UCLA Samueli
Computer Science

Web Development in React

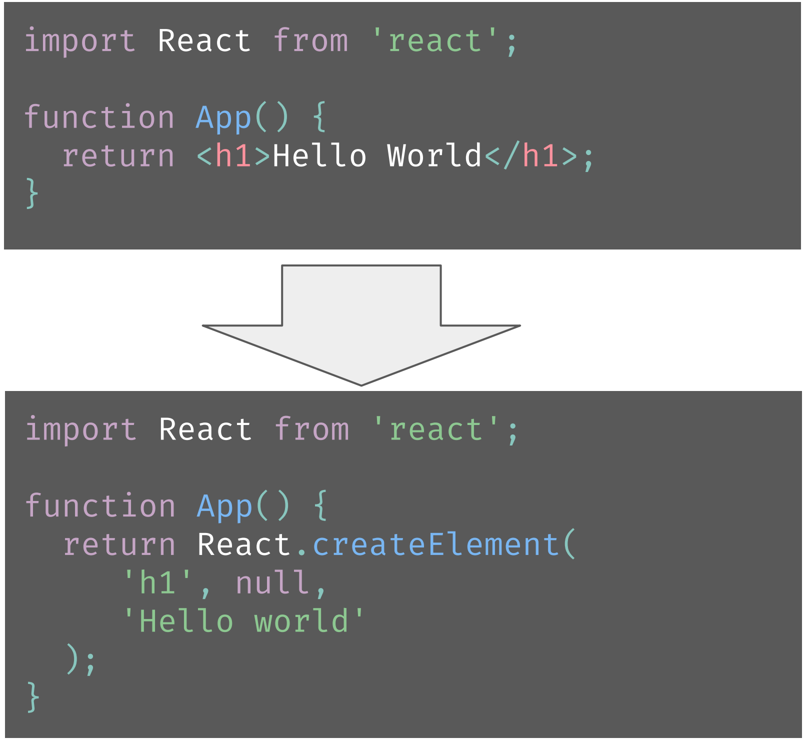
Paul Serafimescu (Winter 2024)

Why Learn React?

- Developed by Facebook (now Meta) in 2013
- Declarative & Component-Based
  - Why is this a good thing?
  - **Easier to maintain, encapsulate logic, use modern features, good for complex web apps**
- Most popular frontend framework as of 2023
  - There are variations such as Next.js which are heavily influenced by React
- How to get started?
  - Make sure you have Node.js installed!
    - `$ npm install -g create-react-app` if you don't have it already
  - `$ npx create-react-app <name>`
  - `$ cd <name>`
  - `$ npm start`

JSX

- Stands for JavaScript XML
- Syntactic Sugar
  - Makes programming easier, why?
- Converted to browser JavaScript by process of **transpilation**
  - Transpiling also allows programmer to use newer features of language than what is supported on a browser. How would this work?



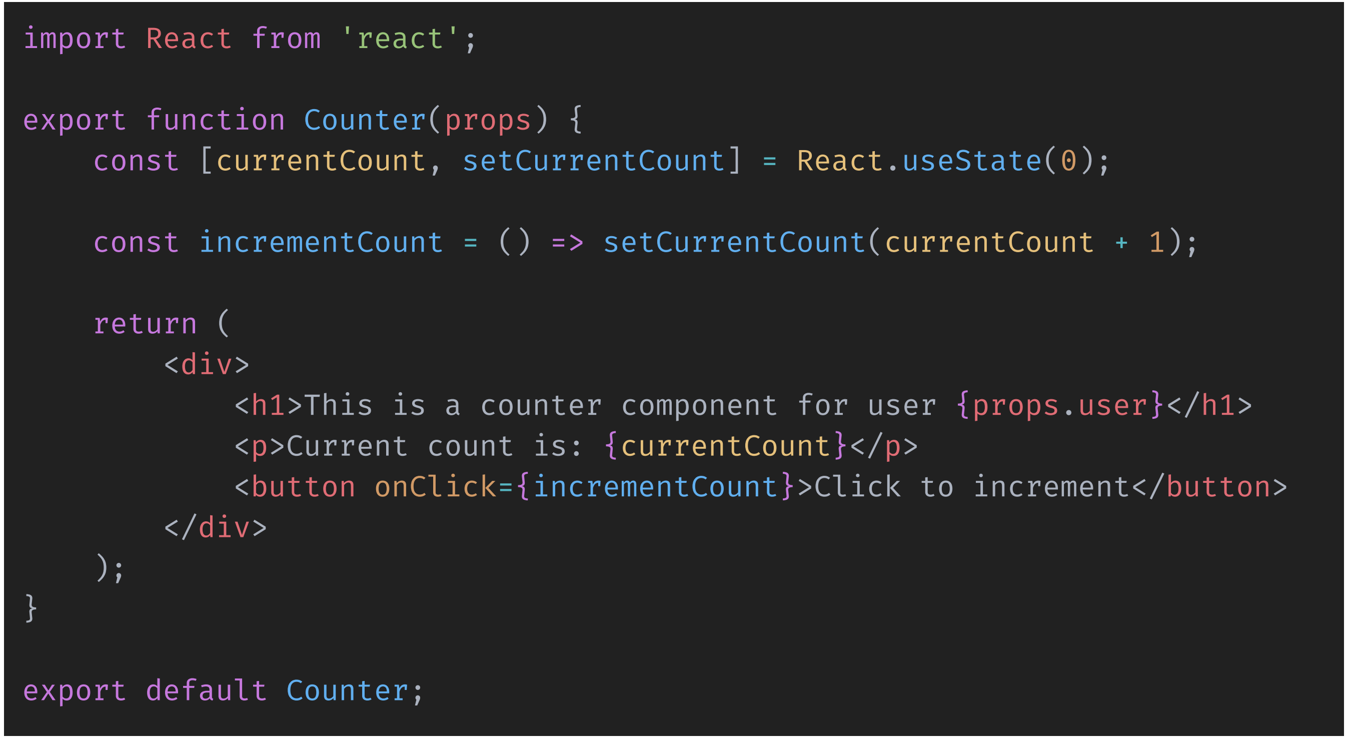
```
import React from 'react';

function App() {
  return <h1>Hello World</h1>;
}
```

```
import React from 'react';

function App() {
  return React.createElement(
      'h1', null,
      'Hello world'
  );
}
```

