Think Beyond Develop

Think Beyond

#มือใหม่ Python เก่งได้ใน 30 วัน

- พื้นฐาน Python สำหรับนักเรียน นักศึกษา และโปรแกรมเมอร์
- ตัวอย่างโค้ดจริงกว่า 150 ตัวอย่าง พร้อมคำอธิบายโดยละเอียด
- วิธีดึงข้อมูลจากเว็บไซต์ สร้างเกม และเว็บแอปพลิเคชัน
- การนำ Python มาใช้กับ Data Science และ Data Visualization
- วิธีใช้งานโมดูลยอดนิยม เช่น turtle, tkinter, flask, Django, pandas และอื่น ๆ

จีราวุธ วารินทร์

SCAN FREE

ไฟล์ประกอบหนังสือ

จัดจำหน่ายโดย IDC PREMIER

คำนำ

Python คือ ภาษาคอมพิเตอร์ยุคใหม่ที่ถูกออกแบบมาให้ใช้งานได้ง่าย ยืดหยุ่น แต่เปี่ยมไปด้วยประสิทธิภาพ เมื่อดูสถิติการใช้งานภาษาคอมพิวเตอร์ทั่วโลก ปฏิเสธไม่ได้เลยว่า Python เป็นภาษาคอมพิวเตอร์ยอดนิยมอันดับหนึ่งในยุคปัจจุบัน

เพื่อให้ผู้อ่านสามารถเข้าใจภาษา Python ได้อย่างรวดเร็ว หนังสือเล่มนี้จึงแบ่งออกเป็น 5 ส่วน เริ่มจากพื้นฐาน Python จนไปถึงตัวอย่างการนำ Python ไปใช้งานในรูปแบบต่าง ๆ ได้

ส่วนแรก พื้นฐาน Python ที่จำเป็นต้องทราบทั้งหมด เช่น การประกาศตัวแปร การเลือกชนิดข้อมูลที่เหมาะสม วิธีคำนวณ การวนลูป การตัดสินใจ วิธีสร้างและใช้ฟังก์ชัน การกำหนดคลาส การสร้างอ็อบเจกต์ และอื่น ๆ อีกมากมาย หลังจากอ่านส่วนนี้จบจะมีความรู้พื้นฐานเกี่ยวกับ Python ครบทุกแง่มุมแล้ว

ส่วนที่ 2 ได้แสดงตัวอย่างการสร้างส่วนติดต่อกับผู้ใช้ โดยใช้โมดูล turtle และ tkinter เช่น การแสดงข้อความ การกำหนดปุ่ม การวาดรูป วิธีสร้างแบบฟอร์ม รวมถึงตัวอย่างการสร้างเกม หลังจากจบในส่วนนี้จะสามารถสร้างแอปพลิเคชันหรือเกมแบบต่าง ๆ

ส่วนที่ 3 เป็นการนำ Python ไปใช้ดึงข้อมูลจากเว็บไซต์ หรือที่เรียกว่า Web Scraping ซึ่งจะเริ่มด้วยการดึงข้อมูลจากเว็บในแบบง่าย ๆ โดยใช้โมดูล BeautifulSoup ตามด้วยการดึงข้อมูลที่ซับซ้อนขึ้นด้วยโมดูล Selenium

ส่วนที่ 4 เป็นตัวอย่างการนำ Python ไปสร้างเว็บแอปพลิเคชัน โดยใช้โมดูลยอดนิยมอย่าง Flask, Sqlite3 และ Django เมื่ออ่านส่วนนี้จบจะสามารถสร้างเว็บเซิร์ฟเวอร์ อ่านเขียนฐานข้อมูล และสร้างเว็บแอปพลิเคชันในแบบที่ต้องการได้ทันที

ส่วนสุดท้าย เป็นการนำ Python ไปประยุกต์ใช้งานด้านอื่น ๆ เช่น การใช้โมดูล pandas สำหรับจัดการและวิเคราะห์ข้อมูล และใช้โมดูล bokeh เพื่อนำข้อมูลที่ได้ไปพล็อตเป็นกราฟแบบต่าง ๆ

จากเนื้อหาทั้งหมดของหนังสือเล่มนี้จะเน้นทฤษฎีพอสังเขป แต่เน้นการยกตัวอย่างโค้ดจำนวนมากเป็นหลัก ผู้เขียนจึงมั่นใจเป็นอย่างยิ่งว่า ผู้อ่านจะมีความรู้เกี่ยวกับ Python ครบทุกด้าน ในระดับที่สามารถนำไปใช้งานจริงได้อย่างแน่นอน

จีราวุธ วารินทร์

jeerawuth@me.com

สารบัญ

Part 1

Basic

บทที่ 1
รู้จักกับ Python

Python (ไพทอน) เป็นภาษาคอมพิวเตอร์ยอดนิยมที่มีผู้ใช้งานทั่วโลก สามารถนำ Python มาสร้างเป็นแอปพลิเคชันหรือใช้งานได้ในหลายลักษณะ เช่น สร้างเว็บแอปพลิเคชัน นำมาใช้วิเคราะห์ข้อมูล ใช้ในงานด้านวิทยาศาสตร์ ใช้ในงานด้าน AI และ Machine Learning

ในบทแรกนี้จะมาทำความรู้จักกับภาษา Python และวิธีการทดสอบโค้ด Python ผ่านทางเว็บเบราว์เซอร์ ซึ่งวิธีนี้มีข้อดี คือ ไม่ต้องติดตั้งอะไรลงไปในเครื่องคอมพิวเตอร์ก็สามารถทดสอบหรือเรียนรู้การใช้ Python ได้ทันที

ภาษาคอมพิวเตอร์

การสั่งงานคอมพิวเตอร์เพื่อให้คอมพิวเตอร์ทำตามที่ต้องการจะต้องใช้คำสั่งที่คอมพิวเตอร์เข้าใจ ซึ่งเรียกว่า **ภาษาเครื่อง (Machine Code)** แต่ภาษาเครื่องเป็นรหัสตัวเลขฐานสองที่ซับซ้อน ทำให้ยากต่อการใช้งาน ดังนั้นจึงมีการพัฒนา**ภาษาคอมพิวเตอร์ (Computer Language)** ขึ้นมาเพื่อให้สามารถสั่งงานคอมพิวเตอร์ได้ง่ายขึ้น

ภาษาคอมพิวเตอร์ที่ใช้ในปัจจุบันมีอยู่ด้วยกันมากมาย เช่น ภาษา C, Java, Swift และ Python โดยแต่ละภาษาคอมพิวเตอร์จะกำหนดกฎหรือโครงสร้างของภาษาที่แตกต่างกัน ผู้ใช้งานจะต้องทราบว่าการสั่งงานคอมพิวเตอร์จะต้องเขียนอย่างไร เช่น ในภาษา Python หากต้องการสั่งให้พิมพ์ข้อความออกมาที่หน้าจอก็จะใช้คำสั่ง print()

เมื่อต้องการสั่งงานคอมพิวเตอร์จะเขียนชุดคำสั่งโดยใช้ภาษาคอมพิวเตอร์ เช่น Java, Python, Swift โดยชุดคำสั่งที่เขียนขึ้นจากภาษาคอมพิวเตอร์จะถูกเรียกว่า **ซอร์สโค้ด (Source Code)** แต่เนื่องจากคอมพิวเตอร์เข้าใจแต่ภาษาเครื่องเท่านั้น ไม่เข้าใจซอร์สโค้ดที่มนุษย์เขียนขึ้น ดังนั้นจึงต้องมีตัวกลางเพื่อแปลงจากซอร์สโค้ดให้กลายเป็นภาษาเครื่อง เรียกว่า **ตัวแปลภาษา (Translator)**

