

สร้างแอปพลิเคชันด้วย Python



# Build Realworld-Project



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

# เครื่องมือพื้นฐาน

- Visual Studio Code
- Python (3.10)
- Package เสริมอื่นๆ



# แผนการสอน (Course Syllabus)

- Basic Python (GUI , OOP)
- Desktop Application
- Web Application



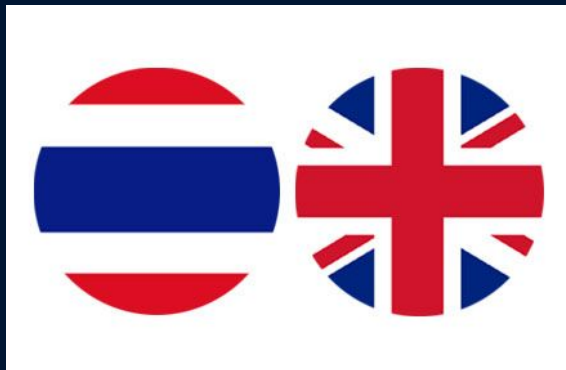


ดาวน์โหลดและติดตั้ง Python



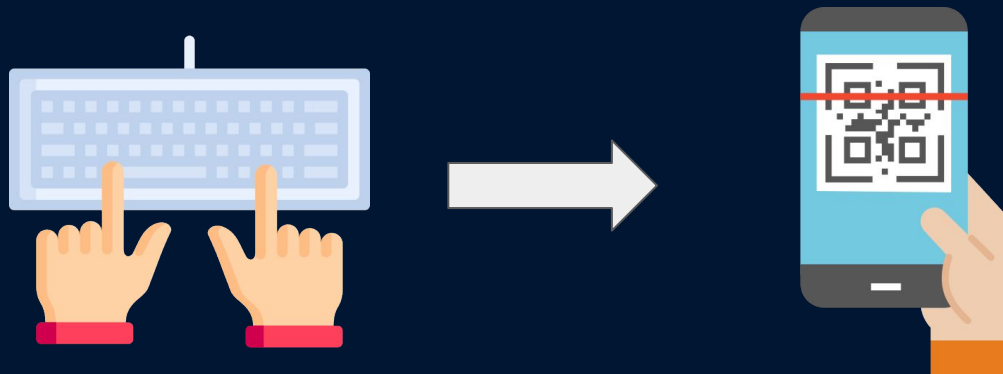
ติดตั้ง Visual Studio Code

# Project 1 : แอปแปลภาษา (EN-TH)



- Desktop Application
- ใช้ Tkinter ร่วมกับ Package เสริม
- รับ Input เป็นข้อความภาษาอังกฤษและส่ง Output เป็นภาษาไทย

# Project 2 : แอปสร้างคิวอาร์โค้ด



- Desktop Application
- ใช้ Tkinter ร่วมกับ Package เสริม
- รับ Input เป็นข้อความ / ลิงก์และส่ง Output เป็น QRCode

# ติดตั้ง Package

- `pyqrcode` - สำหรับสร้าง QRCode
- `pillow` - สำหรับจัดการรูปภาพ
- `pypng` - สร้างภาพ QRCode ในรูปแบบนามสกุล .png

