

The Django logo, featuring the word "django" in a white, lowercase, sans-serif font, centered within a dark green rectangular background.

django

พัฒนาเว็บแอปพลิเคชันด้วย
Django Framework

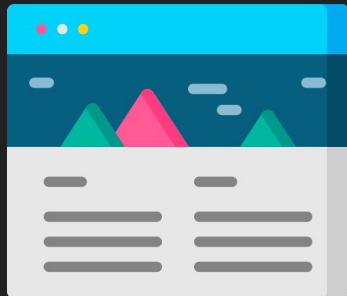
Django Framework คืออะไร



django

Django คือ เว็บเฟรมเวิร์คสำหรับ
สร้าง Web Application ฝั่ง
BackEnd ที่พัฒนาด้วยภาษาไพธ
อน (Python)

Frontend VS Backend



Frontend คือ การพัฒนาโปรแกรมระบบหน้าบ้าน (UI : User Interface หรือ หน้าตาของแอปพลิเคชัน) โดยผู้ใช้งานสามารถมองเห็นและมีส่วนร่วมหรือโต้ตอบภายใน Web Browser ได้



Backend คือ การพัฒนาโปรแกรมหลังบ้าน หรือการทำงานเบื้องหลังในแอป เช่น การทำงานกับฐานข้อมูล เป็นต้น โดยผู้ใช้งานไม่สามารถมีส่วนร่วมหรือโต้ตอบได้

จุดเด่นของ Django Framework

- ใช้ภาษาไพธอนในการพัฒนา มีรูปแบบการเขียนโค้ดที่ไม่ซับซ้อน และเข้าใจง่ายอีกทั้งยังเป็นภาษาที่ได้รับความนิยมในปัจจุบัน
- แยกการทำงานออกเป็นส่วนย่อยๆ โดยใช้สถาปัตยกรรม **MVT (Model-View-Template)**
- ใช้งานได้อิสระ เราสามารถปรับปรุงแก้ไข หรือ กำหนดวิธีการทำงานใหม่ได้ตามความต้องการ



ต้องมีพื้นฐานอะไรบ้าง



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



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

ต้องมีพื้นฐานอะไรบ้าง

- พื้นฐานการเขียนโปรแกรมภาษาไพธอน
- พื้นฐาน SQL
- HTML5
- CSS3
- JavaScript
- Bootstrap 5



ภาพรวมระบบ



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

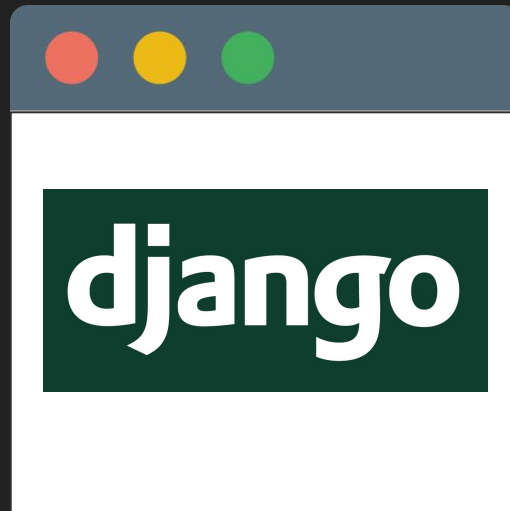


<https://www.facebook.com/KongRuksiamTutorial/>

ภาพรวมระบบ



Frontend (Client)

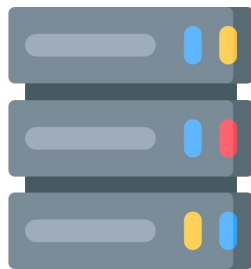
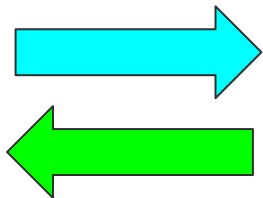


Backend (Server)

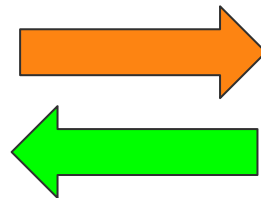
ภาพรวมระบบ



Client



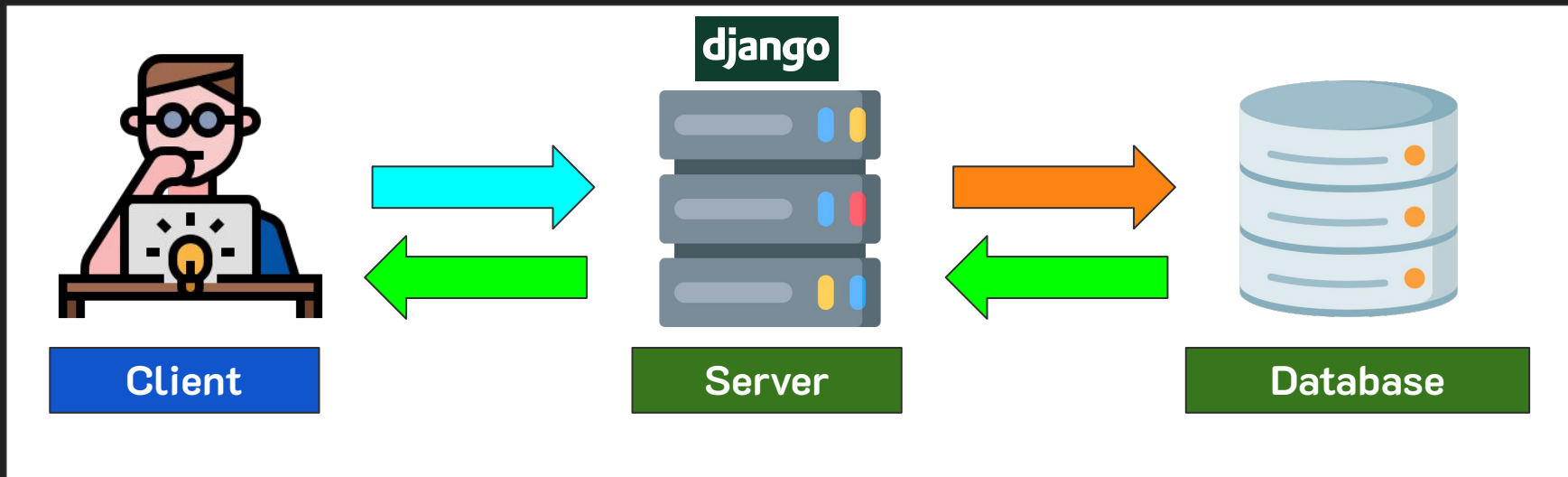
Server



Database



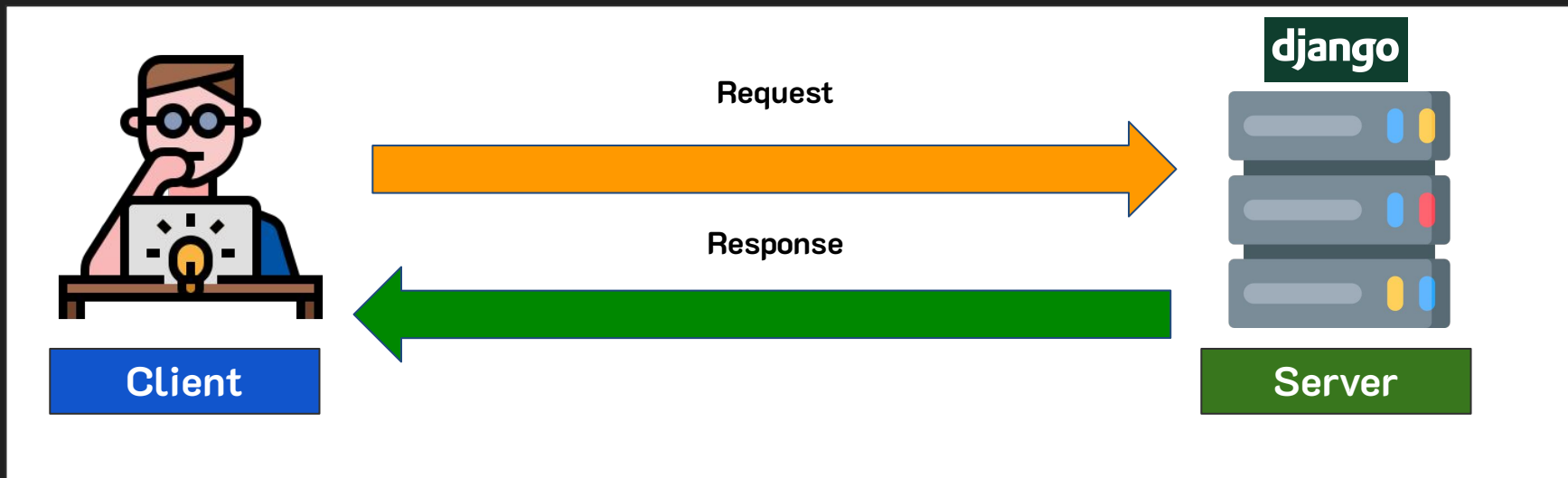
ภาพรวมระบบ



คำศัพท์พื้นฐาน

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

ภาพรวมระบบ



เครื่องมือที่ใช้

เครื่องมือที่ใช้

- Python
- Visual Studio Code
- Google Chrome





ติดตั้ง Python



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



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



คำสั่งสำหรับตรวจสอบเวอร์ชัน

`python --version`



ติดตั้ง Visual Studio Code



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



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

django

ติดตั้ง Django Framework

(pip : python install package)



