Hooks React JS

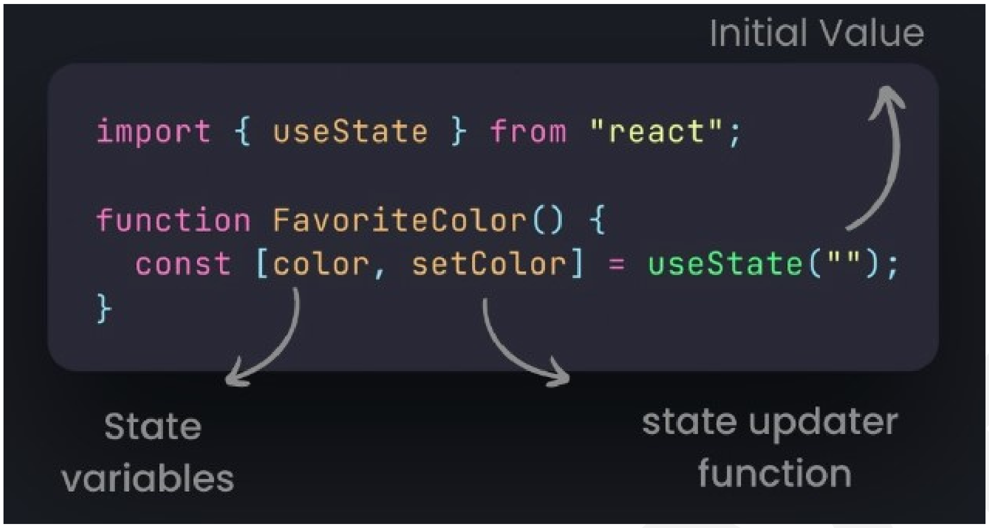
- ## What is Hooks?
  - In React, "**hooks**" are functions that allow you to use state & other React **features (life cycle methods)** in functional components.
  - They were **introduced** in **React version 16.8** and provide a more concise and **reusable way** to manage state and side effects compared to class components.
  - **class components** are generally no longer needed.(no plans to remove)

- ## Hooks Rules
  - You must **import** Hooks from react.
  - Hooks can **only** be called inside React **function components**. (Not in class components)
  - Hooks can only be **called at the top** level of a component.
  - Hooks cannot be conditional.

useState() React Hooks

- This hook allows you to add **state** to functional components. It returns a **state variable** and a **function to update that variable**.
- We initialize our state by **calling useState** in our function component.
- useState accepts an **initial state** and returns two values:
  - The **current state**.
  - A function that **updates the state**.

- **What can State Hold?**
  - The useState Hook can be used to keep track of
    - strings
    - numbers
    - booleans
    - arrays
    - objects, and
    - any combination of these!



```
import { useState } from "react";

function FavoriteColor() {
  const [color, setColor] = useState("");
}
```

- We are **destructuring** useState from **react** as it is a named export. (useState returns two values)
- **variables** that can be named anything you would like

- ## Read & Update State
  - **Read State**
    - After initialize we can include our **state** anywhere in our component.
  - **Update**
    - To **update** our state, we **use our state updater function**.
  - **Note:**
    - We should never directly update state. Ex: **color = "red"** is **not allowed**.