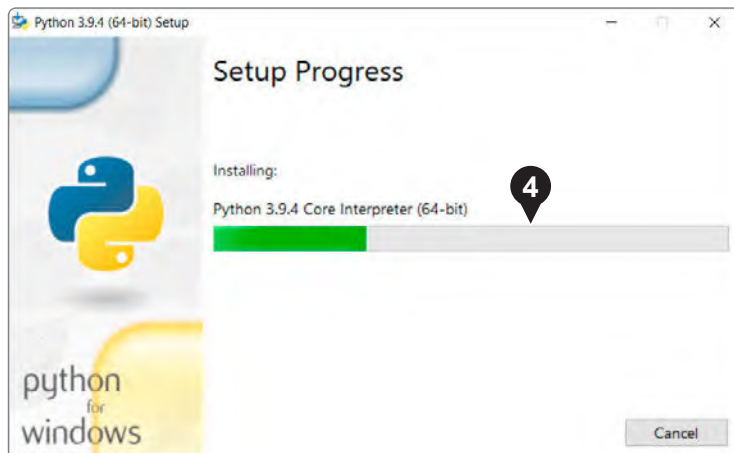
2. คลิกเลือกที่ Windows installer (64-bit)



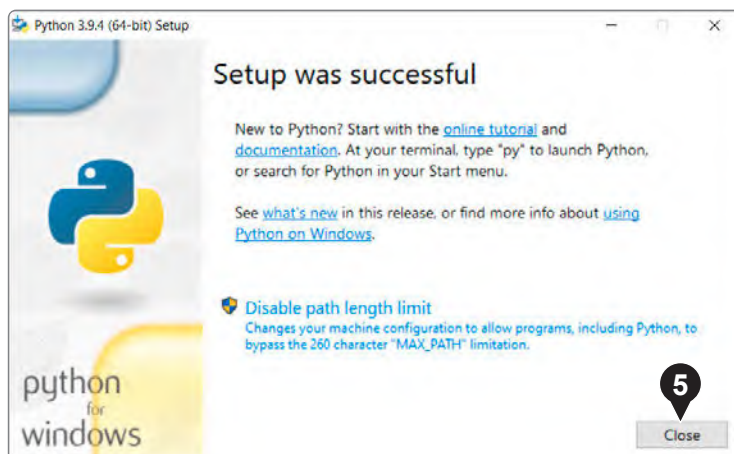
3. เมื่อดาวน์โหลดเสร็จแล้ว ให้คลิกเปิดไฟล์ **python-3.9.4-amd64.exe** ที่ดาวน์โหลดมา และคลิก ✓ ที่ **Add Python 3.9 to PATH** จากนั้นคลิก **Install Now** ดังรูป เพื่อเริ่มติดตั้งโปรแกรม



4. ระหว่างนี้
โปรแกรมจะถูก
ติดตั้งลงที่เครื่อง
ให้รอสักครู่จน
กระทั่งติดตั้งเสร็จ
สมบูรณ์

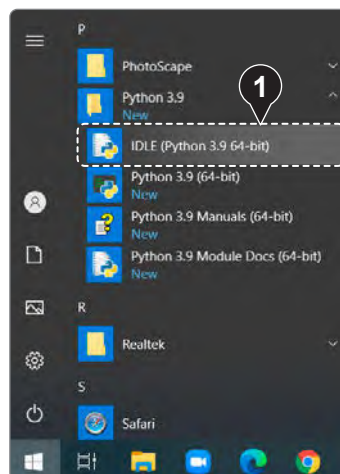


5. เมื่อติดตั้งเสร็จ
ให้คลิกปุ่ม **Close**
เพื่อปิดหน้าต่าง
ดังรูป

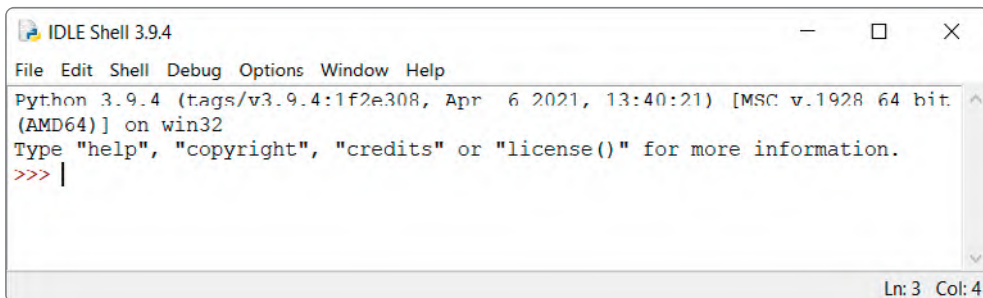


วิธีเรียกใช้งาน Python IDLE

1. คลิกที่ปุ่ม **Start** แล้วเลือกไปที่ **Python 3.9 ▶ IDLE**
(Python 3.9 64-bit) ดังรูป



2. จะมีหน้าต่าง IDLE Shell 3.9.4 ปรากฏขึ้นมา ดังรูป ที่หน้าต่างนี้เราสามารถเขียนโปรแกรมทดสอบและรันโปรแกรมได้



```
IDLE Shell 3.9.4
File Edit Shell Debug Options Window Help
Python 3.9.4 (tags/v3.9.4:1f2e308, Apr 6 2021, 13:40:21) [MSC v.1928 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> |
Ln: 3 Col: 4
```

เขียนโปรแกรมไพธอนโดยใช้ Python IDLE

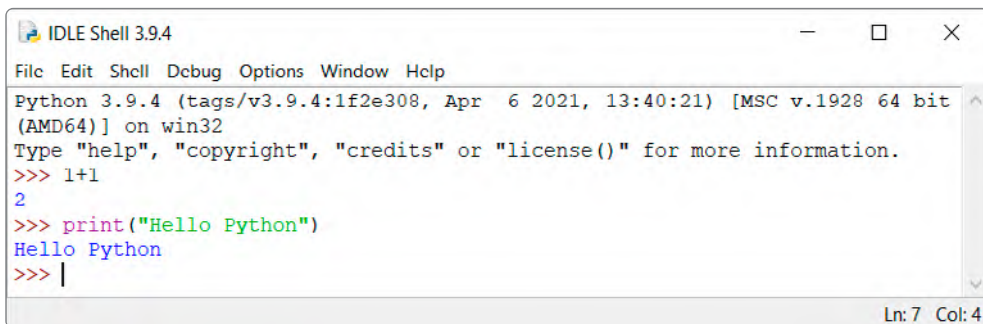
การเขียนโปรแกรมภาษาไพธอนโดยใช้ Python IDLE ทำได้ 2 วิธี คือ

วิธีที่ 1 : เขียนโปรแกรมลงใน Python Shell

วิธีนี้เหมาะกับการทดลองเขียนโปรแกรมเล็กๆ ง่ายๆ หรือใช้สำหรับทดสอบการเขียนคำสั่งโปรแกรมบางอย่างว่าทำงานได้ถูกต้องหรือไม่ ผลลัพธ์เป็นอย่างไร

เครื่องหมาย >>> จะเป็น prompt ของ Python Shell มีไว้สำหรับรับคำสั่งโปรแกรมเพื่อนำไปประมวลผล ลองพิมพ์คำสั่ง 1+1 จะได้ผลลัพธ์เป็น 2 แสดงในบรรทัดถัดมา

จากนั้นพิมพ์คำสั่ง `print("Hello Python")` ก็จะได้ผลลัพธ์เป็นข้อความ Hello Python แสดงออกมา ดังรูป



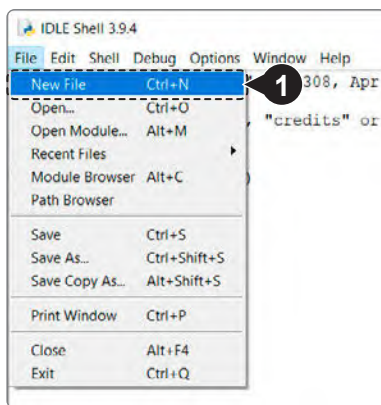
```
IDLE Shell 3.9.4
File Edit Shell Debug Options Window Help
Python 3.9.4 (tags/v3.9.4:1f2e308, Apr 6 2021, 13:40:21) [MSC v.1928 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 1+1
2
>>> print("Hello Python")
Hello Python
>>> |
Ln: 7 Col: 4
```

วิธีที่ 2 : เขียนโปรแกรมในรูปแบบ source code

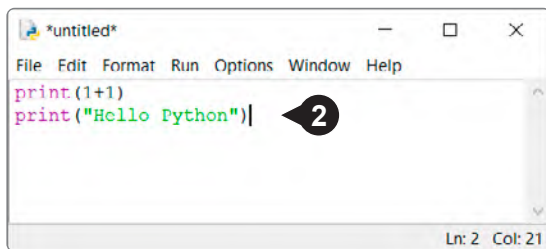
วิธีนี้เหมาะกับการพัฒนาโปรแกรมแบบจริงจังเพื่อนำไปใช้งาน โดยจะต้องเขียนโปรแกรม และบันทึกเป็นไฟล์โปรแกรม จากนั้นเรียกรันผลลัพธ์ของโปรแกรมจากไฟล์

ขั้นตอนการเขียนโปรแกรมในรูปแบบนี้ เป็นดังนี้

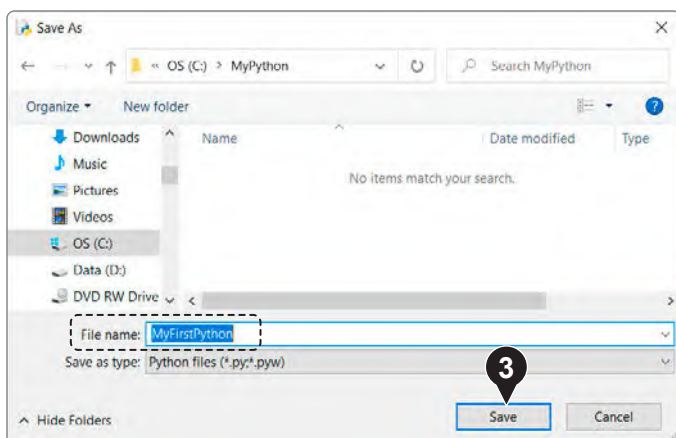
1. ที่หน้าจอ IDLE Shell ให้คลิกที่เมนู **File ▶ New File** (หรือจะกดคีย์ **Ctrl+N** ก็ได้) ดังรูป



