



# ICT12367

การใช้กรอบงานสำหรับการพัฒนาเว็บแอปพลิเคชัน  
เพื่อความมั่นคงปลอดภัย

## Chapter 6

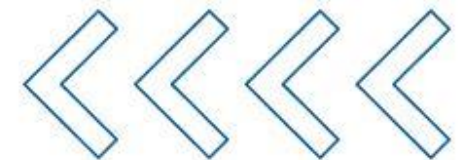
Django Framework



# Django Framework คืออะไร



Django Framework คือ Web Framework สำหรับสร้าง Web Application ฝั่ง Back End ที่พัฒนาด้วยภาษา Python

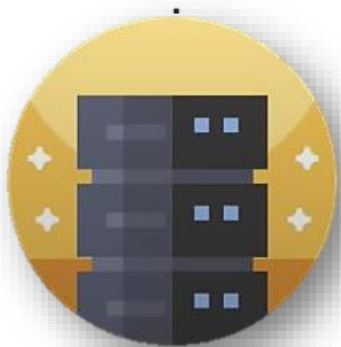




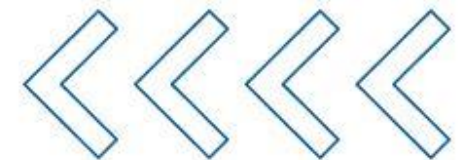
# Frontend VS Backend



**Frontend** คือ การพัฒนาโปรแกรมระบบหน้าบ้าน (UI: User Interface หรือหน้าต่างของแอปพลิเคชัน) โดยผู้ใช้งานสามารถมองเห็นและมีส่วนร่วมหรือโต้ตอบภายใน Web Browser ได้



**Backend** คือ การพัฒนาโปรแกรมระบบหลังบ้าน หรือการทำงานเบื้องหลังในแอป เช่น ทำงานกับฐานข้อมูล เป็นต้น โดยผู้ใช้งานไม่สามารถมีส่วนร่วมหรือโต้ตอบได้

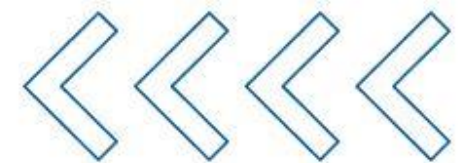




# จุดเด่นของ Django Framework



- ใช้ภาษาไพธอนในการพัฒนา มีรูปแบบการเขียนโค้ดที่ไม่ซับซ้อนและเข้าใจง่าย อีกทั้งยังเป็นภาษาที่ได้รับความนิยมในปัจจุบัน
- แยกการทำงานออกเป็นส่วนย่อยๆ โดยใช้สถาปัตยกรรม **MVT(Model-View-Template)**
- ใช้งานได้อิสระ เราสามารถปรับปรุงแก้ไข หรือ กำหนดวิธีการทำงานใหม่ได้ตามความต้องการ





