

---

# Web Server & Django

August 15, 2024

Abdullah Zayed

[contact@zayedabdullah.com](mailto:contact@zayedabdullah.com)



---

# Agenda

01 How Web Works

02 Request & Response

03 Web Server

04 Python Packages

05 Virtual Environment

06 Pipenv

07 Django

08 Django Directory Structure

09 Django Routing (URL)

10 What's Next...

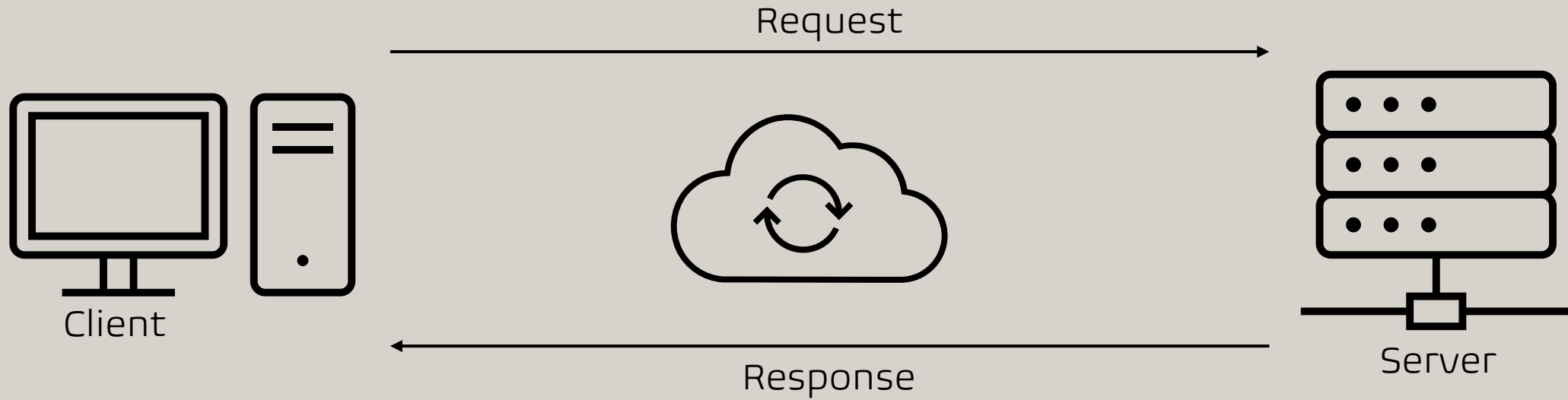
# 01 How Web Works

# 01 How Web Works

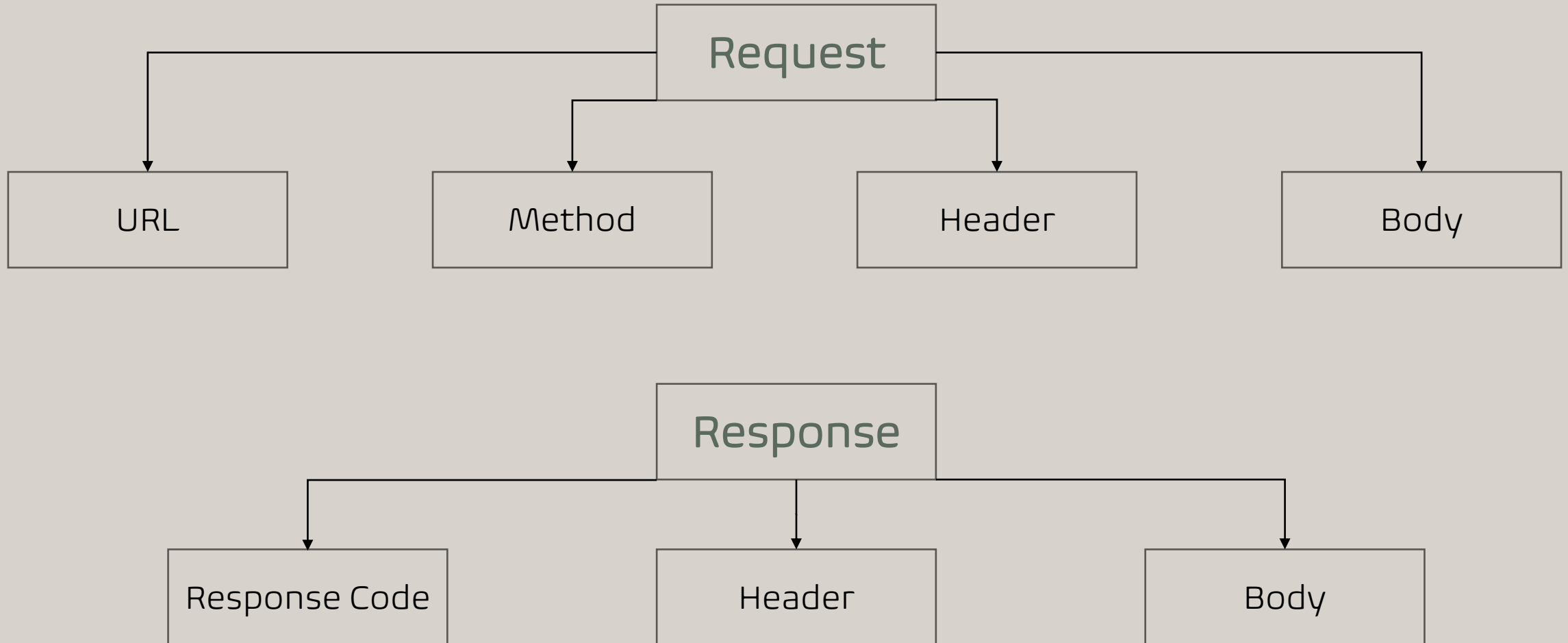


# 02 Request & Response

# 02 Request & Response



# 02 Request & Response



# @ Web Server



# 03 Web Server

A web server is a computer that stores web server software and a website's component files or another web server.