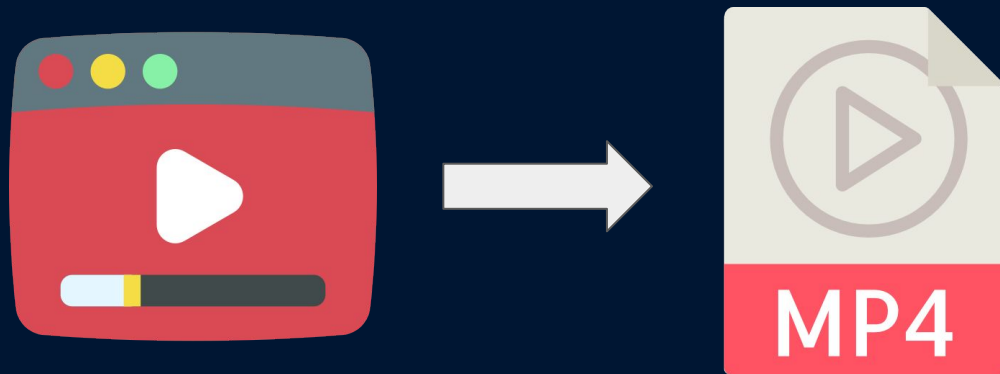


# ขั้นตอนการสร้างคิวอาร์โค้ด

1. กดปุ่มสร้างคิวอาร์โค้ด
2. ดึง Link มาสร้างเป็นคิวอาร์โค้ด
3. สร้างคิวอาร์โค้ดและส่งออกไฟล์ภาพ
4. แสดงภาพคิวอาร์โค้ดในโปรแกรม

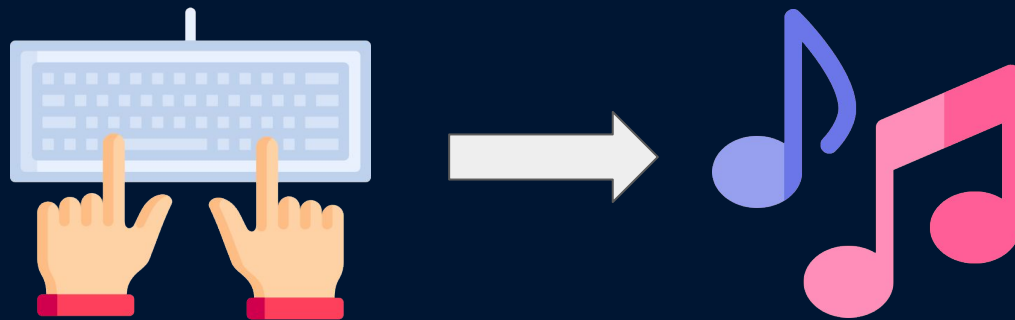


# Project 3 : แอปดาวน์โหลดวิดีโอ



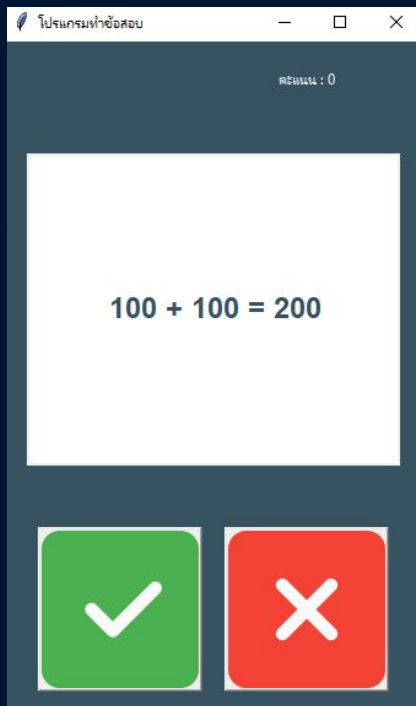
- Desktop Application
- ใช้ Tkinter ร่วมกับ Package เสริม
- รับ Input เป็นลิงก์ Youtube และส่ง Output เป็นไฟล์วิดีโอ

# Project 4 : แปลงข้อความเป็นเสียง



- Desktop Application
- ใช้ Tkinter ร่วมกับ Package เสริม
- รับ Input เป็นข้อความและส่ง Output เป็นไฟล์เสียง

# Project 5 : แอปทำข้อสอบ



- Desktop Application
- ใช้ Tkinter ในการสร้างหน้าจอโปรแกรม
- แก้โจทย์ปัญหาและคะแนนสอบ (Score)

# ความรู้พื้นฐาน

- การเขียนโปรแกรมเชิงวัตถุ (OOP)
- จัดการข้อมูลแบบโครงสร้างด้วย List และ Dictionary
- ออกแบบโจทย์ปัญหาด้วยรูปแบบการเขียนโปรแกรมเชิงวัตถุ
- ออกแบบหน้าจอโปรแกรมด้วย Tkinter (GUI)