# เครื่องมือที่ใช้

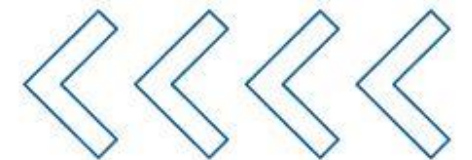


## Frontend (ระบบหน้าบ้าน)

- HTML
- CSS
- JavaScript
- Bootstrap
- Python
- SQL

## Backend (ระบบหลังบ้าน)

- Django Framework
- Database
  - MySQL





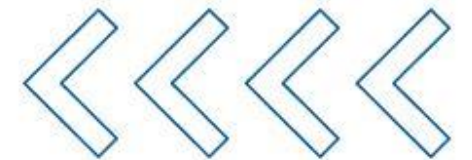
# ກາຟໂວວໂມຣະບຸ



**Frontend (Client)**

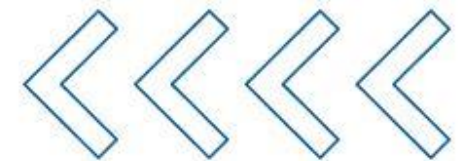
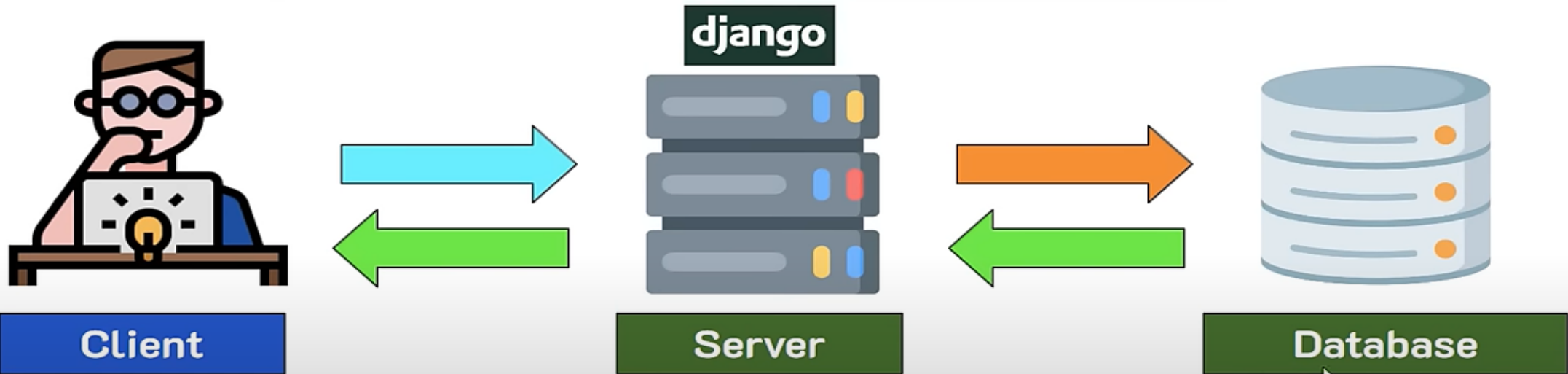


**Backend (Server)**





# ภาพรวมระบบ

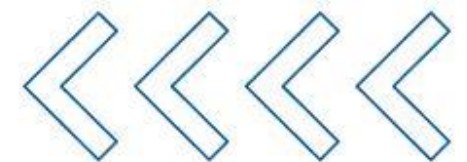






# คำศัพท์พื้นฐาน

- **Client (ผู้ใช้บริการ)**
- **Server (ผู้ให้บริการ)**
- **Request (คำขอในการเข้าถึง)**
- **Response (ตอบกลับคำขอ)**



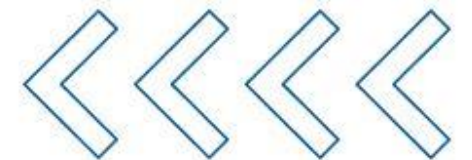




# เครื่องมือที่ใช้เขียนโปรแกรม

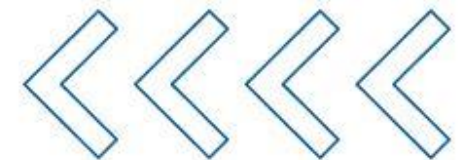
## Frontend (ระบบหน้าบ้าน)

- Visual Studio Code
- Python & Django Framework
- XAMPP & MySQL





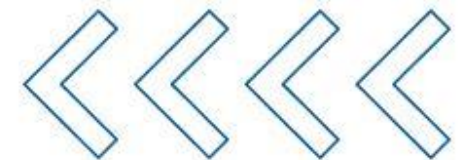
**(programming language)**





# Python คืออะไร

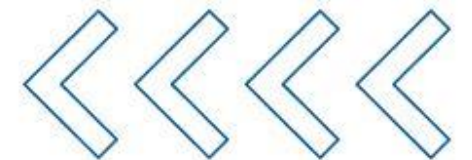
python คือภาษาคอมพิวเตอร์ที่ใช้สำหรับการพัฒนาโปรแกรม โดยผู้พัฒนาต้องเรียนรู้โครงสร้างภาษา Python ว่ามีโครงสร้างการเขียนอย่างไร เพื่อสั่งการให้คอมไพเตอร์นั้นสามารถทำงานตามวัตถุประสงค์ที่ต้องการได้ โดยการสั่งการให้คอมไพเตอร์ทำงานทำงานได้นั้นต้องอาศัยส่วนที่เรียกว่า ตัวแปรภาษา