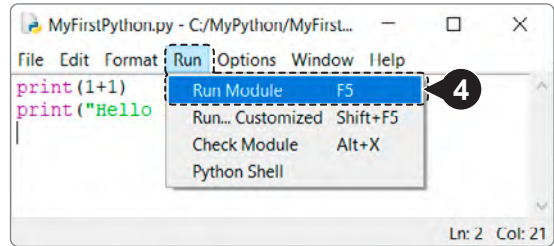
2. จะมีหน้าจอ Editor แสดงขึ้นมา ที่หน้าจอนี้เราสามารถเขียนโปรแกรมไพธอนได้ ให้ผู้อ่านเขียนโปรแกรมตามรูป สิ่งที่ต้องระวัง คือ การเขียนโปรแกรมต้องเขียนชิดด้านซ้ายเท่านั้น ห้ามมีย่อหน้า ช่องว่างหรือเว้นวรรคใดๆ ไม่เช่นนั้น โปรแกรมจะประมวลผลไม่ผ่าน



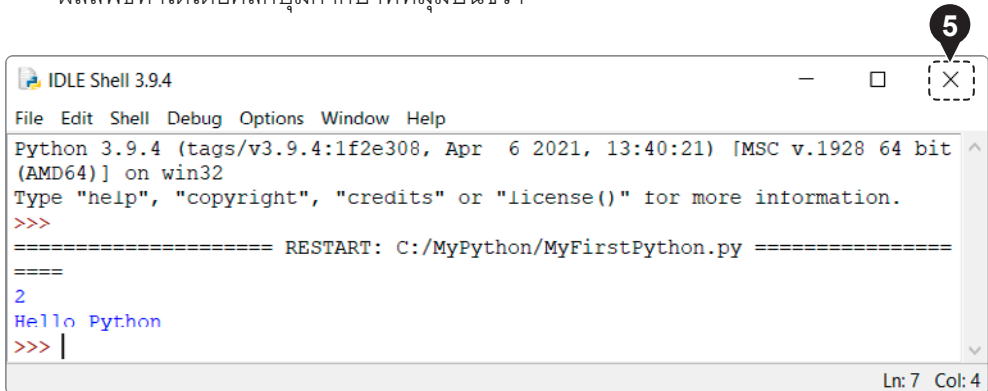
3. ไปที่เมนู **File ▶ Save** (หรือจะกดคีย์ **Ctrl+S** ก็ได้) จะมีหน้าต่างแสดงขึ้นมาดังรูปเพื่อให้บันทึกไฟล์ โดยโปรแกรมที่บันทึกจะมีนามสกุลเป็น .py ในที่นี้ตั้งชื่อโปรแกรมเป็น **MyFirstPython** เก็บไว้ที่ **C:\MyPython** จากนั้นคลิกปุ่ม **Save** ดังรูป



4. ประมวลผลโปรแกรมโดยไปที่เมนู
Run ▶ Run Module (หรือจะกดคีย์
F5 ก็ได้) ดังรูป



5. Python Shell จะแสดงผลลัพธ์ของโปรแกรมออกมาดังรูปด้านล่าง หากต้องการปิดหน้าต่าง
ผลลัพธ์ทำได้โดยคลิกปุ่มกากบาทที่มุมบนขวา



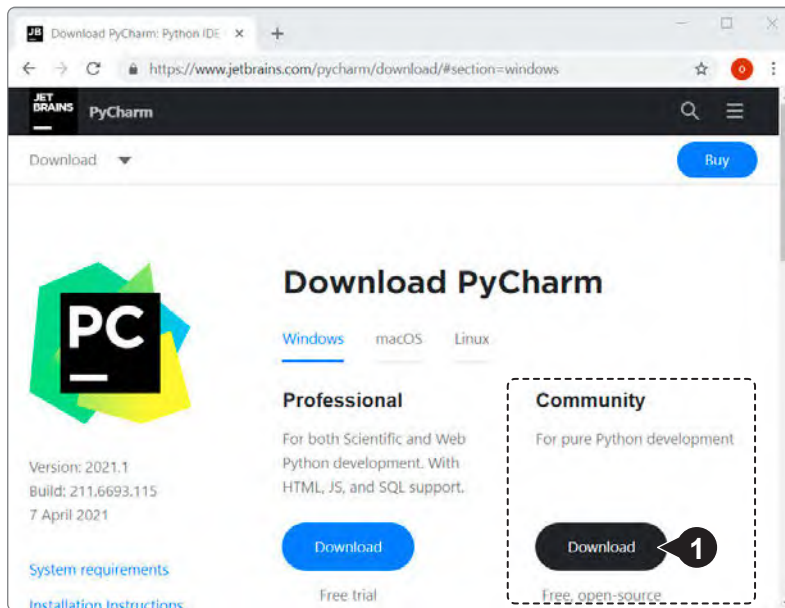
PyCharm

วิธีดาวน์โหลดและติดตั้งโปรแกรม PyCharm

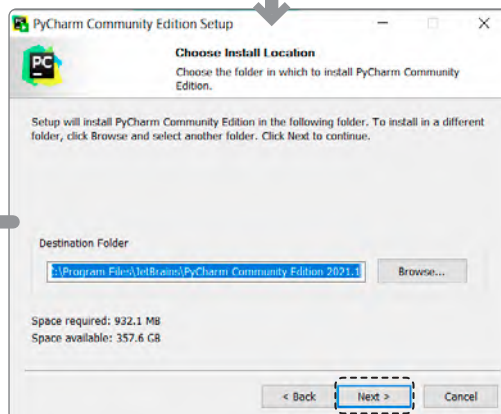
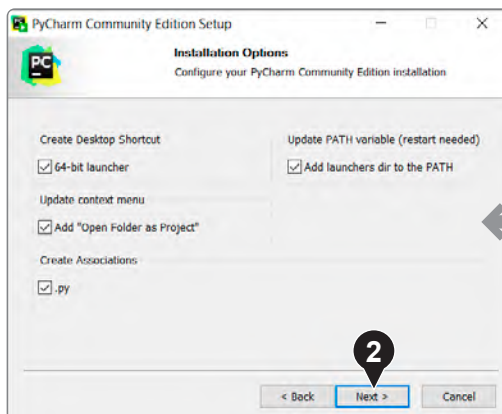
ก่อนติดตั้งโปรแกรม PyCharm จะต้องติดตั้งโปรแกรม Python for Windows ก่อนเสมอ (วิธีติดตั้งสามารถดูได้จากหัวข้อ Python IDLE ค่ะ)

หากไม่ติดตั้ง Python for Windows ก่อนจะไม่สามารถคอมไพล์และรันโปรแกรมภาษาไพธอนที่เขียนขึ้นด้วย PyCharm ได้ โดยเมื่อติดตั้งแล้วให้ดำเนินการติดตั้ง PyCharm ดังขั้นตอนต่อไปนี้จะ

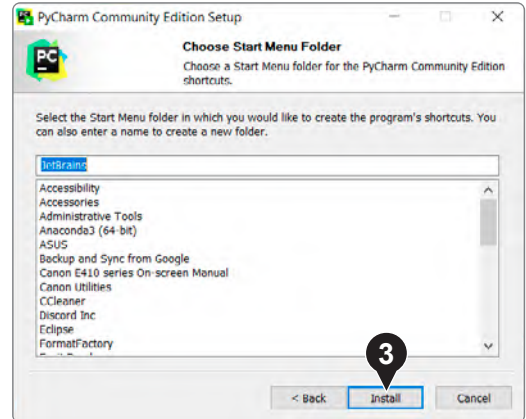
1. ไปที่ URL <https://www.jetbrains.com/pycharm/download> แล้วคลิกปุ่ม **Download** ที่หัวข้อ **Community** เพื่อเริ่มต้นดาวน์โหลดโปรแกรม



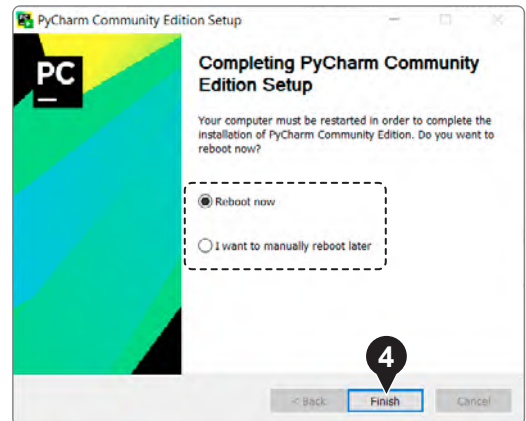
2. ให้เปิดไฟล์ pycharm-community-2021.1.exe ซึ่งเป็นไฟล์ที่ดาวน์โหลดมา จะมีหน้าจอปรากฏดังรูป ให้คลิกปุ่ม Next



3. ให้คลิกปุ่ม **Install** เพื่อเริ่มติดตั้งโปรแกรม

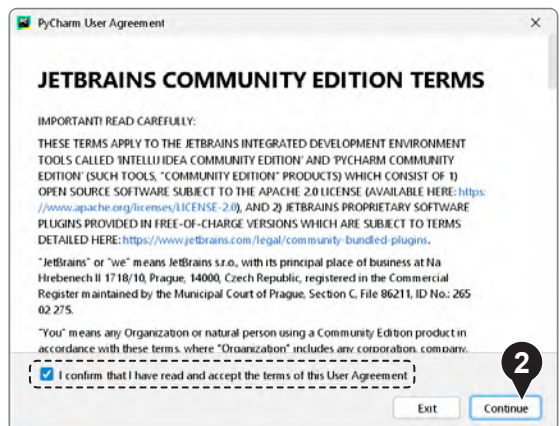


4. การติดตั้งโปรแกรม PyCharm จะเสร็จสมบูรณ์ก็ต่อเมื่อเราสั่งรีสตาร์ทเครื่องคอมพิวเตอร์ ให้เลือก **Reboot now** เพื่อรีสตาร์ทเครื่องทันที หรือเลือก **I want to manually reboot later** เพื่อรีสตาร์ทเครื่องภายหลังก็ได้ จากนั้นคลิกปุ่ม **Finish**

