



comet_11
@payalmanghnani



**What
is
MVC?**



comet_11
@payalmanghnani

Model View Controller

comet_11
@payalmanghnani

The core concept of MVC :-

is to separate the
business logic, user
interface and data
modelling from each
other .





comet_11
@payalmanghnani

Model (Castle Plan):

Imagine you're the chief sandcastle builder. You draw a plan showing where to put the walls, towers, and even the moat. This plan is like the Model, which knows how everything should be built.



comet_11

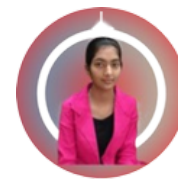
@payalmanghnani

View

(Decorations):

Your friends come to help and bring all kinds of cool seashells, flags, and colorful pebbles. They use these to make your sandcastle look like a real palace. This decorating is like the View, which makes things look super nice.



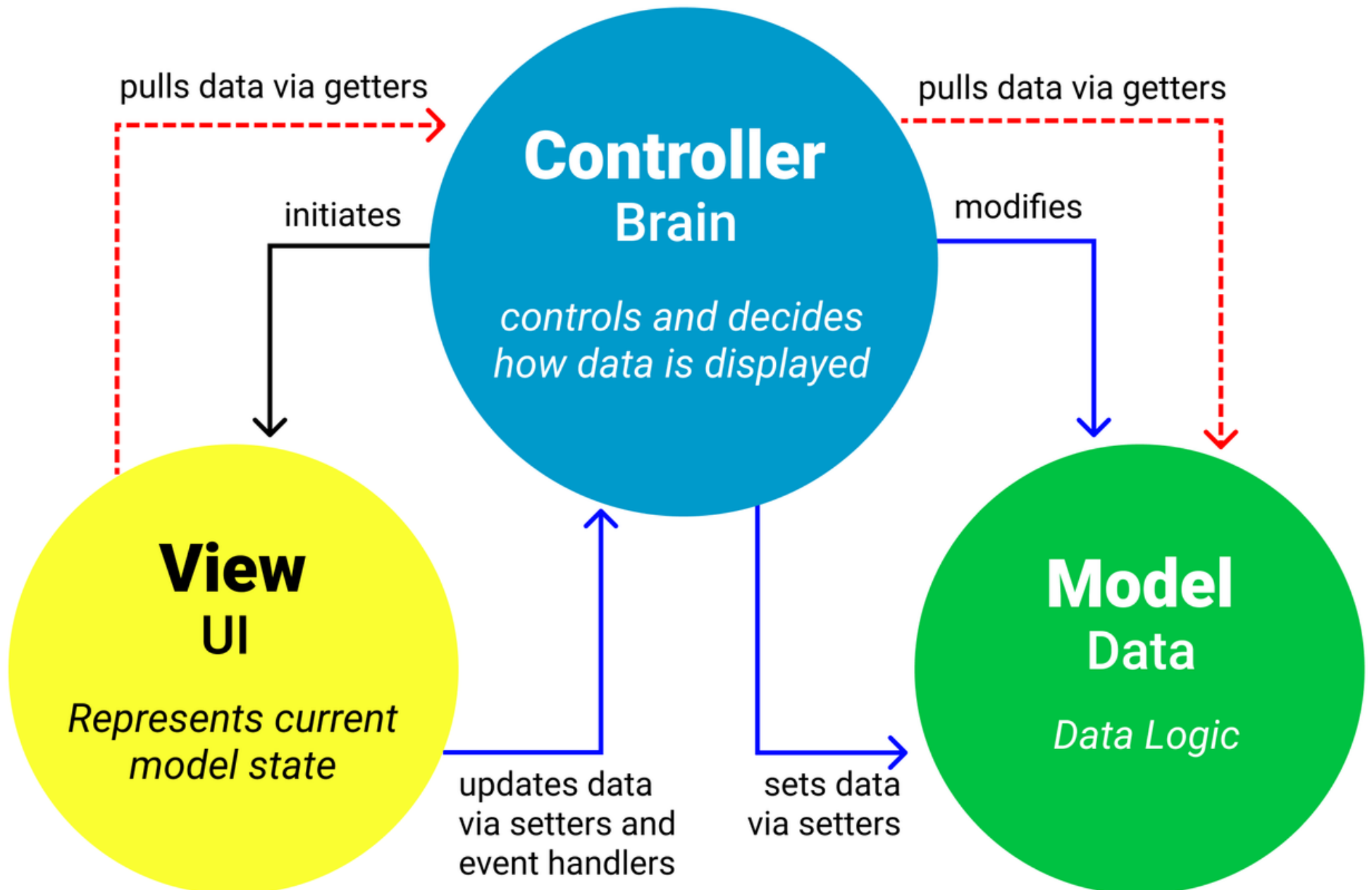


comet_11
@payalmanghnani

Controller (Play Director):

Now, you need someone to lead the fun. Your friend takes charge and tells everyone which part of the castle to work on. They make sure everyone's having fun and that the castle is built just right. This friend is like the Controller, making sure things go smoothly.

MVC Architecture Pattern





comet_11
@payalmanghnani

Why Should You Use MVC?





comet_11
@payalmanghnani

**Using MVC to
build your
sandcastle (or
create software)
makes everything
awesome:**





comet_11

@payalmanghnani



- It helps everyone know what to do (Model).
- It makes your creation look fantastic (View).
- It ensures everything goes smoothly (Controller).



comet_11

@payalmanghnani

Flow Of System



comet_11
@payalmanghnani



**Breaking tasks
into three main
parts helps
engineers write
high-quality
code that can
be used again.**

**This also makes
it simple to
work on a
project –
changing one
part doesn't
mess up the
others.**

comet_11
@payalmanghnani

The user uses the view to see data and talks to the controller through the view to give input.

The controller acts like a brain. It manages user data and model data, sending the final data to the view for the user to see





comet_11
@payalmanghnani

**Example of how the MVC
architecture pattern can
be used to build a simple
calculator application:**





comet_11
@payalmanghnani

- **The View would be responsible for displaying the calculator's user interface, such as the buttons and the display.**



- **The Controller would be responsible for receiving input from the user, such as when the user clicks on a button. The Controller would then send a request to the Model to calculate the result.**

- **The Model would be responsible for storing the numbers and performing the calculations.**





comet_11
@payalmanghnani

**The MVC
architecture
pattern can be used
to build any type of
application, not just
web applications. It
is a versatile and
powerful pattern
that can help you to
build maintainable
and extensible
applications.**





comet_11
@payalmanghnani



Was it *helpful*?



Like



Comment



Share



Save

Thanks for your support!



**Tell me
in the
comments**