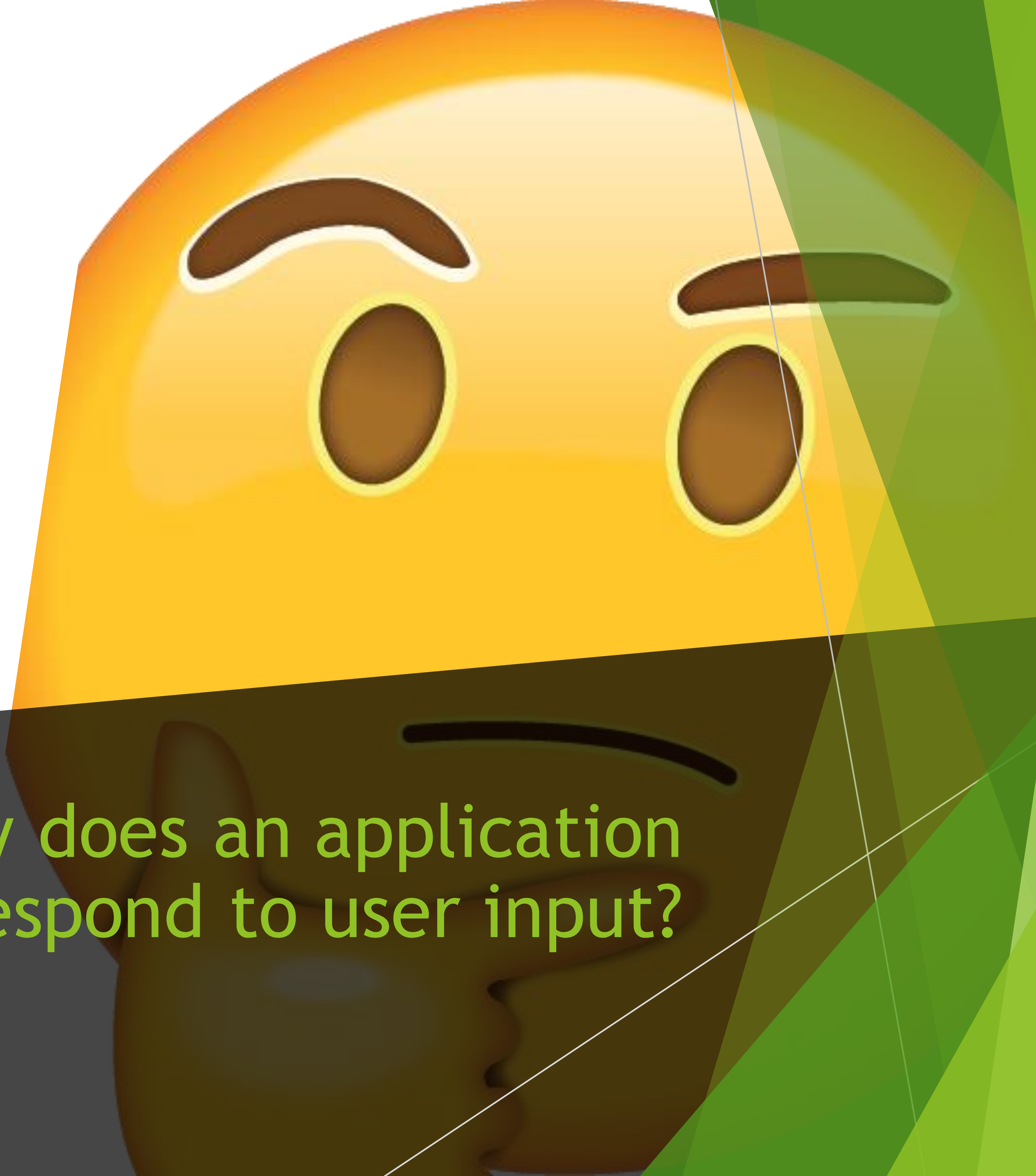
Interaction Programming: Model View Controllers & Event Records