# ตัวแปลภาษา

ตัวแปลภาษาเปรียบเสมือนกับล่าม ทำหน้าที่แปลงโค้ดภาษาคอมพิวเตอร์ที่มนุษย์เขียนขึ้น (ภาษา ) ไปเป็นภาษาที่เครื่องคอมพิวเตอร์เข้าใจว่าต้องการให้ทำงานอะไร

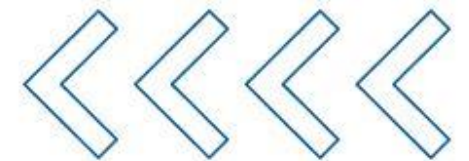
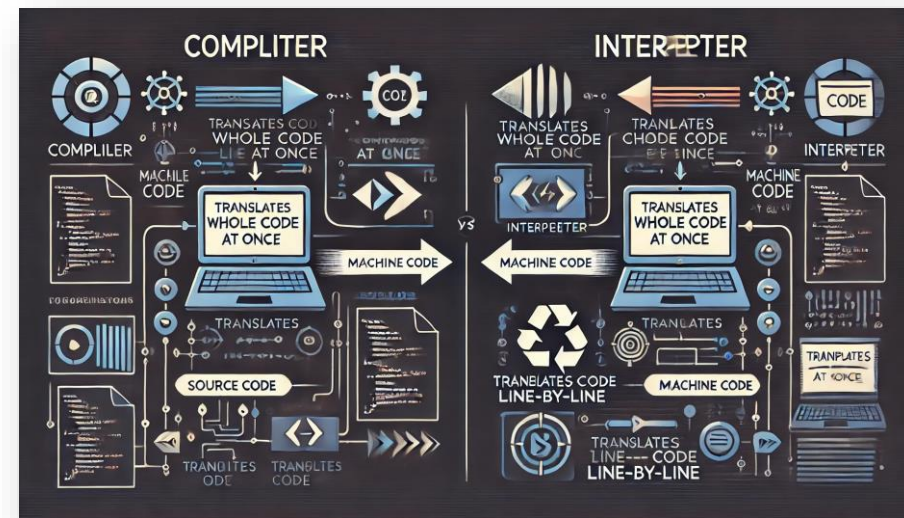




# ประเภทของตัวแปลภาษา

ในปัจจุบันตัวแปลภาษาแบ่งออกเป็น 2 ประเภท

- คอมไพเลอร์ (Compiler)
- อินเตอร์พรีเตอร์ (Interpreter)

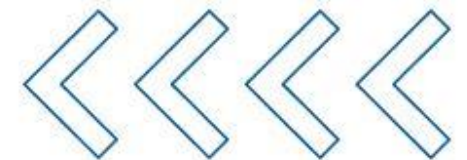




# ข้อดีและข้อเสียของตัวแปลภาษา



	ข้อดี	ข้อเสีย
Compiler	<ul style="list-style-type: none"><li>ทำงานได้เร็ว เนื่องจากจะทำการแปลคำสั่งทั้งหมดในครั้งเดียว แล้วจึงทำงานตามคำสั่งของโปรแกรมในภายหลัง</li></ul>	<ul style="list-style-type: none"><li>เมื่อเกิดข้อผิดพลาดขึ้นจะตรวจสอบหาข้อผิดพลาดได้ยาก เพราะทำการแปลคำสั่งทีเดียวทั้งโปรแกรม</li></ul>
Interpreter	<ul style="list-style-type: none"><li>แปลคำสั่งทีละบรรทัด ทำให้หาข้อผิดพลาดของโปรแกรมได้ง่าย</li><li>เนื่องจากแปลคำสั่งทีละบรรทัด สามารถสั่งให้โปรแกรมทำงานเฉพาะจุดได้</li><li>ไม่เสียเวลารอการแปลคำสั่งเป็นเวลานาน</li></ul>	<ul style="list-style-type: none"><li>ช้า เนื่องจากทำงานทีละบรรทัด</li></ul>