A Basic Component



```
import React from 'react';

export function Counter(props) {
    const [currentCount, setCurrentCount] = React.useState(0);

    const incrementCount = () => setCurrentCount(currentCount + 1);

    return (
        <div>
            <h1>This is a counter component for user {props.user}</h1>
            <p>Current count is: {currentCount}</p>
            <button onClick={incrementCount}>Click to increment</button>
        </div>
    );
}

export default Counter;
```

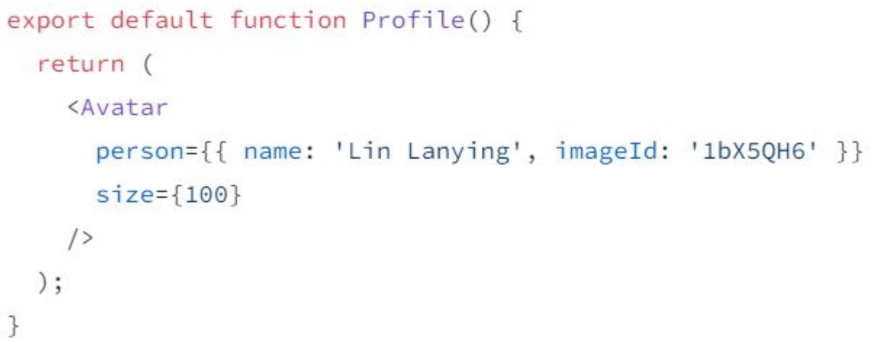
What does this component do?

Counts when button is clicked

What makes that functionality possible?

React Props

- Short for “properties”
- Think of these as immutable values passed into the component “constructor”
- How does the previous component use them?
    - **`{props.user}` comes from the context the component is created**
- Why is this good for a web page?
    - **Logic can be reused without knowing about authentication, etc.**

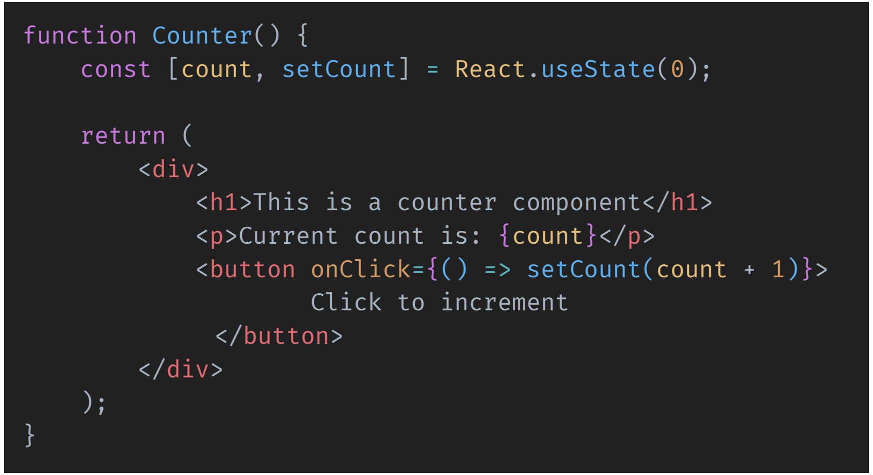


```
export default function Profile() {
  return (
    <Avatar
      person={{ name: 'Lin Lanying', imageId: '1bX5QH6' }}
      size={100}
    />
  );
}
```

How are props passed in the above example?

