Model-Template-View Architecture

In this lesson, we will first describe the MVC architecture, since it makes understanding the MTV architecture easier. Then, we will briefly cover the MTV architecture that most Flask developers favor.

WE'LL COVER THE FOLLOWING

Introduction

The **MTV**, or **Model-Template-View**, architecture is a variation of **MVC**, or **Model-View-Controller**, architecture. In the scope of this course, we will not be covering both of these architectures in detail, as it an extensive discussion on its own. However, we will have a brief introduction to both.

The Model-view-controller (MVC) architecture

The **MVC architecture** is a *software architectural pattern* in which the application logic is divided into **three** components on the basis of *functionality*. These components are called:

- **Models**
- **Views**
- **Controllers**

The MVC architecture is used not only for desktop applications but also for mobile and web applications.

mobile and web applications.

Components of MVC architecture

1. Models

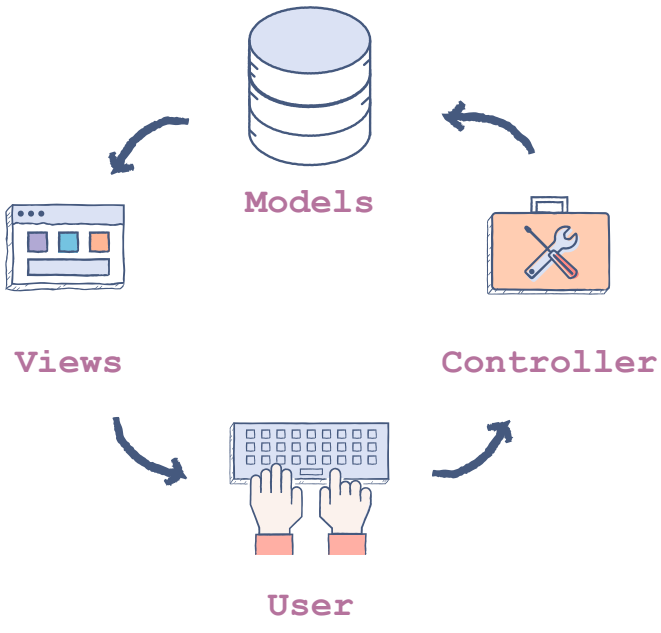
Models represent how data is stored in the database. It contains all the *data definitions* for your application (*the schema*). For example, if you are creating a library management system, then a model for books will be present. It will define all the information concerning each book, such as the name of the author, number of copies, and so on.

2. Views

Views are the components that are visible to the user, such as an output or a Graphical User Interface (GUI). In the case of a website, generally speaking, the views are the HTML pages.

3. Controllers

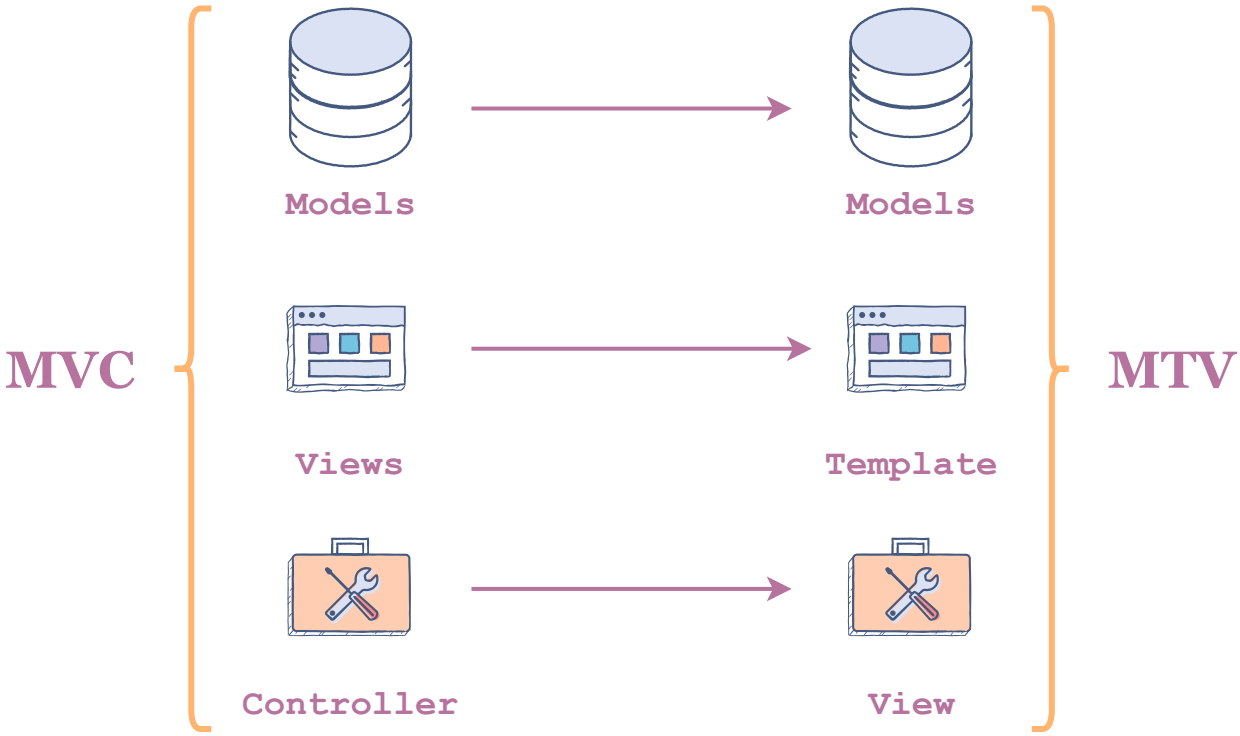
Controllers are the components that act as an interface between models and views. The controller interprets the user interactions, or inputs, and performs tasks on *models* before returning the appropriate data through *views*. In the case of a point-of-sale application, at checkout, the controller will be responsible for calculating the total bill based on the product purchased, *i.e., the model*. Also, it will show you the total amount via a GUI, *i.e., the view*.



MVC Architecture

Model-template-view (MTV) architecture

The **MTV** architecture is a slight variation of **MVC** architecture. On its own, Flask is a microframework; it does not contain built-in support for any architectural framework. However, the programmers who use the `Flask` framework have adopted the MTV architecture because another Python-based web development framework, called `Django`, introduced it. A *very rough* comparison between MVC and MTV is given below.



MVC to MTV

> ⚠ **Disclaimer:** The *"controller"* in **MVC** does **not** directly translate to the *"view"* in **MTV**. They have slightly different functions. To read more about the differences between MVC and MTV, please refer to the docs of the Django Framework (it introduced the term **MTV**).

In the next lesson, we will discover what `WSGI` and `Jinja` are.