
Model View Controller

An introduction for beginners

Separation of concerns

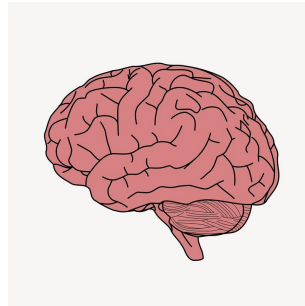
Just like in other levels of our programming we like to keep things organized and separated so working on one area won't affect another.

The three main categories of **MVC** are:

Business



Logic



Display





1. Model

The Model manages data and business logic

Defines the data structure is or more simply what the app should contain.

→ **Updates**

If there is a change the model informs the view

→ **Do you talk to the controller?**

If different logic is needed the model will notify the controller

→ **What data?**

The model will contain everything that is being experienced by the user. The data is whatever information is being dealt with



2. View

The view defines how the apps data should be displayed

→ **What do you see?**

The view is where confirmation of action happens for the user. Clicked on something -entered data - etc. The user should be aware of any changes and this is where it happens

→ **Displaying what they need**

Show what you need to and nothing more.



3. Controller

The controller contains the logic that updates the model and *(or)* the view in response to input from the user

→ **When do you talk to the model?**

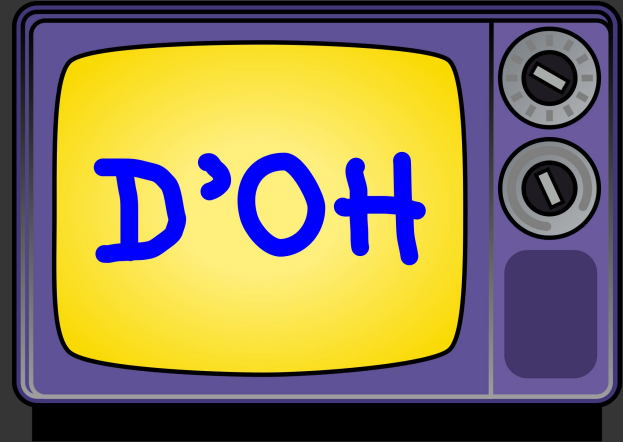
If the user adds information or makes a change or update the controller will tell the model to update its data

→ **When does the controller talk to the view?**

When none of that data changes just how it's being organized. EX; like putting an existing list into alphabetical order

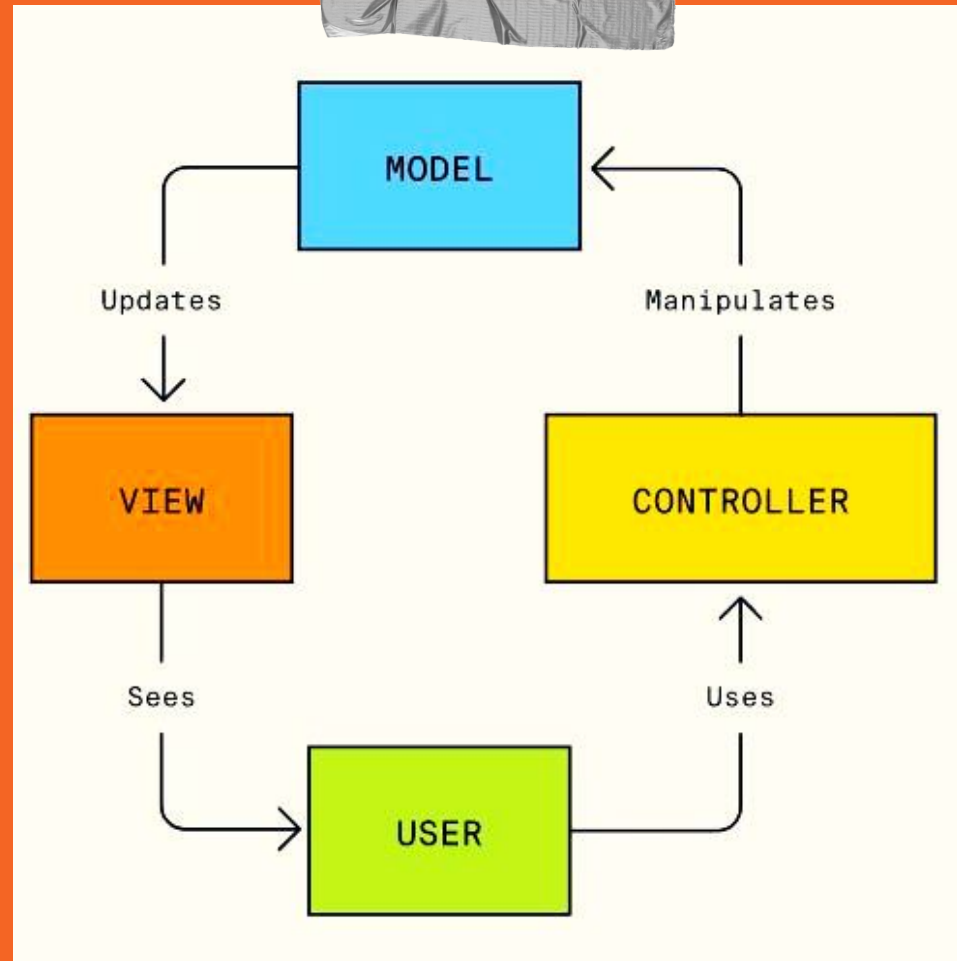
—

Won't someone please **think**
of the **users!!**



From the users perspective

1. User does something with the controller
2. This changes/updates the model
3. These updates are sent to the view
4. The view displays the updates
5. The user sees it





Feel free to check out the resources I used to make this presentation

Glossary

- [MDN](#)
- [Wikipedia](#)
- [Code Academy](#)