ตัวแปลภาษา ทำหน้าที่เป็นเหมือนล่าม แปลความหมายของซอร์สโค้ดที่มนุษย์เขียน เพื่อบอกกับคอมพิวเตอร์ว่ามนุษย์ต้องการทำอะไร ตัวอย่างเช่น หากต้องการสั่งให้คอมพิวเตอร์คำนวณหาพื้นที่สามเหลี่ยม ก็จะต้องเขียนซอร์สโค้ดส่งไปให้ตัวแปลภาษาแปลความหมาย เมื่อคอมพิวเตอร์ได้รับคำสั่งก็จะคำนวณพื้นที่สามเหลี่ยมออกมาตามต้องการ

ตัวแปลภาษาที่ใช้ในปัจจุบัน สามารถแบ่งออกเป็น 2 ชนิด ได้แก่

- **อินเทอร์พรีเทอร์ (Interpreter)** จะใช้วิธีแปลคำสั่งทีละชุดคำสั่ง (Statement) เมื่ออ่านซอร์สโค้ดหนึ่งชุดคำสั่ง ก็จะแปลเป็นภาษาเครื่องออกมาเลย ตัวอย่างภาษาคอมพิวเตอร์ที่แปลคำสั่งในแบบอินเทอร์พรีเทอร์ ได้แก่ JavaScript, Ruby และ Python
- **คอมไพลเลอร์ (Complier)** จะใช้วิธีแปลคำสั่งทั้งหมดทีเดียว โดยจะอ่านซอร์สโค้ดจนครบหมดทั้งโปรแกรมก่อน จากนั้นจึงค่อยแปลคำสั่งทั้งหมดให้กลายเป็นภาษาเครื่องในรวดเดียว ตัวอย่างภาษาคอมพิวเตอร์ที่แปลคำสั่งในแบบคอมไพลเลอร์ ได้แก่ ภาษา C++, Java และ Swift

Note

ตัวแปลภาษาของภาษา Python จะมีลักษณะเป็นอินเทอร์พรีเทอร์ ซึ่งใช้วิธีแปลทีละชุดคำสั่ง ซึ่งมีข้อดีคือ ใช้หน่วยความจำน้อยและสามารถตรวจสอบข้อผิดพลาดได้ง่าย

แนะนำ Python

Python (ไพทอน) เป็นภาษาคอมพิวเตอร์ในแบบโอเพนซอร์ส (เปิดเผยโค้ดต่อสาธารณะ) ที่ถูกออกแบบให้เรียบง่าย เช่น ใช้ย่อหน้าเพื่อจัดโครงสร้างของชุดคำสั่ง ใช้คำสั่งที่สั้น ลดการใช้เครื่องหมายหรือสัญลักษณ์พิเศษโดยไม่จำเป็น ทำให้คำสั่งในภาษา Python กระชับ และมีความใกล้เคียงกับภาษาอังกฤษมาก

```
1  class Main:
2      @staticmethod
3      def main(args = []):
4          for k in range(0, 10):
5              print(k)
```

▲ รูปแสดงตัวอย่างโค้ดในภาษา Python

Python ถือว่าเป็นภาษาคอมพิวเตอร์ยอดนิยมในยุคปัจจุบัน ซึ่งมีจุดเด่นที่น่าสนใจ ดังนี้

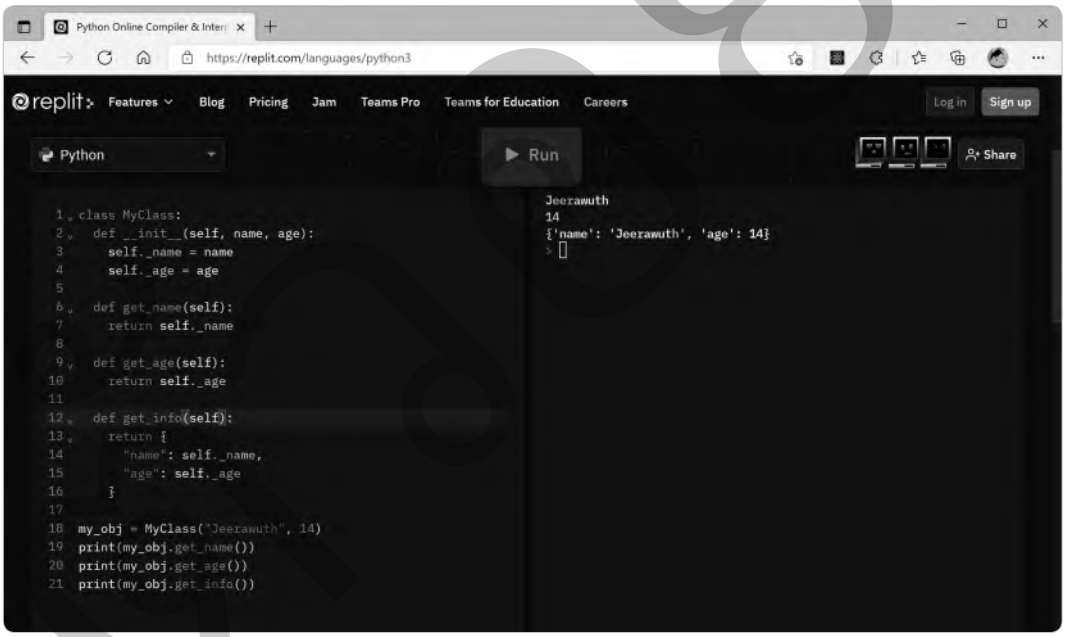
- Python เป็นภาษาคอมพิวเตอร์ที่ถูกออกแบบให้มีโครงสร้างที่ง่าย มีความใกล้เคียงกับภาษาอังกฤษ จึงทำให้เรียนรู้ได้ง่าย เหมาะสำหรับทุกคน ไม่ว่าจะเป็นนักเรียน นักศึกษา โปรแกรมเมอร์ หรือแม้กระทั่งคนทั่วไป
- Python สามารถนำมาใช้งานได้อย่างกว้างขวาง เช่น ใช้สร้างเว็บแอปพลิเคชัน ใช้ในระบบงานวิเคราะห์ข้อมูล ใช้ในงานด้าน IoT (Internet of Things) ใช้ในงานวิทยาศาสตร์ และใช้กับ Machine Learning
- Python มีโมดูลให้เลือกใช้จำนวนมาก ทำให้การสร้างแอปพลิเคชันทำได้ง่ายและรวดเร็ว เนื่องจากสามารถเลือกใช้โมดูลที่คนอื่นพัฒนาไว้แล้วมาประกอบเป็นแอปพลิเคชัน โดยไม่ต้องเขียนชุดคำสั่งเองทั้งหมด

