React-2 Data and State

Dynamic events and how to handle them

- An action(change by user) is an event. Events are used by developers to respond to changes in the application.
- Because events usually rely on some interaction, they need to wait and listen in the background for that interaction to occur before they can be triggered.
- Every HTML element contains a set of events that developers can access by using HTML attributes, commonly referred to as event listeners.
- Events are part of the DOM.
- The attribute of HTML elements are camelCased in JSX. e.g. onClick
- There are different groups of events. mouse, keyboard, clipboard events etc.

These events comes from browser to offer interactions abilities to the user within an Application.

Eventful Issues

Errors are part of a developer's life.

JS offers `try..catch` block to handle errors.

In React, We can use it to do error-handling. As the event-handling logic/code is executed only after the UI is rendered.

Syntax for Handlers

Example:

Event (click) → EventHandler(onClick) → Action (open menu)

In HTML, `<tag eventHandlingAttribute="functionName()"></tag>`

In JS, the same equivalent is:-

- Locate the element in the DOM.
  `e.g document,getElementbyId("id");`
- Attach the Event listener on the object in DOM from above step.
  `e.g element.addEventListener('click', function(){`
  `code})`

In React, the rule is to avoid manipulating the DOM directly.

The best way is to do it declaratively to react and let it modify the DOM.

There is NO function invocation syntax to an event handler function.

In React, Just declare the eventHandler event and pass a reference to it in a JSX expression.

In React, you can use **props** to pass the function declaration in the form on JSX expression.

Event Handling and embedded expressions

- With an inline anonymous ES5 function

```
<button onClick={function() {console.log('first example')}}>
    An inline anonymous ES5 function event handler
</button>
```

- With an inline, anonymous ES6 function (an arrow function)

```
<button onClick={() => console.log('second example')}>
    An inline anonymous ES6 function event handler
</button>
```

- Using a separate ES5 function declaration
  - This syntax makes sense to be used when your onClick logic is too complex to easily fit into an anonymous function.

```
function App() {
    function thirdExample() {
        console.log('third example');
    };
    return (
        <div className="thirdExample">
            <button onClick={thirdExample}>
                using a separate function declaration
            </button>
        </div>
    );
};
export default App;
```

- Using a separate function expression
  - A way to determine if a function is defined as an expression or a declaration is: if it does not start the line with the keyword function, then it’s an expression.
  -

```
function App() {
    const fourthExample = () => console.log('fourth example');

    return (
        <div className="fourthExample">
            <button onClick={fourthExample}>
                using a separate function expression
            </button>
        </div>
  );
};
export default App;
```

User Events

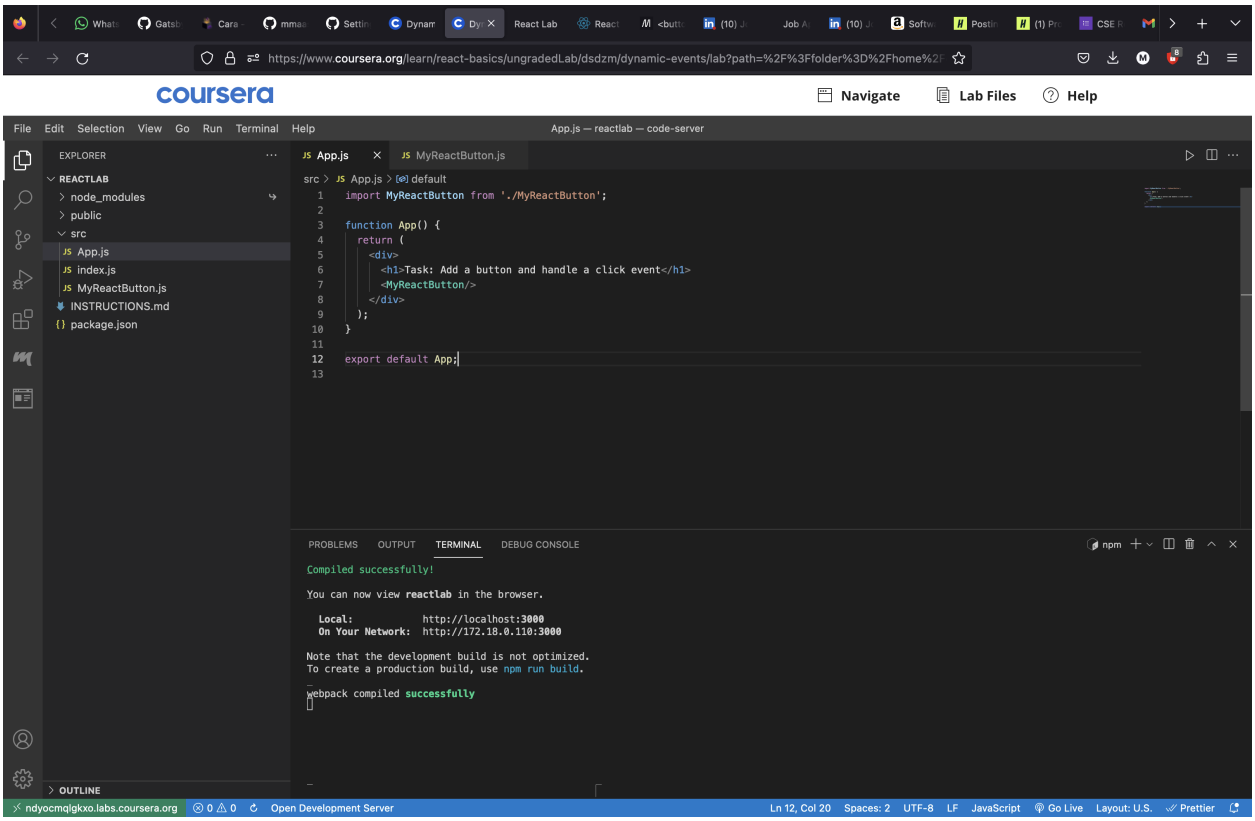
Use of a boolean variable as state to toggle between dark and light mode in a component. To test , I render different content in an h1 tag.

In React, a click handler is placed inside a JSX expression, and only needs a click handler function's name - without the parentheses to invoke it. This is what makes it different from HTML event handing attributes.