React State

- Created using `React.useState(initialValue)`
  - Returns a variable containing the current state, and a function used to set a new state
- How does the component (right) use it?
  - **`count`, button click**
- Why is this useful?
  - **Each component can keep track of its own stateful logic (loading, web forms, change behavior based on interactivity)**



```
function Counter() {
    const [count, setCount] = React.useState(0);

    return (
        <div>
            <h1>This is a counter component</h1>
            <p>Current count is: {count}</p>
            <button onClick={() => setCount(count + 1)}>
                    Click to increment
             </button>
        </div>
    );
}
```

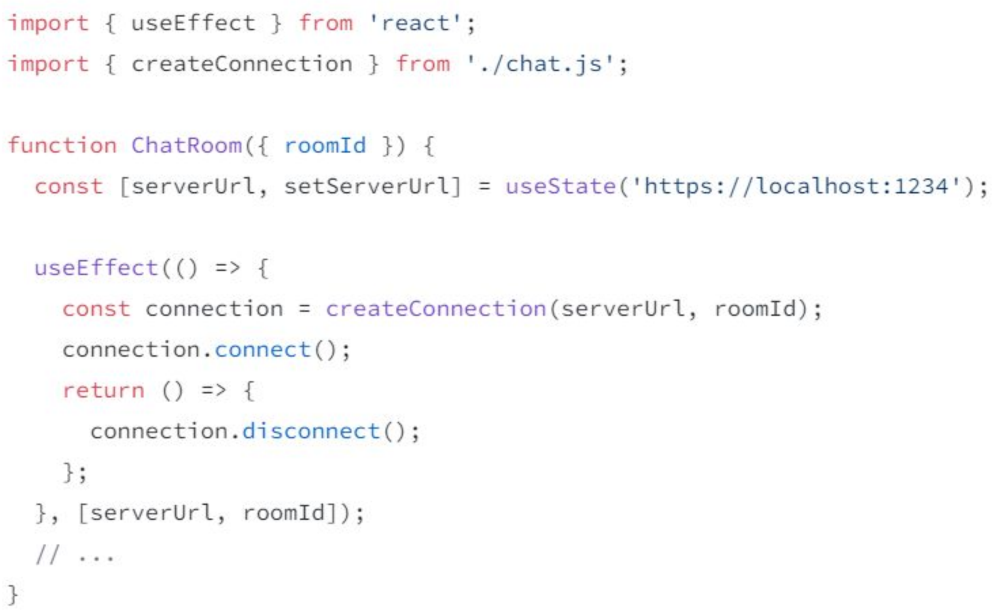
Why do we need to use state here? Why isn't a regular variable enough?

React Effects

- `useEffect` takes in two parameters
  - First is a function with no arguments which contains the code to run
    - If there is “cleanup” logic, this function should return ANOTHER function with no arguments which contains the code to run when the component is removed from the DOM
  - Second is an array which contains the dependencies
    - If the value of one of the dependency variables changes, then the effect is re-triggered
      - An empty array means the effect will run once, when the component is first mounted
- What are some use cases?
  - **Asynchronous data fetching, setting a timer, animations, any others?**

React Effects (Example)

- What does this component do?
  - **Represents chatroom connection**
- What is the purpose of `useEffect` in this component?
  - **Once the component is created on the DOM, then connect to chatroom backend**
  - What does each argument do?
- Why can’t any of the other constructs we’ve covered achieve the same goal?