3-31-2020, Jay Lin (jial8), CSE340 TA



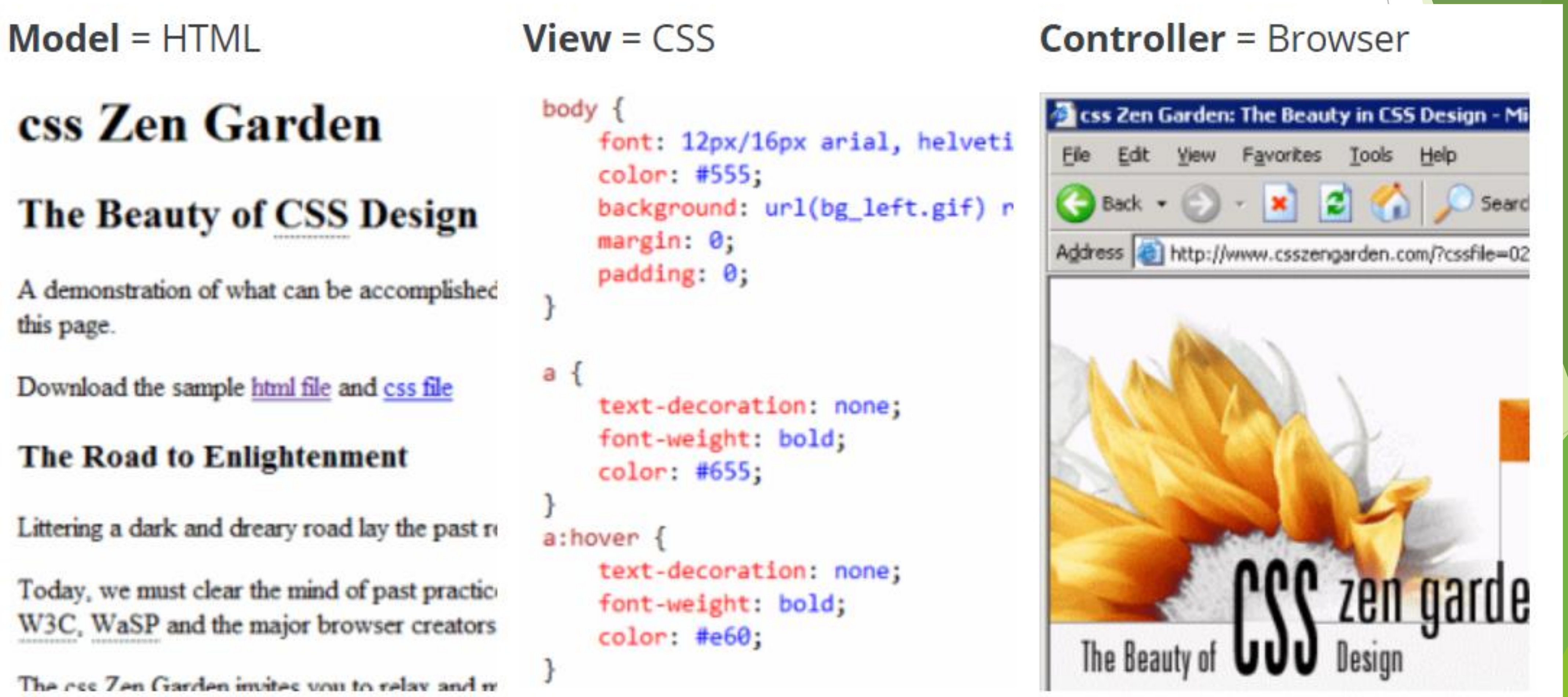
How does an application respond to user input?

How does an application respond to user input?

- We can visualize this process with a **Model View Controller** diagram.
  - MVC is a high-level system architecture
  - Can be used to model app’s overall structure
- Why?
  - Makes your code more reusable in case you need to change/implement a new feature
  - Ex: you want to add this cool button you already made to another app but only copy its appearance (the View component)

- **Model** → bare-bones, the skeleton of content
- **View** → what the user would see/interact with
- **Controller** → accepts input, tells model what to do

Similar to MVC example: HTML/CSS/Browser relationship



Model = HTML

css Zen Garden

The Beauty of CSS Design

A demonstration of what can be accomplished
this page.