Practice: Dynamic Events

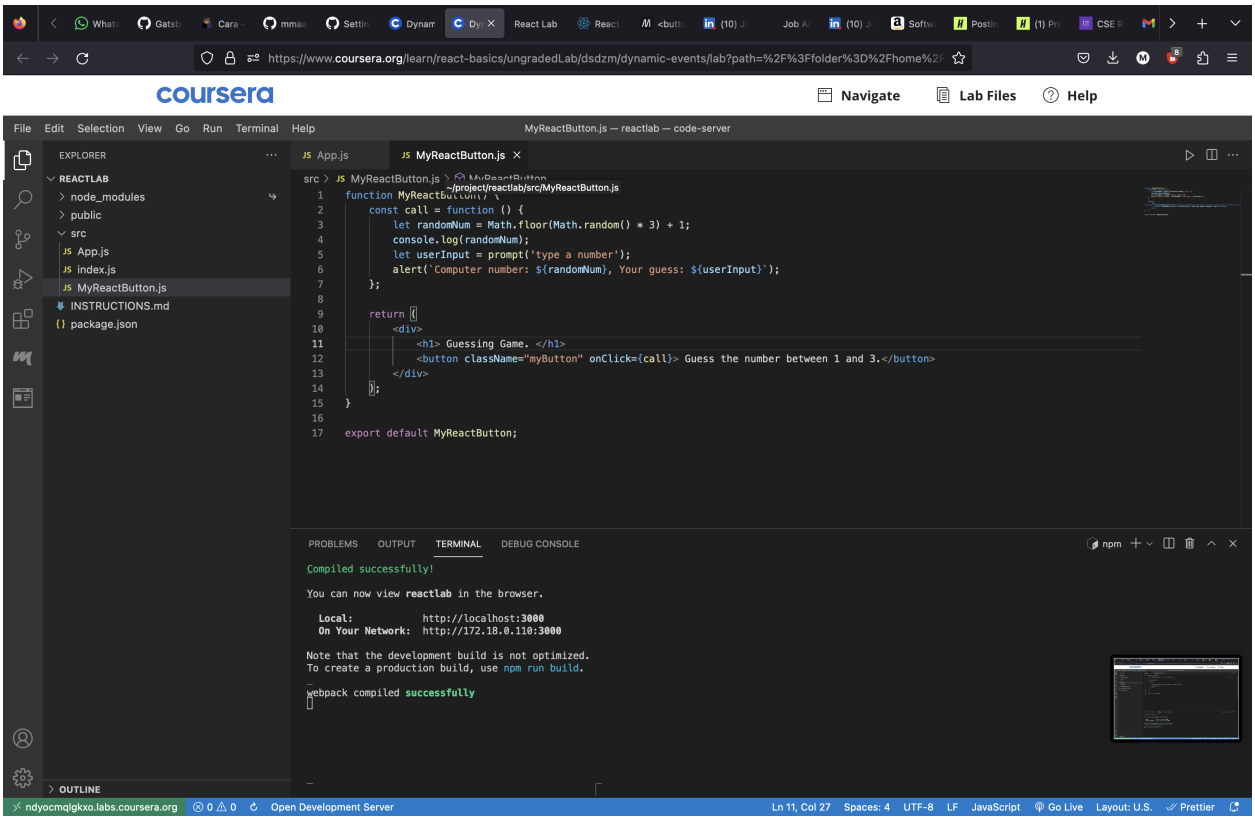
building a simple number-guessing game.

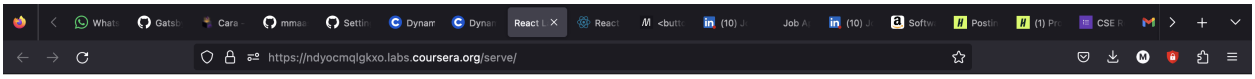
Main task is to add a button component and event handler, that prompt user to guess a number. and verifies if it matched the random number.


coursera
Navigate
Lab Files
Help
File Edit Selection View Go Run Terminal Help
App.js — reactlab — code-server
EXPLORER
REACTLAB
node_modules
public
src
App.js
index.js
MyReactButton.js
INSTRUCTIONS.md
package.json
App.js
MyReactButton.js
src > App.js > default
import MyReactButton from './MyReactButton';
function App() {
return (

<h1>Task: Add a button and handle a click event</h1>
<MyReactButton/>

);
}
export default App;
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
npm
Compiled successfully!
You can now view reactlab in the browser.
Local: http://localhost:3000
On Your Network: http://172.18.0.110:3000
Note that the development build is not optimized.
To create a production build, use npm run build.
webpack compiled successfully
OUTLINE
Open Development Server
Ln 12, Col 20
Spaces: 2
UTF-8
LF
JavaScript
Go Live
Layout: U.S.
Prettier