# รูปแบบโปรแกรม

- กำหนดกลุ่มคำถามด้วย List / Dictionary
- คำตอบมี 2 ตัวเลือก (จริง / ไม่จริง)
- ตอบคำถามถูกได้รับคะแนน
- ตอบคำถามผิดไม่ได้รับคะแนน

# แสดงตัวอย่างโปรแกรม



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

# คลาสโจทย์ปัญหา (Question)

- ออกแบบคลาสสำหรับการจัดการโจทย์ปัญหาหรือคำถาม

ตัวอย่าง

Question (คำถาม) : ประเทศไทยมี 50 จังหวัดใช่หรือไม่ ?

Answer (เฉลย) : False (ไม่ใช่)



# คลาสโจทย์ปัญหา (Question)

ตัวอย่าง

Question (คำถาม) : ประเทศไทยมี 50 จังหวัดใช้หรือไม่ ?

Answer (เฉลย) : False (ไม่ใช่)

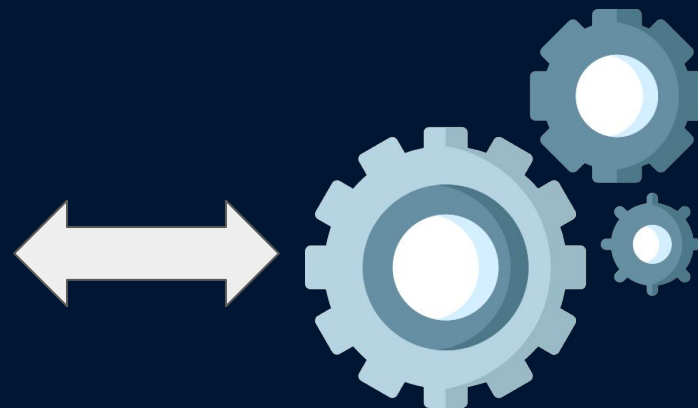
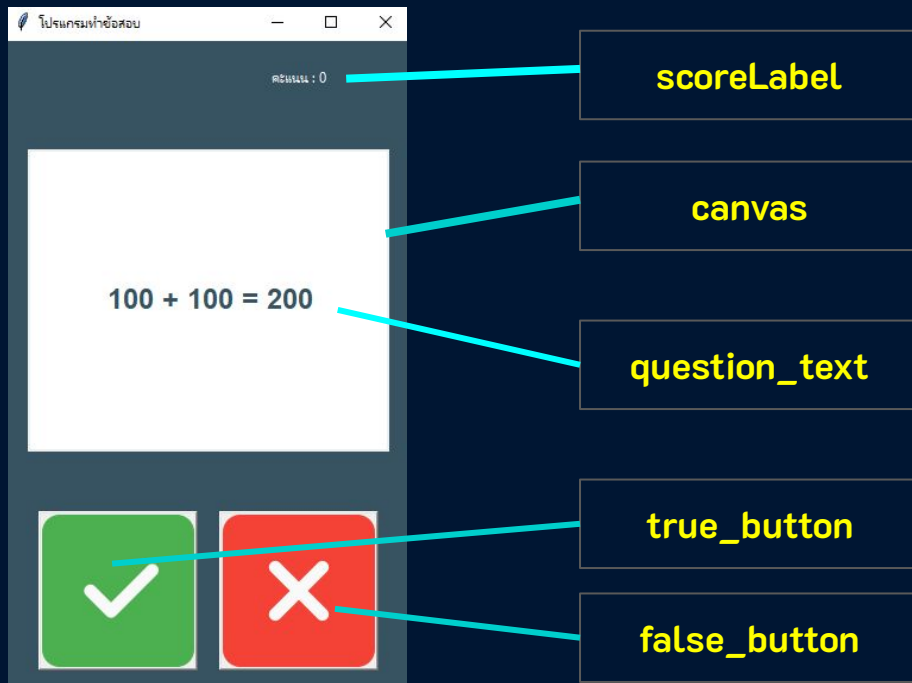
Answer: Attribute