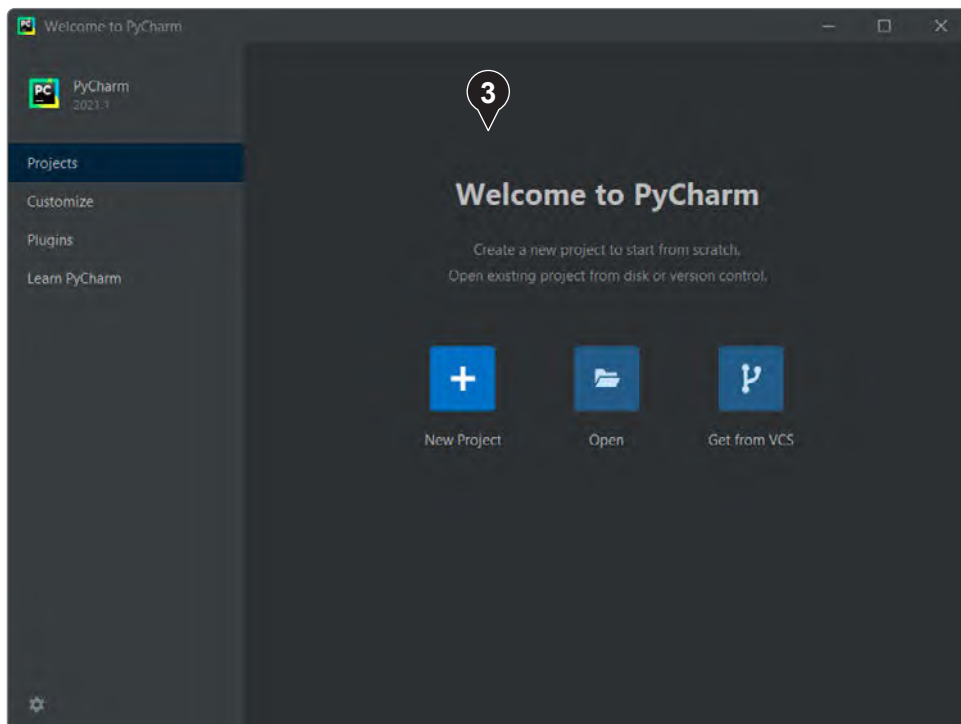


วิธีเรียกใช้งาน PyCharm

1. ไปที่ Desktop แล้วดับเบิลคลิกไอคอนของโปรแกรม PyCharm ดังรูป
2. เลือก **I confirm that I have read and accept the terms of this User Agreement** แล้วคลิกปุ่ม **Continue**

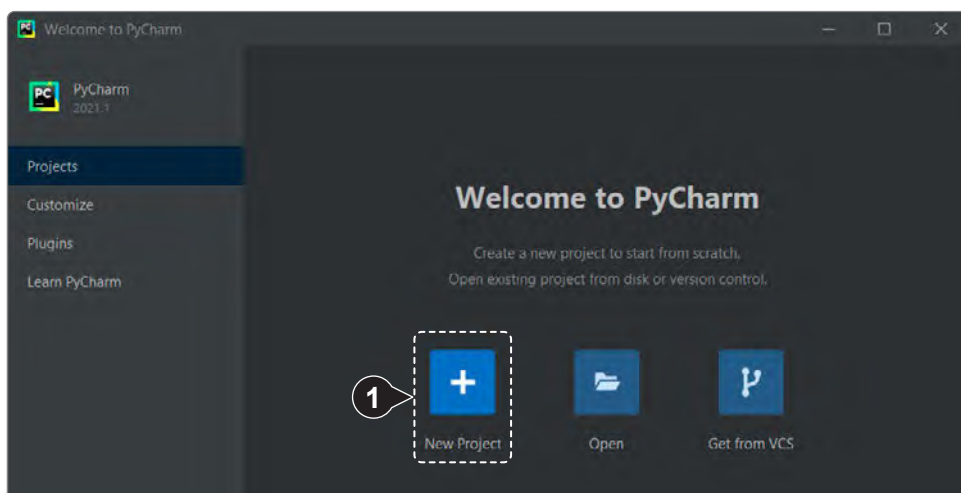


3. จะมีหน้าจอ PyCharm ปรากฏขึ้นมา ดังรูป

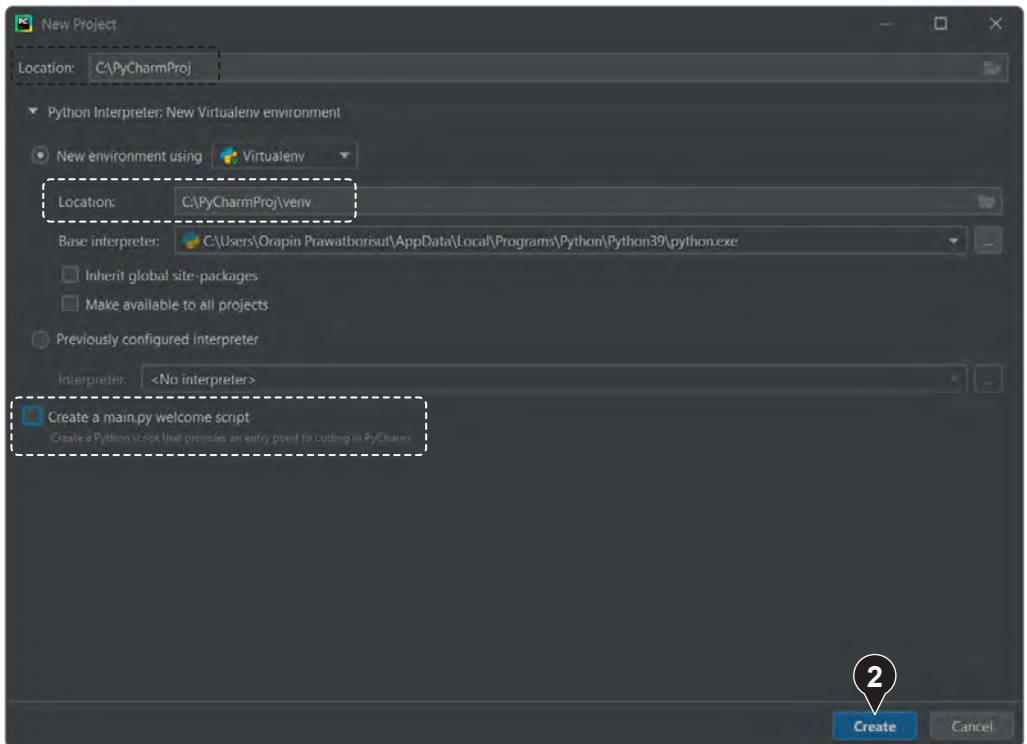


เขียนโปรแกรมไพธอนโดยใช้ PyCharm

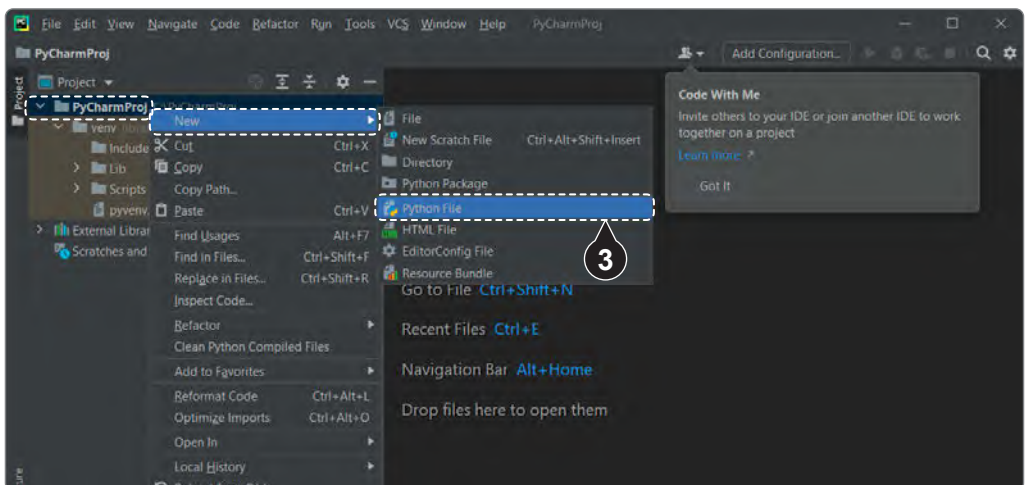
1. คลิกเลือก New Project เพื่อสร้างโปรเจกต์ใหม่



2. ที่ส่วนของ Location ให้กำหนดได้เร็คทอรีสำหรับจัดเก็บโปรเจ็ค ในที่นี้กำหนดเป็น **C:\PyCharmProj** และไม่ทำการคลิกเลือกในส่วนของ Create a main.py welcome script จากนั้นคลิกปุ่ม **Create**

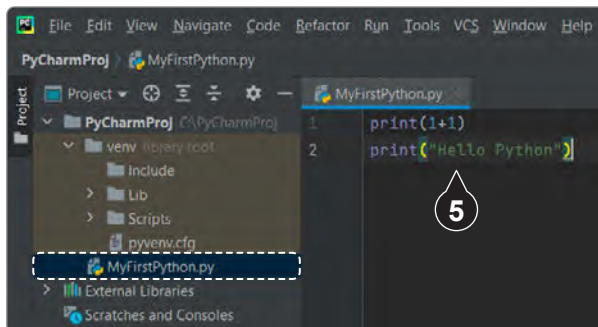
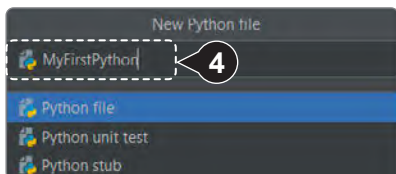


3. คลิกขวาที่ชื่อโปรเจ็ค แล้วเลือก **New ► Python File** เพื่อสร้างไฟล์โปรแกรมภาษาไพธอน