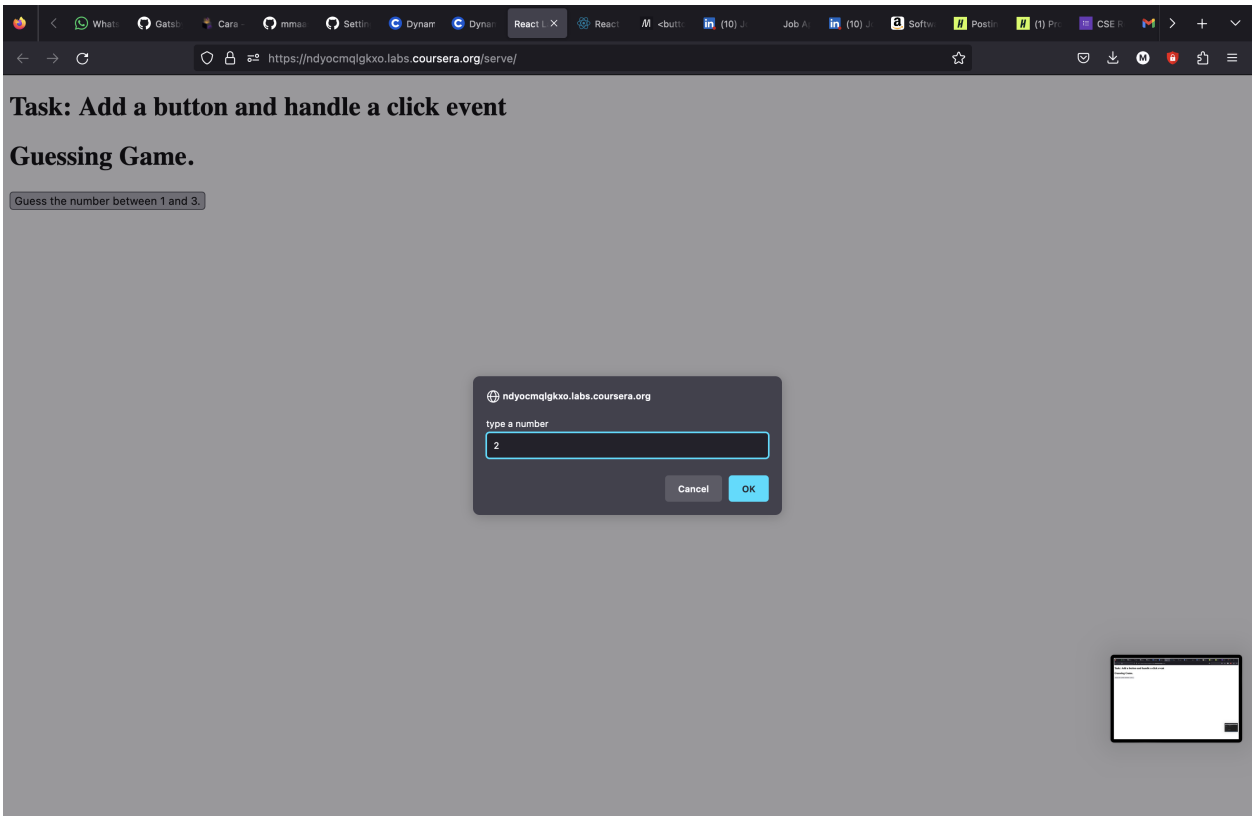


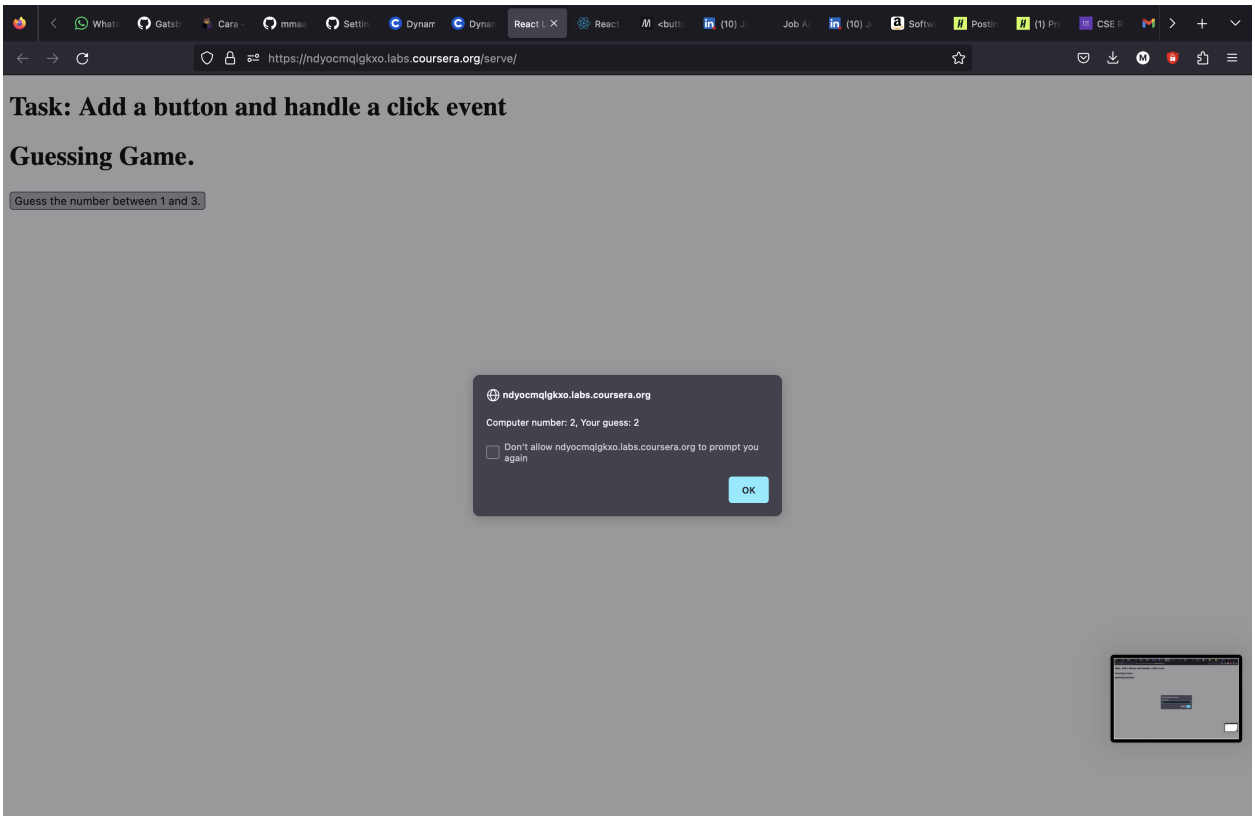


Task: Add a button and handle a click event

Guessing Game.

Guess the number between 1 and 3.


https://ndyocmqlgkxo.labs.coursera.org/serve/
Task: Add a button and handle a click event
Guessing Game.
Guess the number between 1 and 3.
ndyocmqlgkxo.labs.coursera.org
type a number
2
Cancel
OK



- Passing the function name such as “clickHandler” to the onClick handler, and wrapping it in a JSX expression, then coding a clickHandler function declaration, is a common way of handling a click event in React.
- Syntactically, the only difference is that HTML attributes are all lowercased, while React attributes are camelCased.
- The code that would work would need to be as follows: onClick={handleClick}. and this would not work onClick={handleClick()}.

quiz

```
Practice Quiz. • 15 min. • 5 total points available.5 total points
Congratulations! You passed!
Grade received 100%
To pass 80% or higher
1.
Question 1

What code should be added to the element button to make this code snippet valid?
function App() {

  function handleClick() {
    console.log("clicked")
  }

  return (
    <div className="App">
      <button >Click me</button>
    </div>
  );
}
```

1 / 1 point

onClick={handleClick()}

click=handleClick

onClick={handleClick}
Correct

Correct. This is the correct syntax.
2.
Question 2

Imagine that you have a variable named userLoggedIn and it’s set to the boolean of true. How would you complete the below clickHandler fu
nction declaration to toggle the value of the userLoggedIn boolean?

1 / 1 point

userLoggedIn = false

userLoggedIn = true

userLoggedIn = !userLoggedIn
Correct

That’s correct! This will toggle the boolean value from true to false, and from false to true.
3.
Question 3

Is a click handler on its own enough to change the values of variables in your React apps?
1 / 1 point

No

Yes
Correct

That’s correct! You need to also use something known as “hooks”.
4.
Question 4

What are the ways to write an event-handling function in React. Select all that apply.
1 / 1 point

With an inline, anonymous ES6 function (an arrow function)
Correct

That’s right! An inline anonymous ES6 function (an arrow function) is a valid way to handle an event in React.

Using a separate function declaration
Correct

That’s right! A separate function declaration is a valid way to handle an event in React.

With an inline anonymous ES5 function
Correct

That’s right! An inline anonymous ES5 function is a valid way to handle an event in React.

Using a separate function expression
Correct

That’s right! A separate function expression is a valid way to handle an event in React.
5.
Question 5

Choose the appropriate code on line 3 of the following component – to handle a click event.

```
function App() {

  function () {
    console.log("clicked")
  }

  return (
    <div className="App">
      <button onClick={handleClick}>Click me</button>
```

```
    </div>
  );
}

1 / 1 point

function HandleClick() {

function handleClick {

function handleClick() {
Correct

That’s correct. This return statement will show the Example component on the screen.
```

Data and Events

Parent-child data flow

What are the advantages of utilizing a centralized point of data - a "single source of truth" - in your React apps? Choose all that apply.

```
- [ ]  It allows you to edit multiple items from a single point

Correct

That's right! Using a single source of truth allows you to edit multiple items at the same time if they reference the same data, reduces
odds of typing errors, and is more efficient when data changes often.

- [ ]  It offers a more efficient way of working when data frequently changes

Correct

That's right! Using a single source of truth allows you to edit multiple items at the same time if they reference the same data, reduces
odds of typing errors, and is more efficient when data changes often.

- [ ]  Data flows both ways, so it can be edited from anywhere
- [ ]  It reduces the possibility of typing errors in your code

Correct

That's right! Using a single source of truth allows you to edit multiple items at the same time if they reference the same data, reduces
odds of typing errors, and is more efficient when data changes often.
```

PROPS DATA ALWAYS FLOW FROM PARENT COMPONENT TO CHILD COMPONENTS.

The data flow starts at the root and can flow to multiple levels of nesting, from the root component (parent component) to the child component, then the grandchild component, and further down the hierarchy.

Props are immutable (cannot be changed).