Text : Attribute

# คลาสควบคุม (Controller)

- ออกแบบคลาสควบคุมการทำงานของโปรแกรม
- จัดการเกี่ยวกับคำถามหรือโจทย์ปัญหาทั้งหมด
- จัดการคำถามข้อถัดไป (Next Question)
- ตรวจสอบโจทย์ปัญหา (hasQuestion)
- ตรวจสอบคำตอบและเฉลย (CheckAnswer)

# หน้าจอโปรแกรม (GUI)



Controller

# เว็บจัดการบัญชีรายรับ - รายจ่าย



## Back-end

- Flask Micro Framework (Backend)
- DB Browser For SQLite (ฐานข้อมูล)

## Front-end

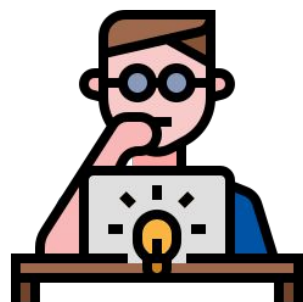
- Jinja2 Template
- Bootstrap 5

# คำศัพท์พื้นฐาน

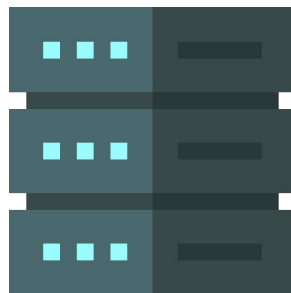
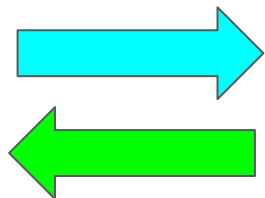
- Client - (ผู้ใช้บริการ)
- Server - (ผู้ให้บริการ)
- Request - (คำขอในการเข้าถึงข้อมูล)
- Response - (ตอบกลับคำขอ)