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



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

แสดงรายการ Package

```
pip list
```

ติดตั้ง Package

```
pip install django
```

สร้างโปรเจกต์ Django



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



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

สร้างโปรเจกต์ใหม่

```
django-admin startproject <ชื่อโปรเจกต์>
```

รันโปรเจกต์

```
python manage.py runserver
```



โครงสร้างโปรเจกต์



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



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

โครงสร้างโปรเจกต์

- **manage.py** คือไฟล์ script สำหรับรันคำสั่งต่างๆ ที่เกี่ยวข้องกับ Django เช่น Run Server , Model & Migration
- **__init__.py** คือ initial ไฟล์หรือไฟล์เปล่าๆมีไว้เก็บ Python Package เราสามารถเพิ่ม Script การทำงานเข้าไปในไฟล์นี้ได้



โครงสร้างโปรเจกต์

- **settings.py** คือไฟล์ที่ใช้สำหรับการตั้งค่าโปรเจค เช่น การตั้งค่า แอป , เวลา , Path,ฐานข้อมูลที่ใช้ เป็นต้น
- **urls.py** คือไฟล์ที่ใช้เก็บการ routing ของ HTTP request หรือ เรียกอีกอย่างว่าการกำหนด urlpattern ของ django project
- **wsgi.py** คือไฟล์ที่ใช้เก็บข้อมูลโปรเจคสำหรับการ Deployment

รู้จักกับ MVT (Model-View-Template)



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

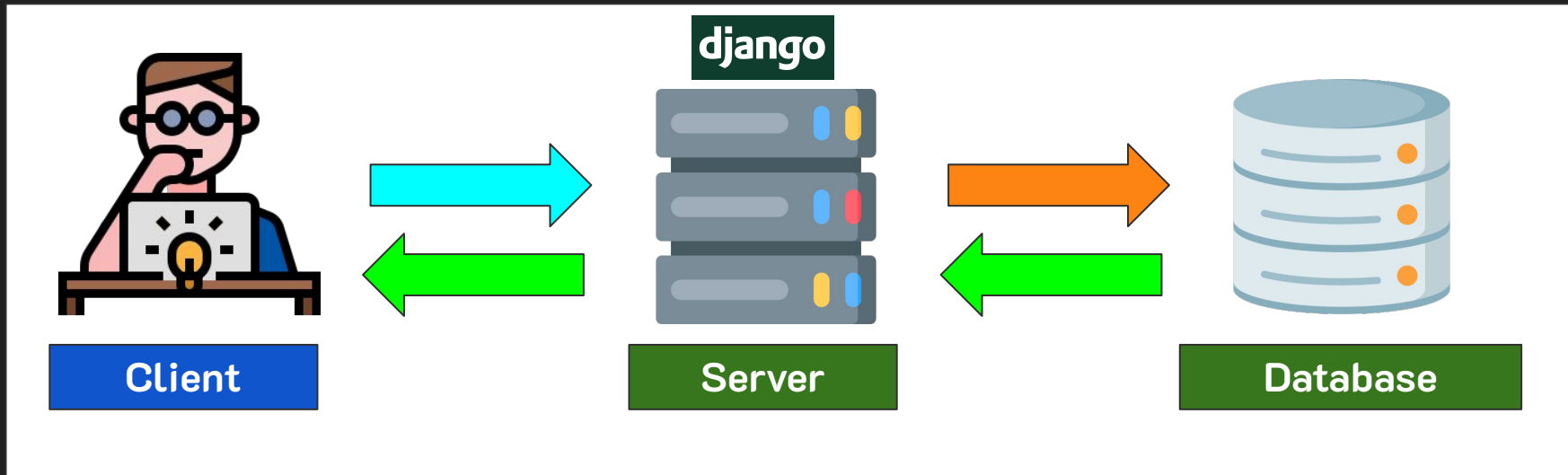


<https://www.facebook.com/KongRuksiamTutorial/>

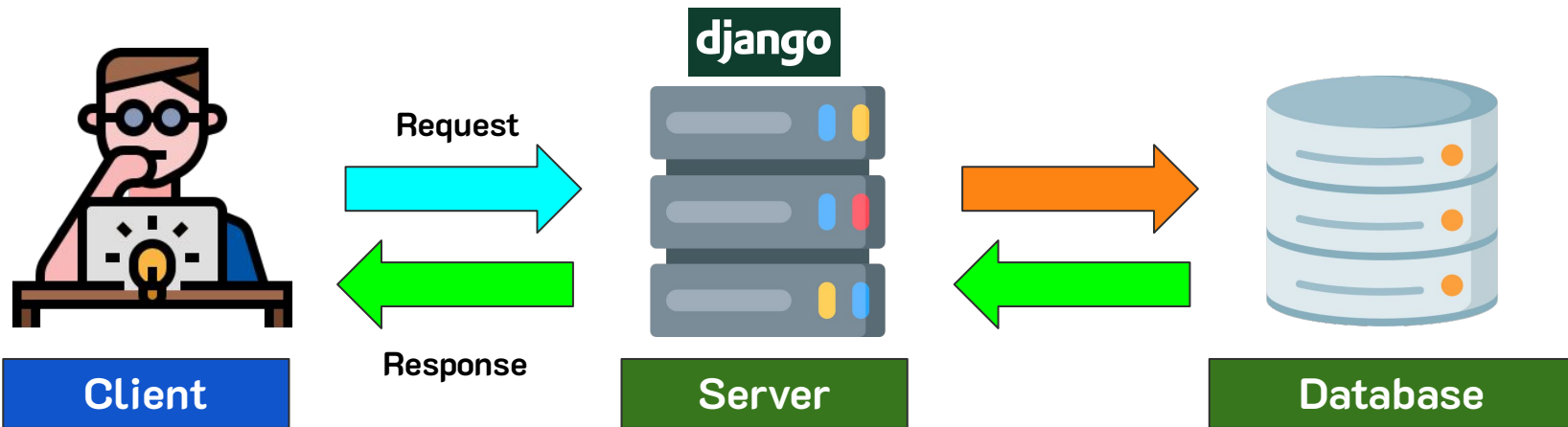
รู้จักกับ MVT

- **Model (M)** คือ ส่วนที่เก็บข้อมูลของ Application
- **View (V)** คือ ส่วนประมวลผลคำสั่งหรือข้อมูลต่างๆ โดยควบคุมการทำงานระหว่าง Model และ Template
- **Template (T)** คือ หน้าตา Application เป็นส่วนที่ไว้ใช้แสดงผลข้อมูลผลลัพธ์จากการประมวลผลข้อมูลในหน้าเว็บร่วมกับ HTML