Download the sample html file and css file

The Road to Enlightenment

Littering a dark and dreary road lay the past r

Today, we must clear the mind of past practic
W3C, WaSP and the major browser creators

The css Zen Garden invites you to relax and m

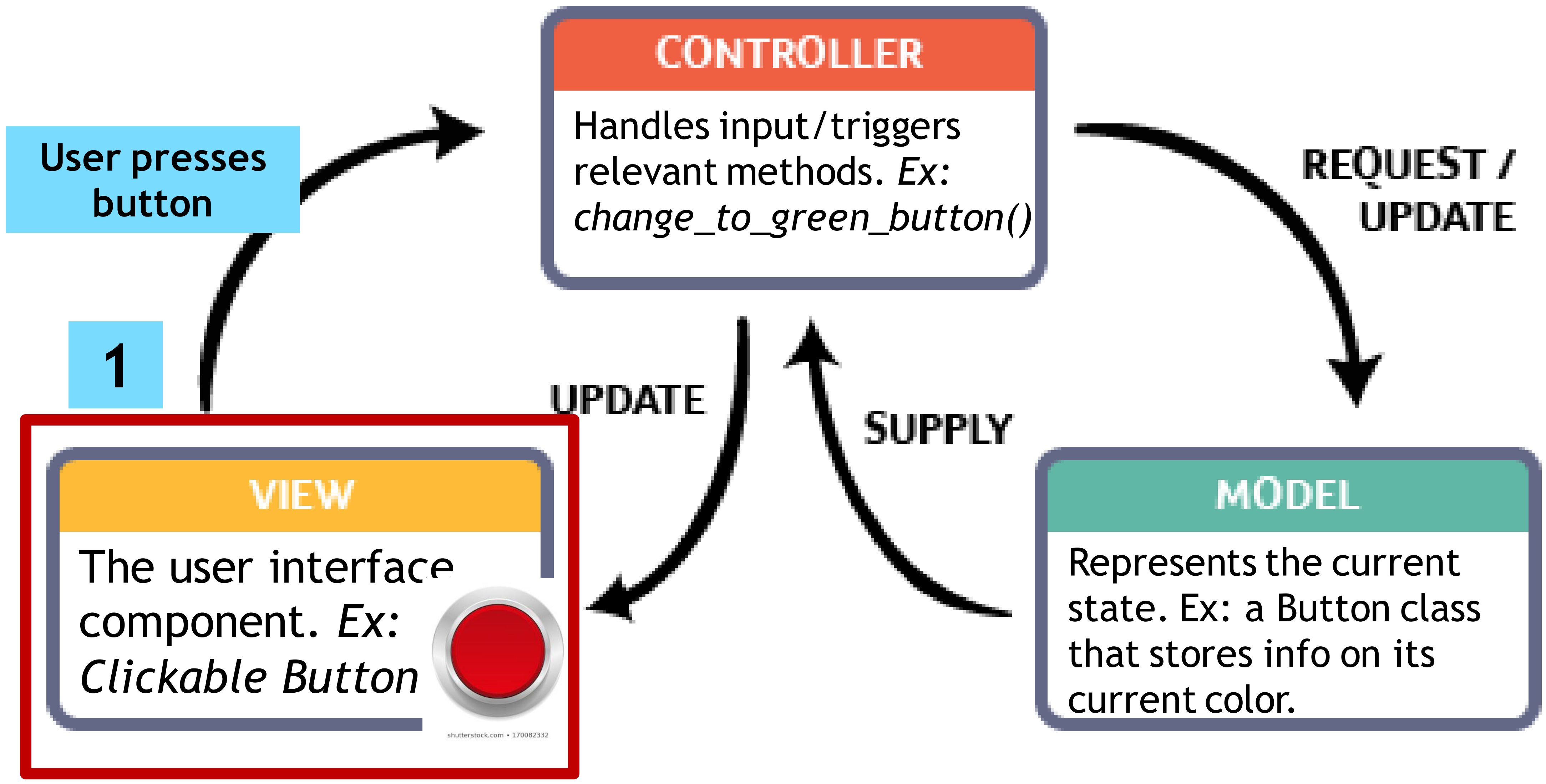
View = CSS