ภาพรวมของระบบ

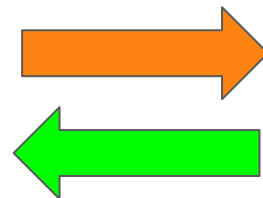
# ภาพรวมระบบ



Client

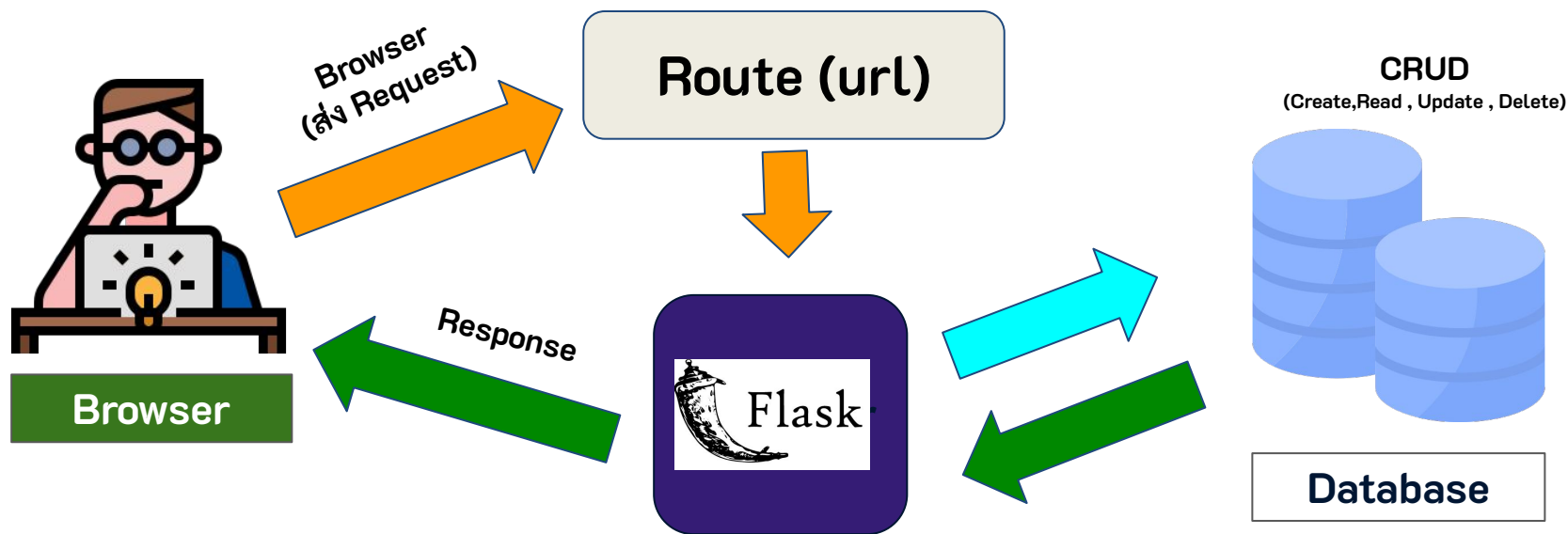


Server



Database

# แผนภาพการทำงาน





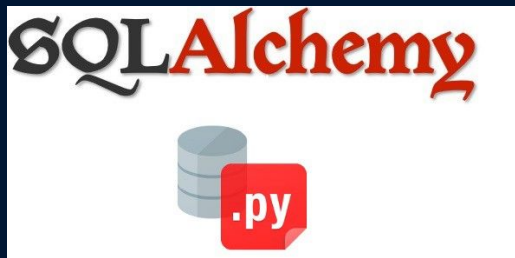
# ข้อมูลธุรกรรม (Statement)

ชื่อคอลัมน์	ชนิดข้อมูล	ความหมาย
ID (PK)	Integer	รหัสธุรกรรม
Date	String	วันที่บันทึก
Name	String	ชื่อธุรกรรม
Amount	Integer	จำนวนเงิน
Category	String	หมวดหมู่

# ติดตั้ง Package

- sqlalchemy - สำหรับจัดการฐานข้อมูล
- flask-sqlalchemy - เชื่อมฐานข้อมูลกับเว็บ

# SQLAlchemy คืออะไร



เป็น Package ที่ใช้จัดการกับฐานข้อมูล  
โดยไม่จำเป็นต้องใช้คำสั่ง SQL แต่จะใช้รูปแบบ  
**Object Relational Mapper (ORM)** ใน  
การดำเนินการกับฐานข้อมูลทั้งหมดผ่านส่วน  
ที่เรียกว่า โมเดล (Model)

# เชื่อมต่อกับฐานข้อมูล

```
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///mydb.db'  
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
```

# สร้างโมเดล (Model)

ข้อมูลธุรกรรม (Statement)

ชื่อคอลัมน์	ชนิดข้อมูล	ความหมาย
ID (PK)	Integer	รหัสธุรกรรม
Date	String	วันที่บันทึก
Name	String	ชื่อธุรกรรม
Amount	Integer	จำนวนเงิน
Category	String	หมวดหมู่

# ดึงผลห่วยจากเว็บไซต์ด้วย Python



# Web Scrapping คืออะไร

คือ การดึงข้อมูลจากเว็บไซต์ โดยใช้ภาษาโปรแกรม (Python) สำหรับรัน Script เพื่อสกัดเอาข้อมูลที่ต้องการในหน้าเว็บเพจ ไปใช้งานตามวัตถุประสงค์ที่ต้องการ โดยข้อมูลดังกล่าวจะไม่ได้ถูกเรียกมาจากฐานข้อมูลโดยตรง (Back-end) แต่จะถูกสกัดและถูกกลั่นกรองเอาเฉพาะส่วนที่ต้องการในหน้าเว็บเพจเท่านั้น (Front-end)