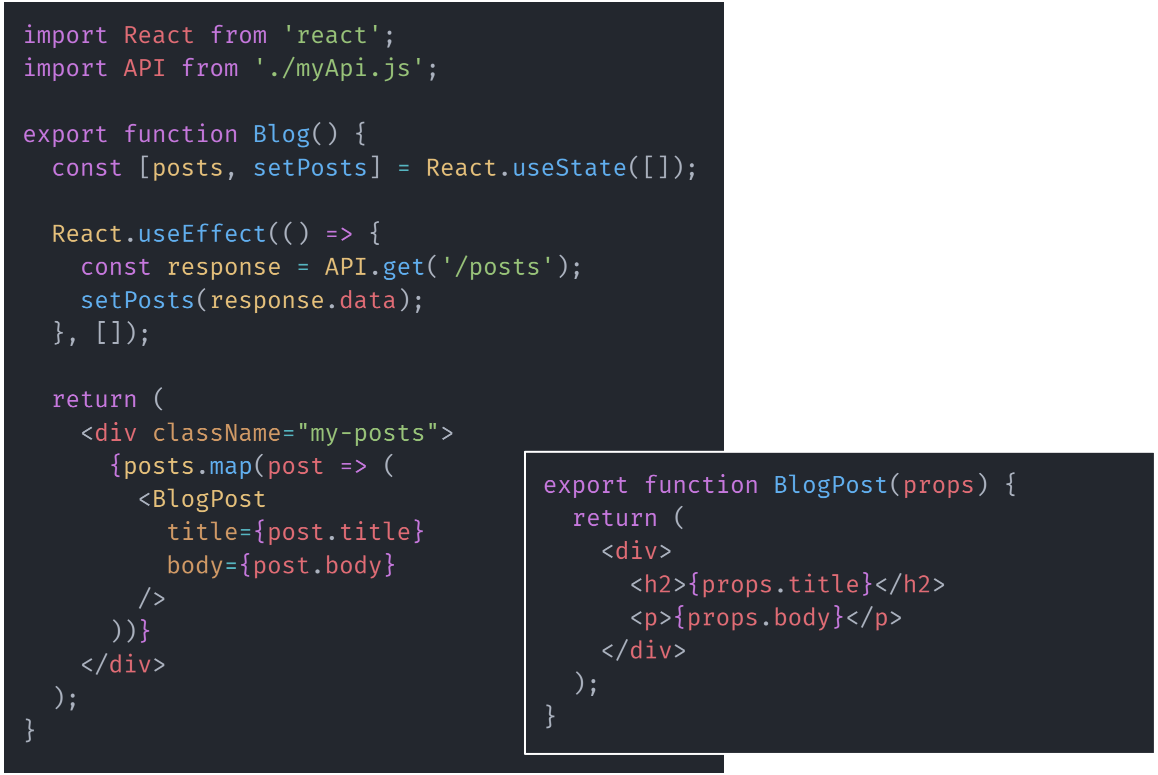


```
import { useEffect } from 'react';
import { createConnection } from './chat.js';

function ChatRoom({ roomId }) {
  const [serverUrl, setServerUrl] = useState('https://localhost:1234');

  useEffect(() => {
    const connection = createConnection(serverUrl, roomId);
    connection.connect();
    return () => {
      connection.disconnect();
    };
  }, [serverUrl, roomId]);
  // ...
}
```

Fetching Data



```
import React from 'react';
import API from './myApi.js';

export function Blog() {
  const [posts, setPosts] = React.useState([]);

  React.useEffect(() => {
    const response = API.get('/posts');
    setPosts(response.data);
  }, []);

  return (
    <div className="my-posts">
      {posts.map(post => (
        <BlogPost
          title={post.title}
          body={post.body}
        />
      ))}
    </div>
  );
}
```

```
export function BlogPost(props) {
  return (
    <div>
      <h2>{props.title}</h2>
      <p>{props.body}</p>
    </div>
  );
}
```

With the person sitting nearest to you, discuss how this component fetches data, and pay particular attention to the order in which lifecycle functions are executed.

How would the component appear on the web page over time?

Hint: .map() method applies a function passed as an argument to each element of the array, and returns a new array with the returned values

Brief Intro to Selenium

- Used to perform automation testing on web pages
  - Automation testing can be used to make sure UI behavior and appearance is correct, check cross-browser compatibility, and regression test (what does this mean?)
    - **Regression testing makes sure no new code has introduced a bug in old features**
- https://web.cs.ucla.edu/classes/winter24/cs35L/assign/test_chorus_lapilli.py
  - What does each method in the script do? What does the `WebElement` class represent? How do you think Selenium is able to find these elements on the webpage?
  - **`WebElement` represents a single HTML element on the DOM. Selenium can take advantage of WebDriver protocol which has more functionality than simple Javascript**
- `Useful Documentation Link`

Useful Links for the Project

- https://react.dev/learn/passing-props-to-a-component
- https://react.dev/learn/managing-state
- https://react.dev/reference/react/useEffect
- https://tanstack.com/query/v3/docs/framework/react/overview
- https://mui.com/material-ui/ (component library -> speeds up development)
- https://www.typescriptlang.org/ (for those who prefer static typing)
- https://github.com/paul-serafimescu/f22-cs35L-frontend-ws (frontend LA example)
- https://github.com/paul-serafimescu/s22-cs35l-backend-ws (backend LA example)
- https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise
- https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/async_function (async/await and Promises should be read together)

UCLA Samueli Computer Science

Have a good weekend!

We hope you learned something too