

---

---

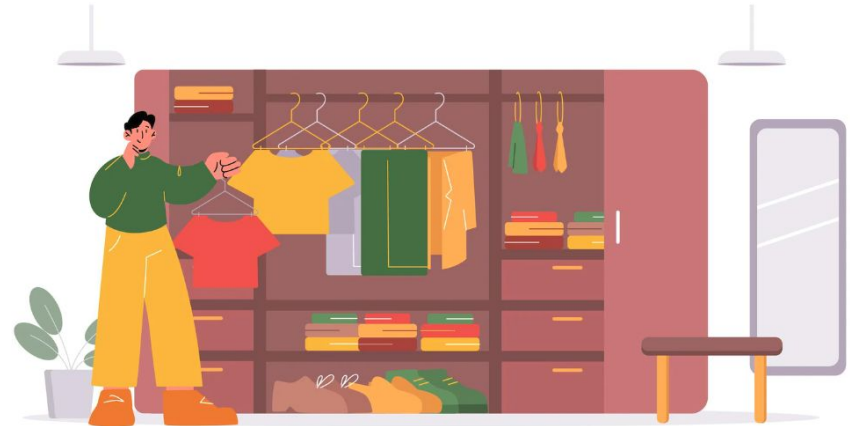
# MVC in Programming

A brief overview to help understanding

---

# MVC Architecture

We use ideas like MVC (Model, View, Controller) in our daily lives when we don't even know it's happening. Subconscious processes that we take for granted follow this model, view, controller architecture.



—

**Let's take a look at a process  
that most of us perform  
every day!**

# — **Scenario:** You've begun your day...

You've eaten breakfast.

You've showered.

You've brushed your teeth.

Now what?

We need to get dressed for  
our day!

What to wear???



# Let's break this down

- 1) Pick out pants, shorts, dress, etc
- 2) Pick out shirt, sweatshirt, sweater, etc
- 3) Pick out some socks
- 4) Pick out some shoes



**WAIT!!!**

What in the world does this have to do with MVC?

# Let's imagine your closet:

Holds all of these clothes, all of these different materials, to keep us warm in the winter, cool in the summer, all of these different patterns, colors, etc. to pull out when we need them. We have put all of these clothes in our closets, to store until we want or need them...sort of like a Database?? Well in this case, this is going to be our **Model**.







## How are these clothes getting on us now?

Seems simple, but it is a very complex series of events.

What do we need to pick...what's the temperature outside, what are we doing today...does my sock go on my hand...do I wear this "shirt" on my legs?

# That brain of ours is surely miraculous:

Our brain tells us to look at the weather, check our calendars, place the shirt over our heads and push our arms through, “is this a left shoe or a right shoe?”...that’s for the brain to tell us. It tells us where these items we are grabbing from the Model are needing to go...sounds like the **Controller**, huh?



# Controller

The controller directs traffic!

It decides where the information, or clothes in this case, need to go. It sends the information where it needs to go for the use that it has.

The controller is the most important part of the process, but without it or the model (or the view, next), none of this will work!

## Ok, we have the Model and the Controller, but how does the View tie in?

After you get dressed for the day, you have your shirt and pants on, you've tied your shoes...you're ready to go! Look down.

Do you see how everything looks together with you in there? No. You have to look into the mirror and get the whole picture of what others are seeing.

# The View:

What does the user see?  
How has the information been  
rendered (or put on in this case)?



# View

The view is how all the information has been rendered and how it is perceived by the end-user

How does the information look displayed?  
That is where the view has taken over.

The Controller has taken the information from the Model, whether it be the information from the database, the way it's supposed to look, or both, and put all of these things together in the View so that the end-user can visualize all of the materials together.

# But what if it rains, or if it is snowing outside?

We can change the components to fit what we need.

Need an umbrella? Take it.  
Need a knit hat?  
Put it on your head.

The MVC architecture allows us to “plug and play” different pieces to get the end result we want.