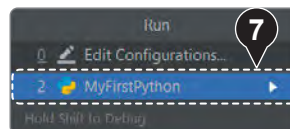
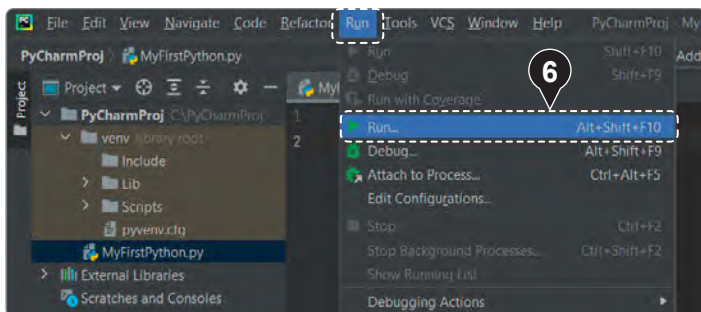
4. กำหนดชื่อไฟล์ของโปรแกรมภาษาไพธอน (ในที่นี้กำหนดเป็น **MyFirstPython**) แล้วกดคีย์ **Enter**

5. จะมีหน้าต่างของไฟล์ **MyFirstPython.py** ปรากฏขึ้นมา ให้พิมพ์โค้ดโปรแกรม ดังรูป

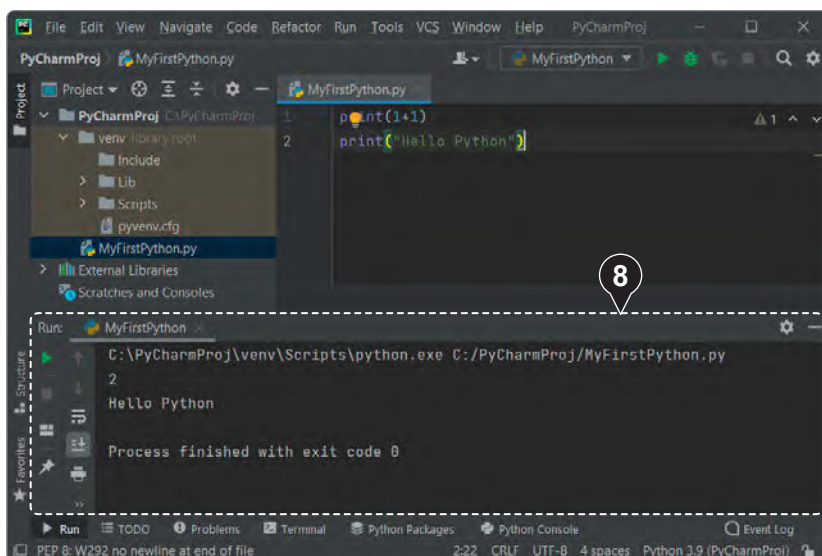


6. ไปที่เมนู **Run** ▶ **Run** หรือกดคีย์ **Alt+Shift+F10** เพื่อรันดูผลลัพธ์ของโปรแกรม

7. ให้คลิกที่ชื่อไฟล์ **MyFirstPython**



8. จะมีหน้าต่างผลลัพธ์ของโปรแกรม แสดงขึ้นมา



Jupyter Notebook

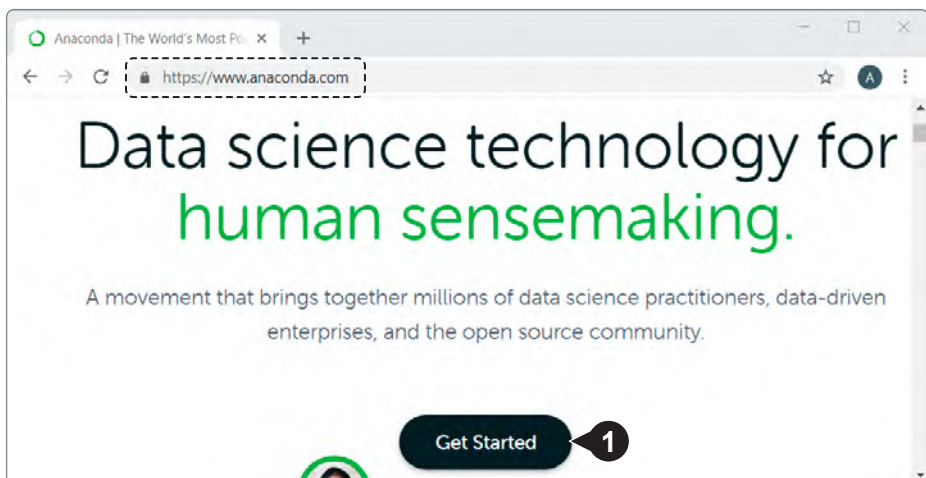
Jupyter Notebook เป็น IDE ตัวหนึ่งที่ยิมนำมาใช้ในการเขียนโปรแกรมของ Data Science (ใน Part 3 ของหนังสือเล่มนี้จะสอนการเขียนโปรแกรมภาษาไพธอนสำหรับงาน Data Science หากผู้อ่านต้องการเรียนรู้การเขียนโปรแกรมสำหรับงาน Data Science สามารถเลือกใช้ Jupyter Notebook เป็น IDE ได้ค่ะ) ซึ่ง Jupyter notebook เป็น open-source web application ที่ทำงานบนเว็บเบราว์เซอร์ในลักษณะโครงสร้างแบบ client-server application

การใช้งาน Jupyter Notebook ในหนังสือเล่มนี้จะสอนติดตั้งโปรแกรม Anaconda เนื่องจาก Anaconda ติดตั้งง่าย ใช้งานไม่ยาก มีไลบรารีการทำงานต่าง ๆ ค่อนข้างครบสำหรับการเขียนโปรแกรมภาษาไพธอน รวมถึงไลบรารีต่าง ๆ สำหรับการเขียนโปรแกรม Data Science ด้วย เช่น ไลบรารี NumPy, Pandas, Matplotlib เป็นต้น โดยที่เราไม่ต้องติดตั้งไลบรารีเพิ่มเติมด้วยตัวเองให้ยุ่งยาก และที่สำคัญ Anaconda มี Jupyter Notebook ผูกติดมากับมันด้วย เรียกได้ว่าลง Anaconda ตัวเดียวก็พร้อมสำหรับการเขียนโปรแกรมภาษาไพธอน และการเขียนโปรแกรมภาษาไพธอนสำหรับงาน Data Science เลยค่ะ

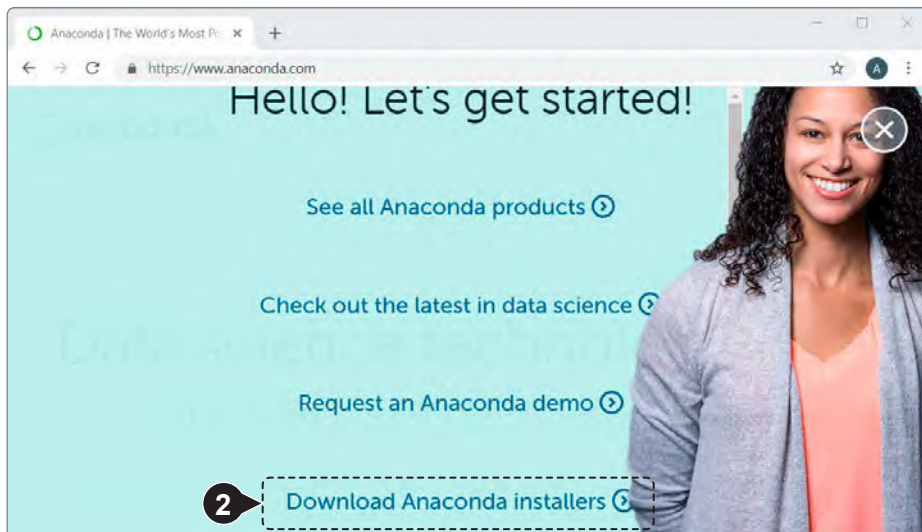
โดยปกติเมื่อติดตั้ง Anaconda แล้ว Jupyter Notebook server จะถูกสร้างขึ้นมาบนเครื่องของเราโดยอัตโนมัติ ดังนั้น การเรียกใช้งาน Jupyter Notebook จึงไม่จำเป็นต้องเชื่อมต่อกับระบบเครือข่ายอินเทอร์เน็ต เมื่อเราสั่งเรียก Jupyter notebook ให้ทำงาน นอกจาก server จะถูกสตาร์ทให้ทำงานแล้ว มันจะเปิด Jupyter Notebook client บนเว็บเบราว์เซอร์ให้อัตโนมัติด้วย เราสามารถเขียนโปรแกรมต่าง ๆ ลงบน Jupyter Notebook client และสั่งรันเพื่อดูผลลัพธ์ได้

วิธีดาวน์โหลดและติดตั้งโปรแกรม Anaconda

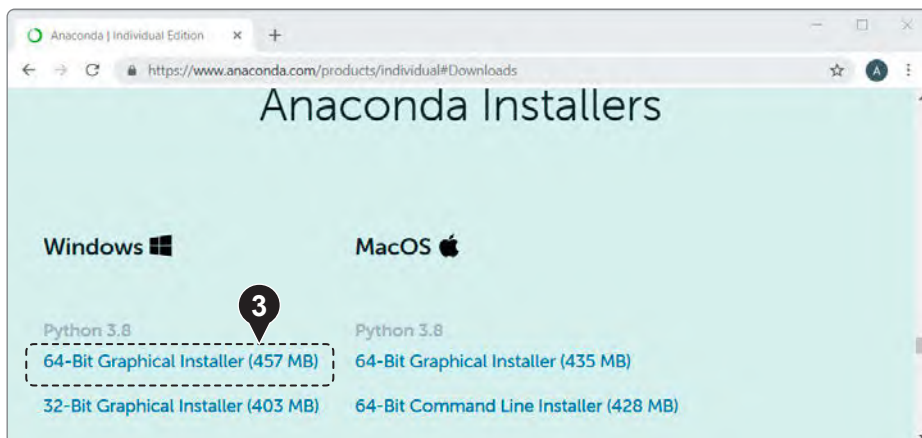
1. เปิดเว็บเบราว์เซอร์ไปที่ <http://www.anaconda.com> แล้วคลิกเลือกที่ **Get Started**