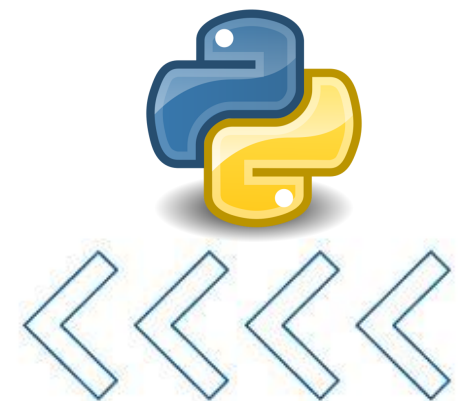






# ข้อดีของภาษา Python

- ทำงานได้หลายระบบปฏิบัติการ (Windows, Mac, Linux)
- เป็น Open Source (ใช้งานได้ฟรี ไม่มีค่าใช้จ่าย)
- โครงสร้างไวยากรณ์ภาษา เข้าใจง่าย
- รองรับการเขียนโปรแกรมเชิงวัตถุ (OOP)
- มีไลบรารีให้ใช้งานจำนวนมาก (เว็บ, เกม, กราฟิก, AI, ML)
- รองรับการทำงานกับฐานข้อมูล (Database)

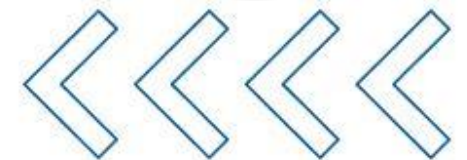
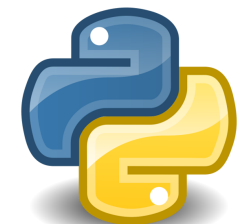






# ตัวอย่างการประยุกต์ใช้งาน

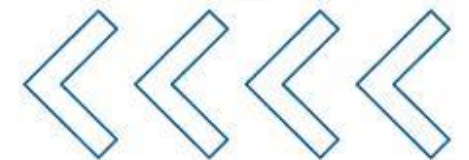
- สามารถพัฒนาแอปพลิเคชันในรูปแบบ GUL ได้ (Tkinter, PyQt)
- สามารถพัฒนาเว็บแอปพลิเคชันได้ (Django Framework, Flask)
- สามารถพัฒนาเกมได้ (Pygame)
- Data Science (Numpy, Pandas, Matplotlib)
- Machine Learning & AI





# ตัวอย่างการประยุกต์ใช้งาน (ต่อ)

- งานด้านการประมวลผลภาพ (Image Processing)
- การพัฒนาระบบแผนที่
- การสกัดข้อมูลจากเว็บไซต์ (Web Scrapping)
- การพัฒนา API (Fast API, Django REST Framework)
- ทำงานร่วมกับระบบจัดการฐานข้อมูล (SQL, NOSQL)

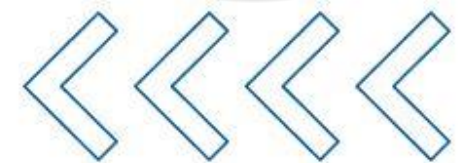




# องค์ประกอบพื้นฐานของ ภาษา Python



- ไฟล์ที่เก็บโค้ดภาษา Python (Source Code) จะมีนามสกุลไฟล์ **.py**
- การย่อหน้าโค้ดโปรแกรม (Indentation) คือการจัดรูปแบบคำสั่งในภาษา Python โดยแบ่งโค้ดออกเป็นกลุ่มย่อยๆ ซึ่งแต่ละกลุ่ม **จะต้องจัดย่อหน้าให้เท่ากัน** ไม่เช่นนั้นจะเกิดข้อผิดพลาดและไม่สามารถรันโปรแกรมได้