The two main benefits of this unidirectional data flow are that it allows developers to:

1. comprehend the logic of React apps more quickly and
2. simplify the data flow.

If data were moving everywhere, all the time, then it would be much harder to comprehend its logical flow. Any optimization you tried to implement would likely not be as efficient as it could be, especially in modern React.

ONE WAY IS PROPS, AND ANOTHER WAY TO INTERACT WITH DATA IS WITH STATE.

PROPS DATA LIES OUTSIDE THE COMPONENTS, THIS IS THE DATA THAT IT RECEIVES AND IT BELONGS TO PARENT COMPONENT & IS NOT MUTABLE.

WHEREAS,

THE STATE DATA LIES WITHIN THE COMPONENT AND THE COMPONENT CAN CONTROL AND MUTATE.

QUIZ

```
Week 2

    Knowledge check: Data flow

Knowledge check: Data flow
Practice Quiz18 minutes • 18 min
Submit your assignment
Receive grade
To Pass 80% or higher
Your grade
100%

We keep your highest score
Knowledge check: Data flow

Practice Quiz. • 18 min. • 6 total points available.6 total points
Congratulations! You passed!
Grade received 100%
To pass 80% or higher
1.
Question 1

Usually, a React app consists of many components, organized as a component tree.
1 / 1 point

True

False
Correct

That’s correct, a React app is organized as a tree of components.
2.
Question 2

Uni-directional data flow is...
1 / 1 point

The term that describes the one-way flow of data in a React app.

The term that describes the one-way flow of DOM updates in a React app

The term that describes the one-way flow of components in a React app
Correct

Correct. Uni-directional flow of data is synonymous with one-way data flow.
3.
Question 3

A component can, at any given time_______. Select all that apply.
```

```
1 / 1 point

Receive data as props
Correct

That’s right. You can receive data as props.

Pass data as props
Correct

That’s right. You can pass data as props.

Pass data as props and receive data as props at the same time
Correct

That’s right. You can pass data as props and receive data as props at the same time.
4.
Question 4

You can only pass a single prop to a component.
1 / 1 point

True

False
Correct

You can pass more than a single prop.
5.
Question 5

The props parameter is:
1 / 1 point

A string

An object

A boolean

An array
Correct

The props parameter is an object.
6.
Question 6

Consider the following piece of code:


Which element of this code represents a child component?
1 / 1 point

<Appetizers />

return

<div>

MyMenu()
Correct

That’s right! <Appetizers /> is the name of the child component. In this case, it is defined in a separate file.
```

---

core concepts of React acquired ✅

Learn how to add interactivity, maintain state within a React component and explore hooks.

What are hooks?

complex component where state need to be tracked.

Hooks - lets us plug into a component, and **track & manage** its state.

how to use hooks?

- import `useState` from react.
- declare state variable within a component.
  - when a variable is initialized with useState function, it return a pair of values
  - const [var1, var2] = useState(false) is best way; known as **array destructuring.**
  - first var1 is state variable, and var2 is function that sets/updates the var1 value.
- **the useState hook MUST be called at the top level of the component code.**
- **Hooks can be called only at the top level and only from React functions.**
- It can be used to track any kind of data.
- You can either use hooks from react or build your own hooks.

In React, state is always referred to the local state of a component.

Rules with Hooks:

Hooks also come with a set of rules, that you need to follow while using them. This applies to all React hooks

- You can only call hooks at the top level of your component or your own hooks.
- You cannot call hooks inside loops or conditions.
- You can only call hooks from React functions, and not regular JavaScript functions.

Example:

- 1 state variable

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

export default function InputComponent() {
  const [inputText, setText] = useState('hello');

  function handleChange(e) {
    setText(e.target.value);
  }

  return (
    <>
      <input value={inputText} onChange={handleChange} />
      <p>You typed: {inputText}</p>
      <button onClick={() => setText('hello')}>
        Reset
      </button>
    </>
  );
}
```

- Instead of pair of state variable and function, for every state variable that we wanna track, we can use a single object.

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

export default function RegisterForm() {
  const [form, setForm] = useState({
    firstName: 'Luke',
    lastName: 'Jones',
    email: 'lukeJones@sculpture.com',
  });

  return (
    <>
      <label>
        First name:
        <input
          value={form.firstName}
          onChange={e => {
            setForm({
              ...form,
              firstName: e.target.value
            });
          }}
        />
      </label>
      <label>
        Last name:
        <input
          value={form.lastName}
          onChange={e => {
            setForm({
              ...form,
              lastName: e.target.value
            });
          }}
        />
      </label>
      <label>
        Email:
        <input
          value={form.email}
          onChange={e => {
            setForm({
              ...form,
              email: e.target.value
            });
          }}
        />
      </label>
      <p>
        {form.firstName}{' '}
        {form.lastName}{' '}
        ({form.email})
      </p>
    </>
  );
}
```

In addition to the useState hook, there are other hooks that come in handy such as useContext, useMemo, useRef, etc. When you need to share logic and reuse the same logic across several components, you can extract the logic into a custom hook. Custom hooks offer flexibility and can be used for a wide range of use-cases such as form handling, animation, timers, and many more.

---

what is state in React?

- Data in a component that determines behavior.

- It allows components to stay in sync and ensure consistent behavior. this means that the **state** is shared with a component's children and successor components.
- when a component is created , it is assigned an initial state, and the state is used to initialize some of its properties.
- Components can be stateful or stateless.
    - stateful has state variable(s) in it done using hook(s).
    - stateless is just static content.

Observing State : We cannot use the state update/setting function directly, it has to be tied with another event handing attribute and it is fired/called when it happens.

quiz

```
    Knowledge Check: State the concept

Knowledge Check: State the concept
Practice Quiz15 minutes • 15 min
Submit your assignment
Receive grade
To Pass 80% or higher
Your grade
100%

We keep your highest score
Knowledge Check: State the concept

Practice Quiz. • 15 min. • 5 total points available.5 total points
Congratulations! You passed!
Grade received 100%
To pass 80% or higher
1.
Question 1

In React, can state be considered data?
1 / 1 point

Yes

No
Correct

Correct. In React, an app’s data can be expressed as state.
2.
Question 2

In React, can props be considered data?
1 / 1 point

Yes

No
Correct

Correct. In React, an app’s data can be expressed as props.
3.
Question 3

Choose the correct statement.
1 / 1 point

The props object represents data that is external to a component, and state represents data that is internal to a component.

The props object represents data that is internal to a component, and state represents data that is external to a component.
Correct

That’s correct! The props data is controlled outside of a component, and the state data is controlled by the component itself.
4.
```