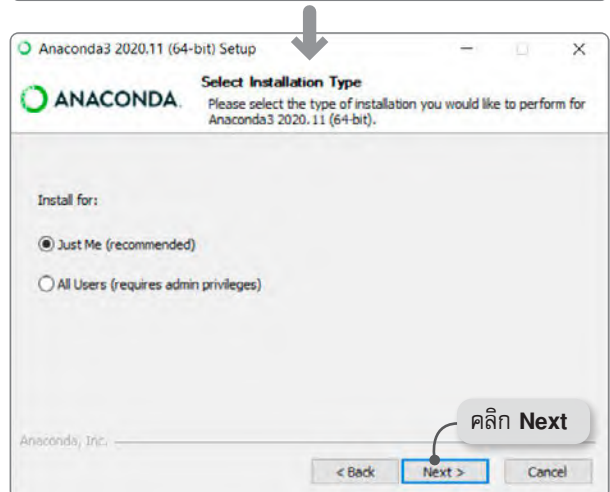
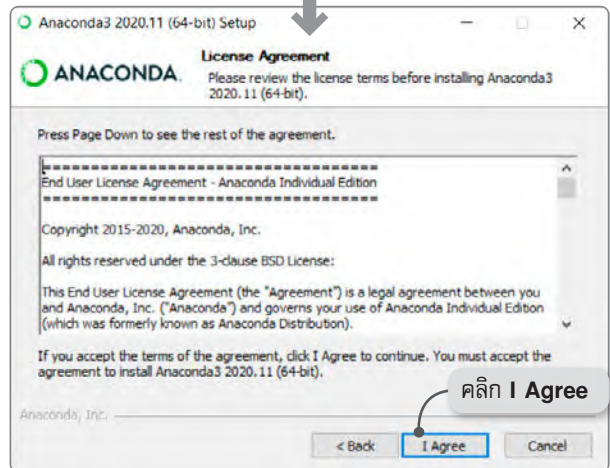
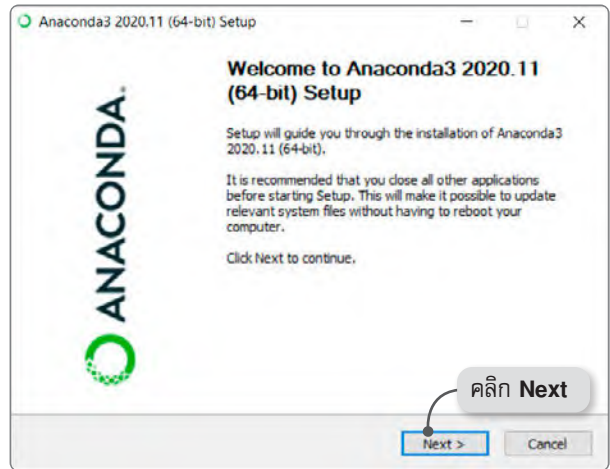
2. คลิกที่ Download Anaconda Installers

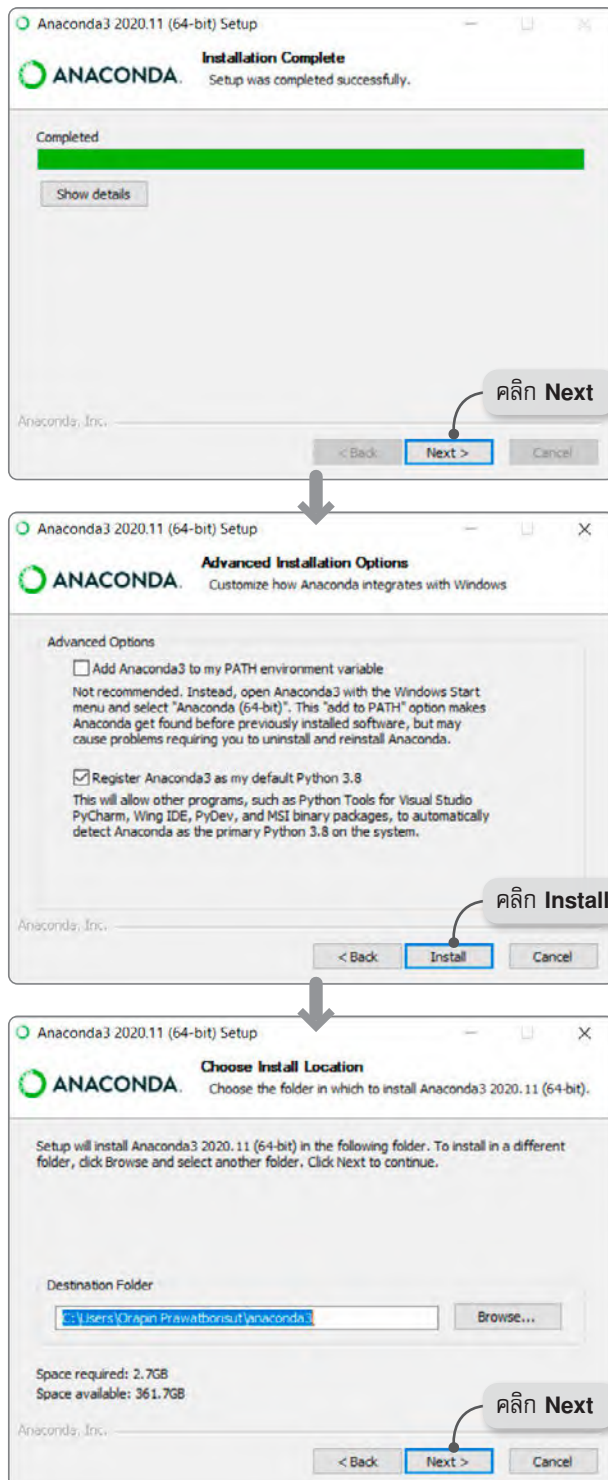


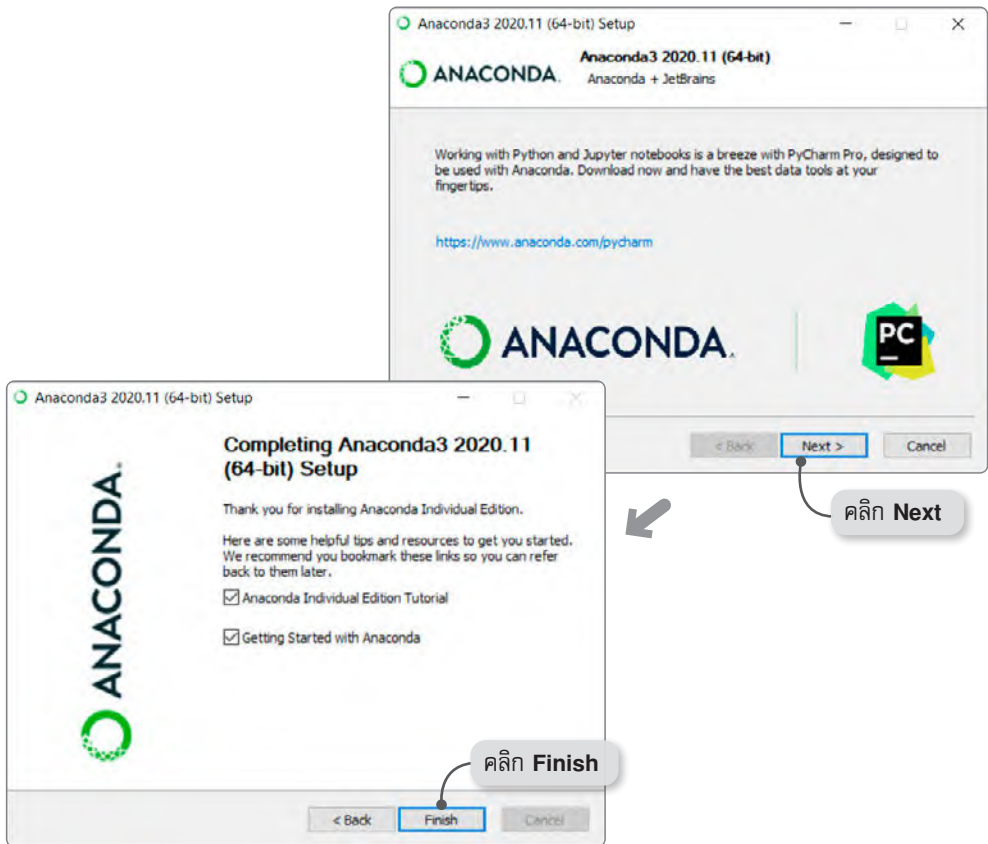
3. เลื่อนไปที่ระบบปฏิบัติการ **Windows** แล้วคลิกเลือก **64-Bit Graphical Installer (457 MB)** แต่ถ้ระบบปฏิบัติการที่เครื่องคอมพิวเตอร์ของผู้อ่านเป็นแบบ 32 บิต ก็ให้คลิกเลือกที่ 32 Bit Graphical Installer (403 MB) จากนั้นรอจนกระทั่งดาวน์โหลดไฟล์สำเร็จ



4. ดับเบิลคลิกที่ไฟล์ที่ดาวน์โหลดมาเพื่อเริ่มการติดตั้ง โดยการติดตั้งให้ทำตามขั้นตอนดังรูป







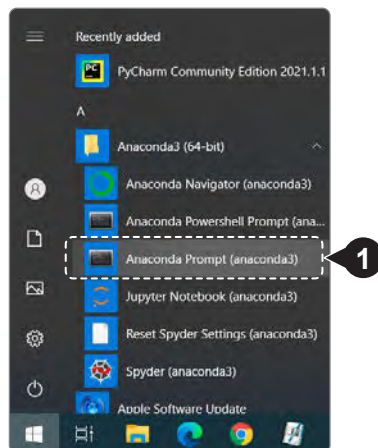
การสร้าง Virtual Environment สำหรับ Python 3.9 บน Anaconda

เนื่องจากเวอร์ชันของไพธอนที่ผูกติดมากับ Anaconda3 จะยังคงเป็น python 3.8 ขั้นตอนต่อไปเราจะมาอัปเดตเวอร์ชันของไพธอนเป็นเวอร์ชันล่าสุด (3.9.4) กัน

