What are the Frameworks?

An introduction to frameworks and their relationship to the front-end and back-end of a website.

WE'LL COVER THE FOLLOWING

In the previous chapters, we discussed both front-end and back-end, as well as the difference between the two. Now, we will look at how development on both ends and their integration can be made easier through web frameworks.

Introduction

A **web framework** is defined as a package made up of a structure of files and folders of standardized code (HTML, CSS, JS documents, etc.), which can be used to support the development of websites as a basis to start building a site. Essentially, frameworks provide some basic, standard starter code that allows developers to build the meat of their website on.

Most websites share a very similar structure, and the aim of frameworks is to provide this common structure as a starting point so that developers don't have to redo it from scratch and can reuse the provided code. Frameworks, therefore, serve to simplify the web development process.

MVC architecture

Before we introduce some of the common types, frameworks, can be categorized in, let's discuss the underlying architecture of frameworks. Most,

if not all, web application frameworks rely on the **Model View Controller** architecture. The pattern serves to separate the application logic from the user interface, thus forming the three components that the name is comprised of.

Model

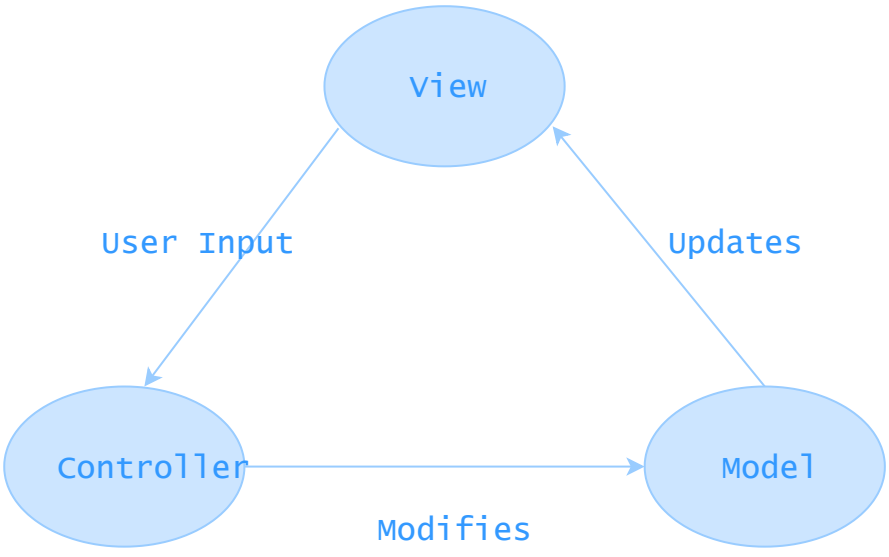
The Model stores all information about the content and structure of an application. Upon receiving user input data from the Controller, it communicates any required interface updates to the View component.

View

This refers to the application's front-end, or what is more commonly known as the *user interface*. It contains information about the layout and the way users can interact with any of its parts. The View receives user input, communicates it to the Controller for processing, and updates itself according to the instructions it, in turn, receives from the Model.

Controller

The Controller is an intermediary between the Model and the View. It receives user input from the View, processes it, and mediates required changes between the two.



The MVC Architecture

Quiz on the MVC architecture

Test your understanding of the MVC Architecture!

1 Which of the following best defines what the View component in an MVC application is?

COMPLETED 0%

1 of 2

Types of frameworks

There are both front-end and back-end frameworks that are very popularly used, and you may already have heard of them. Additionally, there are multiple isomorphic frameworks available that work on both the front-end and the back-end and serve as a bridge between the two.

Front-end (client-side) frameworks

1. Angular JS
2. Bootstrap
3. React.js
4. Backbone
5. Semantic-UI

Back-end (server-side) frameworks

1. Express (JavaScript)
2. Symfony (PHP)
3. Django (Python)

3. Django (Python)
4. Ruby on Rails (Ruby)
5. ASP .NET (C#)

Isomorphic (client-server) frameworks

1. Meteor JS
2. Lazo.js
3. Rendr

Now that we know which web development frameworks are the most widely used, in the following lessons, we will be discussing the specific features of each of them so you can make an informed decision about which framework you might want to begin learning for your web application.