It connects with the network and supports data interchange with other devices.

# 04 Python Packages

# 04 Python Packages

Python Package is a **directory** (folder) that contains related Python files (modules) that work together to provide **certain functionality**.  
Examples: Django, Requests, Pillow, etc.

PIP is the **package manager** that lets us **install** and **manage** Python packages.

# 05 Virtual Environment

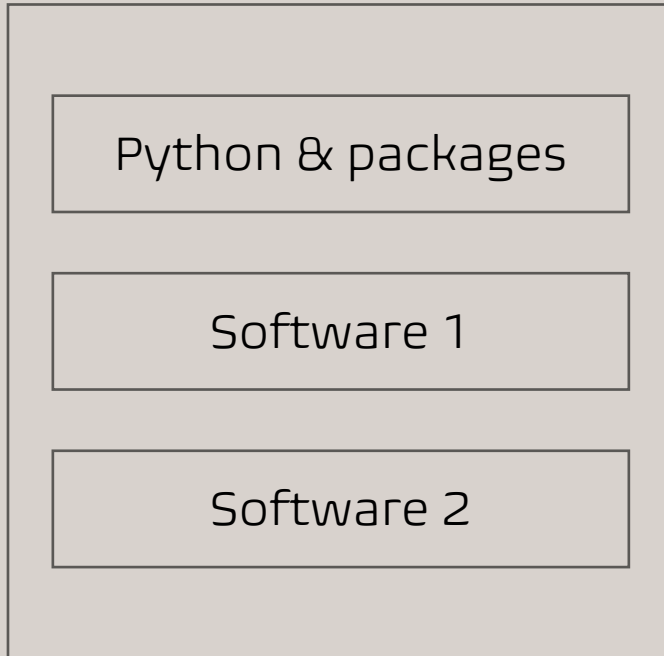
# 05 Virtual Environment

Virtual Environment is simply a **directory** (folder) containing “**Python Interpreter**” and **packages**.

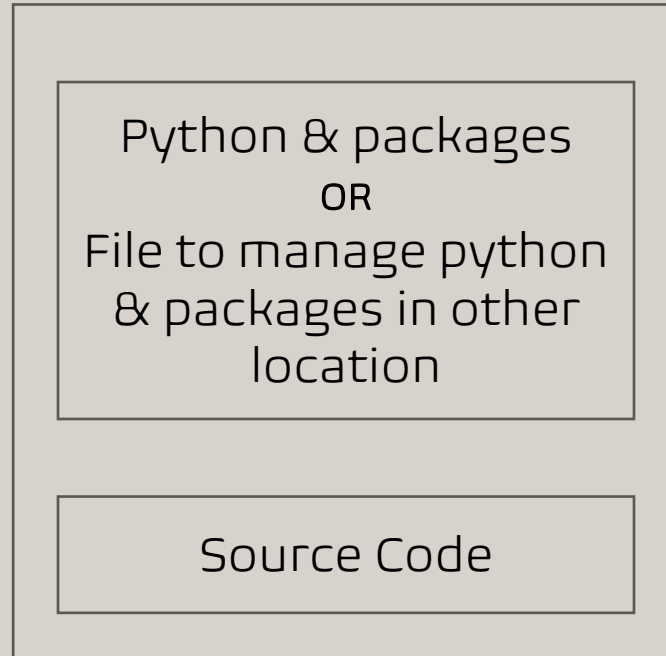
It contains the packages in a **single directory** (folder) specific to a project to avoid package clashes.

