Salvatore Ventrone (aka Ventrosky)

Sviluppatore Web Full Stack

Introduzione agli Hooks di React

Develer Webinar

14/09/2022

develer

DI COSA PARLEREMO

- Che problemi risolvono
- Tre hooks base:
  - useState
  - useEffect
  - useContext

Cos’è React?

- Una libreria JavaScript lato client
- Dichiarativa
- Basata su Componenti
- Tipicamente utilizzata per sviluppare interfacce reattive per il web

Perchè gli hooks - Ciclo di vita dei componenti

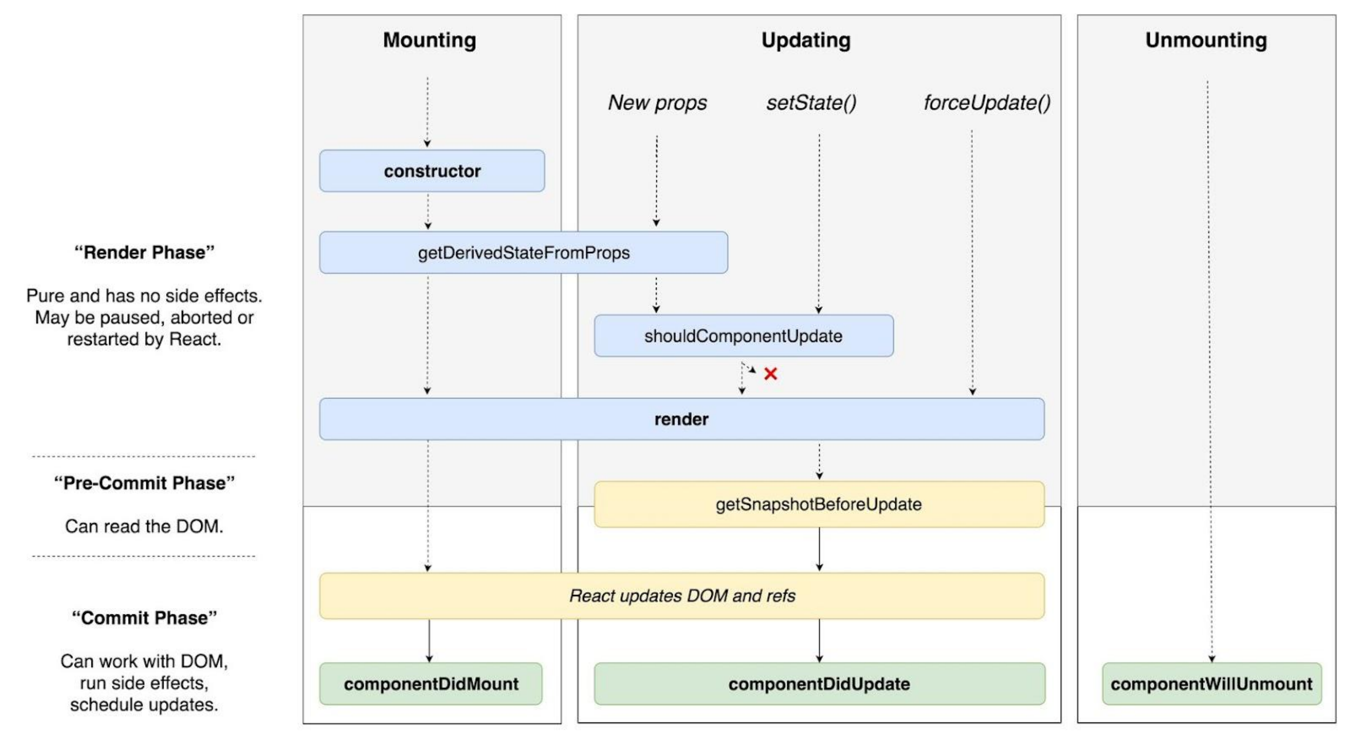
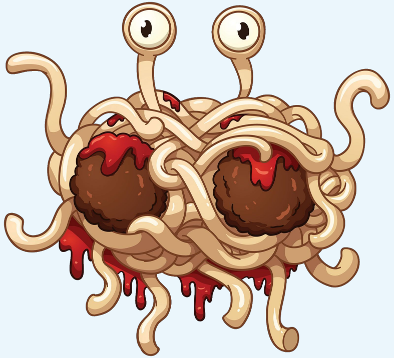


