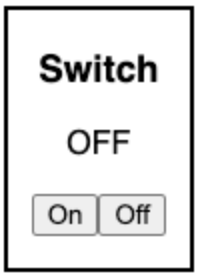
REACT WORKSHOP EXERCISES

EXERCISE 1: NON-FUNCTIONING SWITCH COMPONENT

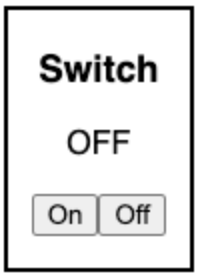


Build a component named `Switch` that looks roughly like this. It doesn't do anything yet. It just always says OFF. Add a couple of these switches to the App.js component.

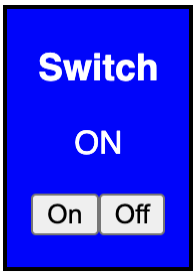
Steps to Create a Component

1. Create new JS file and CSS files. (Switch.js, Switch.css)
2. In JS file, add imports: 1) React. 2) Your css file.
3. Create a function (named Switch). Export it at the bottom with `export default`.
4. In the function, return JSX code. It's a good idea to give the root element the className equal to your component name (`<div className="Switch">`).
5. Fill in any JSX and CSS you want.
6. Add your component to App.js: 1) import it. 2) include it in JSX via tag (`<Switch />`).

EXERCISE 2: WORKING SWITCH



Task 1: Add a state variable to your Switch. Use that state to show "ON" or "OFF" in the JSX. Update the state appropriately when the On and Off buttons are clicked.

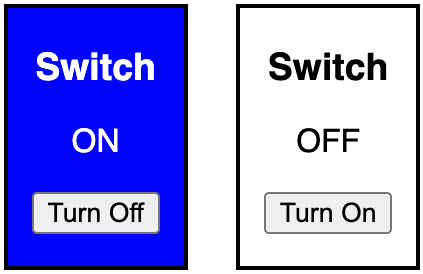


Task 2: Modify the Switch so that the background changes color when the switch is on. To do this, add a CSS class in the JSX only when the state is on. Add a CSS rule for this class to give it the color. This may require some creative thinking, but see if you can figure it out!

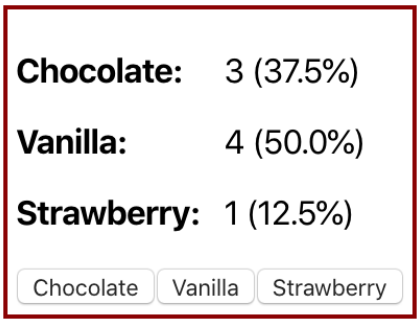
EXERCISE 3: OPTIMIZED SWITCH

If you didn't finish getting the background color working from exercise 2, do that now.

There's no reason to show both buttons all the time. When the switch is on, only show a Turn Off button, and when it is off, only show a Turn On button.



EXERCISE 4: VOTES



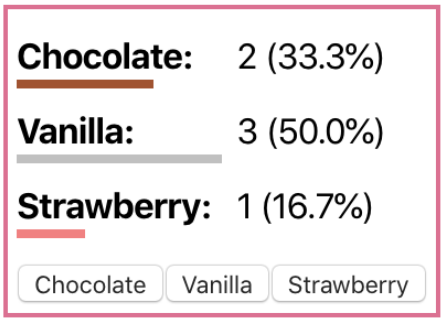
Create a new component named Votes and add it to your page in App.js. Make it look something like this image. (Style as you like. Don't spend too much time on styling.)

Numbers for all three flavors should start at zero. Each button adds a vote for the corresponding flavor. Display the number of votes and the percentage of the total vote for each.

Hints:

- In order to track 3 different state variables, just repeat useState on multiple lines.
- Percentage should not be in state. It should be calculated as needed in the component function.

Want more? Try this: Add colored bars indicating the percentage of the vote for each. You might need to search online how to use inline styles in React for the bar width.



EXERCISE 5: QUOTES

Create a `Quote` component with three props, all of which are strings.

- text
- author
- date (optional)

Display the values in the component. Note that if there is no value for date, do not include the date paragraph at all.

In App.js, add several of these Quotes, passing different values for the props. The image here shows three Quote components included in App.js.

EXERCISE 6: QUOTE LIST

Create another component named `QuoteList`. It will display a list of `Quote`s (from exercise 5) based on an array. You can use this array:

```
const quotes = [
  {
    author: "Nelson Mandela",
    text: "It always seems impossible until it's done.",
    date: "2001"
  },
  {
    author: "Yoda",
    text: "Do. Or do not. There is no try.",
    date: "A long time ago"
  },
  { author: "Bugs Bunny", text: "What's up, doc?" }
];
```

Include QuoteList on the page in App.js. To avoid confusion, you might want to remove or comment out the Quotes added to App.js in exercise 5.

EXERCISE 7: QUOTES FROM API

Modify your `QuoteList` component. Use instead of the hard-coded array, load the data from this API: https://type.fit/api/quotes (Note: there are a LOT of quotes here.)

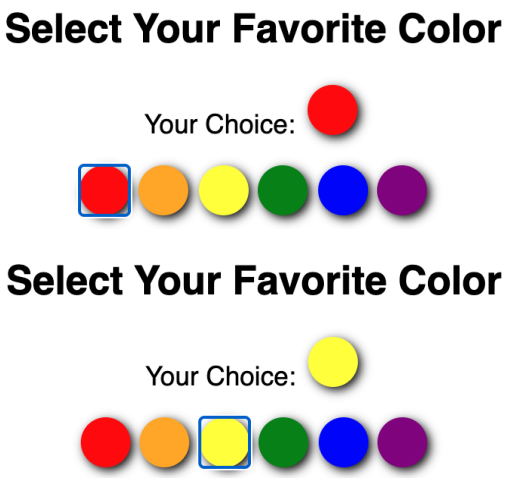
Finished early? Try this:

- Option 1: Can you limit your component to only show the first 10 quotes from the API?
- Option 2: Load quotes from this API instead. The structure is more complicated, and it includes dates for some quotes. https://dwolverton.github.io/juneteenth/public/api/quotes.json

EXERCISE 8: FAVORITE COLOR

Create a component named `Color` using the JavaScript and CSS below. This component displays as a circle with the specified `color` prop. As you can see in the JavaScript, it is a button, but if you click it nothing happens yet.

Create another component named `ColorPicker`. This component should display several of the `Color` components using a variety of colors. The user can click any of these colors to select their favorite color, which is also displayed.



JavaScript

```
import React from "react";
import './Color.css';

function Color({ color }) {
  return (
    <button
      className="Color"
      style={{ backgroundColor:
color }}
      onClick={null}
    ></button>
  );
}

export default Color;
```

CSS

```
.Color {
  width: 30px;
  height: 30px;
  border-radius: 50%;
  border: none;
  box-shadow: 2px 2px 6px black;
  margin: 0 3px;
}
```