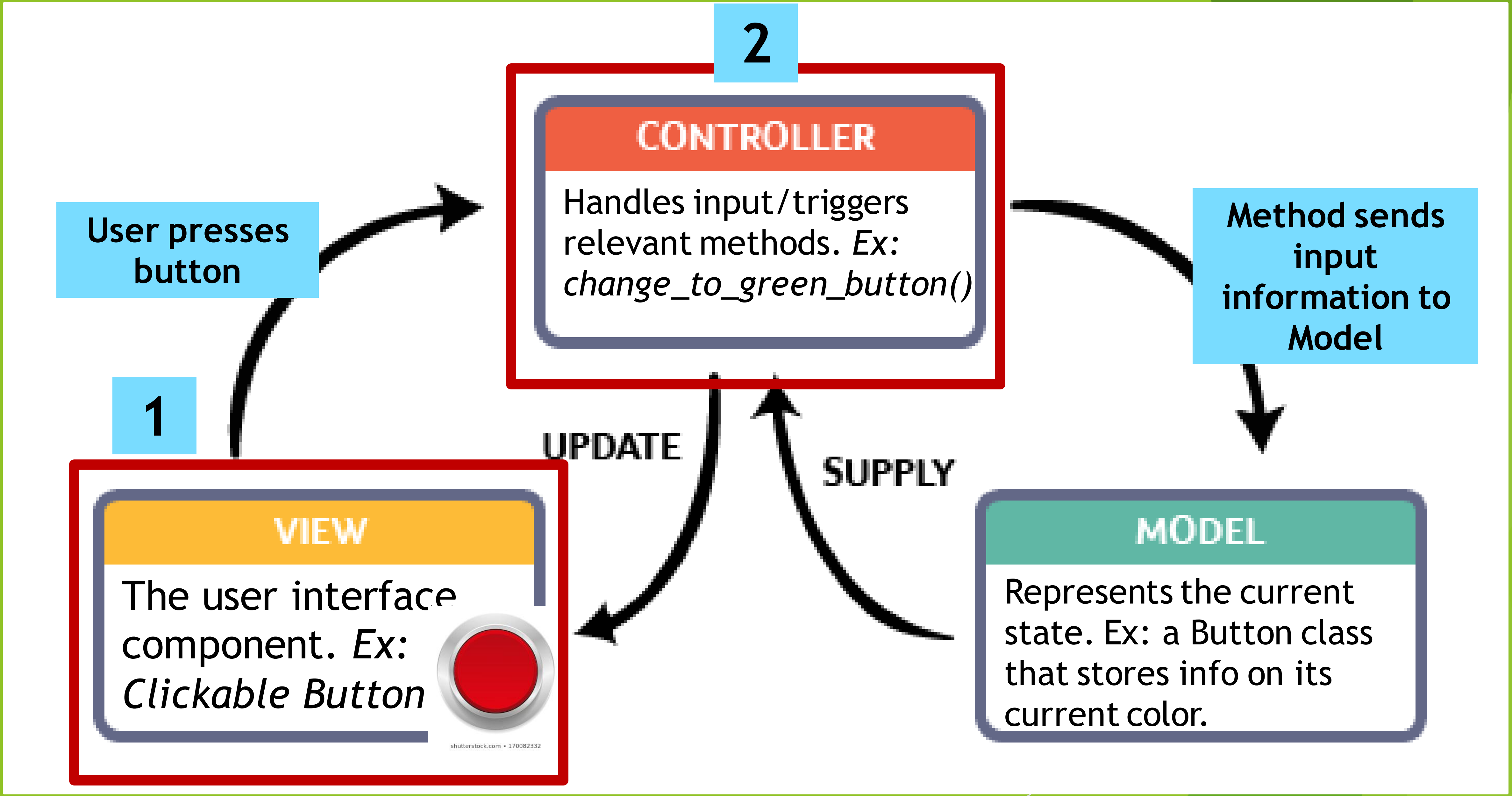
```
body {
    font: 12px/16px arial, helveti
    color: #555;
    background: url(bg_left.gif) r
    margin: 0;
    padding: 0;
}


a {
    text-decoration: none;
    font-weight: bold;
    color: #655;
}
a:hover {
    text-decoration: none;
    font-weight: bold;
    color: #e60;
}
```