- Python สามารถใช้งานได้กับหลากหลายแพลตฟอร์ม เช่น Windows, Unix/Linux, macOS, iOS, iPadOS, Solaris และ AIX
- Python มีชุมชนขนาดใหญ่ มีผู้ใช้จำนวนมาก ทำให้มีเอกสาร หนังสือ คู่มือ วิดีโอการสอน หรือเว็บบอร์ดจำนวนมาก หากมีปัญหาเกี่ยวกับการใช้งาน Python ก็สามารถโพสต์สอบถามในเว็บบอร์ดต่าง ๆ หรือถ้าต้องการศึกษาความรู้ใหม่ ๆ ก็สามารถทำได้ง่ายเช่นกัน
- Python เป็นโอเพนซอร์สที่ถูกพัฒนาจากทั่วโลก จึงเป็นภาษาคอมพิวเตอร์ที่ถูกพัฒนาให้ดีกว่าเดิมตลอดเวลา ดังนั้น การเลือกใช้ Python ถือว่าเป็นทางเลือกที่มีอนาคตและทันกับเทคโนโลยีที่เปลี่ยนแปลงอย่างรวดเร็วในปัจจุบัน

ทดสอบ Python ในแบบออนไลน์

การศึกษาภาษา Python สามารถทำได้หลายช่องทาง ซึ่งวิธีหนึ่งที่นิยมในปัจจุบัน คือ การศึกษา Python ผ่านระบบออนไลน์ โดยสามารถกรอกชุดคำสั่งภาษา Python ลงในเบราว์เซอร์ ผลลัพธ์การรันชุดคำสั่งก็จะปรากฏบนเบราว์เซอร์ได้ทันที

เว็บไซต์ที่สามารถทดสอบ Python ในแบบออนไลน์มีหลายเว็บไซต์ เช่น replit.com และ online-python.com



▲ รูปแสดงตัวอย่างการทดสอบโค้ด Python บนเบราว์เซอร์

ในบทนี้จะเริ่มต้นศึกษา Python ผ่านทาง Repl (ที่เพจ https://replit.com) โดยเหตุผลที่เลือกทดสอบ Python ผ่านทาง Repl เนื่องจากเป็นระบบออนไลน์ที่ไม่ต้องติดตั้งอะไรให้ยุ่งยาก สามารถบันทึกโค้ดไว้ดูในภายหลัง สามารถแชร์โค้ดคำสั่งไปให้คนอื่น ๆ ได้ และสามารถปรับแต่งคุณสมบัติพื้นฐานบางอย่างได้ เช่น กำหนดระยะแท็บ กำหนดสีของหน้าจอ

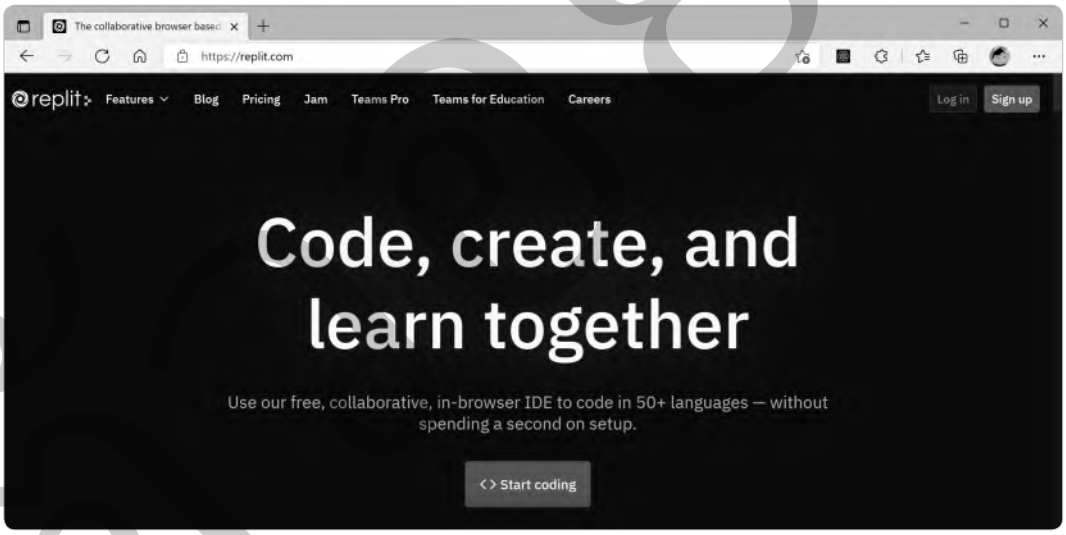
Note

Repl ย่อมาจาก read-eval-print loop เป็น IDE ในแบบออนไลน์ ที่ประกอบไปด้วยคอมไพลเลอร์และอินเทอร์พรีเทอร์ ตลอดจนเครื่องไม้เครื่องมือ สำหรับการพัฒนาแอปพลิเคชันในภาษาต่าง ๆ (รองรับกว่า 60 ภาษา)

ลงทะเบียนสมัคร https://replit.com/

ก่อนทดสอบโค้ด Python กับ Repl จะต้องสมัครสมาชิกก่อน เพื่อให้มีโปรไฟล์สำหรับเก็บโค้ดที่ได้เขียนไว้ และสามารถแชร์โค้ดไปให้สมาชิกคนอื่น ๆ สามารถนำโค้ดไปใช้งานได้

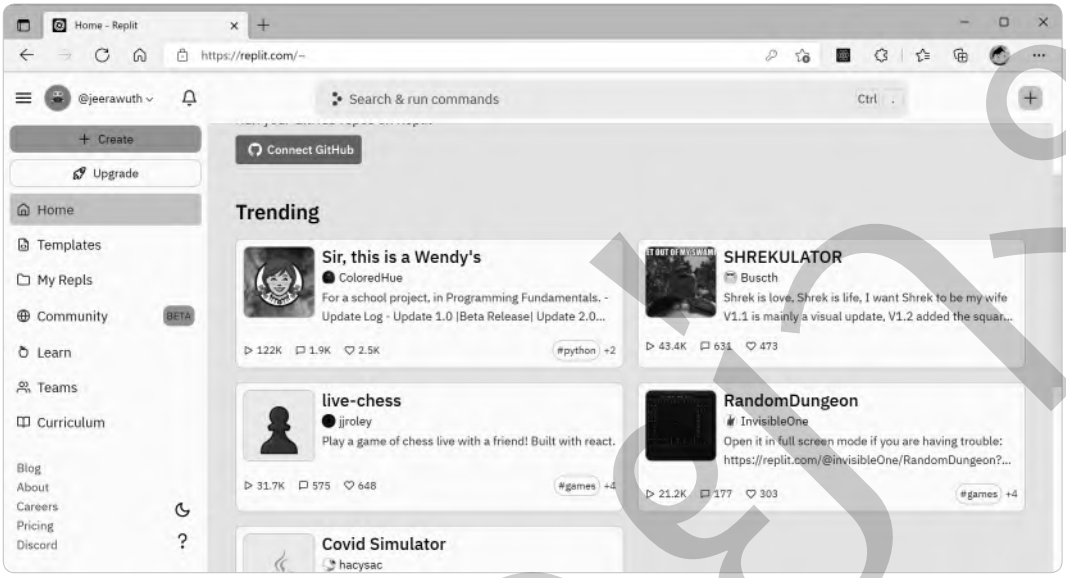
1. เปิดเบราว์เซอร์ และไปยังเว็บไซต์ https://replit.com แล้วคลิกปุ่ม **Sign up** เพื่อสมัครสมาชิก



2. ทำตามขั้นตอนต่าง ๆ ตามที่ทางเว็บไซต์แนะนำ จากนั้นให้เข้าสู่ระบบให้เรียบร้อย

สำรวจหน้าจอโฮมของ Repl

หลังจากลงทะเบียนกับ Repl เรียบร้อย ให้กลับไปยัง https://replit.com อีกครั้ง เมื่อเข้าสู่ระบบก็จะพบหน้าจอโฮม (Home) ดังรูป