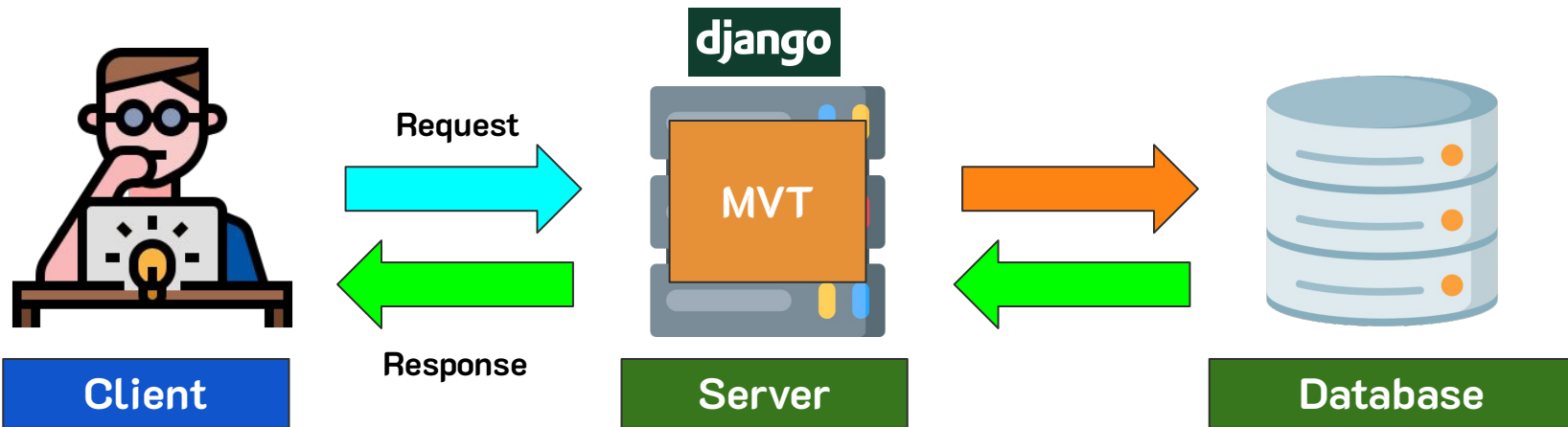
รู้จักกับ MVT



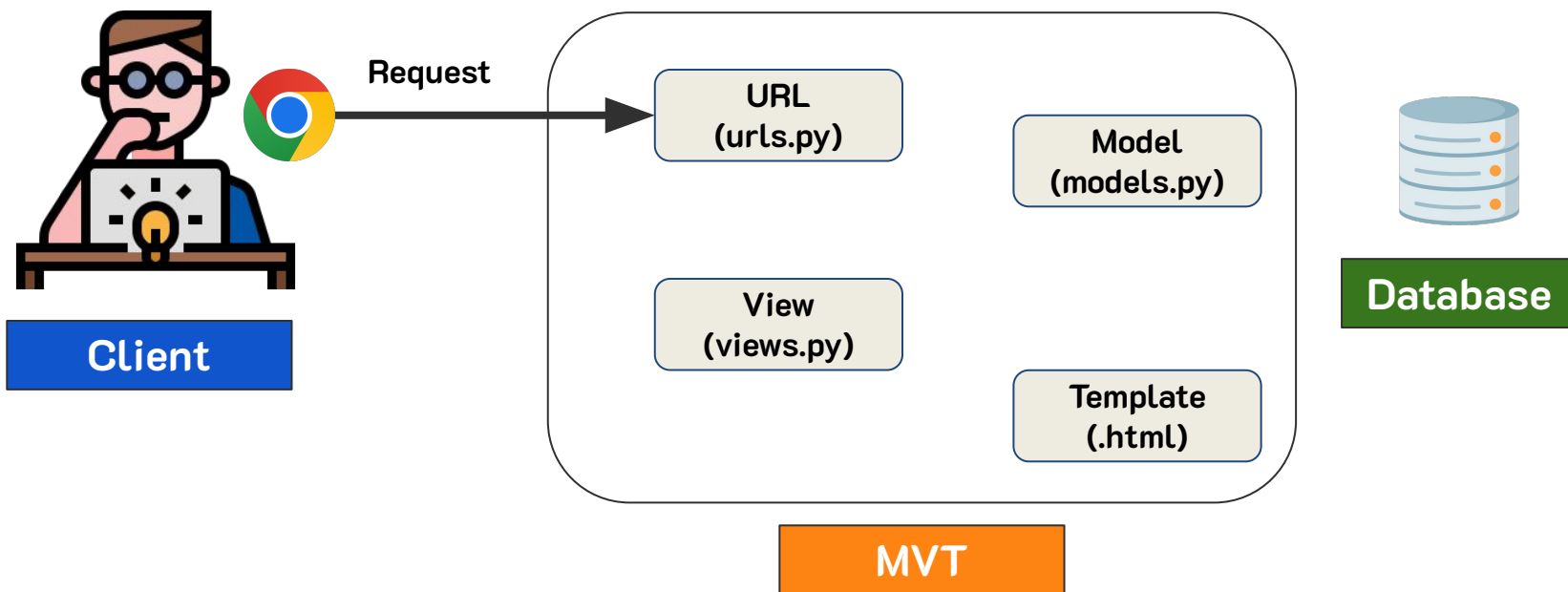
รู้จักกับ MVT



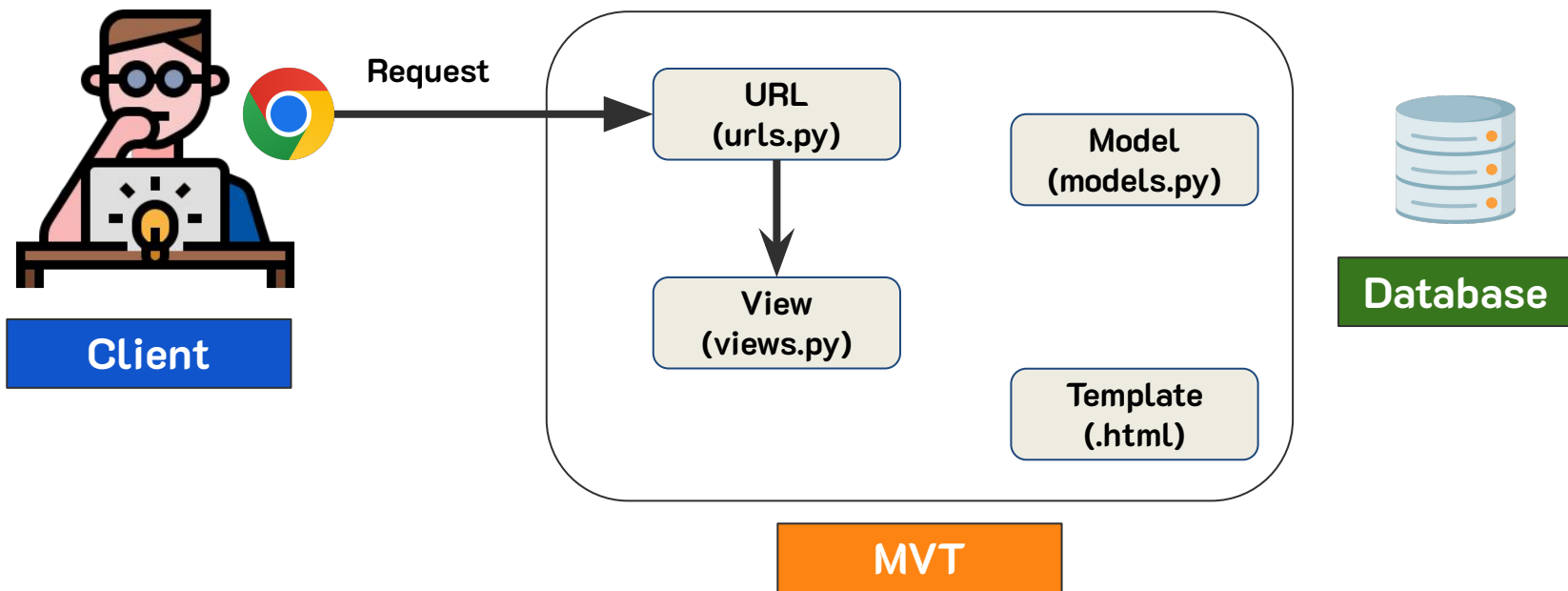
รู้จักกับ MVT



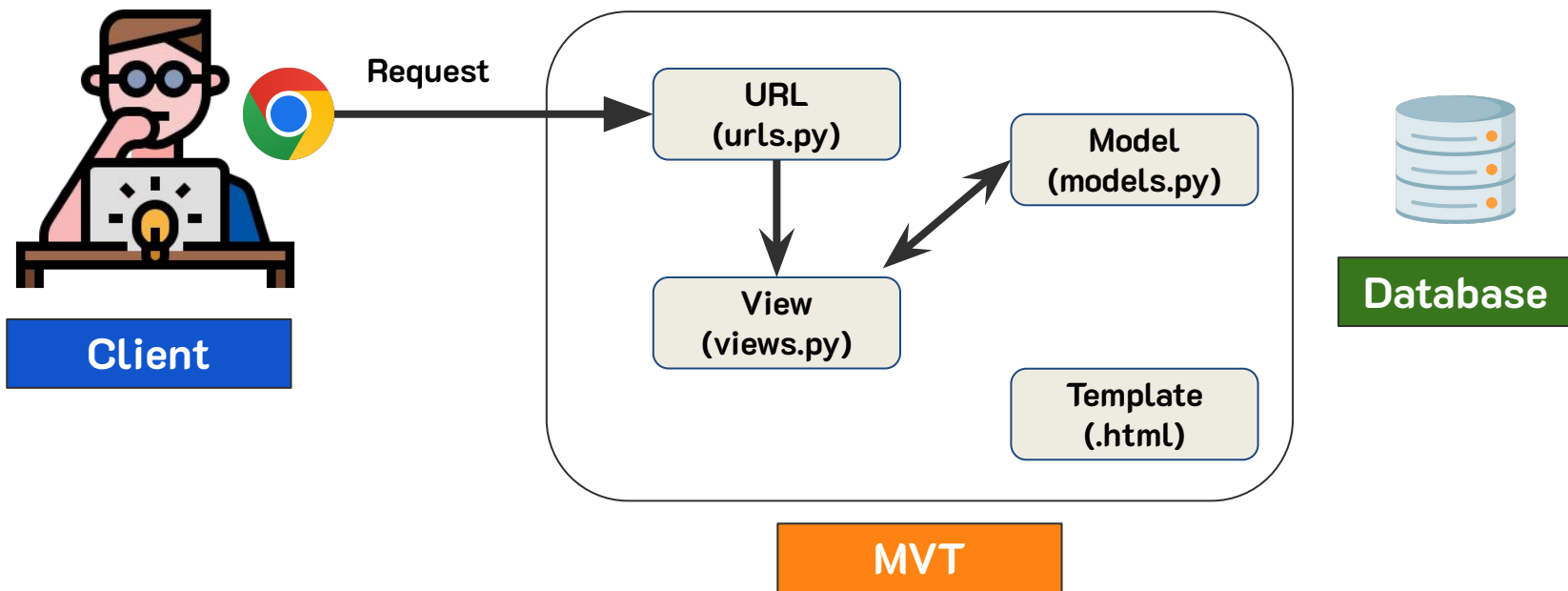
รู้จักกับ MVT



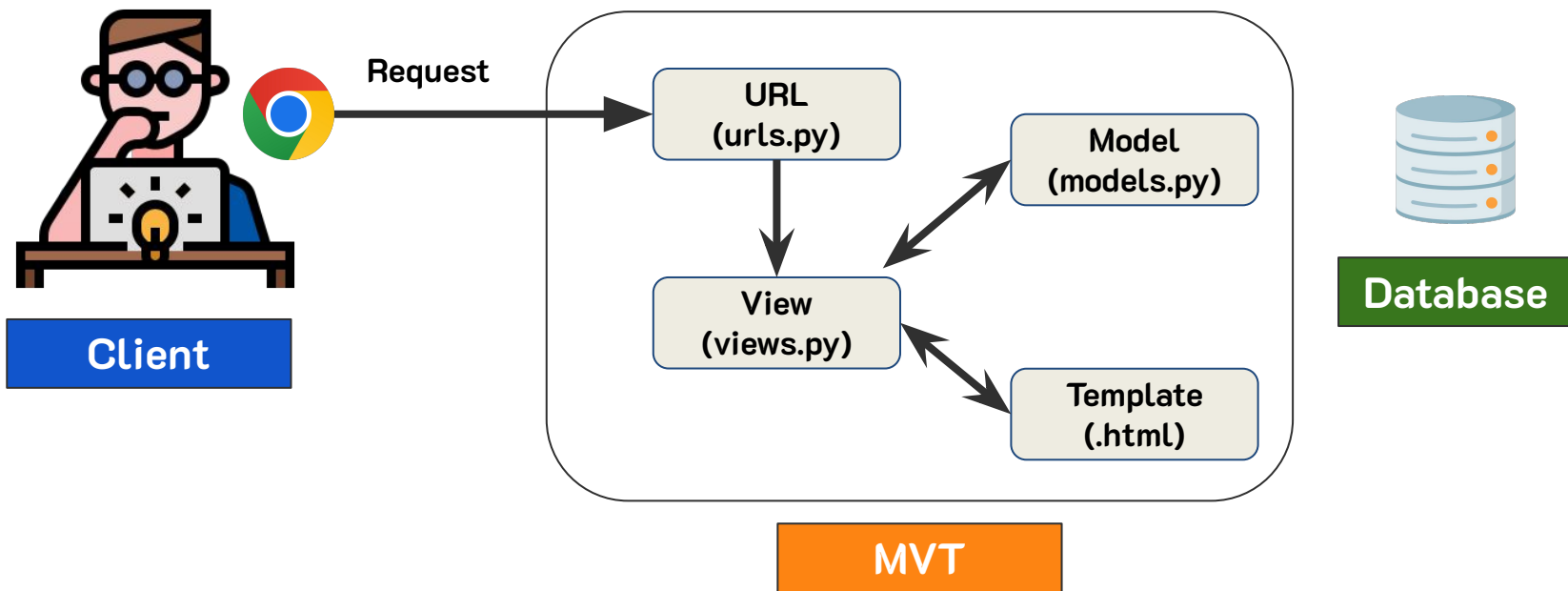
รู้จักกับ MVT



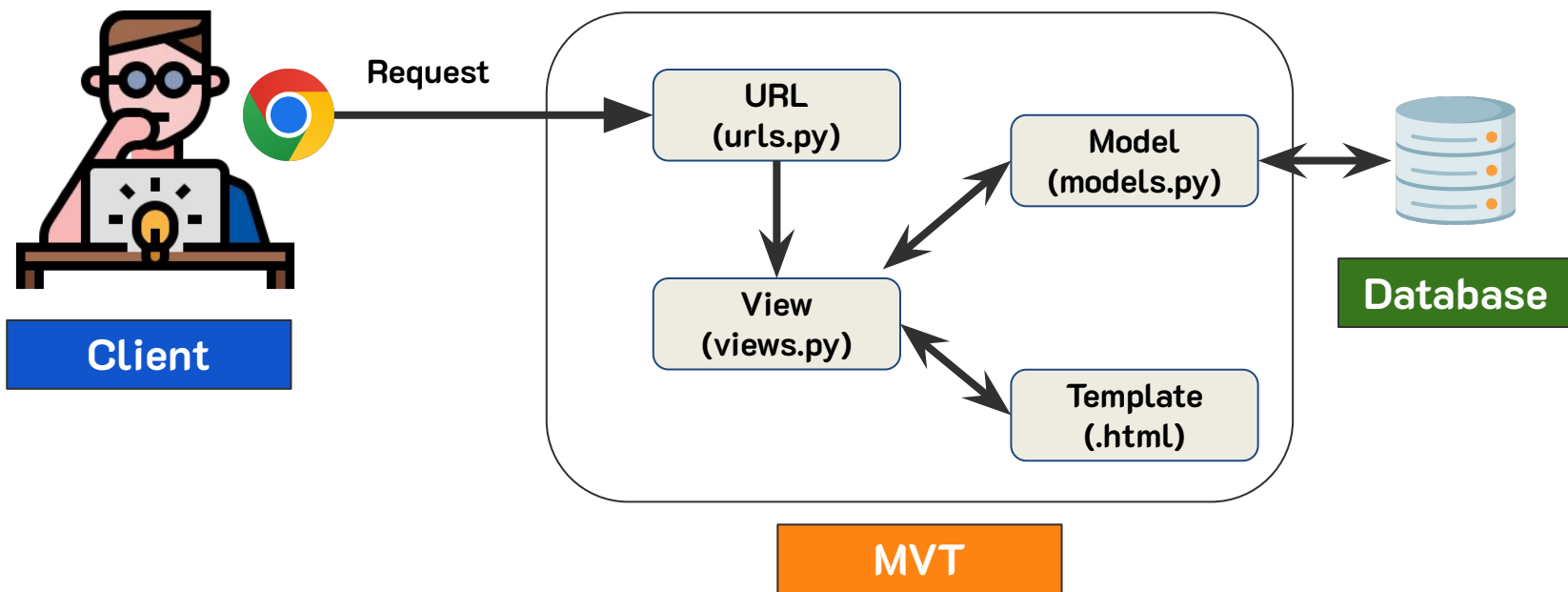
รู้จักกับ MVT



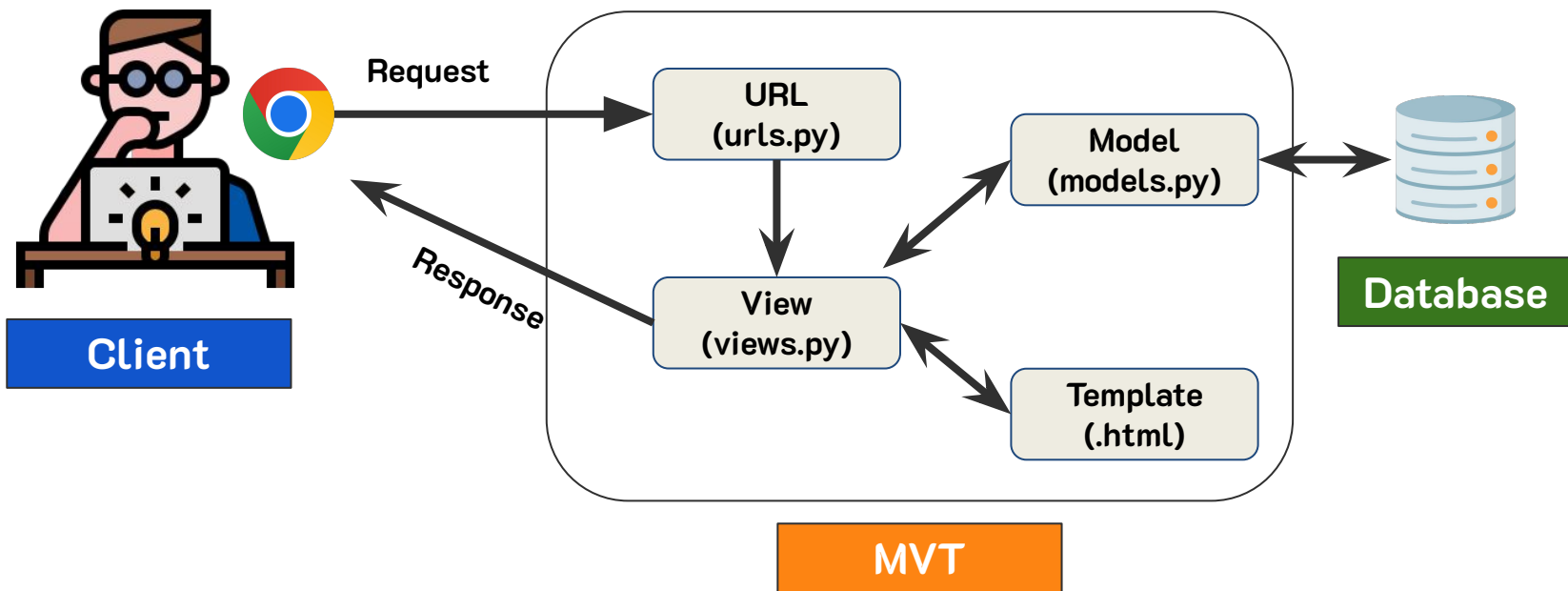
รู้จักกับ MVT



รู้จักกับ MVT



รู้จักกับ MVT



สร้างแอปพลิเคชัน (Django Application)



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



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

สร้างแอปพลิเคชัน

ในโปรเจกต์ สามารถแบ่งองค์ประกอบออกเป็นส่วนย่อยๆแล้วนำมาทำงานร่วมกันในภายหลังได้ เรียกว่า แอป เพื่อจัดการเกี่ยวกับระบบย่อยต่างๆ ภายในโปรเจกต์ ซึ่งภาพรวมของระบบสามารถแบ่งการทำงานออกเป็น 2 ระดับ คือ

- **Project Level** คือ ระบบหลัก สามารถดำเนินการกับโปรเจกต์ได้โดยตรง
- **App Level** คือ ระบบย่อย เป็นการดำเนินการกับระบบย่อยต่างๆในโปรเจกต์ เช่น ระบบสมาชิก , ระบบหมวดหมู่ เป็นต้น