การสร้าง Virtual Environment ของไพธอนบน Anaconda จะมีประโยชน์ในการแยกสภาพแวดล้อมในการทำงานของคอมไพเลอร์และชุดแพ็คเกจ (โมดูล) ที่เป็นไลบรารีของตัวภาษาไพธอนคนละเวอร์ชันออกจากกัน (Isolated Python Environment) โดยในที่นี้จะเป็นการแยกเพื่อให้เราสามารถใช้งานฟังก์ชันการทำงานใหม่ๆ ของ Python 3.9 ได้โดยไม่กระทบกับโปรแกรมอื่นที่อาจจะเขียนอ้างอิงกับ Python 3.8 หรือเวอร์ชันก่อนหน้านี้อย่างไรก็ตาม แต่ละ environment จะแยกติดตั้งชุดแพ็คเกจโดยไม่มีส่วนเกี่ยวข้องกัน เช่น หากทำการติดตั้งชุดแพ็คเกจ django บน environment ของ python3.9 แต่เราไปเรียกใช้งาน environment ของ python 3.8 ก็จะไม่สามารถใช้งาน django ได้ เพราะ django ถูกติดตั้งไว้แค่ที่ environment ของ python 3.9 เป็นต้น

ขั้นตอนการติดตั้ง Python 3.9 บน Anaconda

1. คลิกปุ่ม **Start** แล้วเลือกไปที่ **Anaconda3 (64-bit)** จากนั้นคลิกเลือก **Anaconda Prompt (anaconda3)**



2. พิมพ์คำสั่งดังต่อไปนี้

```
conda create -c conda-forge -n python39 python=3.9.4
```

แล้วกดคีย์ **Enter** เพื่อเริ่มทำการติดตั้ง Python 3.9.4 ซึ่งจะเป็นการติดตั้ง Python 3.9.4 ลงบน environment ใหม่ที่ชื่อว่า python39 ภายใต้ Anaconda3

```
Anaconda Prompt (anaconda3) - conda create -c conda-forge -n python39 python=3.9.4

(base) C:\Users\Orapin Prawatborisut>conda create -c conda-forge -n python39 python=3.9.4
WARNING: A space was detected in your requested environment path
'C:\Users\Orapin Prawatborisut\anaconda3\envs\python39'
Spaces in paths can sometimes be problematic.
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

environment location: C:\Users\Orapin Prawatborisut\anaconda3\envs\python39

added / updated specs:
- python=3.9.4

The following packages will be downloaded:
```

package	build	size	channel
ca-certificates-2020.12.5	h5b45459_0	173 KB	conda-forge
certifi-2020.12.5	py39hcbf5309_1	144 KB	conda-forge
openssl 1.1.1k	h8fffc710_0	5.7 MB	conda-forge
pip-21.1.1	pyhd8ed1ab_0	1.1 MB	conda-forge
python-3.9.4	h6244533_0	16.4 MB	
python_abi-3.9	1_cp39	4 KB	conda-forge
setuptools-49.6.0	py39hcbf5309_3	954 KB	conda-forge
sqlite 3.35.5	h8fffc710_0	1.2 MB	conda-forge
tzdata-2021a	he74cb21_0	121 KB	conda-forge
vc-14.2	hb210afc_4	13 KB	conda-forge

3. Anaconda จะทำการตรวจสอบและแสดงแพ็คเกจของไพธอนว่าต้องทำการติดตั้งแพ็คเกจใดบ้าง ในขั้นนี้ให้พิมพ์ “y” แล้วกดคีย์ **Enter** เพื่อเริ่มการติดตั้ง

```
Anaconda Prompt (anaconda3) - conda create -c conda-forge -n python39 python=3.9.4

python_abi-3.9                1_cp39                4 KB    conda-forge
setuptools-49.6.0             py39hcbf5309_3        954 KB  conda-forge
sqlite-3.35.5                 h8ffe710_0            1.2 MB  conda-forge
tzdata-2021a                  he74cb21_0            121 KB  conda-forge
vc-14.2                        hb210afc_4            1.1 KB  conda-forge
vs2015_runtime-14.28.29325    h5e1d092_4            2.3 MB  conda-forge
wheel-0.36.2                  pyhd3deb0d_0          31 KB   conda-forge
wincertstore-0.2              py39hcbf5309_1006     15 KB   conda-forge
-----
Total:                        28.1 MB

The following NEW packages will be INSTALLED:

ca-certificates                conda-forge/win-64::ca-certificates-2020.12.5-h5b45459_0
certifi                       conda-forge/win-64::certifi-2020.12.5-py39hcbf5309_1
openssl                       conda-forge/win-64::openssl-1.1.1k-h8ffe710_0
pip                           conda-forge/noarch::pip-21.1.1-pyhd8ed1ab_0
python                        pkgs/main/win-64::python-3.9.4-h6244533_0
python_abi                    conda-forge/win-64::python_abi-3.9-1_cp39
setuptools                     conda-forge/win-64::setuptools-49.6.0-py39hcbf5309_3
sqlite                         conda-forge/win-64::sqlite-3.35.5-h8ffe710_0
tzdata                        conda-forge/noarch::tzdata-2021a-he74cb21_0
vc                             conda-forge/win-64::vc-14.2-hb210afc_4
vs2015_runtime                 conda-forge/win-64::vs2015_runtime-14.28.29325-h5e1d092_4
wheel                          conda-forge/noarch::wheel-0.36.2-pyhd3deb0d_0
wincertstore                   conda-forge/win-64::wincertstore-0.2-py39hcbf5309_1006

Proceed ([y]/n)? y
```

4. เมื่อติดตั้งเสร็จ ให้พิมพ์คำสั่งดังต่อไปนี้ เพื่อทำการ activate environment ของ python39 ให้สามารถใช้งานได้

conda activate python39

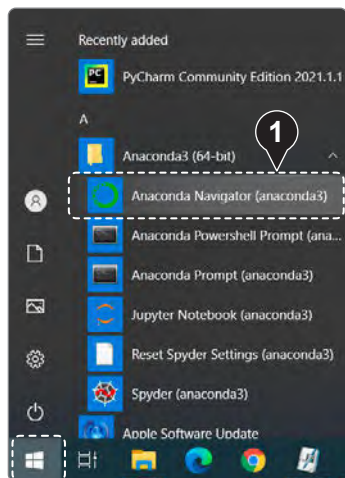
```
Anaconda Prompt (anaconda3)

Downloading and Extracting Packages
openssl-1.1.1k                5.7 MB |#####| 100%
vs2015_runtime-14.28          2.3 MB |#####| 100%
sqlite-3.35.5                 1.2 MB |#####| 100%
wincertstore-0.2              15 KB  |#####| 100%
wheel-0.36.2                  31 KB  |#####| 100%
pip-21.1.1                    1.1 MB |#####| 100%
setuptools-49.6.0             954 KB |#####| 100%
certifi-2020.12.5             144 KB |#####| 100%
vc-14.2                        13 KB  |#####| 100%
python-3.9.4                  16.4 MB|#####| 100%
tzdata-2021a                  121 KB |#####| 100%
python_abi-3.9                 4 KB   |#####| 100%
ca-certificates-2020          173 KB |#####| 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate python39
#
# To deactivate an active environment, use
#
#     $ conda deactivate
#

(base) C:\Users\Orapin Prawatborisut>conda activate python39
```

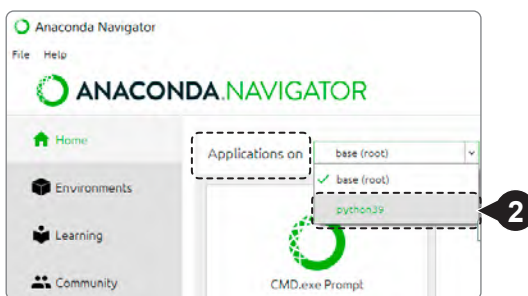
การติดตั้ง Jupyter Notebook ของ Python 3.9 uu Anaconda

เนื่องจากเรามีการสร้าง Virtual Environment ใหม่สำหรับ Python 3.9 ดังที่ได้กล่าวไปแล้ว ดังนั้นแพ็คเกจหรือเครื่องมือต่าง ๆ บน environment นี้ 아직ยังไม่มีติดตั้ง ในครั้งแรกเราจึงต้องทำการติดตั้ง Jupyter Notebook ด้วย ดังนี้

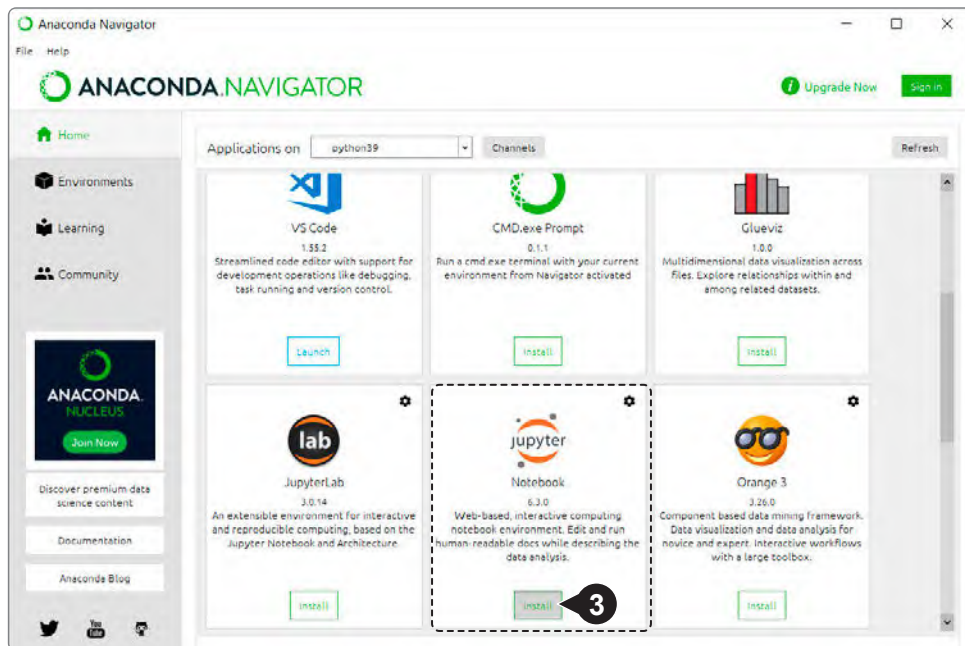


1. คลิกปุ่ม **Start** แล้วเลือกไปที่ **Anaconda3 (64-bit)** จากนั้นคลิกเลือก **Anaconda Navigator (anaconda3)**