# ข้อดีและข้อเสีย

## ข้อดี

- สามารถดึงข้อมูลจากหน้าเว็บเพจได้ฟรี
- สามารถนำข้อมูลไปใช้ประโยชน์ตามวัตถุประสงค์ที่ต้องการได้

## ข้อเสีย

- ถ้ามีการเปลี่ยนแปลงโครงสร้างหน้าเว็บก็จะส่งผลให้มีการปรับปรุงแก้ไข Script การทำงานของโปรแกรมใหม่ทั้งหมด





# ต้องมีพื้นฐานอะไรบ้าง

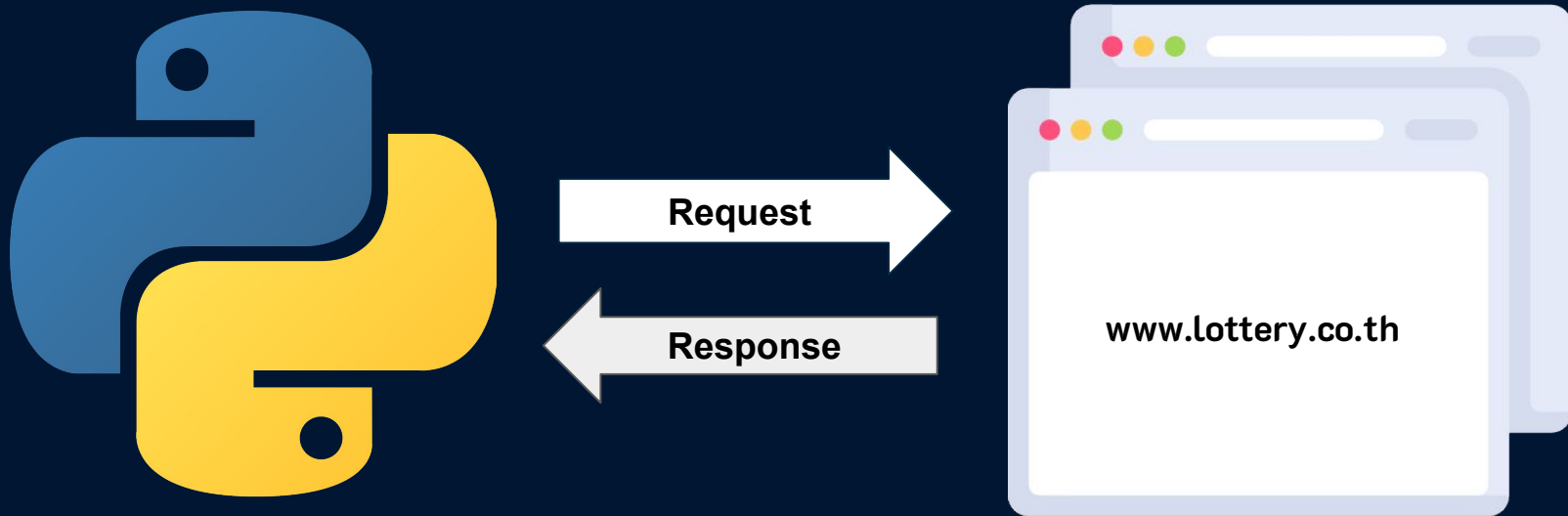
- Python เบื้องต้น
- HTML5 เบื้องต้น\*
- CSS3 เบื้องต้น\*



# คำศัพท์พื้นฐาน

- Client - (ผู้ใช้บริการ)
- Server - (ผู้ให้บริการเว็บไซต์)
- Request - (คำขอในการเข้าถึงข้อมูล)
- Response - (ตอบกลับคำขอ)

# ดึงผลหวยจากเว็บไซต์ด้วย Python



# ติดตั้ง Package

- requests - ส่งคำขอการเข้าถึงข้อมูลหน้าเว็บ
- beautifulsoup4 - ดึงโครงสร้างหน้าเว็บ (HTML)

# การเซ็นเซอร์ใบหน้าด้วย Python