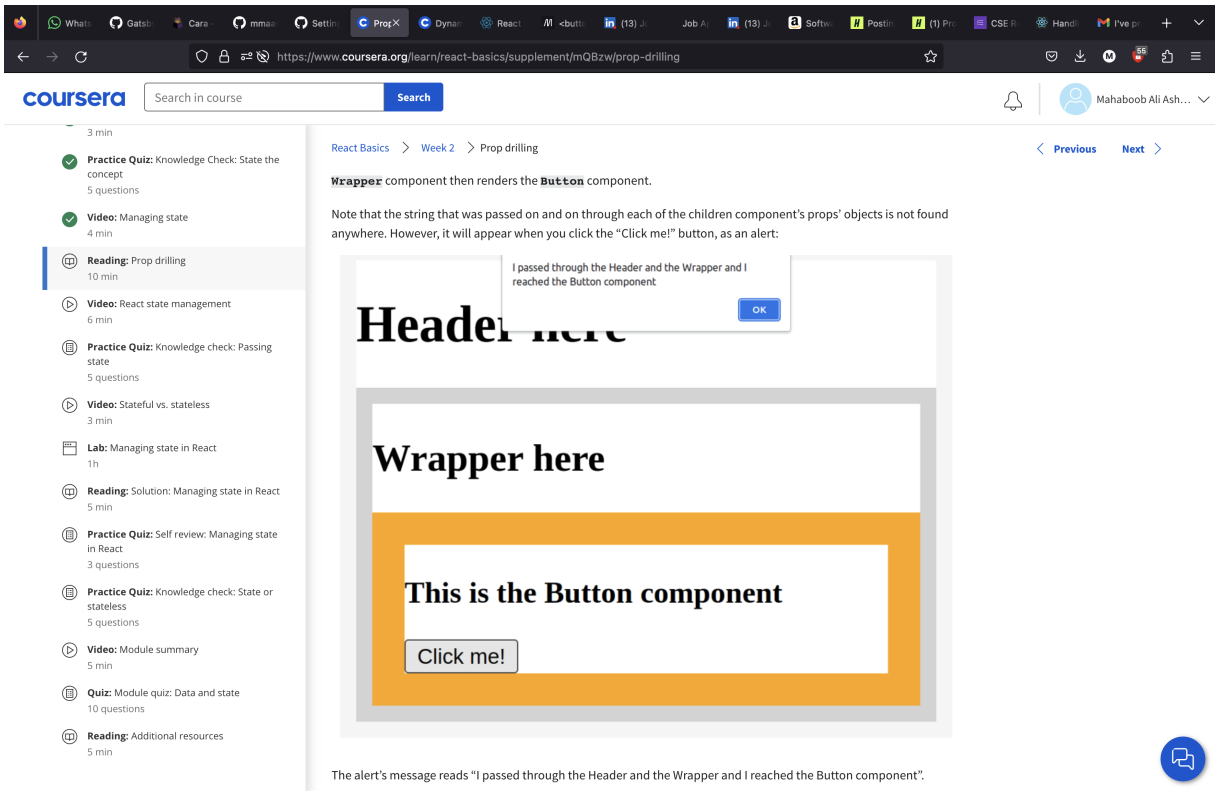
```
Question 4

What does the useState hook do?
1 / 1 point

It allows a component to have its own state.

It allows a component to receive state from its parent.
Correct

That’s right! The useState hook allows a component to have its own state.
5.
Question 5

Based on the code below, is the userName variable external or internal data of the DisplayUser component?
1
2
3
4
5
6

1 / 1 point

The userName value is data that is external to the DisplayUser component

The userName value is data that is internal to the DisplayUser component
Correct

That’s correct. The userName value comes from the parent component, and thus is not controlled by the DisplayUser component.
```

Managing State in React Applications

Lifting state up is about cutting the state from the child component and moving it to the parent component's code, with the intent of making the state available in sibling components.

Prop drilling: prop drilling is a situation where you are passing data from a parent to a child component, then to a grandchild component, and so on, until it reaches a more distant component further down the component tree, where this data is required.

- **Props drilling simply means passing a prop through props objects through several layers of components**. The more layers there are, the more repetitive and unnecessary this feels.



Global State: In react, we can use context API. extracts state into a separate file that hold the context, and any file/component that needs it can simple import it and use it.

Props can be used to pass Data one component at a time.

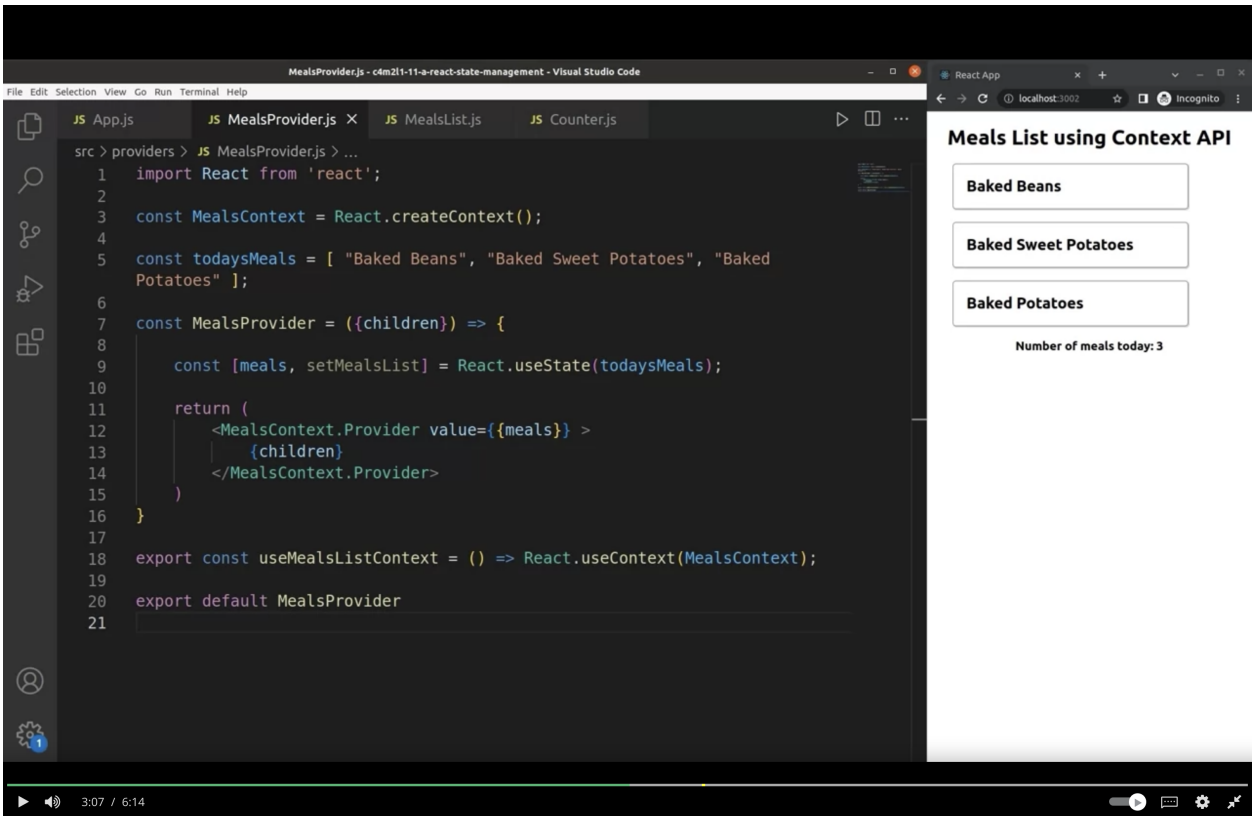
Tools to manage State of React components

Context API can be used to perform state management across multiple components.

