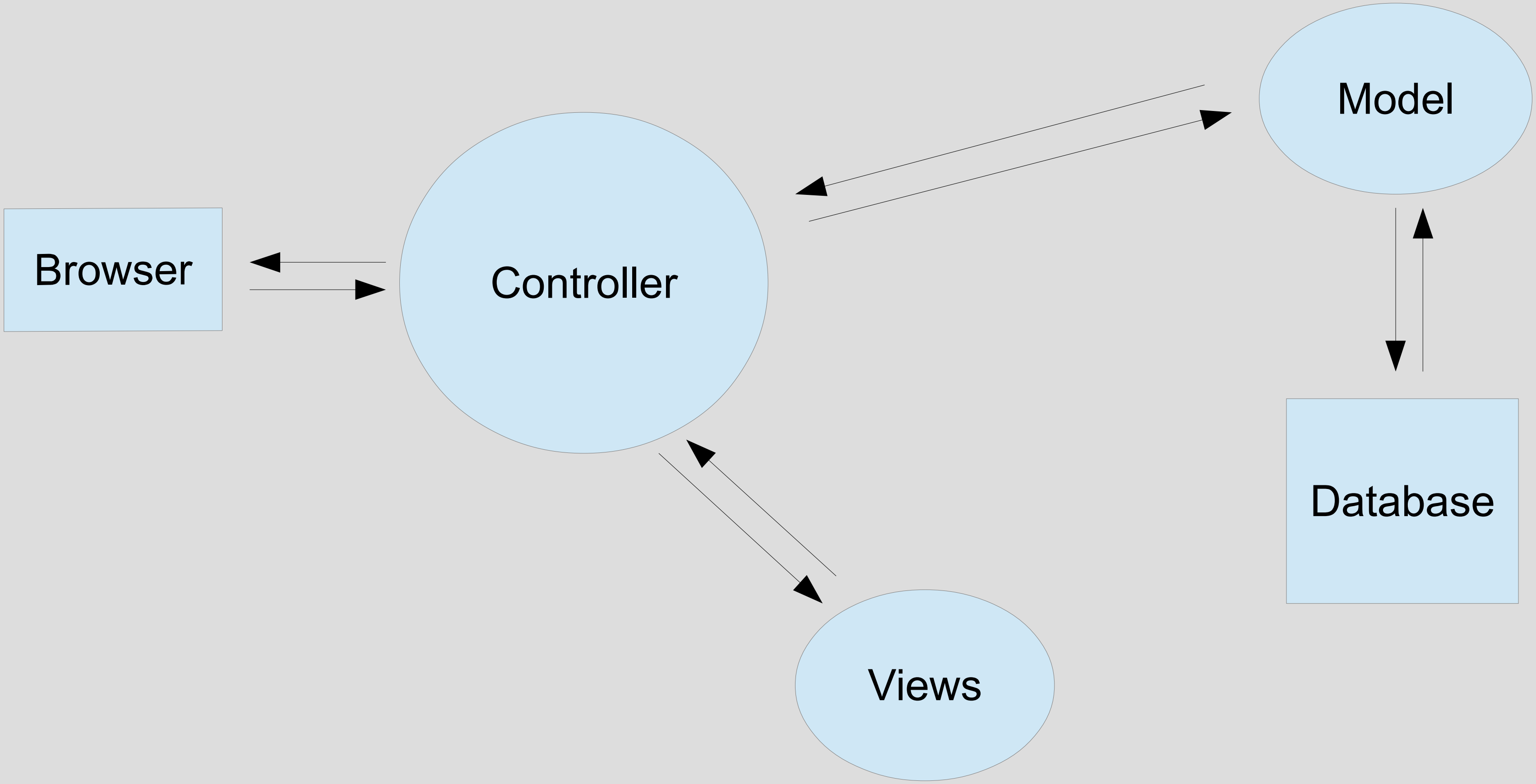
MVC: Model, View, Controller

MVC is an architectural pattern that separates an application into 3 main logical components: the model, the view and the controller.



Model

The model represents all the **data-related** logic.

- It retrieves information from the database.
- It writes information to the database.
- It is responsible for maintaining the state and integrity of the data: it needs to control what goes in the database and HOW it is formulated/constructed.
- The model should not interact directly with the view.

View

View is everything that pertains to client-side presentation.

- It is a “template” to render to the client.
- It includes all user-side logic/components.
- It gets its data from the controller.

Controller

It is the intermediary between the other components.

- It dispatches the requests to the right places.
- It feeds the information to the model and the model is responsible of feeding it into the database.
- It collects the requested information from the model.
- It feeds the information to the view and tells it what to render.

Why?

As you have “Separation of concerns” in HTML/CSS/JS, you have “Separation of concepts” for the back-end.

It allows:

- Better code organization.
- More modularity.
- Easier maintainability.

By keeping everything separated, developers can work together on different pieces without playing all in the same file. It also helps code reusability as every part is responsible for a small action, so you can plug-n-play the different snippets of codes more easily.

Troubleshooting

- If you have a problem with the rendering on the webpage, you go to the views.
- If the problem is in data entry, you go to the model.
- Everything is separated so you can isolate the components and test them individually.

Schemas

Data schemas are the skeletons of the data.

When storing information in the database, it is important to format the data in a way that protects the integrity of the data. It also needs to make it retrievable and usable.

A schema is mostly there to enforce and validate fields before data is stored in the database.

A schema in “human words” could be:

- Field “name” is mandatory and needs to be a string.
- Field “date” is mandatory and needs to be “YYYY-MM-DD”
- Field “age” is optional BUT needs to be a number if entered.