จากรูปหน้าจอหลักของ Repl จะมีอยู่ 2 ส่วน คือ

- ด้านซ้ายจะเป็นเมนูคำสั่ง เช่น เมนู Home ใช้แสดงหน้าจอโฮมที่เป็นหน้าจอหลัก เมนู Templates ใช้กำหนดภาษาเริ่มต้นสำหรับโปรเจกต์ใหม่ เมนู My Repls สำหรับแสดงข้อมูลเกี่ยวกับโค้ดที่ได้เขียนเอาไว้ หรือเมนู Learn สำหรับเรียนรู้วิธีใช้งาน Repl
- ด้านขวาจะแสดงเนื้อหาที่เกี่ยวข้องกับเมนูที่ได้เลือกไว้ เช่น เมื่อเลือกเมนู Learn ก็จะพบข้อมูลเกี่ยวกับวิธีใช้งาน หรือคำแนะนำเกี่ยวกับการใช้งาน Repl

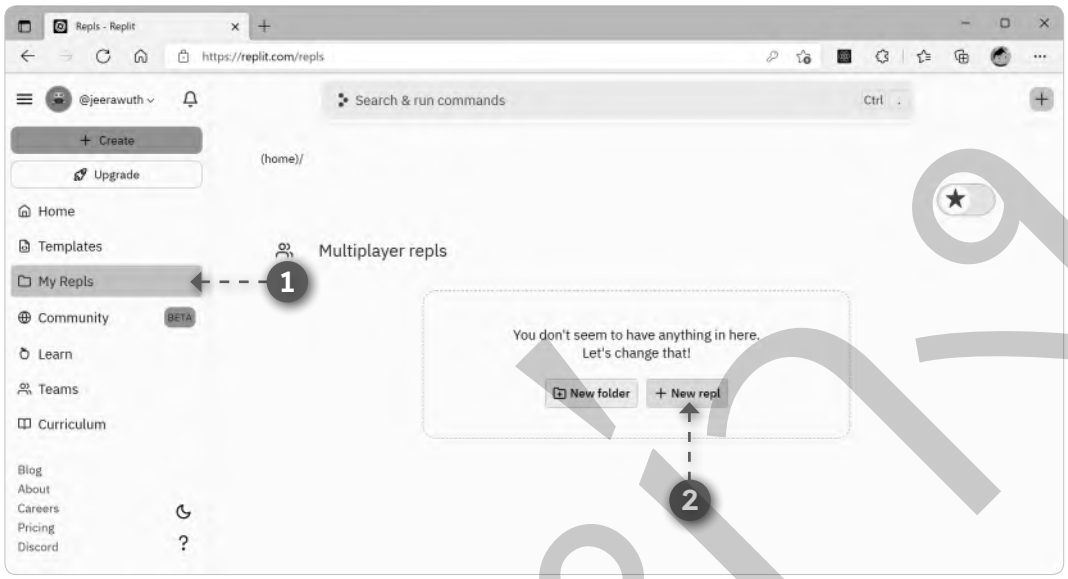
โค้ดแรกกับ Python

คำสั่งแรกของ Python ที่ต้องทราบ คือ คำสั่ง print ใช้เพื่อพิมพ์ข้อความออกมาที่หน้าจอ เช่น เมื่อต้องการพิมพ์ข้อความ "Hello Python" ก็ให้ใช้คำสั่ง print("Hello Python")

เริ่มต้นด้วยการสร้างไฟล์ขึ้นมาใหม่ จากนั้นเลือกภาษาที่ใช้งานเป็น Python และกรอกโค้ดคำสั่งลงไป ดังรายละเอียดต่อไปนี้

1. เปิดเบราว์เซอร์และไปยัง https://replit.com ให้เข้าสู่ระบบ จากนั้นคลิกที่เมนู My Repls เพื่อแสดงพื้นที่ส่วนตัว ซึ่งเก็บโค้ดที่ได้เขียนเอาไว้ (ในการใช้งานครั้งแรกจะเป็นพื้นที่ว่าง ๆ ที่ยังไม่มีข้อมูลอะไรอยู่)

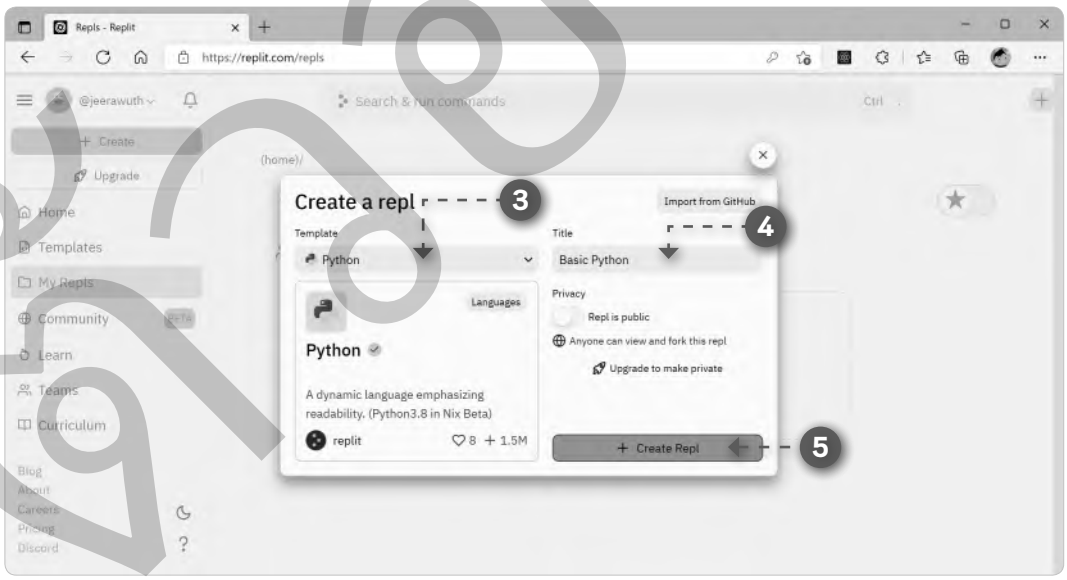
2. คลิกที่ปุ่ม + New repl



3. เลือกเทมเพลตที่จะใช้งาน ให้เลือกเป็น Python

4. ที่ Title สามารถตั้งชื่อได้ตามต้องการ เช่น Basic Python

5. คลิกที่ปุ่ม + Create Repl



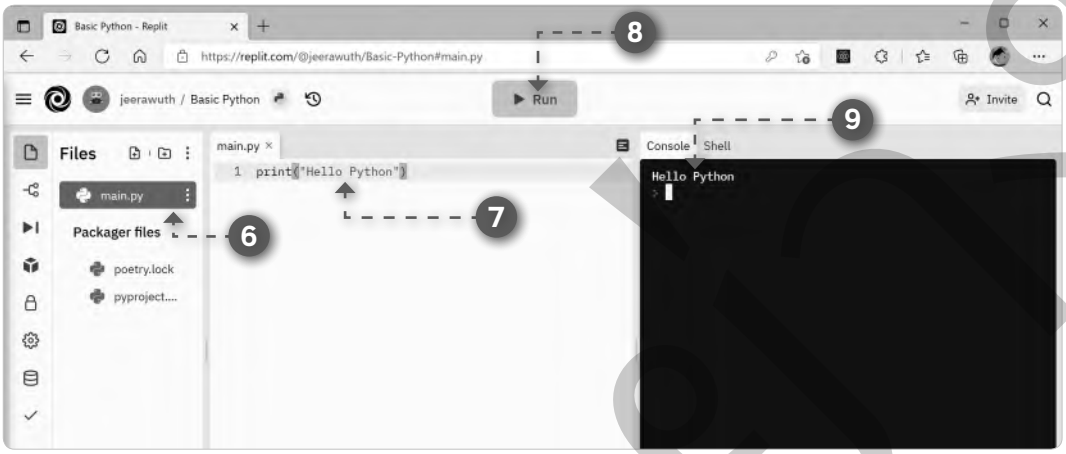
6. ระบบจะสร้างไฟล์ main.py มาให้อัตโนมัติ (นามสกุลไฟล์จะเป็น .py ซึ่งเป็นนามสกุลไฟล์ของ Python)