Working:

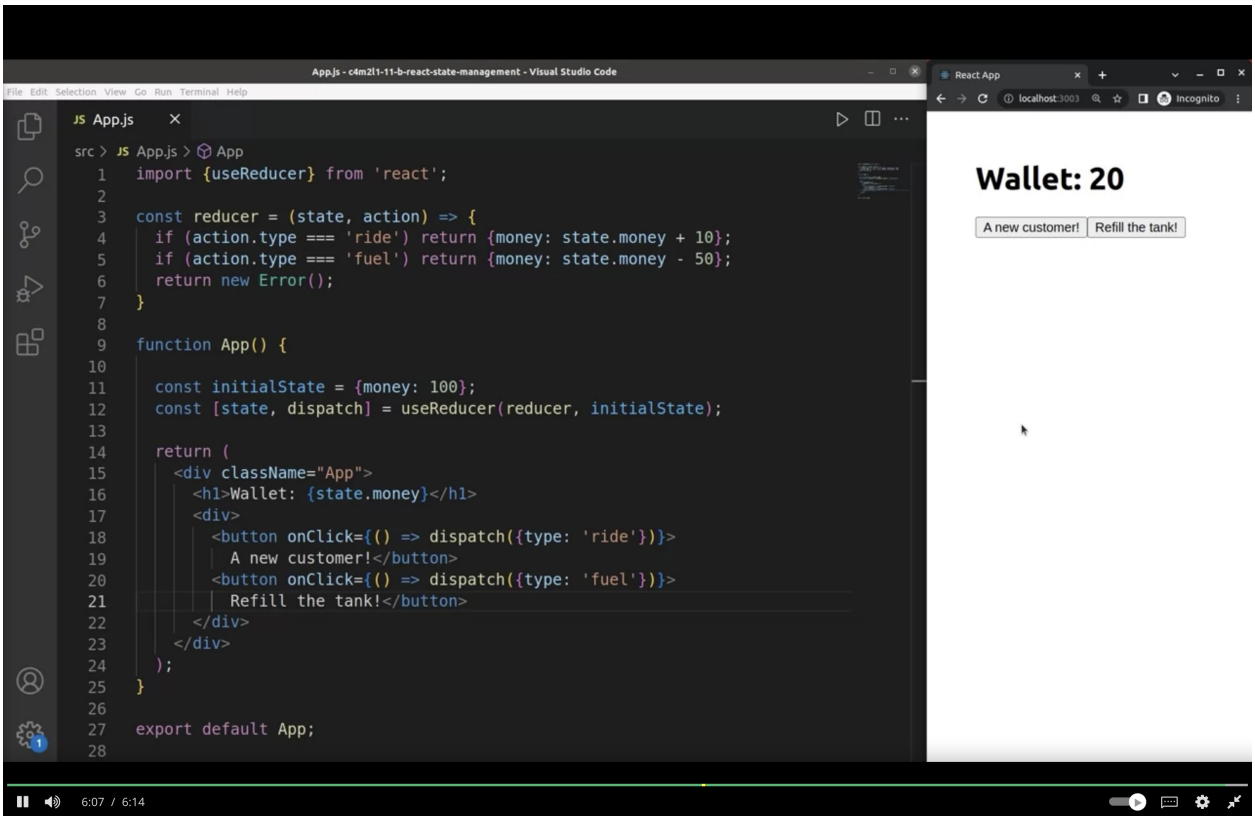
- Context Provider
  - Component that stores the state.
- Context Consumer
  - Component that will use the state.

`React.createContext()` returns an object , in which `Context.Provider` element has JSX attribute `value` which has a state variable name as JSX expression.



`UseReducer`

similar to useState hook, it return a pair, state variable, and a dispatcher.
we can define a (state, action)and use dispatcher to perform the action.



quiz: Passing State

```
Skip to Main Content
Coursera
Search in course

Video: VideoParent-child data flow
. Duration: 5 minutes5 min
Reading: ReadingData flow in React
. Duration: 10 minutes10 min
Video: VideoChildren and data
. Duration: 3 minutes3 min
Practice Quiz: Knowledge check: Data flow
6 questions
Video: VideoWhat are hooks?
. Duration: 3 minutes3 min
Reading: ReadingUsing hooks
. Duration: 10 minutes10 min
Video: VideoWhat is state?
. Duration: 4 minutes4 min
Video: VideoObserving state
. Duration: 3 minutes3 min
Practice Quiz: Knowledge Check: State the concept
5 questions
Video: VideoManaging state
. Duration: 4 minutes4 min
Reading: ReadingProp drilling
. Duration: 10 minutes10 min
Video: VideoReact state management
. Duration: 6 minutes6 min
Practice Quiz: Knowledge check: Passing state
5 questions
```

Video: VideoStateful vs. stateless
. Duration: 3 minutes3 min
Lab: Managing state in React
. Duration: 1 hour1h
Reading: ReadingSolution: Managing state in React
. Duration: 5 minutes5 min
Practice Quiz: Self review: Managing state in React
3 questions
Practice Quiz: Knowledge check: State or stateless
5 questions
Video: VideoModule summary
. Duration: 5 minutes5 min
Quiz: Module quiz: Data and state
10 questions

        Reading: ReadingAdditional resources
        . Duration: 5 minutes5 min

    React Basics

Week 2

    Knowledge check: Passing state

Knowledge check: Passing state
Practice Quiz15 minutes • 15 min
Submit your assignment
Receive grade
To Pass 80% or higher
Your grade
100%

We keep your highest score
Knowledge check: Passing state

Practice Quiz. • 15 min. • 5 total points available.5 total points
Congratulations! You passed!
Grade received 100%
To pass 80% or higher
1.
Question 1

What is the Context API?
1 / 1 point

An alternative way to working with state in React.

A way to change the execution context of a function in JavaScript.
Correct

Correct. Context API is used to work with state from a global store in react.
2.
Question 2

When working with useState to keep state in a variable, you should not use array destructuring.
1 / 1 point

True

False
Correct

Correct. You should always use array destructuring when working with useState.
3.
Question 3

If a state variable is destructured from useState, and set to variable name of user, the name of the function to update the user state va
riable should be...
1 / 1 point

setUser

useUser

userSetter

useState
Correct

```
That’s correct! The convention is to name the state-setting function using the word “set” plus whatever the name of the state variable i
s, all written in camelCase.
4.
Question 4

What does the concept of “lifting up state” entail?
1 / 1 point

It involves moving the state from the child component to the parent component.

It involves moving the state from the parent component to the child component.
Correct

That’s right! Lifting up state means moving state up from a child to the parent component.
5.
Question 5

What is a negative result of lifting up state in a React app?
1 / 1 point

Prop drilling.

It can significantly increase the number of components that you need to create.

There are no negatives from lifting up state in React.
Correct

That’s correct. Lifting up state can sometimes lead to prop drilling, which lowers maintainability and modularity of a React app.
```

Stateful and Stateless

Identifying needs and then decide which kind of component is needed.

Stateful hold state within component, & changes based on interactions.

Stateless does not hold state, it just gets/inherent data/state from parent component via props.

Rules to decide state of component:

1. Stateless: when there is no needed of State for the component to work.
2. Stateful: State must be maintained to work.

A common approach is to have a stateful parent and stateless component. the children receive parent’s state via props. and the children component can have event Handlers to update the state value in parent component.