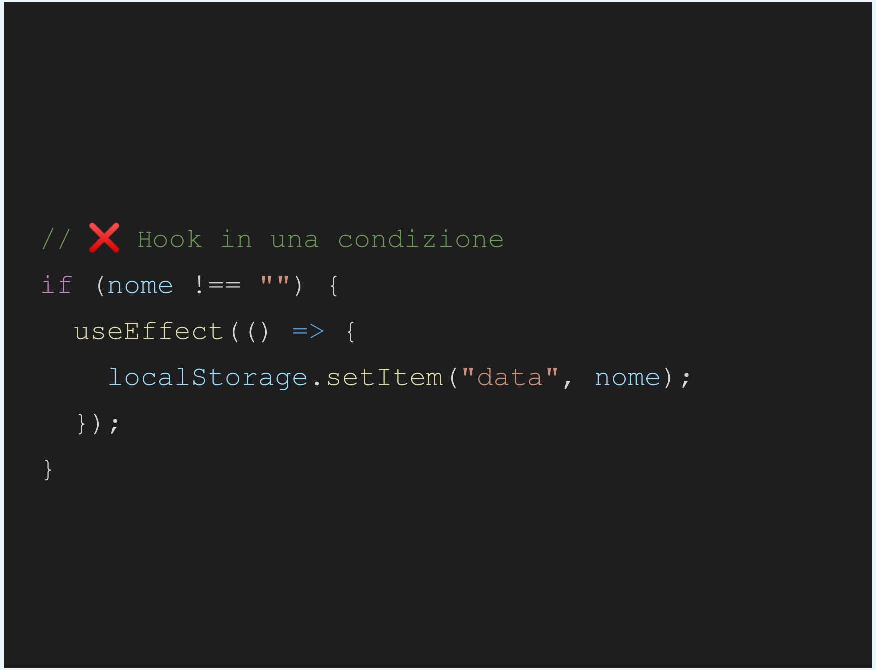
Diagramma da Dan Abramov



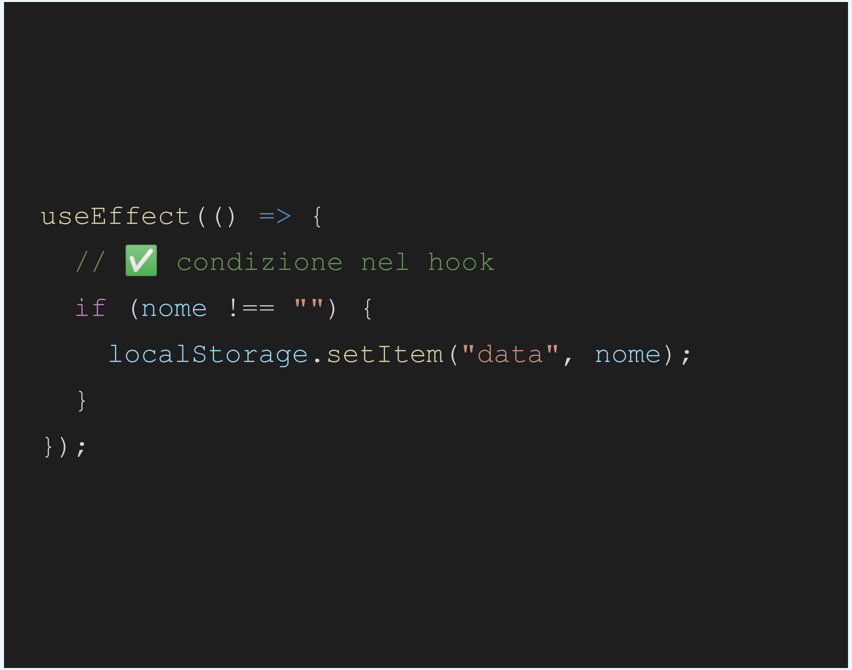
Le regole degli hooks

- Utilizza gli Hooks solo al Top Level
- Utilizza gli Hooks da funzioni React

Regola 1 - solo al Top Level

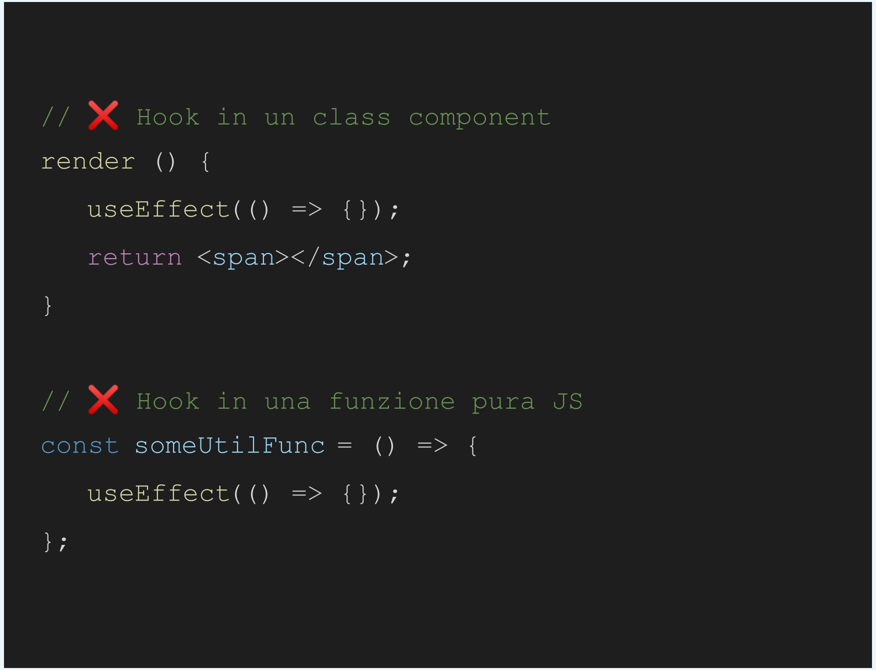


```
