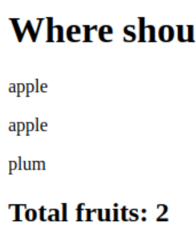
Managing state in React
Open Lab

Instructions

Task

In the starter code of this code lab, you are given a **`Fruits`** component that has its own state. Based on this state, it outputs three fruits on the screen. Additionally, you have a **`FruitsCounter`** component which shows a message that reads: "Total fruits: 2".
Your task is to lift state up from the **`Fruits`** component to the **`App`** component, so that you can then pass the state information to both the **`Fruits`** component and the **`FruitsCounter`** component.
This change to the app should fix the previously incorrect message of "Total fruits: 2". The new message should be "Total fruits: 3". However, the new message will not be just a hard-coded string. Instead, it should reflect the number of fruits that exist in the state variable, so based on how many fruits there are in the state array, this information should affect the output of the total number of fruits - as returned from the **`FruitsCounter`** component.



Where should the state go?

apple

apple

plum

Total fruits: 2

Steps
Step 1. This task's starting point is the stateless **`App`** component's code:

```
import Fruits from "./Fruits";
import FruitsCounter from "./FruitsCounter";

function App() {
  return (
    <div className="App">
      <h1>Where should the state go?</h1>
      <Fruits />
      <FruitsCounter />
    </div>
  );
};

export default App;

```

The first step of this task is to move the state from the **`Fruits`** component:

```
import React from "react";

function Fruits() {
    const [fruits] = React.useState([
        {fruitName: 'apple', id: 1},
        {fruitName: 'apple', id: 2},
        {fruitName: 'plum', id: 3},
    ]);

    return (
        <div>
            {fruits.map(f => <p key={f.id}>{f.fruitName}</p>)}
        </div>
    );
};

export default Fruits;
```

The state should be moved to the **`App`** component. That is the first step of this task.

Step 2.

The **`Fruits`** component should be updated so that it accepts state from the **`App`** component.

Step 3.

Once you've lifted the state up from the **`Fruits`** component to the **`App`** component, you also need to update the **`FruitsCounter`** component.
Just like the **`Fruits`** component, the **`FruitsCounter`** component should also receive state from the **`App`** component, so that it can display the number of the available fruits using the **`length`** property of the array of fruits from the **`fruits`** state variable. This state variable comes from **`App`** and is received in **`FruitsCounter`** as **`prop.button`** , to which you'll add the following text: *Guess the number between 1 and 3*.

Step 3.

Save your updates and serve the app so that it shows the correct count of fruits, while the rest of the output should remain unchanged.