Practice Lab: Managing State

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?

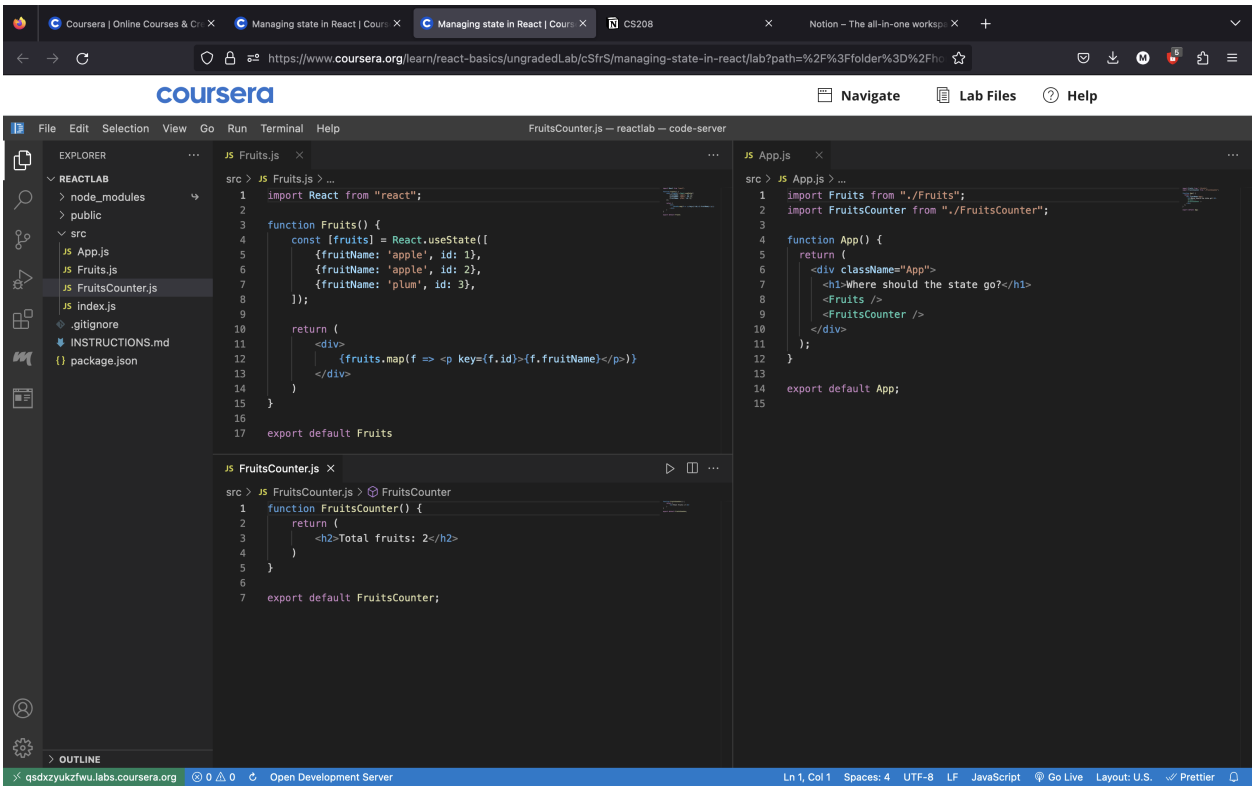
apple

apple

plum

Total fruits: 2

Unmodified Code:



My Idea:

have the fruits as state in app component, and pass it as props to fruits, and fruitscounter.

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

function App() {
  const [fruits] = React.useState([
```

```
      {fruitName: 'apple', id: 1},
      {fruitName: 'apple', id: 2},
      {fruitName: 'plum', id: 3},
  ]);

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

export default App;
```

```
function Fruits(props) {
    return (
        <div>
            {props.fruits.map(f => <p key={f.id}>{f.fruitName}</p>)}
        </div>
    )
}

export default Fruits
```

```
function FruitsCounter(props) {
    return (
        <h2>Total fruits: {props.fruits.length}</h2>
    )
}

export default FruitsCounter;
```

quiz- managing state

```
Skip to Main Content
Coursera
Search in course

    React Basics

Week 2

    Self review: Managing state in React

Video: VideoParent-child data flow
. Duration: 5 minutes5 min
Reading: ReadingData flow in React
. Duration: 10 minutes10 min
Video: VideoChildren and data
. Duration: 3 minutes3 min
Practice Quiz: Knowledge check: Data flow
6 questions
Video: VideoWhat are hooks?
. Duration: 3 minutes3 min
Reading: ReadingUsing hooks
. Duration: 10 minutes10 min
```

```
Video: VideoWhat is state?
. Duration: 4 minutes4 min
Video: VideoObserving state
. Duration: 3 minutes3 min
Practice Quiz: Knowledge Check: State the concept
5 questions
Video: VideoManaging state
. Duration: 4 minutes4 min
Reading: ReadingProp drilling
. Duration: 10 minutes10 min
Video: VideoReact state management
. Duration: 6 minutes6 min
Practice Quiz: Knowledge check: Passing state
5 questions
Video: VideoStateful vs. stateless
. Duration: 3 minutes3 min
Lab: Managing state in React
. Duration: 1 hour1h
Reading: ReadingSolution: Managing state in React
. Duration: 5 minutes5 min
Practice Quiz: Self review: Managing state in React
3 questions
Practice Quiz: Knowledge check: State or stateless
5 questions
Video: VideoModule summary
. Duration: 5 minutes5 min
Quiz: Module quiz: Data and state
10 questions

        Reading: ReadingAdditional resources
        . Duration: 5 minutes5 min

Self review: Managing state in React
Practice Quiz9 minutes • 9 min
Submit your assignment
Receive grade
To Pass 80% or higher
Your grade
100%

We keep your highest score
Self review: Managing state in React

Practice Quiz. • 9 min. • 3 total points available.3 total points
Congratulations! You passed!
Grade received 100%
To pass 80% or higher
1.
Question 1

True or false? When lifting state up, you need to move the useState from a child component to a parent component.
1 / 1 point

False.

True.
Correct

Correct. When lifting state up, you need to move the useState from a child component to a parent component.
2.
Question 2

If the state variable holds an array or a string value, once you pass that state via props from a parent to a child, you can then read the
1 / 1 point

False

True
Correct

Correct. If the state variable holds an array or a string value, once you pass that state via props from a parent to a child, you can then
3.
Question 3

What's wrong with this code?
12
13
14
```

```
15
16
17
18
19
20
21
11
10
9
7
8
6
5
3
4
2
1

1 / 1 point

There's nothing wrong with the provided code.

The useState call should be React.useState.