// ❌ Hook in una condizione
if (nome !== "") {
  useEffect(() => {
    localStorage.setItem("data", nome);
  });
}
```

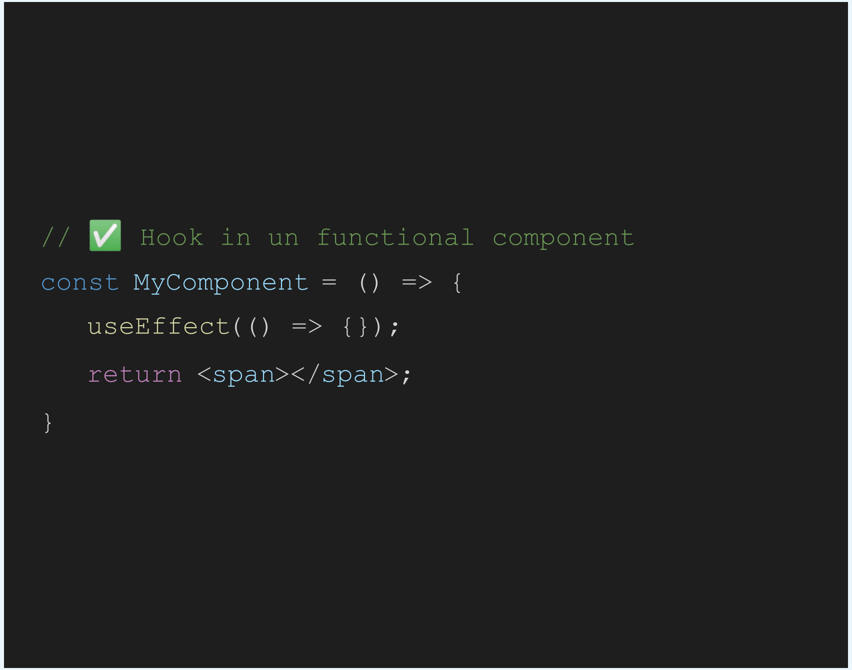


```
useEffect(() => {
  // ✅ condizione nel hook
  if (nome !== "") {
    localStorage.setItem("data", nome);
  }
});
```