สร้างแอปพลิเคชัน

```
python manage.py startapp <ชื่อแอป>
```



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



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

เพิ่มแอปพลิเคชันในโปรเจกต์

settings.py

```
INSTALLED_APPS = [  
    'ชื่อแอป'  
]
```

URL & Views

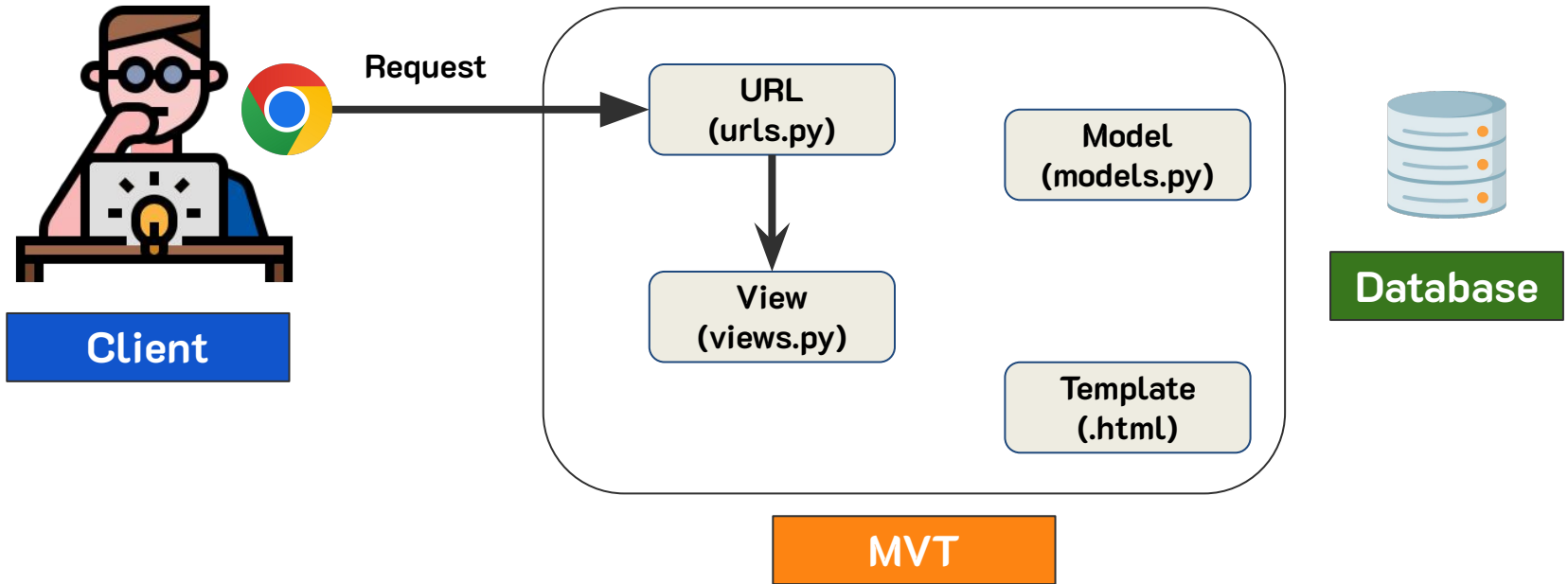


<https://www.youtube.com/c/KongRuksiamOfficial/>

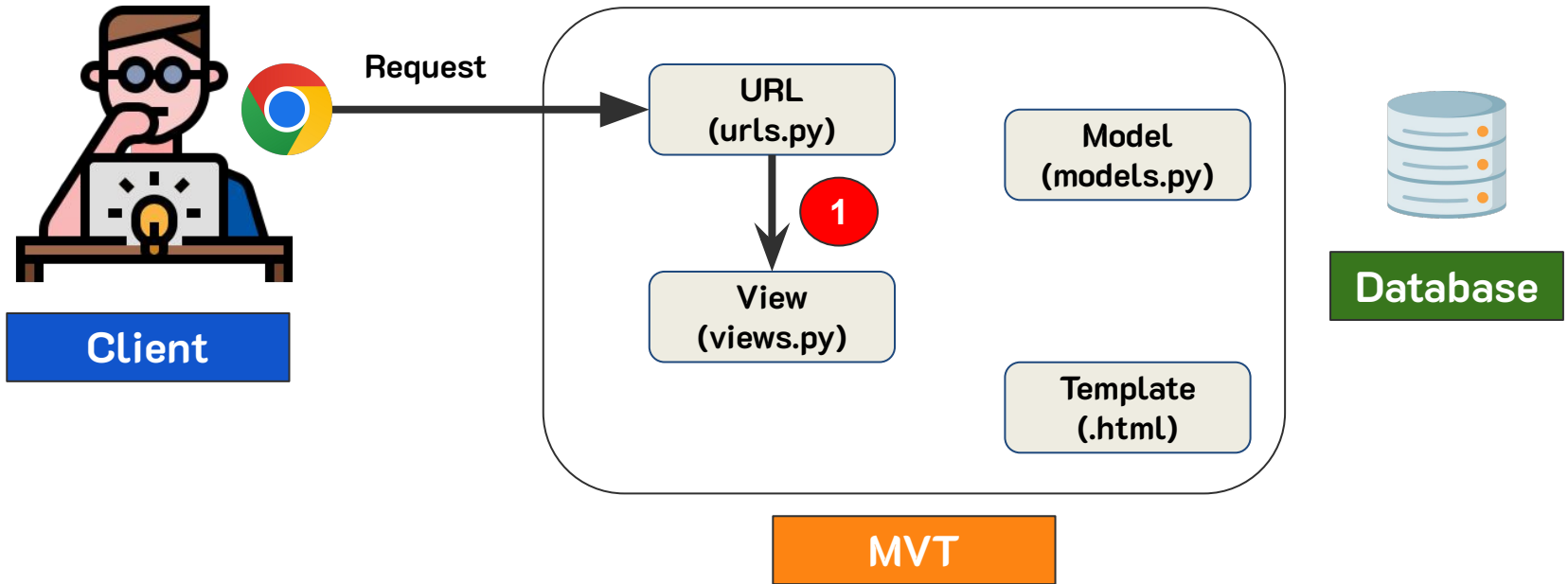


<https://www.facebook.com/KongRuksiamTutorial/>

URL & View



URL & View



URL & View

- **URL** คือ ส่วนที่ใช้ระบุเส้นทางในการรับส่งข้อมูล
- **View** คือ ศูนย์กลางสำหรับรับส่งข้อมูล โดยเชื่อมโยงการทำงานระหว่าง Model และ Template

โครงสร้าง URL

<https://www.example.com/project/computer>

URL & View

- **URL** คือ ส่วนที่ใช้ระบุเส้นทางในการรับส่งข้อมูล
- **View** คือ ศูนย์การสำหรับรับส่งข้อมูล โดยเชื่อมโยงการทำงานระหว่าง Model และ Template

โครงสร้าง URL

Protocol

Domain

Path

<https://www.example.com/project/computer>

urls.py

ไฟล์สำหรับการจัดการเส้นทางหรือพาร (path) การทำงาน

องค์ประกอบของ urls.py

- **from django.urls import path , include** คือการนำเอาพาร (url) มาใช้งาน
- **urlpatterns** คือ การกำหนดกลุ่มรูปแบบพาร (List)

views.py

ไฟล์สำหรับกำหนดรูปแบบการทำงานโดยเชื่อมโยงกับไฟล์ urls.py

views.py

```
from django.http import HttpResponse
```

```
def index(request):
```

```
    return HttpResponse("KongRuksiam Studio")
```

ชื่อโปรเจกต์ / urls.py ของโปรเจกต์

```
from django.urls import path , include

urlpatterns = [

    path('/',include("ชื่อแอป.urls")),

]
```

ชื่อแอป / urls.py

```
from ชื่อแอป import views

from django.urls import path

urlpatterns = [

    path('/',views.index),

]
```

ชื่อแอป / views.py

```
from django.http import HttpResponse

def index(request):

    return HttpResponse("KongRuksiam")
```

ชื่อโปรเจกต์ / urls.py ของโปรเจกต์

```
from django.urls import path , include
```

```
urlpatterns = [
```

```
    path('/',include("ชื่อแอป.urls")),
```

```
]
```

ชื่อแอป / urls.py

```
from ชื่อแอป import views
```

```
from django.urls import path
```

```
urlpatterns = [
```

```
    path('/',views.index),
```

```
]
```

ชื่อแอป / views.py

```
from django.http import HttpResponse
```

```
def index(request):
```

```
    return HttpResponse("KongRuksiam")
```


ชื่อโปรเจกต์ / urls.py (ของโปรเจกต์)

```
from django.urls import path , include

urlpatterns = [

    path('/',include("ชื่อแอป.urls")),

]
```

ชื่อแอป / urls.py (ของแอป)

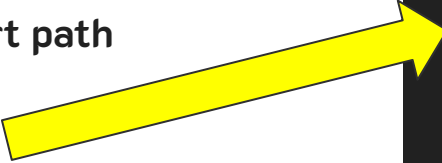
```
from ชื่อแอป import views

from django.urls import path

urlpatterns = [

    path('/',views.index),

]
```



ชื่อแอป / views.py

```
from django.http import HttpResponse

def index(request):

    return HttpResponse("KongRuksiam")
```