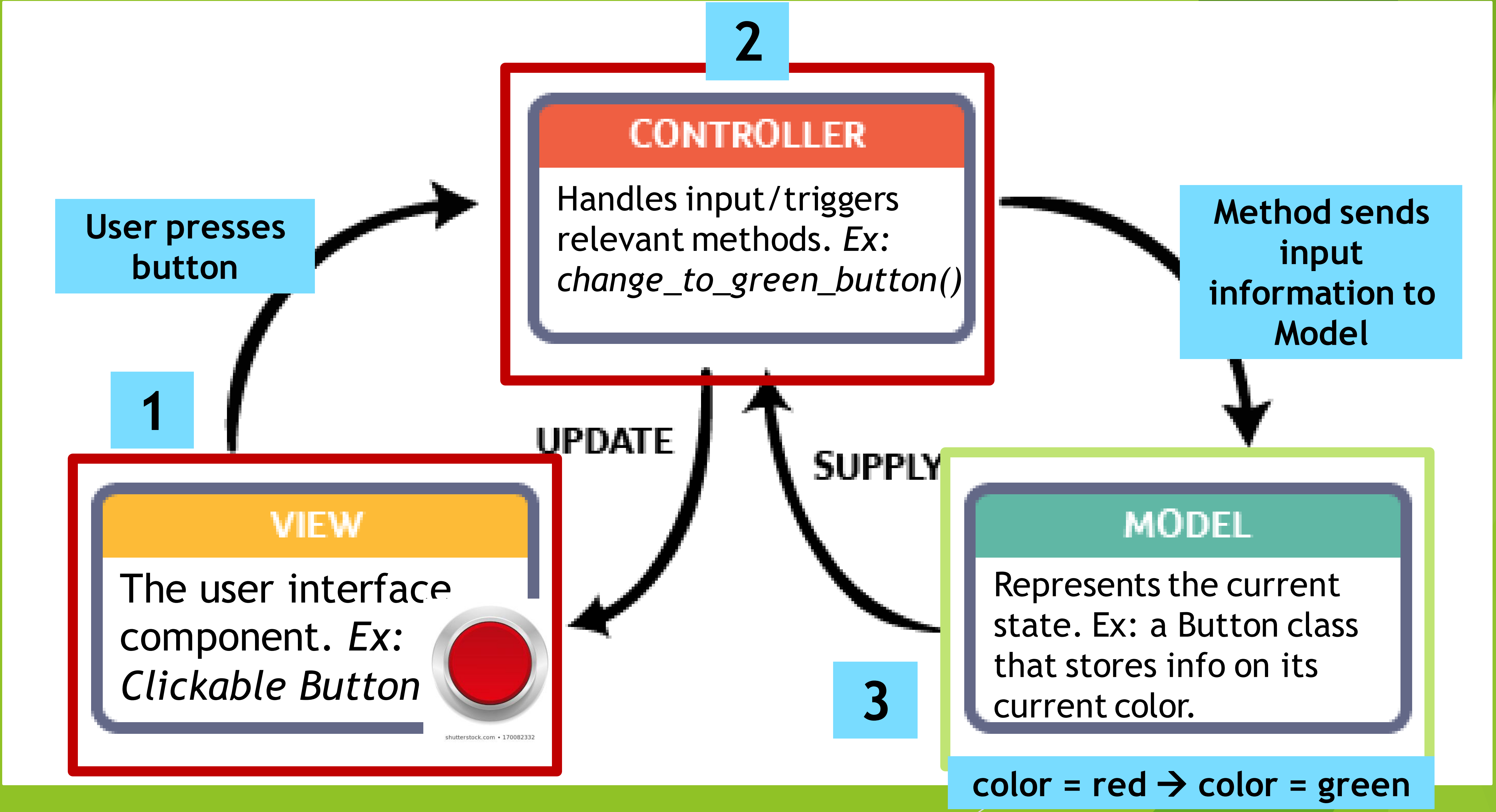
Controller = Browser