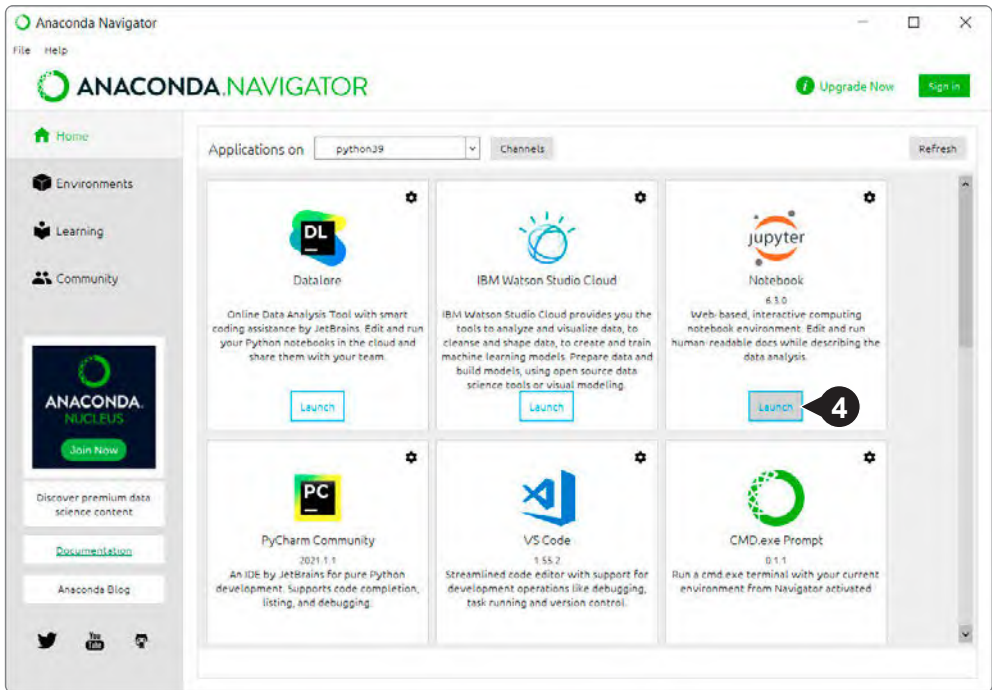
2. คลิกที่ **Applications on** แล้วเลือก **python39**



3. เลือก **Jupyter Notebook** แล้วคลิกที่ **Install**

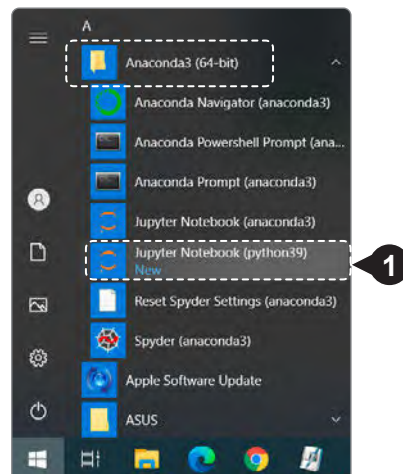


4. เมื่อติดตั้งเสร็จแล้วที่ส่วนของ Jupyter Notebook จะแสดงปุ่ม **Launch** สามารถคลิกเพื่อเปิดโปรแกรม Jupyter Notebook ได้ทันที

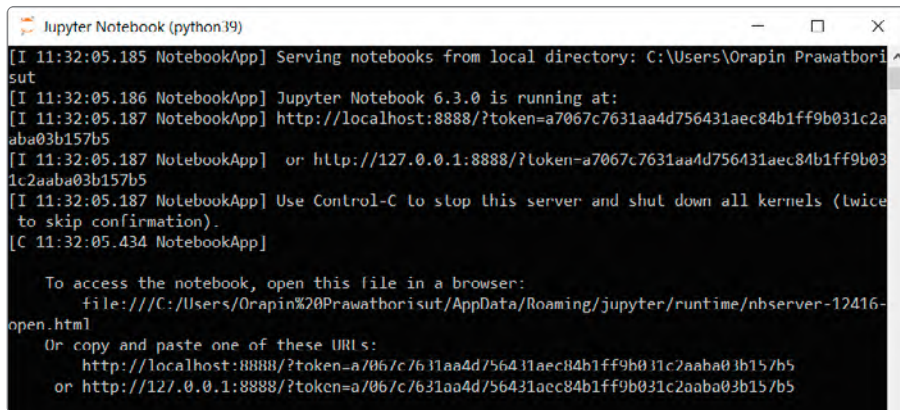


วิธีเรียกใช้งาน Jupyter Notebook

1. คลิกปุ่ม **Start** แล้วเลือกไปที่ **Anaconda3 (64-bit)** จากนั้นคลิกเลือก **Jupyter Notebook (python39)**



2. Jupyter Notebook Server จะถูกสตาร์ทขึ้นมา ดังรูป

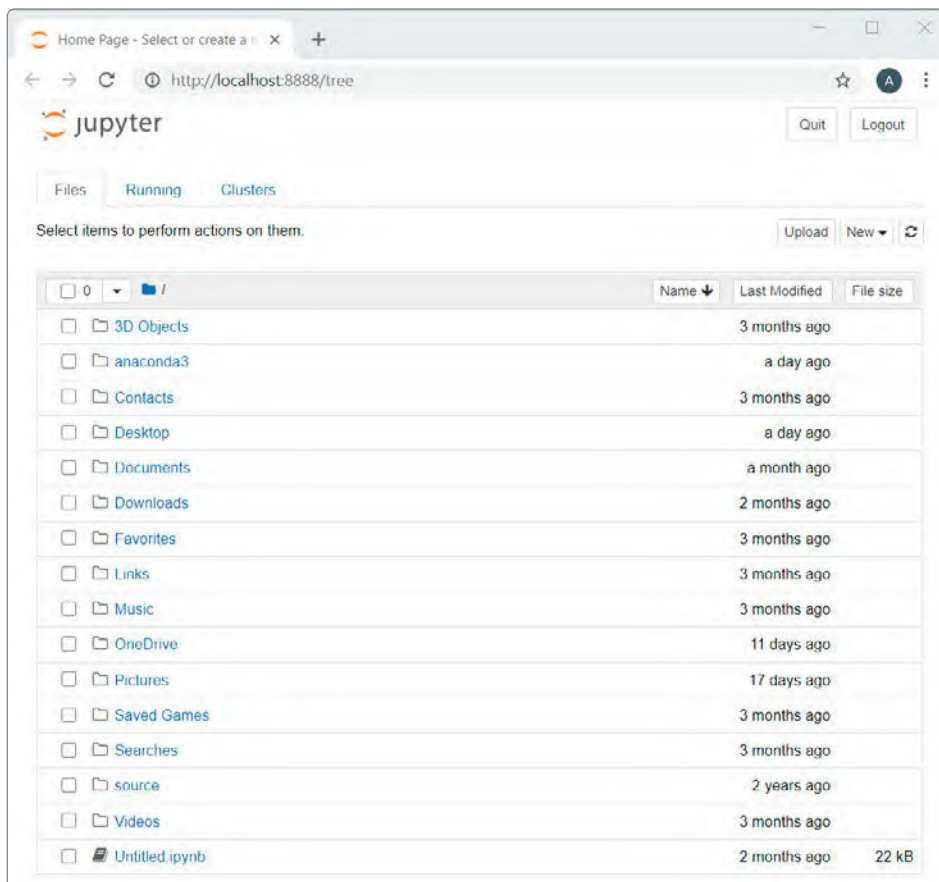


```

Jupyter Notebook (python39)
[I 11:32:05.185 NotebookApp] Serving notebooks from local directory: C:\Users\Orapin Prawathorisut
[I 11:32:05.186 NotebookApp] Jupyter Notebook 6.3.0 is running at:
[I 11:32:05.187 NotebookApp] http://localhost:8888/?token=a7067c7631aa4d756431aec84b1ff9b031c2aaba03b157b5
[I 11:32:05.187 NotebookApp] or http://127.0.0.1:8888/?token=a7067c7631aa4d756431aec84b1ff9b031c2aaba03b157b5
[I 11:32:05.187 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 11:32:05.434 NotebookApp]

To access the notebook, open this file in a browser:
file:///C:/Users/Orapin%20Prawathorisut/AppData/Roaming/jupyter/runtime/nbserver-12416-open.html
Or copy and paste one of these URLs:
http://localhost:8888/?token=a7067c7631aa4d756431aec84b1ff9b031c2aaba03b157b5
or http://127.0.0.1:8888/?token=a7067c7631aa4d756431aec84b1ff9b031c2aaba03b157b5
  
```

3. เมื่อ Jupyter Notebook Server สตาร์ทเสร็จเรียบร้อยแล้ว Jupyter Notebook Client ก็จะถูกเปิดขึ้นมาบนเว็บเบราว์เซอร์โดยอัตโนมัติ ดังรูป