If you don't add the setFruits state-updating function when descructuring values from useState, the app won't compile.
Correct

Correct. In the context of the give code snippet, you need to access the useState hook using the React.useState syntax.
```

Practice Knowledge check: State or stateless

```
Skip to Main Content
Coursera
Search in course

    React Basics

Week 2

    Knowledge check: State or stateless

Video: VideoParent-child data flow
. Duration: 5 minutes5 min
Reading: ReadingData flow in React
. Duration: 10 minutes10 min
Video: VideoChildren and data
. Duration: 3 minutes3 min
Practice Quiz: Knowledge check: Data flow
6 questions
Video: VideoWhat are hooks?
. Duration: 3 minutes3 min
Reading: ReadingUsing hooks
. Duration: 10 minutes10 min
Video: VideoWhat is state?
. Duration: 4 minutes4 min
Video: VideoObserving state
. Duration: 3 minutes3 min
Practice Quiz: Knowledge Check: State the concept
5 questions
Video: VideoManaging state
. Duration: 4 minutes4 min
Reading: ReadingProp drilling
. Duration: 10 minutes10 min
Video: VideoReact state management
. Duration: 6 minutes6 min
Practice Quiz: Knowledge check: Passing state
```

5 questions
Video: VideoStateful vs. stateless
. Duration: 3 minutes3 min
Lab: Managing state in React
. Duration: 1 hour1h
Reading: ReadingSolution: Managing state in React
. Duration: 5 minutes5 min
Practice Quiz: Self review: Managing state in React
3 questions
Practice Quiz: Knowledge check: State or stateless
5 questions
Video: VideoModule summary
. Duration: 5 minutes5 min
Quiz: Module quiz: Data and state
10 questions

        Reading: ReadingAdditional resources
        . Duration: 5 minutes5 min

Knowledge check: State or stateless
Practice Quiz15 minutes • 15 min
Submit your assignment
Receive grade
To Pass 80% or higher
Your grade
100%

We keep your highest score
Knowledge check: State or stateless

Practice Quiz. • 15 min. • 5 total points available.5 total points
Congratulations! You passed!
Grade received 100%
To pass 80% or higher
1.
Question 1

What is a stateless component?
1 / 1 point

A component that doesn’t track its parent’s state.

A component that doesn’t track its own state.
Correct

Correct. A stateless component doesn’t track its own state.
2.
Question 2

A stateful component must have a props object.
1 / 1 point

False

True
Correct

Correct. Whether or not a stateful component has a props object is irrelevant.
3.
Question 3

To turn a stateless component into a stateful component, you must pass it a props object.
1 / 1 point

True

False
Correct

Correct. Whether or not a stateful component receives a props object is irrelevant.
4.
Question 4

The process of lifting up state can lead to:
Select all that apply.
1 / 1 point

A stateless component becoming a stateful component.

```
Correct

That’s right! Lifting up state means moving state up from a child to the parent component - meaning that a previously stateless parent comp

A stateful child component controlling the state of a stateless parent component.

A stateful child component controlling the state of a stateful parent component.

A stateful component becoming a stateless component.
Correct

That’s right! Lifting up state means moving state up from a child to the parent component - meaning that a previously stateless parent comp
5.
Question 5

A prop doesn't have to always pass state.
1 / 1 point

True

False
Correct

That’s correct. A prop doesn't have to always pass state.
```

graded quiz

```
Skip to Main Content
Coursera
Search in course

    React Basics

Week 2

    Module quiz: Data and state

Video: VideoParent-child data flow
. Duration: 5 minutes5 min
Reading: ReadingData flow in React
. Duration: 10 minutes10 min
Video: VideoChildren and data
. Duration: 3 minutes3 min
Practice Quiz: Knowledge check: Data flow
6 questions
Video: VideoWhat are hooks?
. Duration: 3 minutes3 min
Reading: ReadingUsing hooks
. Duration: 10 minutes10 min
Video: VideoWhat is state?
. Duration: 4 minutes4 min
Video: VideoObserving state
. Duration: 3 minutes3 min
Practice Quiz: Knowledge Check: State the concept
5 questions
Video: VideoManaging state
. Duration: 4 minutes4 min
Reading: ReadingProp drilling
. Duration: 10 minutes10 min
Video: VideoReact state management
. Duration: 6 minutes6 min
Practice Quiz: Knowledge check: Passing state
5 questions
Video: VideoStateful vs. stateless
. Duration: 3 minutes3 min
Lab: Managing state in React
. Duration: 1 hour1h
```

Reading: ReadingSolution: Managing state in React
. Duration: 5 minutes5 min
Practice Quiz: Self review: Managing state in React
3 questions
Practice Quiz: Knowledge check: State or stateless
5 questions
Video: VideoModule summary
. Duration: 5 minutes5 min
Quiz: Module quiz: Data and state
10 questions

Reading: ReadingAdditional resources
. Duration: 5 minutes5 min

Module quiz: Data and state
Quiz30 minutes • 30 min
Submit your assignment
Due September 24, 11:59 PM PDTSep 24, 11:59 PM PDT
Attempts 3 every 8 hours
Receive grade
To Pass 80% or higher
Your grade
90%

We keep your highest score
Module quiz: Data and state

Graded Quiz. • 30 min
DueSep 24, 11:59 PM PDT
Congratulations! You passed!
Grade received 90%
Latest Submission Grade 90%
To pass 80% or higher
1.
Question 1

In React, data flows in one way: from a parent component to a child component.
1 / 1 point

True

False
Correct

That’s correct. The above statement is true.
2.
Question 2

Why is one-way data flow important in React?
1 / 1 point

It ensures that the data is flowing from top to bottom in the component hierarchy.

It ensures that the data is flowing from bottom to top in the component hierarchy.
Correct

Correct. This ensures that the data is flowing from top to bottom in the component hierarchy.
3.
Question 3

True or false? State data is the data inside a component that a component can mutate.
1 / 1 point

True

False
Correct

Correct. State data is data inside a component that a component, which that component controls and can mutate.
4.
Question 4

What is prop drilling?
1 / 1 point

Prop drilling is a situation where you are passing data from a parent to a child component, then to a grandchild component, and so on, unti

Prop drilling is a situation where you are passing data from a child, to a parent component, then to a grandparent component, and so on, un

Correct

Correct. Prop drilling is a situation where you are passing data from a parent to a child component, then to a grandchild component, and so
5.
Question 5

The distinction between stateful and stateless components is that the latter doesn't have its own state.
1 / 1 point

True

False
Correct

That’s correct! This statement is true.
6.
Question 6

Choose the correct statement.
1 / 1 point

Remember that you should always change the values of props in children components; you should never work with them as they are. In other wo

Remember that you should never change the values of props in children components; you should only work with them as they are. In other word
Correct

That’s correct! Props are immutable and thus you should not attempt to update them in children components.
7.
Question 7

Is this code valid?
11

1 / 1 point

Yes

No
Correct

Correct! This code is an example of a valid onClick event handler.
8.
Question 8

Is this code valid?
1
2
3

1 / 1 point

Yes

No
Correct

Correct! This code is an example of a valid onClick event handler.
9.
Question 9

Select the correct statement: The useState hook...
0 / 1 point

... lets you hook into React state and lifecycle features from a component.

...is not part of React; you must import it from a third-party package.

... has a convention that if the state variable is named, for example, counter, the function to update this counter variable should be name

... should never be called at the top level of a React component.
Incorrect

Not quite. Please go back and revise the lesson item titled Reading: What are Hooks?
10.
Question 10

The Context API allows you to:
1 / 1 point

```
Avoid having to pass state down through multiple levels of components.

Avoid having to use the return statement in a child component.

Avoid having to keep your components modular.
Correct

Correct. Using Context API allows you to bypass having to pass state down through multiple levels of components.
```