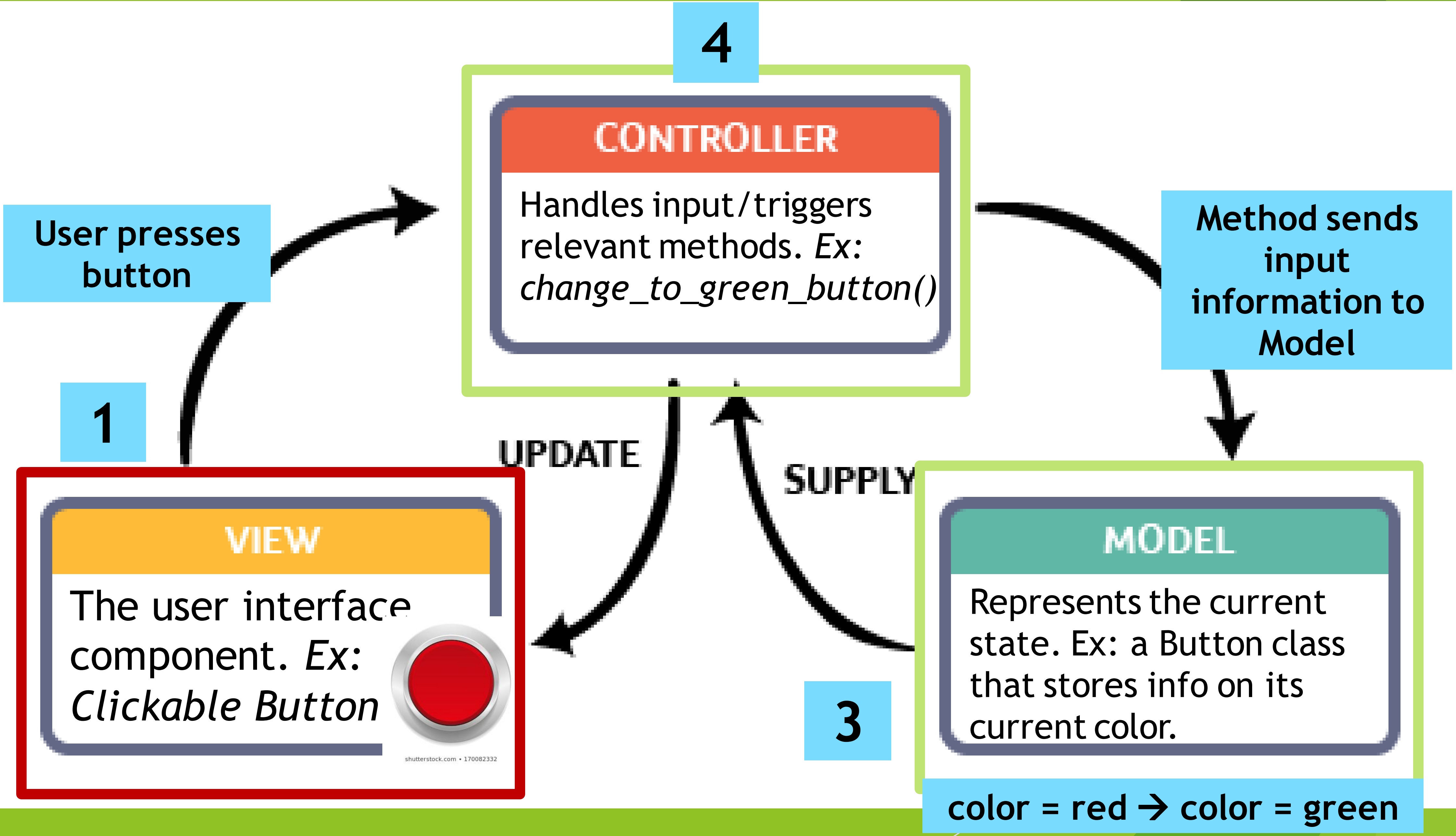
css Zen Garden: The Beauty in CSS Design - Mi