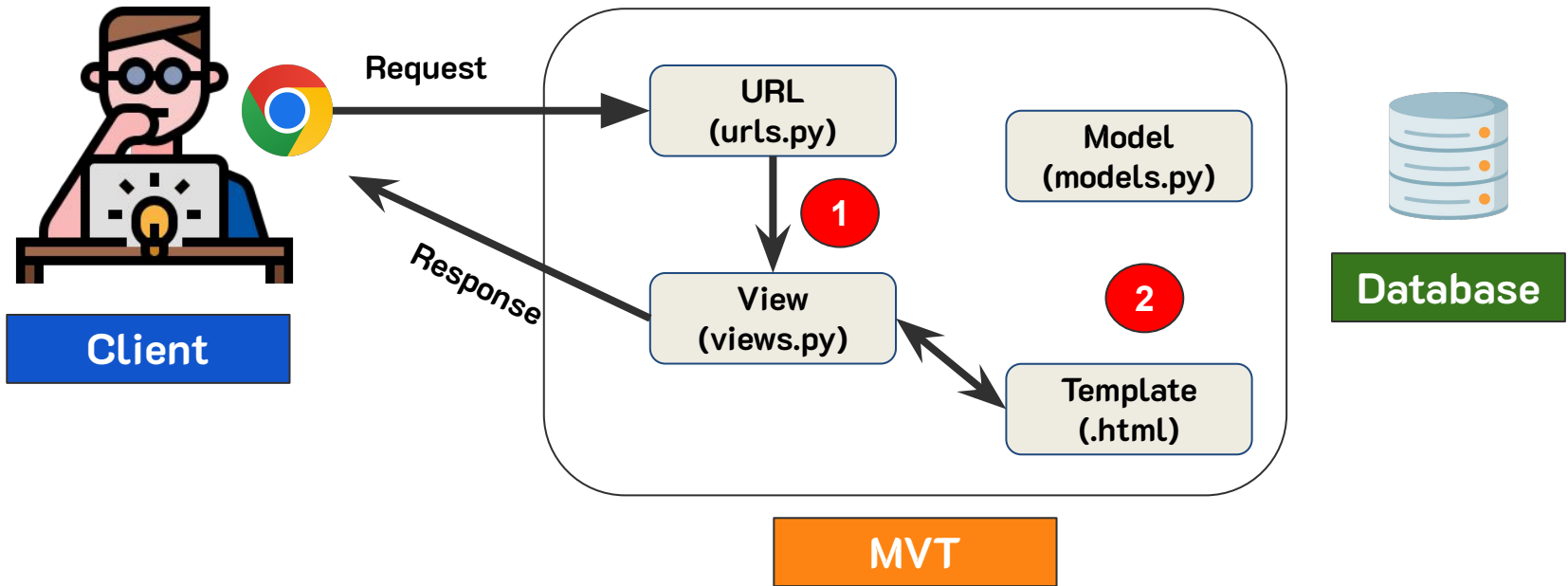
Template

Template



Template คือหน้าต่าง Application
เป็นส่วนที่ไว้ใช้แสดงผลข้อมูล
ผลลัพธ์จากการประมวลผลข้อมูลใน
หน้าเว็บร่วมกับ HTML

Template



แสดงผล Template

1. สร้างโฟลเดอร์ `templates` ในแอปพลิเคชันพร้อมไฟล์ `.html`

2. ชื่อแอป/`views.py`

```
from django.shortcuts import render
```

```
def index(request):
```

```
    return render(request, "ชื่อtemplate.html")
```

Django Template Language

Django Template Language

คือ การแทรกไวยากรณ์ภาษา Python ลงไปในเว็บเพจ (html) ซึ่งมีทั้งหมด 3 รูปแบบ

- **{{variable}}** คือ การแทรกตัวแปรลงไปในส่วนที่ต้องการในหน้าเว็บเพจ
- **{% tag %}** คือการแทรก tag หรือโครงสร้างควบคุมลงไปในหน้าเว็บ
- **{{variable | filter }}** คือ การแทรกตัวกรองข้อมูลในตัวแปร



แทรกตัวแปรในหน้าเว็บ

View.py

```
def index(request):  
    return render(request,  
        "index.html", {ชื่อตัวแปร:ค่าที่เก็บในตัวแปร}  
    )
```



index.html

```
<body>
```

```
  {{ชื่อตัวแปร}}
```

```
</dody>
```



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



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

แทรก Tag ในหน้าเว็บ

แทรก Tag ในหน้าเว็บ

โครงสร้างคำสั่ง

```
{% tag %}
```

.....

```
{% endtag %}
```

ตัวอย่าง

```
{% if name=="admin" %}
```

.....

```
{% endif %}
```

แทรก Tag ในหน้าเว็บ

โครงสร้างคำสั่ง

```
{% tag %}
```

.....

```
{% endtag %}
```

ตัวอย่าง

```
{% for item in list %}
```

.....

```
{% endfor %}
```

แทรกตัวกรองในหน้าเว็บ

index.html

```
<body>
```

```
  {{ชื่อตัวแปร | ตัวกรอง}}
```

```
</dody>
```



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



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

ตัวอย่างตัวกรอง (Filter)

ชื่อตัวกรอง (Filter)	ความหมาย
lower	แปลงข้อความเป็นตัวพิมพ์เล็ก
upper	แปลงข้อความเป็นตัวพิมพ์ใหญ่
wordcount	นับจำนวนคำ



การสร้าง Template หลัก (Base Template)

การสร้าง Template หลัก

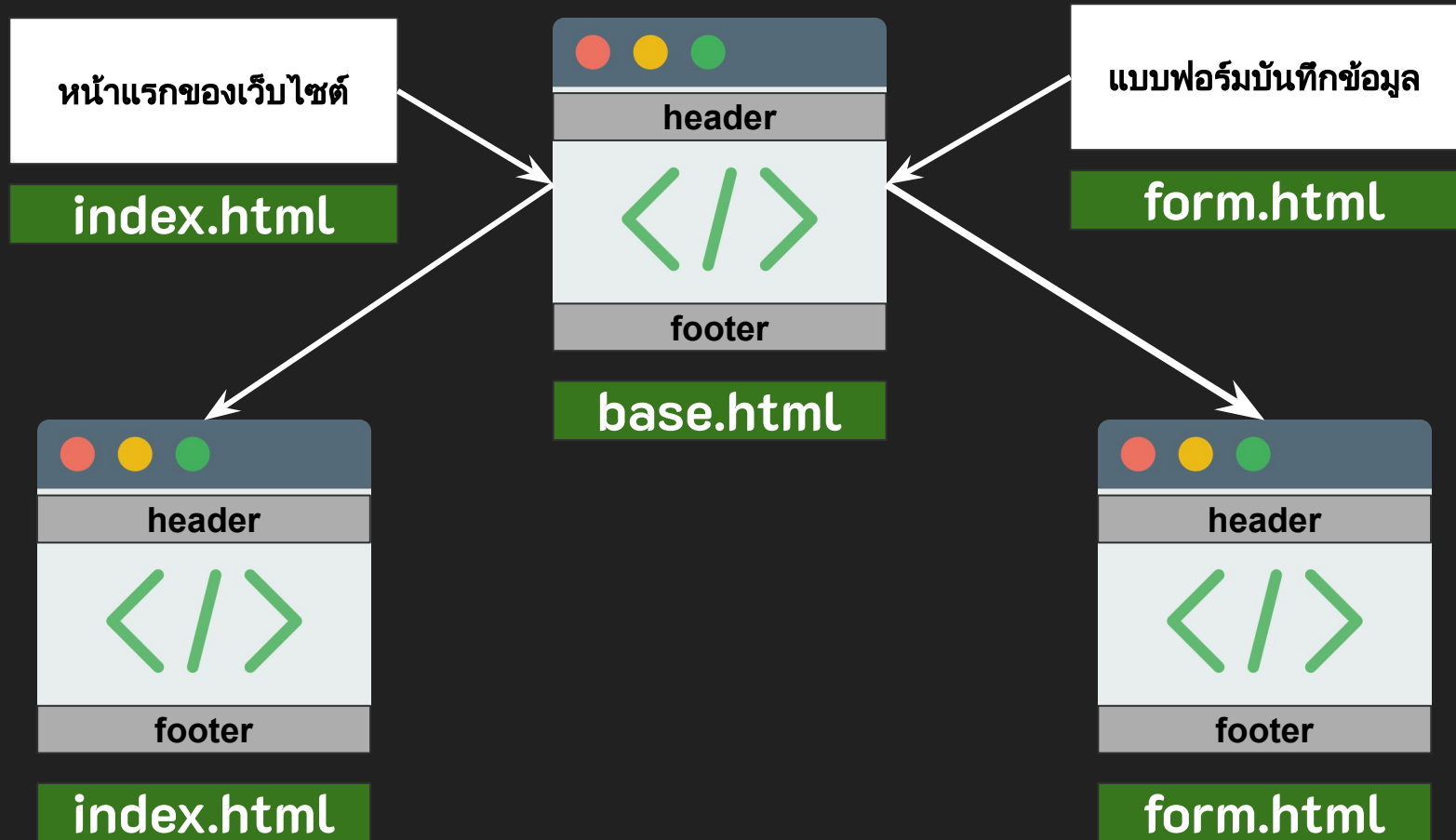
คือ การกำหนดโครงสร้างหลักในหน้าเว็บเพจที่ทุกๆหน้า ใช้งานร่วมกัน เพื่อลดความซ้ำซ้อนของโค้ด โดยการแทรกไฟล์เนื้อหาหลักเข้าไปทำงานในหน้าเว็บเพจ และกำหนดตำแหน่งแทรกเนื้อหาลงไปตามชื่อพื้นที่ (block)