7. ที่หน้าต่างตรงกลาง (เรียกว่า Code Editor) ให้กรอกโค้ดภาษา Python ดังนี้

```
1  print("Hello Python")
```

8. คลิกที่ Run เพื่อรันโค้ด

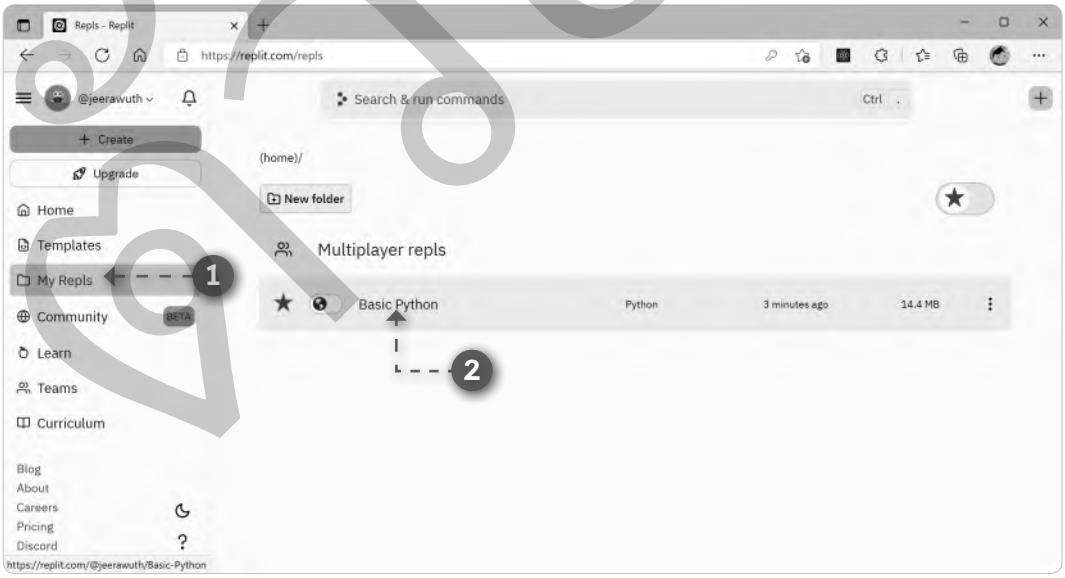
9. ผลลัพธ์คำว่า Hello Python จะถูกแสดงที่หน้าต่างด้านขวามือ (หน้าต่างคอนโซล)



หลังจากได้ทดลองเขียนโค้ด Python กับ Repl แล้ว โค้ดจะถูกบันทึกโดยอัตโนมัติ ในครั้งต่อไปสามารถสร้าง Repl ขึ้นใหม่ หรือเลือก Repl ที่เคยสร้างไว้ก็ได้ ดังนี้

1. คลิกที่ My Repls

2. จะปรากฏรายการ Repl ที่เคยทำเอาไว้ให้คลิกเลือก Repl ที่ต้องการใช้งาน (ในตัวอย่างให้คลิกที่ Basic Python)



ตัวแปร (Variable)

ตัวแปร (Variable) คือ ชื่อที่ใช้อ้างอิงไปยังพื้นที่ในหน่วยความจำ เพื่อใช้งานข้อมูลที่อยู่ในหน่วยความจำนั้น ๆ เช่น กำหนดชื่อตัวแปร PI เพื่อใช้อ้างอิงค่าตัวเลข 3.14159 หรือกำหนดตัวแปร my_box เพื่ออ้างอิงไปยังตัวเลข 10

การกำหนดค่าให้กับตัวแปร จะเริ่มจากกำหนดชื่อตัวแปรไว้ด้านซ้าย ตามด้วยเครื่องหมายเท่ากับ (=) และกำหนดค่าไว้ด้านขวา

```
variable_name = data
```

- **variable_name** คือ ชื่อของตัวแปร
- **data** คือ อ็อบเจกต์ที่ผูก (bind) ไว้กับตัวแปร ซึ่งอาจเป็นข้อความ ตัวเลข หรือข้อมูลรูปแบบอื่น ๆ

ตัวอย่าง 1.1 การประกาศตัวแปร

ตัวอย่างการประกาศตัวแปร (ตั้งชื่อตัวแปร) และกำหนดค่าให้กับตัวแปร

1. ที่หน้าจอ Repl ให้คลิกที่ไฟล์ main.py
2. กรอกโค้ดคำสั่ง ดังนี้

```
1 first_name = "Sombat"
2 age = 14
3 age = 40
```

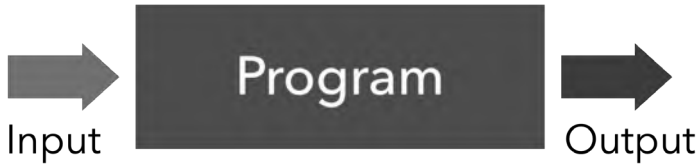
- บรรทัดที่ 1 ประกาศตัวแปรโดยใช้ชื่อว่า first_name และกำหนดค่าไปยังตัวแปร first_name ด้วยข้อความ "Sombat" (กำหนดให้ตัวแปร first_name ชี้ไปยังหน่วยความจำที่เก็บข้อความ "Sombat")
- บรรทัดที่ 2 ประกาศตัวแปรโดยใช้ชื่อว่า age และกำหนดค่าตัวแปร age ด้วยตัวเลข 14 (กำหนดให้ตัวแปร age ชี้ไปยังหน่วยความจำที่เก็บตัวเลข 14)
- บรรทัดที่ 3 กำหนดค่าตัวแปร age ด้วยตัวเลข 40 (กำหนดให้ตัวแปร age ชี้ไปยังหน่วยความจำที่เก็บตัวเลข 40)

Note

การกำหนดค่าให้กับตัวแปรใน Python อาจต่างกับภาษาคอมพิวเตอร์อื่น ๆ โดยค่าที่กำหนดให้กับตัวแปรหมายถึง การกำหนดชื่อตัวแปร เพื่อชี้ไปยังหน่วยความจำที่เก็บอ็อบเจกต์เอาไว้ ตัวอย่างเช่น age = 14 จะหมายถึง การกำหนดชื่อตัวแปรเพื่อใช้อ้างอิงไปยังอ็อบเจกต์ซึ่งเก็บค่า 14 ไม่ใช่การนำตัวเลข 14 ไปกำหนดให้กับตัวแปร age