# ตัวอย่างการย่อหน้า

ต่างกันอย่างไร

**If username == "admin":**

**Print("สวัสดีผู้ดูแลระบบ")**

**else:**

**print("ข้อมูลไม่ถูกต้อง")**

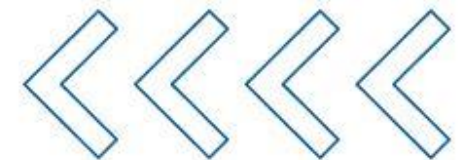


**If username == "admin":**

➡ **print("สวัสดีผู้ดูแลระบบ")**

**else:**

➡ **print("ข้อมูลไม่ถูกต้อง")**



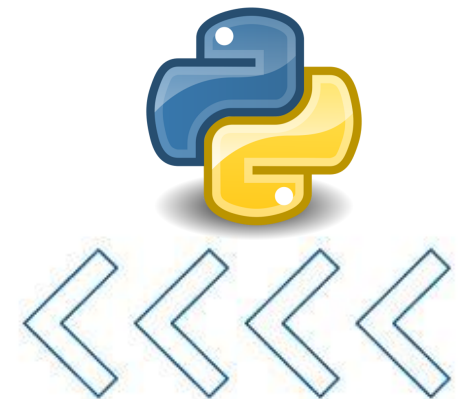


# องค์ประกอบพื้นฐานของ ภาษา Python



- **การเขียนคอมเมนต์ (Comment)** คือส่วนหมายเหตุของโปรแกรมเพื่ออธิบายหน้าที่หรือความหมายของโค้ดที่เขียนหรือยกเลิกโค้ดชั่วคราว ส่งผลให้ตัวแปลภาษาไม่สนใจโค้ดในบรรทัดที่ถูกทำหมายเหตุ

\* หากต้องการระบุคอมเมนต์ในโปรแกรมให้นำหน้าบรรทัดนั้นด้วยเครื่องหมาย #



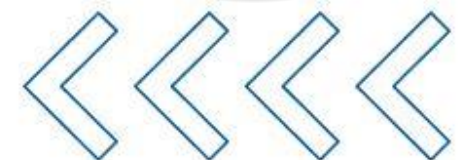


# โครงสร้างควบคุมพื้นฐาน ภาษา Python



คือกลุ่มคำสั่งที่ใช้ควบคุมการทำงานของโปรแกรม

- คำสั่งแบบลำดับ (Sequence)
- คำสั่งเงื่อนไข (Condition)
- คำสั่งทำซ้ำ (Loop)





# หมายเหตุ (Comment) ภาษา Python



## จุดประสงค์

- อธิบายหน้าที่หรือความหมายของโค้ดที่เขียน
- ยกเลิกโค้ดชั่วคราว ส่งผลให้ตัวแปลภาษาไม่สนใจโค้ดในบรรทัดที่ถูกทำหมายเหตุ





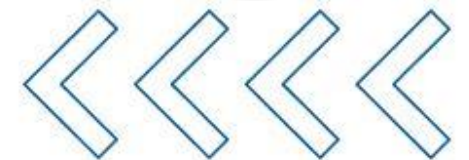


# หมายเหตุ (Comment) ภาษา Python

วิธีที่ 1 ใช้เครื่องหมาย # ใช้ในการอธิบายคำสั่งสั้นๆ

ในรูปแบบบรรทัดเดียว(เขียนด้านบน หรือ ด้านหลัง)

```
program.py - C:/Users/kongr/Desktop/LearningPython/program.py (3.12.7)
File Edit Format Run Options Window Help
1 #แสดงความออกจากทางจอภาพ
2 print("Hello Python") #แสดงความ (String)
3 print(100+400) #แสดงผลบวกของตัวเลข
4
```



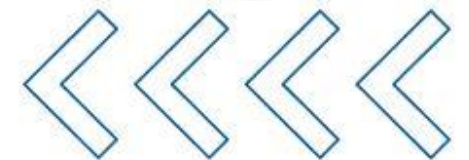


# หมายเหตุ (Comment) ภาษา Python

วิธีที่ 2 เขียนคำอธิบายไว้ในเครื่องหมาย `"""` (Triple Quote)

ใช้ในการอธิบายคำสั่งยาวๆ หรือแบบหลายบรรทัด

```
*program.py - C:/Users/kongr/Desktop/LearningPython/program.py (3.12.7)*
File Edit Format Run Options Window Help
1 """
2 แสดงข้อความออกทางจอภาพ
3 ผู้พัฒนาโปรแกรมคือ Python
4 เวอร์ชันโปรแกรม v1.0
5 """
6
7 print("Hello Python") #แสดงข้อความ (String)
8 print(100+400) #แสดงผลบวกของตัวเลข
9
```





Q&A