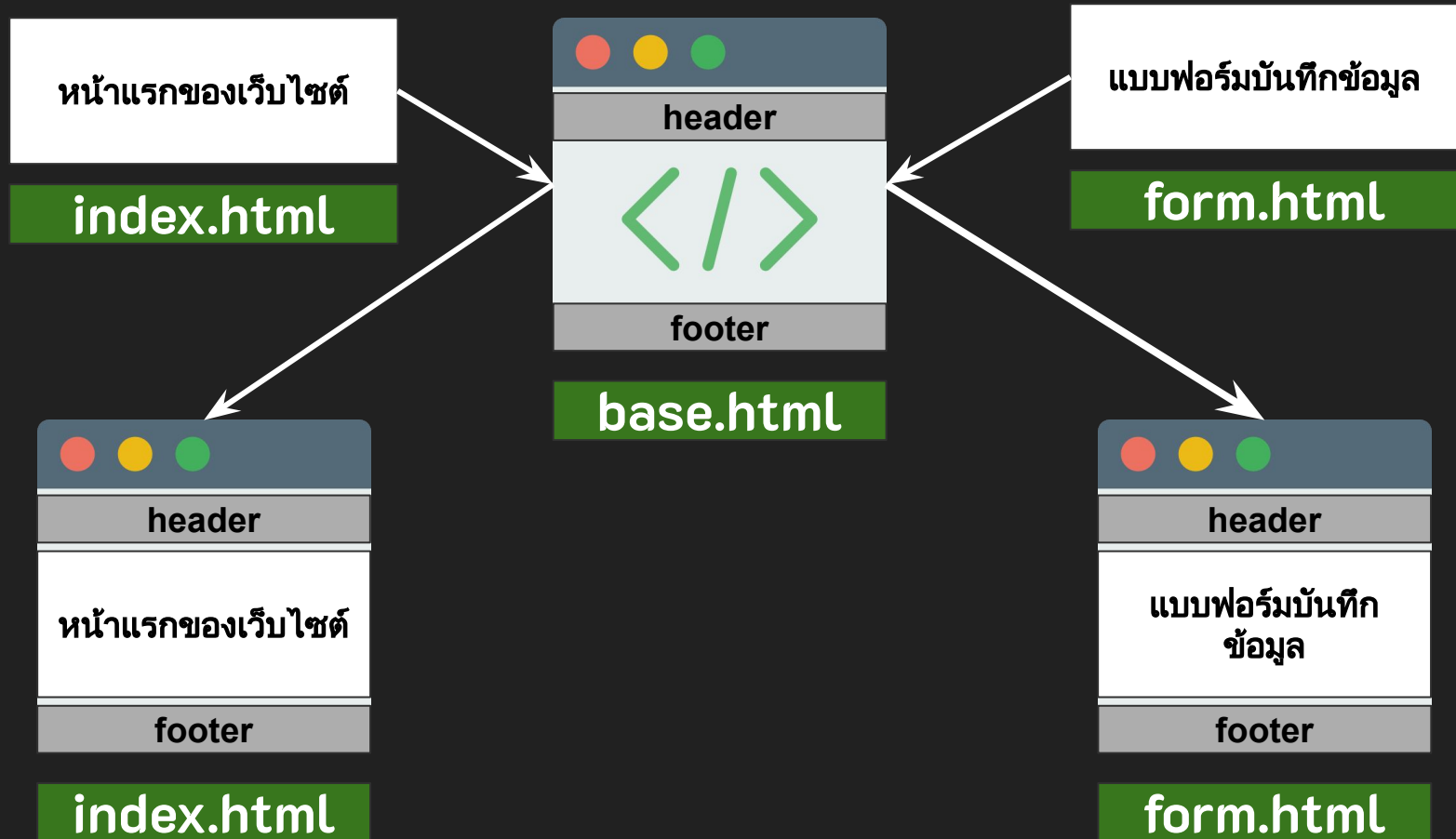
โครงสร้างคำสั่ง

```
{% block ชื่อพื้นที่ %}
```

เนื้อหา

```
{% endblock %}
```





Bootstrap



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



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

Bootstrap คืออะไร



Frontend Framework ที่รวม HTML, CSS และ Javascript เข้าด้วยกัน เป็นชุดคำสั่งที่ถูกพัฒนาขึ้นมาเพื่อกำหนดมาตรฐานหรือรูปแบบการพัฒนาเว็บไซต์ในส่วนของ User Interface (UI) และทำให้เว็บรองรับการแสดงผลขนาดหน้าจอที่แตกต่างกันอีกด้วย

จุดเด่นของ Bootstrap

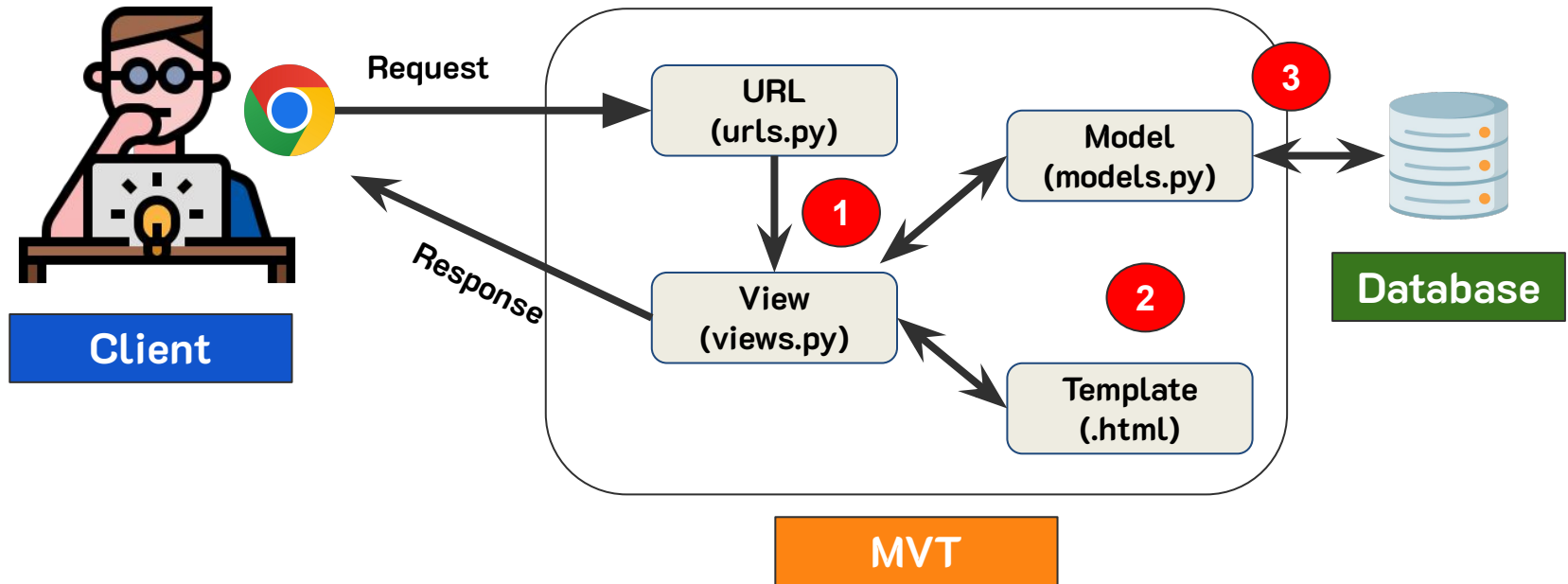


- มี UI ที่สวยงาม
- เรียนรู้ง่ายและเป็นที่นิยมใช้ทั่วโลกมีการพัฒนาและปรับปรุงต่อเนื่องจนถึงเวอร์ชัน 5
- ประหยัดเวลาในการพัฒนาเว็บไซต์
- รองรับการแสดงผลในอุปกรณ์ที่มีหน้าจอต่างขนาดกัน



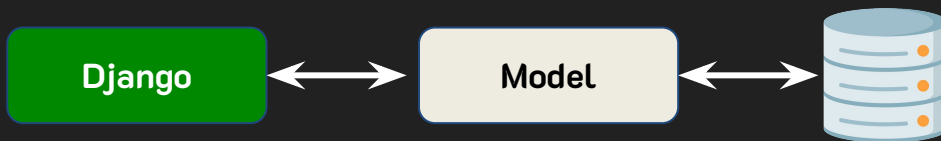
รู้จักกับโมเดล (Model)

รู้จักกับโมเดล (Model)



โมเดลคืออะไร

Model (M) คือ ส่วนที่เก็บข้อมูลของ Application โดยจะทำหน้าที่จัดการฐานข้อมูลแทนการใช้คำสั่ง SQL โดยตรง ผ่านการกำหนด Class และ Object ซึ่งจะเรียกส่วนนี้ว่า ORM (Object Relational Mapping)



การเชื่อมต่อฐานข้อมูล

โมเดลของ Django สามารถทำงานกับฐานข้อมูลได้หลากหลาย
เช่น MySQL , Oracle , PostgreSQL, SQLite เป็นต้น

ใน Django จะมี Driver Build-in ของ SQLite ซึ่งสามารถใช้งานได้เลย โดยไม่ต้องติดตั้ง Driver เพิ่ม ซึ่ง SQLite นั้นจะเก็บข้อมูลเป็นไฟล์เดียวชื่อว่า db.sqlite3 ซึ่งถูกสร้างอัตโนมัติในโปรเจกต์ของเรา