อินพุตและเอาต์พุต

แอปพลิเคชันหรือโปรแกรมจะมีพื้นฐานที่สำคัญอยู่ 2 ส่วน ได้แก่ อินพุตและเอาต์พุต โดย **อินพุต (Input)** คือ การที่ผู้ใช้งานโปรแกรมส่งข้อมูลบางอย่างเข้าไปยังโปรแกรม เช่น กรอกข้อความ คลิกปุ่ม เลื่อนเมาส์ ส่วน **เอาต์พุต (Output)** คือ ข้อมูลบางอย่างที่ส่งออกมาจากโปรแกรม เช่น การแสดงข้อความ การแสดงปุ่ม รูปภาพ



▲ รูปแสดงอินพุตและเอาต์พุต

ในการพัฒนาโปรแกรมจะต้องกำหนดอินพุตและเอาต์พุตรูปแบบต่าง ๆ เพื่อใช้ติดต่อสื่อสารกับผู้ใช้ ซึ่งอินพุตขั้นพื้นฐาน คือ การพิมพ์ข้อความโดยใช้คีย์บอร์ดส่งเข้าไปยังโปรแกรม ส่วนเอาต์พุตขั้นพื้นฐาน คือ การแสดงข้อความบางอย่างออกมาที่หน้าจอ

แสดงข้อความเอาต์พุตด้วยคำสั่ง print

ใน Python หากต้องการแสดงข้อความออกมาที่หน้าจอ (เป็นการกำหนดเอาต์พุต) จะใช้คำสั่ง print ซึ่งการใช้ print ใน Python มีรูปแบบพื้นฐานดังนี้

```
print(*value)
```

- **print** คือ ฟังก์ชันที่ใช้แสดงข้อความออกมาที่หน้าต่างคอนโซล (เป็นเอาต์พุต)
- **value** คือ ข้อความ (หรือค่าจากตัวแปร) ที่ต้องการนำมาแสดงบนหน้าจอ โดยข้อความจะต้องอยู่ภายในเครื่องหมาย " " (เรียกว่า Double quote) เช่น "Hello World" หรืออยู่ภายในเครื่องหมาย ' ' (เรียกว่า Single quote) เช่น 'Hello World'

ตัวอย่าง 1.2 การใช้งานฟังก์ชัน print

หากต้องการแสดงข้อความว่า Good morning teacher. ก็สามารถใช้คำสั่ง print ได้ดังนี้

1. ที่หน้าจอ Repl ให้คลิกที่ไฟล์ main.py
2. กรอกโค้ดคำสั่ง ดังนี้

```
1 print("Good morning teacher.")
2 print('Good afternoon teacher.')
```

- บรรทัดที่ 1 ใช้คำสั่ง print แสดงข้อความ "Good morning teacher." สังเกตว่าใช้เครื่องหมาย " " ครอบข้อความ
- บรรทัดที่ 2 ใช้คำสั่ง print แสดงข้อความ 'Good afternoon teacher.' สังเกตว่าสามารถใช้เครื่องหมาย ' ' ครอบข้อความ แทนการใช้ " " ก็ได้

3. คลิกปุ่ม Run

4. พบข้อความ 2 บรรทัด เนื่องจากมีการเรียกใช้คำสั่ง print จำนวน 2 ครั้ง

รับข้อความที่เป็นอินพุตด้วยคำสั่ง input

หากต้องการส่งอินพุตที่เป็นข้อความเข้าไปยังโปรแกรมจะใช้ฟังก์ชัน input ซึ่งการใช้ input ใน Python มีรูปแบบพื้นฐานดังนี้

```
variable = input(text)
```

- **variable** คือ ตัวแปรที่ใช้เก็บข้อความซึ่งผู้ใช้ได้กรอกลงไป
- **input** คือ ฟังก์ชันที่ใช้รับข้อความจากคีย์บอร์ดเพื่อนำไปใช้ในโปรแกรม
- **text** คือ ข้อความที่ต้องการนำมาแสดงบนหน้าจอ เพื่อบอกจุดประสงค์ว่าจะนำข้อความที่ผู้ใช้กรอกไปทำอะไร โดยข้อความจะต้องอยู่ภายในเครื่องหมาย Double quote หรือ Single quote เช่น "กรุณากรอกอายุ" หรือ 'กรุณากรอกอายุ' ตามลำดับ

ตัวอย่าง 1.3
การใช้งานฟังก์ชัน input

ตัวอย่างการใช้งานฟังก์ชัน input เช่น สั่งให้แสดงข้อความถามชื่อผู้ใช้ว่า "กรุณากรอกชื่อ " เมื่อผู้ใช้กรอกชื่อแล้วกด <Enter> ชื่อของผู้ใช้ก็จะถูกนำไปผูกกับตัวแปร your_name

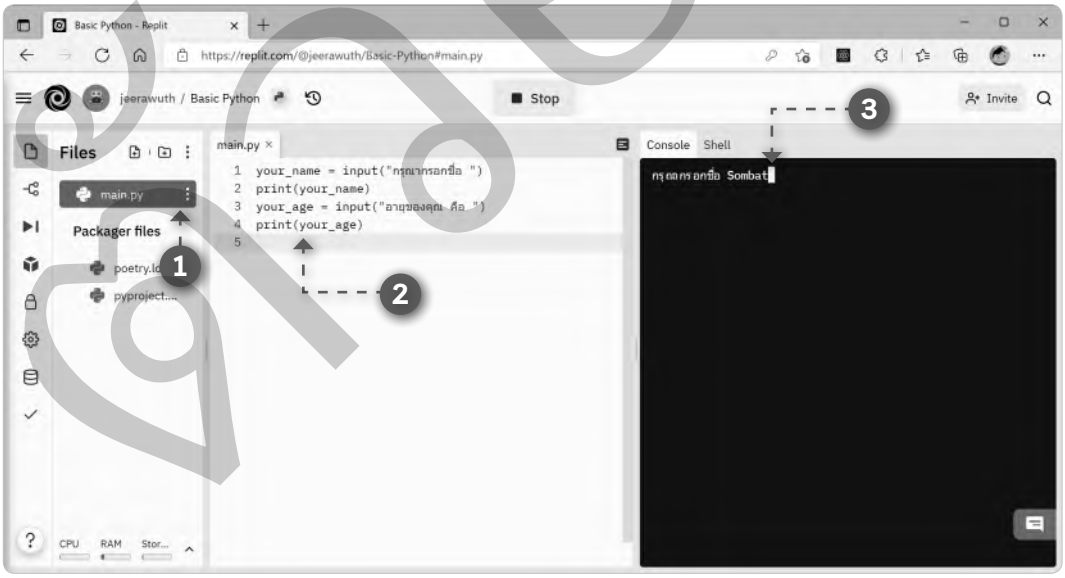
1. ที่หน้าจอ Repl ให้คลิกที่ไฟล์ main.py

2. กรอกโค้ดคำสั่งดังนี้

```
1 your_name = input("กรุณากรอกชื่อ ")
2 print(your_name)
3 your_age = input("อายุของคุณ คือ ")
4 print(your_age)
```