Regola 2 - solo da functional components



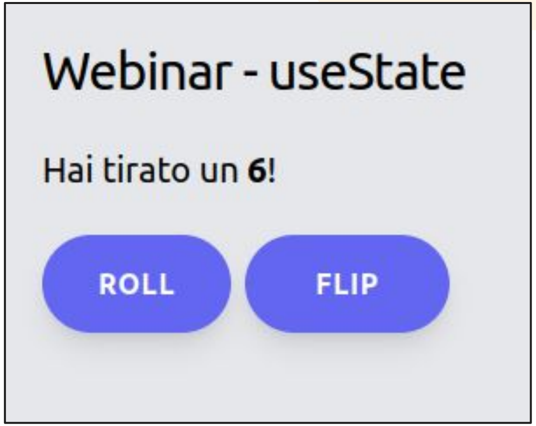
```
// ❌ Hook in un class component
render () {
   useEffect(() => {});
   return <span></span>;
}

// ❌ Hook in una funzione pura JS
const someUtilFunc = () => {
   useEffect(() => {});
};
```



```
// ✅ Hook in un functional component
const MyComponent = () => {
   useEffect(() => {});
   return <span></span>;
}
```

Hooks Base

useState

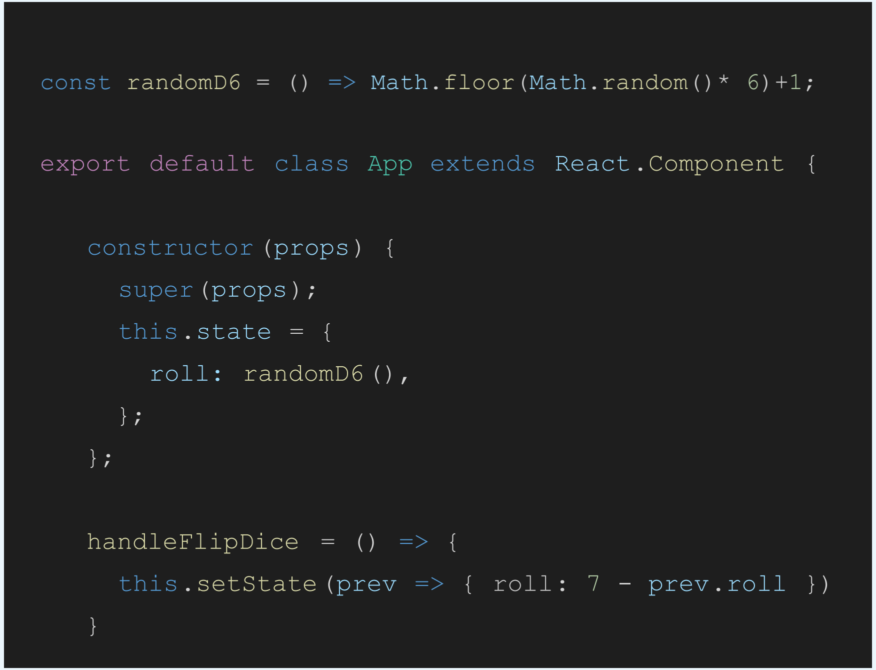


Webinar - useState

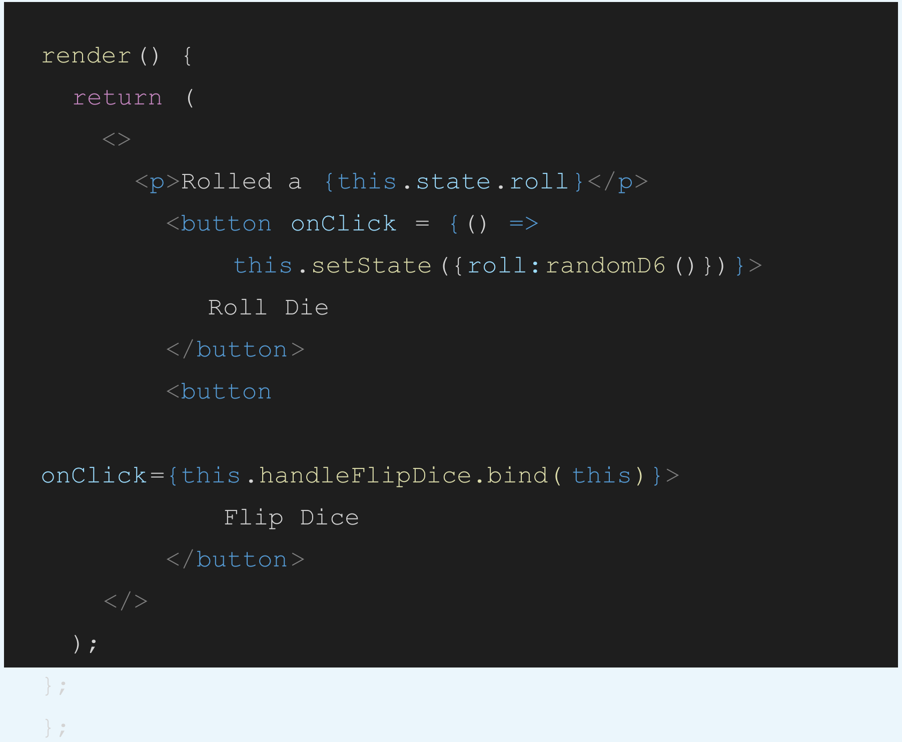
Hai tirato un **6**!