Settings.py

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.sqlite3',  
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),  
    }  
}
```

สร้างโมเดล (Model)



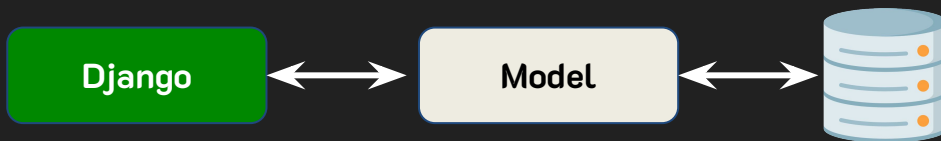
<https://www.youtube.com/c/KongRuksiamOfficial/>



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

สร้างโมเดล (Model)

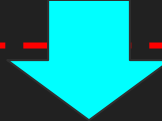
ขั้นตอนการสร้างโมเดลใน Django คือการกำหนดโครงสร้างตารางที่จะเก็บในฐานข้อมูล ผ่านการสร้าง Class ซึ่งจะเขียนในไฟล์ `models.py` ซึ่ง Class ที่สร้างนั้นจะสืบทอดคุณสมบัติมาจาก Class `models.Model`



models.py

```
from django.db import models
```

```
class ชื่อclass(models.Model):
```



```
from django.db import models
```

```
class Person(models.Model):
```

Django Field



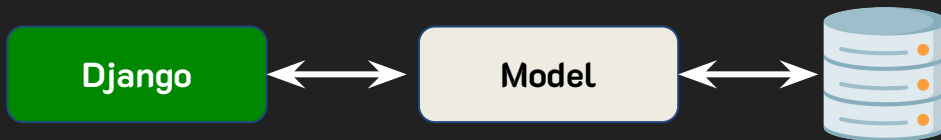
<https://www.youtube.com/c/KongRuksiamOfficial/>



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

Django Field

ฟิลด์หรือแอตทริบิวต์ คือ การกำหนดคอลัมน์ภายในตารางซึ่งจะถูกกำหนดใน Model พร้อมระบุรูปแบบให้สอดคล้องกับการจัดเก็บข้อมูลในคอลัมน์นั้นๆ ด้วย



ตัวอย่าง

Field	ความหมาย
CharField	เก็บข้อมูลอักขระ
IntegerField	เก็บตัวเลขจำนวนเต็ม
FloatField	เก็บตัวเลขทศนิยม
DateField	เก็บข้อมูลวันเดือนปี
BooleanField	เก็บข้อมูลค่าทางตรรกศาสตร์ (True/False)

ตัวอย่าง

```
from django.db import models

class Person(models.Model):

    name = models.CharField(max_length=50)
    age = models.IntegerField()
    date = models.DateField(auto_now_add=True)
```



Migration



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



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

Migration

คือการนำเอาโครงสร้างที่ระบุในโมเดลไปสร้างเป็นตารางในฐานข้อมูล โดยมีทั้งหมด 2 ขั้นตอน คือ

1. make migrations

สร้างไฟล์ migration ที่เก็บโครงสร้างตารางจากโมเดล

2. migrate

นำไฟล์ migration ไปใช้งาน

Admin Panel

Admin Panel

สร้างผู้ดูแลระบบ

```
python manage.py createsuperuser
```

username : ชื่อผู้ดูแลระบบ

email : อีเมลผู้ดูแลระบบ

password : รหัสผ่านเข้าสู่ระบบ

Register Model

คือ การกำหนด Admin ทำหน้าที่จัดการโมเดลในฐานข้อมูลได้

admin.py

```
from django.contrib import admin
```

```
from ชื่อแอป.models import ชื่อโมเดล
```

```
admin.site.register.(ชื่อโมเดล)
```



View Data

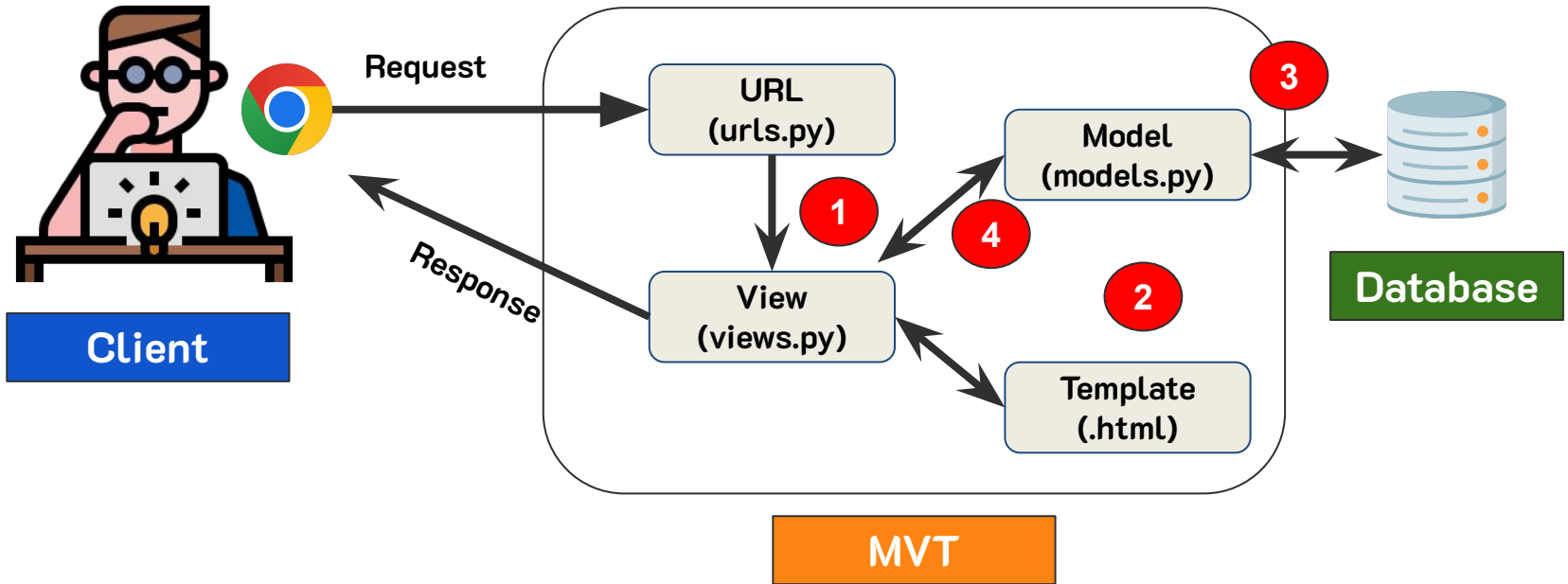


<https://www.youtube.com/c/KongRuksiamOfficial/>



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

View Data



View Data

ดึงข้อมูลหลายรายการ (QuerySet)

- all() - ดึงข้อมูลทั้งหมด
- filter(เงื่อนไข) - ดึงชุดข้อมูลพร้อมระบุเงื่อนไข

ดึงข้อมูลรายการเดียว (Single Object)

- get(เงื่อนไข) - ดึงข้อมูลพร้อมระบุเงื่อนไข

รับ-ส่งข้อมูล จากแบบฟอร์ม



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



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

การส่งข้อมูลจากแบบฟอร์ม

```
<form method="get|post" action="path">
```

```
// Element
```

```
</form>
```

การส่งข้อมูลจากแบบฟอร์ม

method="get หรือ post" (รูปแบบการส่งข้อมูล)

- **get** ส่งข้อมูลพร้อมแนบข้อมูลไปพร้อมกับ url (ไม่มีความปลอดภัยเพราะข้อมูลถูกมองเห็นและไม่ควรใช้งานร่วมกับข้อมูลที่เป็นแบบ sensitive data)
- **post** ส่งข้อมูลพร้อมซ่อนค่าข้อมูลระหว่างทางที่ส่งไป (มีความปลอดภัย)

การรับข้อมูลจากแบบฟอร์ม

หากส่งข้อมูลแบบ post ต้องระบุ `csrf-token` เพื่อป้องกันการโจมตี (Hack) ผ่านการป้อนข้อมูลในแบบฟอร์มจากผู้ไม่หวังดี เช่น ไม่สามารถแนบหรือรับ Input ที่เป็นรูปแบบ Script ได้ เป็นต้น

คำสั่งตรวจสอบและรับข้อมูล

- `request.method` คำสั่งสำหรับตรวจสอบรูปแบบการส่งข้อมูล
- `request.POST["name"]` คำสั่งรับค่าจากข้อมูลที่ส่งมาในรูปแบบ POST Method

คอร์สเรียนพัฒนาเว็บด้วย Django Framework (Real-World Projects)

การพัฒนา > การพัฒนาเว็บ > Django


พัฒนาเว็บด้วย Django Framework (Real-World Projects)

เรียนรู้การพัฒนาเว็บภาษา Python ด้วย Django Framework ตั้งแต่เริ่มต้นจนใช้งานจริง

ใหม่ 0.0 ★★★★★ (0 คะแนน) ผู้เรียน 13 คน

สร้างโดย [Kong.Ruksiam](#)

🌐 อัปเดตล่าสุด 12/2565 🇹🇭 Thai



ดูตัวอย่างหลักสูตรนี้

THB1,200

เพิ่มไปยังรถเข็น

ซื้อเลย

รับประกันยินดีคืนเงินภายใน 30 วัน

สิ่งที่คุณจะได้เรียนรู้

- ✓ เรียนรู้พื้นฐานการใช้งาน Django Framework
- ✓ ใช้งาน Bootstrap 5 ร่วมกับ Django Framework
- ✓ จัดการฐานข้อมูลด้วย Django Framework
- ✓ สร้างแอปแสดงผลข้อมูลแบบกราฟ
- ✓ สร้างระบบ Login & Logout (Authentication)
- ✓ สร้างระบบหมายเลขหน้า (Pagination)

<https://www.udemy.com/course/django-framework-real-world-projects/>