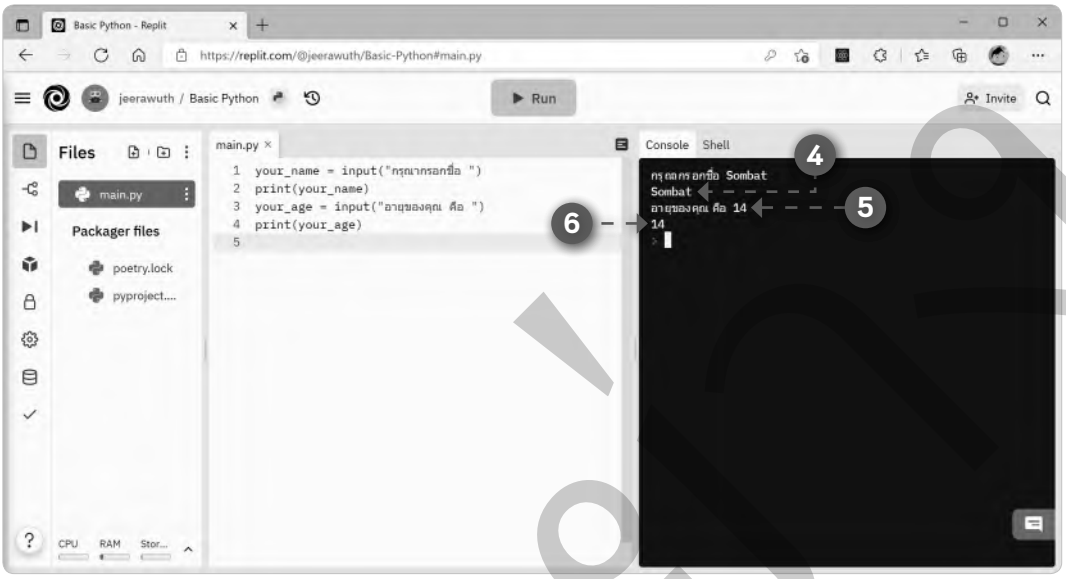
- บรรทัดที่ 1 ใช้คำสั่ง input รับข้อความจากผู้ใช้ เมื่อผู้ใช้กรอกชื่อแล้วกด <Enter> ชื่อที่เป็นข้อความจะถูกนำมาเก็บยังตัวแปร your_name
- บรรทัดที่ 2 ใช้คำสั่ง print แสดงข้อความที่อยู่ในตัวแปร your_name ออกมา
- บรรทัดที่ 3 ใช้คำสั่ง input รับข้อความจากผู้ใช้ เมื่อผู้ใช้กรอกอายุแล้วกด <Enter> อายุที่ผู้ใช้กรอกจะถูกนำมาเก็บยังตัวแปร your_age สังเกตว่า แม้ว่าอายุจะเป็นตัวเลข แต่การกรอกตัวเลขด้วยคีย์บอร์ดตัวเลขนี้จะถือว่าเป็นข้อความชนิดหนึ่ง (ไม่ใช่ตัวเลขที่จะนำมาคำนวณค่าได้)
- บรรทัดที่ 4 ใช้คำสั่ง print แสดงข้อความที่อยู่ในตัวแปร your_age ออกมา

3. คลิกปุ่ม Run จะปรากฏข้อความ "กรุณากรอกชื่อ " ให้กรอกชื่อ (เช่น Sombat) แล้วกด <Enter>



4. ชื่อที่ผู้ใช้กรอกจะถูกนำมาแสดงที่หน้าจอ (ในตัวอย่างแสดงข้อความ Sombat)

5. จะปรากฏข้อความ "อายุของคุณ คือ " ให้กรอกอายุ (เช่น 14) แล้วกด <Enter>
6. อายุที่ผู้ใช้กรอกจะถูกนำมาแสดงที่หน้าจอ (ในตัวอย่างคือเลข 14)



เชื่อมข้อความด้วยเครื่องหมายบวก (+)

เมื่อต้องการนำข้อความมาต่อกันสามารถใช้เครื่องหมายบวก (+) เชื่อมข้อความเข้าด้วยกัน

```
text1 + text2
```

- **text1** คือ ข้อความในส่วนแรก
- **text2** คือ ข้อความในส่วนที่สอง
- **text1** + **text2** คือ การนำข้อความจาก text1 มาเรียงต่อกับข้อความใน text2

ตัวอย่างการเชื่อมระหว่างข้อความ "My name is: " กับข้อความที่อยู่ในตัวแปร my_name

```
1 my_name = "Lisa"
2 print("My name is: " + my_name)
```

- บรรทัดที่ 1 ประกาศตัวแปร my_name เก็บข้อความ "Lisa"
- บรรทัดที่ 2 นำข้อความ "My name is: " มาต่อกับข้อความที่อยู่ในตัวแปร my_name

Note

ข้อความที่ใช้ในชุดคำสั่งของ Python จะต้องอยู่ภายใต้เครื่องหมาย " " เช่น "Hello Python" หรืออยู่ภายใต้เครื่องหมาย ' ' ก็ได้ เช่น 'Hello Python' ส่วนชื่อตัวแปรไม่ต้องอยู่ภายใต้เครื่องหมาย " " หรือ ' ' เช่น my_name

ความสำคัญของการเยื้องใน Python

การเขียนโค้ดโดยใช้ภาษา Python จะไม่มีการใช้เครื่องหมายปีกกาหรือเครื่องหมายอื่น ๆ เพื่อกำหนดขอบเขตของบล็อกของคำสั่งเหมือนกับบางภาษา (เช่น ภาษา C, Java หรือ PHP) แต่ Python จะใช้การเยื้องเพื่อบอกขอบเขตในแต่ละบล็อกแทน ดังนั้น การเยื้อง (Indentation) ในภาษา Python ถือว่ามีความสำคัญมาก หากเยื้องไม่ถูกต้องก็จะเกิด error ขึ้นทันที ดังนี้

```
1  print("The first indent")
2  print("The second indent")
3   print("The third indent")
```

- บรรทัดที่ 1 ใช้คำสั่ง print เพื่อแสดงข้อความออกมาที่หน้าจอ สังเกตว่า คำสั่ง print จะอยู่ชิดกับขอบด้านซ้าย (ไม่มีการเว้นวรรค ไม่มีการเยื้องเมื่อนับจากขอบด้านซ้าย)
- บรรทัดที่ 2 ใช้คำสั่ง print เพื่อแสดงข้อความออกมาที่หน้าจอ สังเกตว่า คำสั่ง print จะอยู่ชิดกับขอบด้านซ้าย ซึ่งเป็นแนวเดียวกับคำสั่งในบรรทัดที่ 1
- บรรทัดที่ 3 ใช้คำสั่ง print เพื่อแสดงข้อความออกมาที่หน้าจอ สังเกตว่า คำสั่ง print จะขยับจากขอบด้านซ้ายมา 1 เคาะ ซึ่งบรรทัดนี้จะเกิด error เนื่องจากเยื้องไม่ตรงกับคำสั่งอื่น ๆ ที่อยู่ในบล็อกเดียวกัน

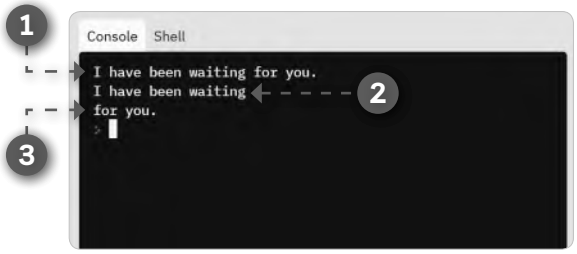
ขึ้นบรรทัดใหม่_ด้วย \n

การใช้คำสั่ง print เพื่อแสดงข้อความออกมาบนหน้าจอนั้น หากต้องการแสดงข้อความในหลายบรรทัดจะใช้ \n แทรกไปยังตำแหน่งที่ต้องการขึ้นบรรทัดใหม่ ดังนี้

```
1 first_sentence = "I have been waiting for you."
2 second_sentence = "I have been waiting \nfor you."
3 print(first_sentence)
4 print(second_sentence)
```