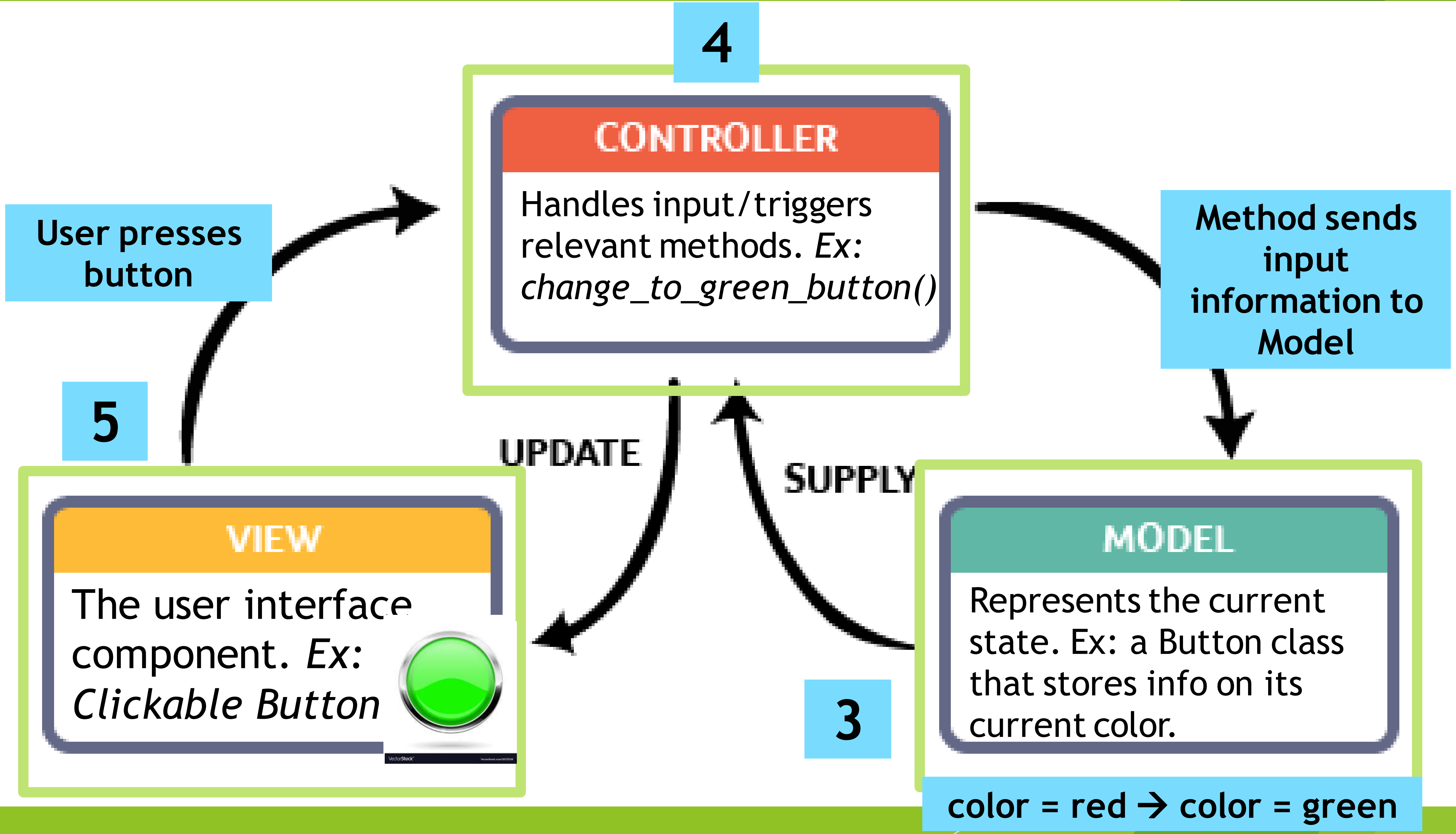
[Image source](#)


User presses button
1
CONTROLLER
Handles input/triggers relevant methods. *Ex: change_to_green_button()*
REQUEST / UPDATE
UPDATE
SUPPLY
VIEW
The user interface component. *Ex: Clickable Button*
shutterstock.com • 170082332
MODEL
Represents the current state. Ex: a Button class that stores info on its current color.


2
CONTROLLER
Handles input/triggers relevant methods. *Ex: change_to_green_button()*
User presses button
Method sends input information to Model
1
UPDATE
SUPPLY
VIEW
The user interface component. *Ex: Clickable Button*
shutterstock.com • 170082332
MODEL
Represents the current state. Ex: a Button class that stores info on its current color.


2
CONTROLLER
Handles input/triggers relevant methods. *Ex: change_to_green_button()*
User presses button
Method sends input information to Model
1
UPDATE
SUPPLY
VIEW
The user interface component. *Ex: Clickable Button*
shutterstock.com • 170082332
MODEL
Represents the current state. Ex: a Button class that stores info on its current color.
3
color = red → color = green


4
CONTROLLER
Handles input/triggers relevant methods. *Ex: change_to_green_button()*
User presses button
Method sends input information to Model
1
UPDATE
SUPPLY
VIEW
The user interface component. *Ex: Clickable Button*
shutterstock.com • 170082332
MODEL
Represents the current state. Ex: a Button class that stores info on its current color.
3
color = red → color = green



Handles input/triggers relevant methods. *Ex: change_to_green_button()*

The user interface component. *Ex: Clickable Button*

Represents the current state. Ex: a Button class that stores info on its current color.