# 05 Virtual Environment

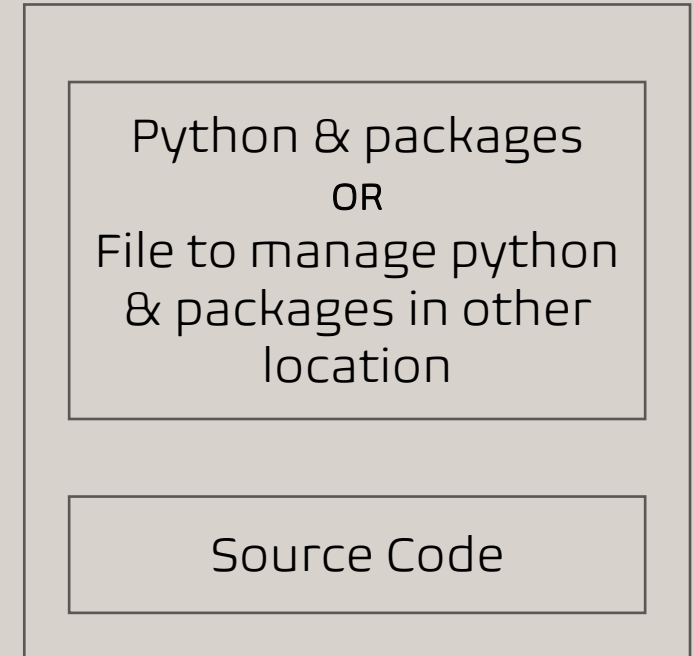
System



Specific Project 1



Specific Project 2



06  
Pipenv

# Pipenv

Pipenv is a Virtual Environment management **tool**.

It creates a folder in a **separate directory (folder)** for the virtual environment and creates ``pipfile`` and ``pipfile.lock`` to manage packages for the specific project.



# Pipenv

## Basic Commands

Basic Commands to manage virtual environment using pipenv:

- Pipenv shell start  
`"python -m pipenv shell"`
- Pipenv package install  
`"pipenv install <package_name>"`

@Django

# 07 Django

Django is a **Battery Included Python Framework** for the Web.

It forces a **structure** to follow for an easy and consistent development experience.

# 07 Django

## Basic Commands

Basic Commands to initialize and run a Django project:

- Django Project Initialize

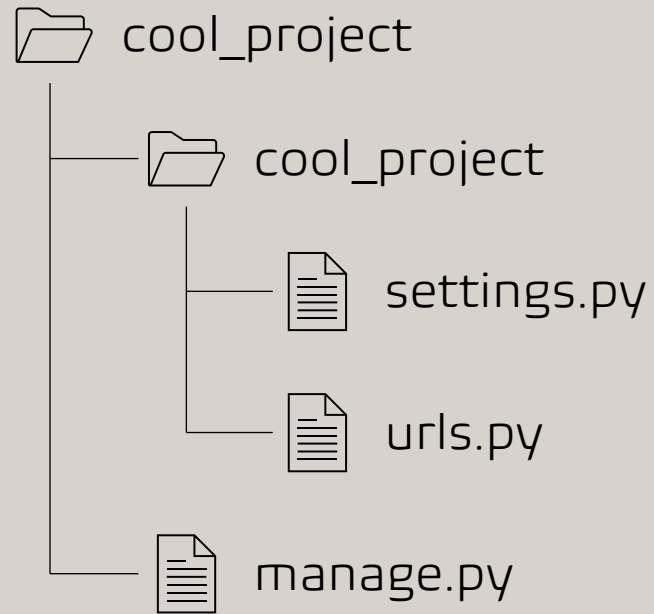
```
"django-admin startproject <project_name> <directory:optional>"
```

- Django Server Start

```
"python manage.py runserver"
```

# 08 Django Directory Structure

# 08 Django Directory Structure



# 09 Django Routing (URL)

# 09 Django Routing (URL)

Django manages routing by providing some value to a list called ``urlpatterns``.

It supports dividing the routes into **multiple** paths.



# 09 Django Routing (URL)

We need to know two functions for basic routing. Both functions are available in ``django.urls``.

1. `path` – Creates a URLPattern / Resolver
2. `include` – Includes another URL pattern list

# 09 Django Routing (URL)

## Path Function

Path Function takes the route (i.e. /the/url/path/) as the **first argument**.

For the second argument,

- It can take a **function** that handles the request-response.

OR

- Or it can take another list of **`URLPattern`**.

# 09 Django Routing (URL)

## Include Function

Include Function takes

- A list of **URL Pattern**

OR

- A string containing the **list** location of the URL pattern as its first parameter.

# 10 What's Next?

# 10 What's Next?

- Separating URL and Views
- Django Apps
- Using the app to structure files
- Django Templates
- Etc....

---

# The End



Thanks.

Zayed | <https://zayedabdullah.com>

[contact@zayedabdullah.com](mailto:contact@zayedabdullah.com)

[linkedin.com/in/abdullahzayed01](https://www.linkedin.com/in/abdullahzayed01)