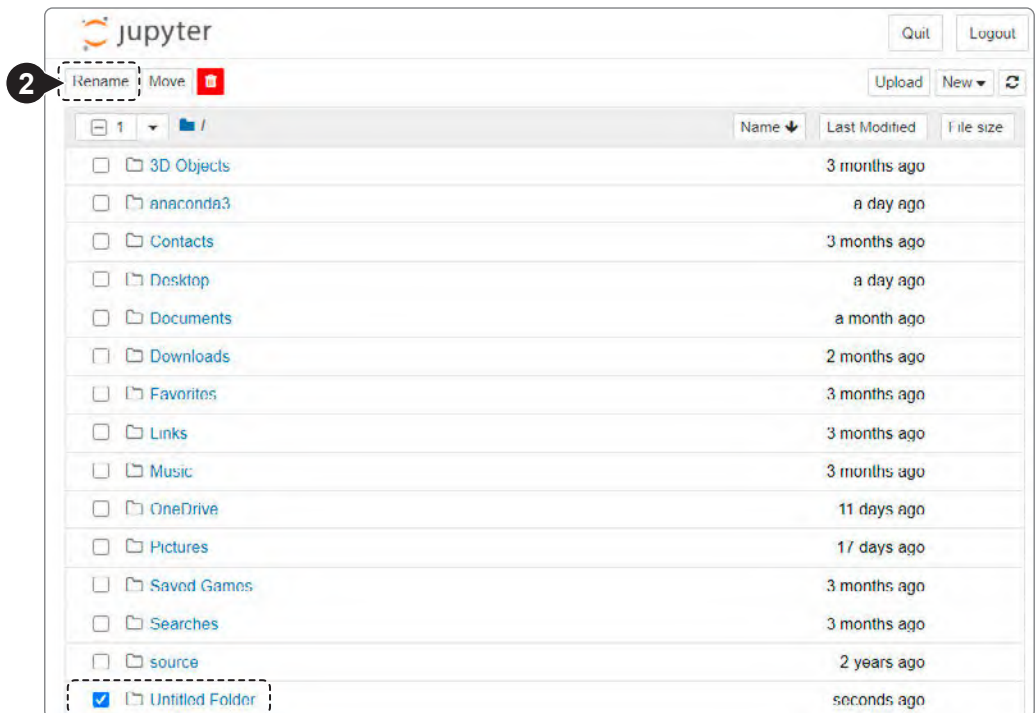
การสร้าง folder เพื่อจัดเก็บโค้ดโปรแกรม

เราสามารถสร้าง folder ใหม่ขึ้นมาใน Jupyter Notebook เพื่อจัดเก็บโค้ดโปรแกรมของเราได้ดังขั้นตอนต่อไปนี้

1. คลิกที่ New แล้วเลือก Folder



2. โฟลเดอร์ใหม่จะถูกสร้างขึ้นมาด้วยชื่อ Untitled Folder ให้คลิกเลือกที่โฟลเดอร์นั้น แล้วคลิกที่ปุ่ม Rename เพื่อเปลี่ยนชื่อโฟลเดอร์



3. กำหนดชื่อโฟลเดอร์ที่ต้องการ และคลิกปุ่ม **Rename**

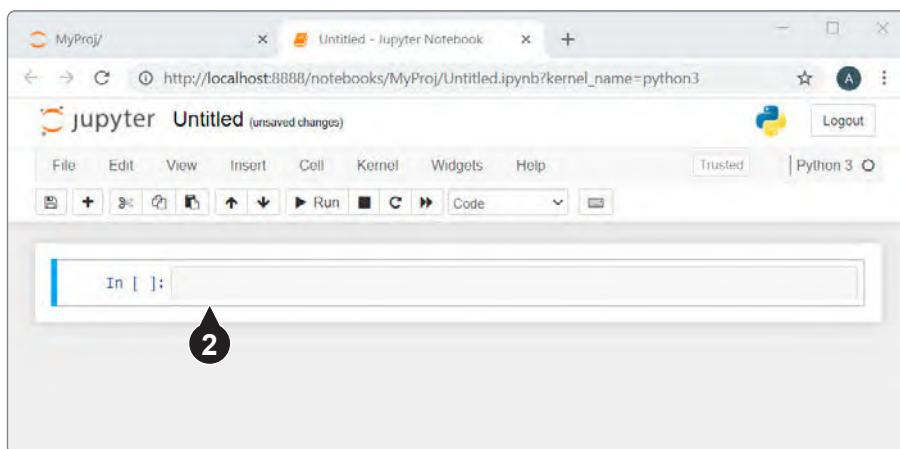


เขียนโปรแกรมไพธอนโดยใช้ Jupyter Notebook

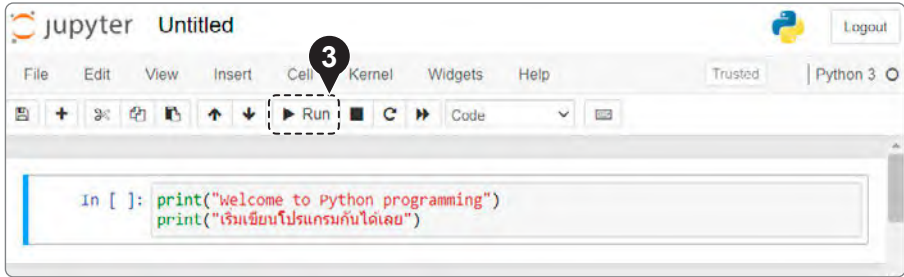
1. หากเราจะเก็บโค้ดโปรแกรมไว้ที่โฟลเดอร์ใด ให้คลิกเข้าไปที่โฟลเดอร์นั้นก่อน (ในที่นี้คลิกเข้าไปที่โฟลเดอร์ **MyProj** ที่สร้างขึ้นมา) จากนั้นคลิกที่ **New** แล้วเลือก **Python3**



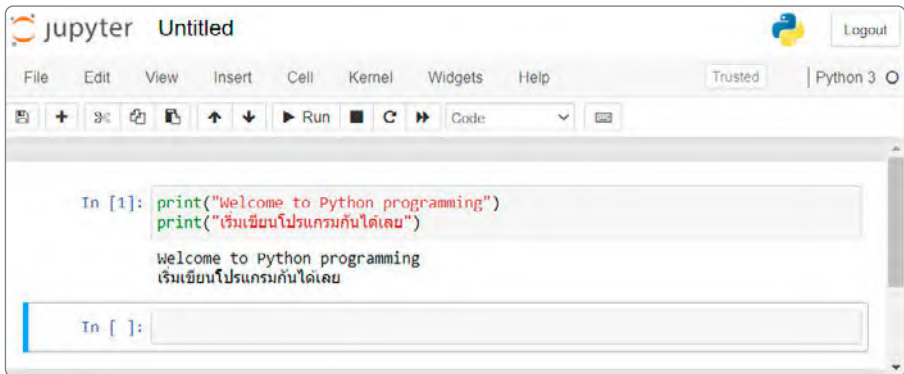
2. จะมีหน้าต่างแสดงขึ้นมาดังรูป เราสามารถเขียนโค้ดโปรแกรมลงไปที่ช่องเซลล์ที่ชื่อว่า **In []:** ได้



3. ให้ลองพิมพ์โค้ดโปรแกรมดังรูป จากนั้นรันเซลล์ปัจจุบันที่เลือกโดยคลิกปุ่ม **Run** (หรือกดคีย์ **Ctrl + Enter** ก็ได้) เพื่อรันดูผลลัพธ์



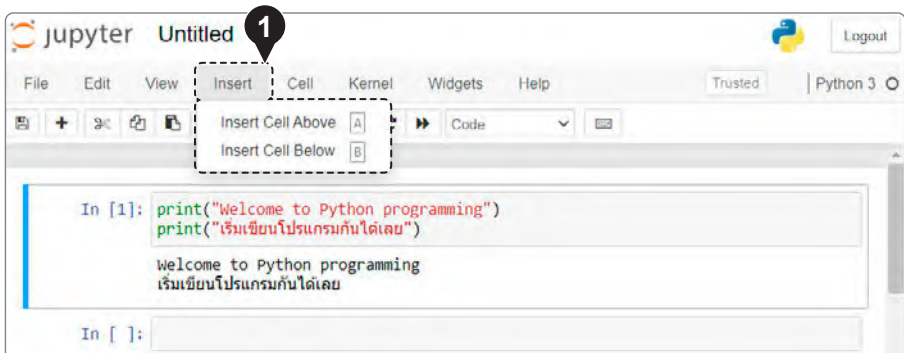
4. เมื่อสั่งรันโปรแกรมแล้ว จะมีผลลัพธ์แสดงออกมา ดังรูป



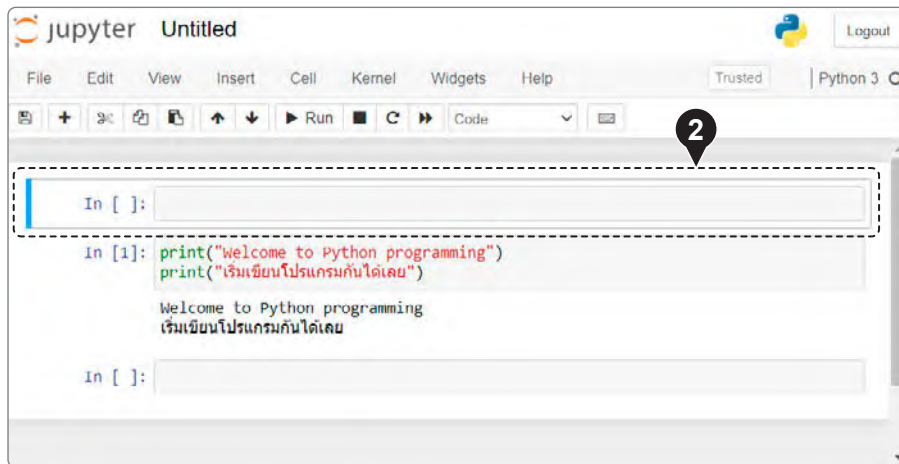
จะพบว่าเมื่อเรารันโค้ดโปรแกรมหนึ่ง ๆ แล้ว จะมีเซลล์ In []: ใหม่แสดงขึ้นมา เราสามารถเขียนโค้ดโปรแกรมอื่นๆ ลงในแต่ละเซลล์เพื่อดูผลลัพธ์ได้ เซลล์แต่ละเซลล์จะแยกการทำงานออกจากกัน

การเพิ่มและลบเซลล์

1. ในกรณีที่ต้องการเพิ่มเซลล์ให้คลิกที่เมนู **Insert**



2. หากต้องการเพิ่มเซลล์ใหม่ให้อยู่ด้านบนของเซลล์ปัจจุบันที่ทำงานอยู่ก็เลือกที่ **Insert Cell Above** จะได้ผลลัพธ์ดังรูป



3. แต่ถ้าต้องการให้เซลล์ใหม่อยู่ด้านล่างของเซลล์ปัจจุบันที่ทำงานอยู่ก็ให้เลือกที่ **Insert Cell Below** หรือจะคลิกปุ่ม **+** ก็ได้ จะได้ผลลัพธ์ดังรูป