- บรรทัดที่ 1 ประกาศตัวแปร first_sentence เก็บข้อความ
- บรรทัดที่ 2 ประกาศตัวแปร second_sentence เก็บข้อความ สังเกตว่าข้อความจะคล้ายกับบรรทัดที่ 1 แต่มีสิ่งที่แตกต่าง คือ ได้แทรก \n นำหน้าข้อความ "for you." เป็นการบังคับให้ขึ้นบรรทัดใหม่ในตำแหน่งนี้
- บรรทัดที่ 3 ใช้คำสั่ง print แสดงข้อความที่อยู่ในตัวแปร first_sentence
- บรรทัดที่ 4 ใช้คำสั่ง print แสดงข้อความที่อยู่ในตัวแปร second_sentence

เมื่อรันคำสั่งตามตัวอย่างข้างต้น จะพบข้อความจำนวน 3 บรรทัด ดังรูป



1. ข้อความ I have been waiting for you. เกิดจากคำสั่งในบรรทัดที่ 3
2. ข้อความ I have been waiting เกิดจากคำสั่งในบรรทัดที่ 4
3. ข้อความ for you. จะถูกบังคับให้ขึ้นบรรทัดใหม่ เกิดจากคำสั่ง \n ในบรรทัดที่ 2

เครื่องหมาย " " หรือ ' ' ใช้แบบไหนดี

การกำหนดข้อความให้กับตัวแปรและผ่านค่าข้อความไปยังฟังก์ชัน input และ print จะต้องอยู่ภายใต้เครื่องหมาย " " (Double quote) หรืออยู่ภายใต้เครื่องหมาย ' ' (Single quote) เสมอ

สิ่งที่ต้องระวัง คือ หากภายในข้อความมีการใช้เครื่องหมาย ' ' หรือ " " แทรกอยู่ ก็อาจเกิด error ขึ้นได้ เนื่องจาก Python ไม่ทราบว่าข้อความนั้นมีจุดเริ่มต้นและจุดสิ้นสุดที่ตรงไหน ดังตัวอย่างนี้

ตัวอย่าง 1.4
เมื่อต้องแสดงข้อความ She said: "Love me love my dog."

สังเกตว่าข้อความนี้มีการใช้เครื่องหมาย " " แทรกอยู่ ดังนั้นหากใช้เครื่องหมาย " " ครอบข้อความนี้ก็จะเกิด error เนื่องจาก Python ไม่สามารถกำหนดจุดเริ่มต้นหรือจุดสิ้นสุดของข้อความได้อย่างถูกต้อง

```
1  sentence_a = 'She said: "Love me love my dog."'
2  sentence_b = "She said: "Love me love my dog.""
```

- บรรทัดที่ 1 ประกาศตัวแปร sentence_a เก็บข้อความ โดยใช้เครื่องหมาย ' ' ครอบข้อความ บรรทัดนี้ใช้งานได้ตามปกติ เพราะ Python สามารถระบุขอบเขตของข้อความที่จะกำหนดไปยังตัวแปร sentence_a ได้อย่างถูกต้อง
- บรรทัดที่ 2 ประกาศตัวแปร sentence_b เก็บข้อความ โดยใช้เครื่องหมาย " " ครอบข้อความ ในบรรทัดนี้จะเกิด error ขึ้น เนื่องจากภายในข้อความที่กำหนดนั้นมีเครื่องหมาย " " ทำให้ไม่สามารถกำหนดขอบเขตของข้อความในส่วนนี้ได้

สาเหตุที่ทำให้ Python ไม่สามารถกำหนดข้อความได้อย่างถูกต้อง เนื่องจาก Python ไม่ทราบว่าส่วนไหนคือคำสั่ง ส่วนไหนคือส่วนที่เป็นข้อความนั่นเอง ดังรูป

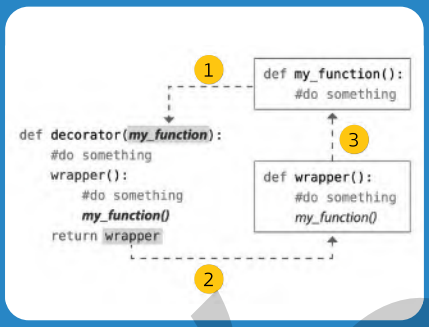
Think Beyond Develop

30 วันก็เก่งได้

ด้วยเนื้อหาที่กระชับ
ปรับแต่งให้เข้าใจได้ง่าย
อ่านและทำตามได้แบบ

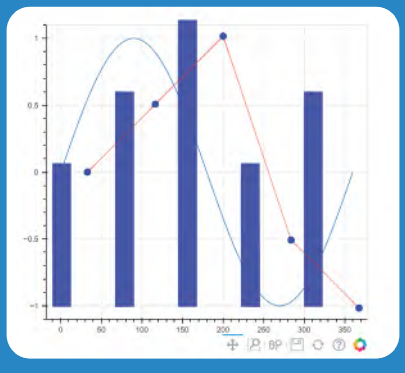
Step-By-Step

เก่งได้ แม้ไม่มีพื้นฐานมาก่อน



- สรุปพื้นฐาน Python ที่จำเป็นต้องทราบ
- อธิบาย Python ด้วยโค้ดตัวอย่าง พร้อมคำอธิบายในทุกขั้นตอน
- การใช้งาน Python Shell, PyCharm และ Visual Studio Code
- พื้นฐานเกี่ยวกับ Scope, Closure และ Decorator
- วิธีสร้างอ็อบเจกต์ การใช้งานเมธอด แอตทริบิวต์ และพรอปเพอร์ตี
- พื้นฐานการใช้งานฐานข้อมูล พร้อมตัวอย่างแอปพลิเคชัน
- การกำหนด Virtual Environment สำหรับแต่ละโปรเจกต์
- วิธีสร้างส่วนติดต่อกับผู้ใช้ เช่น การสร้างแบบฟอร์ม และการวาดรูปกราฟิก

- ตัวอย่างการสร้างเกมด้วยโมดูล turtle พร้อมคำอธิบายแต่ละส่วนโดยละเอียด
- วิธีดึงข้อมูลจากเว็บไซต์ (Web Scraping) และตัวอย่างการใช้งาน
- พื้นฐานการสร้างเว็บแอปพลิเคชันในฝั่งเซิร์ฟเวอร์ด้วย flask
- พื้นฐานการสร้างเว็บแอปพลิเคชันโดยใช้ Django
- การใช้งานโมดูล pandas จัดการกับข้อมูลจำนวนมาก
- วิธีอ่านและเขียนไฟล์ Excel, JSON และ CSV
- การนำข้อมูลจากไฟล์ Excel มาพล็อตเป็นกราฟแบบต่าง ๆ
- ตัวอย่างการเข้ารหัส (Encrypt) และถอดรหัสข้อมูล (Decrypt)
- ตัวอย่างการอ่านเขียนข้อมูลลงใน Google Sheet



ISBN(eBook) 885-909-931-021-5

ราคา 550 บาท