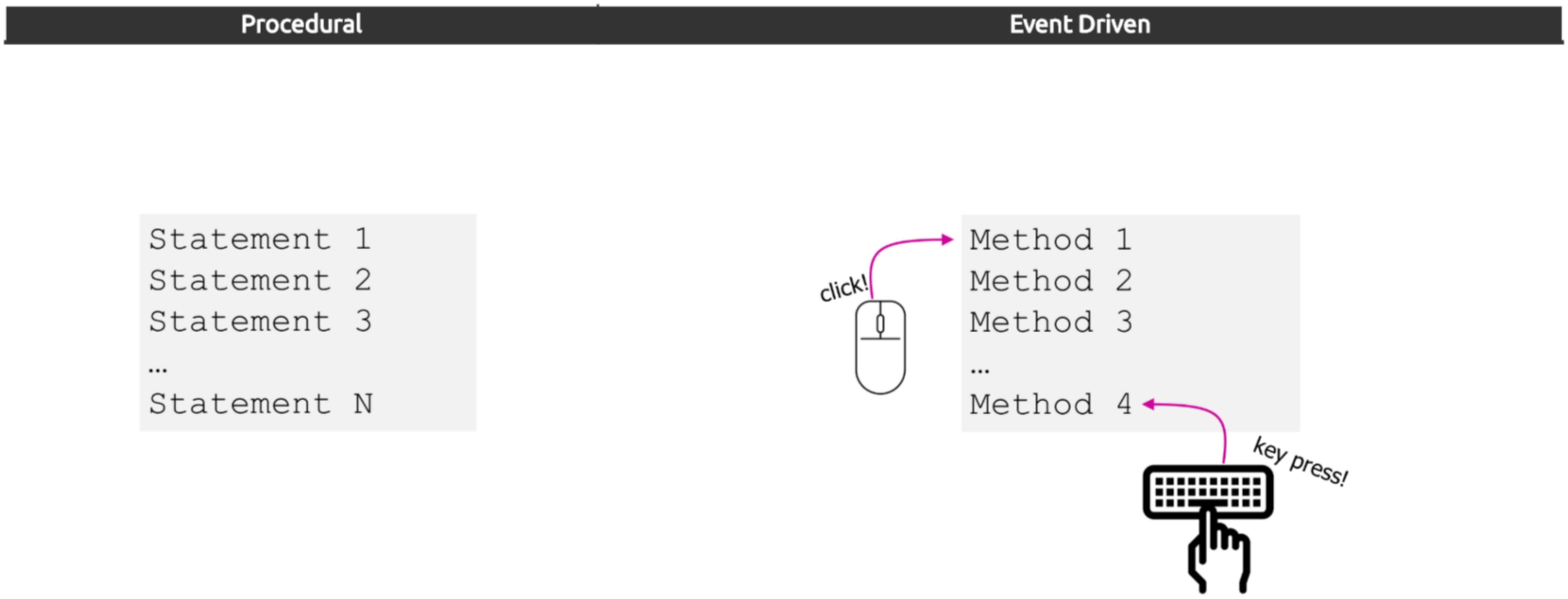
color = red → color = green

But… (generally speaking) how did the Controller know that the user pressed *the button* and not anything else?

Is there an…

Event Record (1/2)

- Apps are event-driven, meaning as the user interacts with the app, the code doesn't just go Line 1 Line 2 Line 3…



Taken from: https://cse340-20wi.pages.cs.washington.edu/website/slides/wk04/events.html#11

Event Record (2/2)

- What does the controller need to know?
  - What: Event Type (ex: finger press down (on computer → mouse moved))
  - Where: Event Target (ex: Button)
  - When: Timestamp (ex: 00:00:05 after opening the app)
  - Value: Event-specific variable (ex: user's touch coordinates)
  - Context: modifiers or "what was going on when this happened??" (ex: two-finger tap, (or if on computer → CTRL SHIFT ALT…etc))

What if we want: User presses button → *button turns green AND a ☺ View appears on the screen?*

Solution: Have the Button View and ☺ View have “listeners” for Value: user’s touch in that specific location, looking at the Event Record

Summary

- **MVC**: a way to structure your app, shows how user interaction affects the app
- **Events**: the user did something within your app
- **Event Records**: A method of storing the information describing the circumstances of that user interaction