https://codepen.io/BuccaneerDev/pen/Poezmaa

useState - il problema

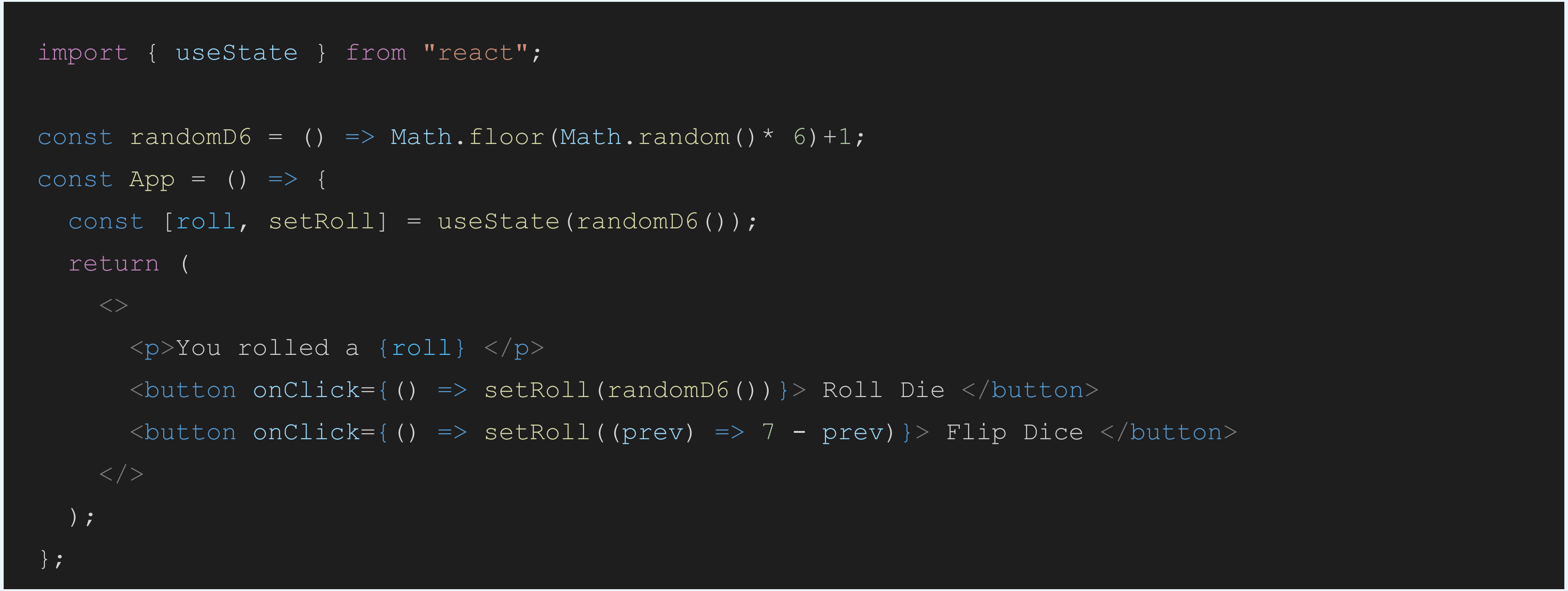


```
const randomD6 = () => Math.floor(Math.random()* 6)+1;

export default class App extends React.Component {

   constructor(props) {
     super(props);
     this.state = {
       roll: randomD6(),
     };
   };

   handleFlipDice = () => {
     this.setState(prev => { roll: 7 - prev.roll })
   }
```



```
render() {
  return (
    <>
      <p>Rolled a {this.state.roll}</p>
        <button onClick = {() =>
            this.setState({roll:randomD6()})}>
           Roll Die
        </button>
        <button

onClick={this.handleFlipDice.bind( this)}>
            Flip Dice
        </button>
    </>
  );
};
};
```

useState - la soluzione

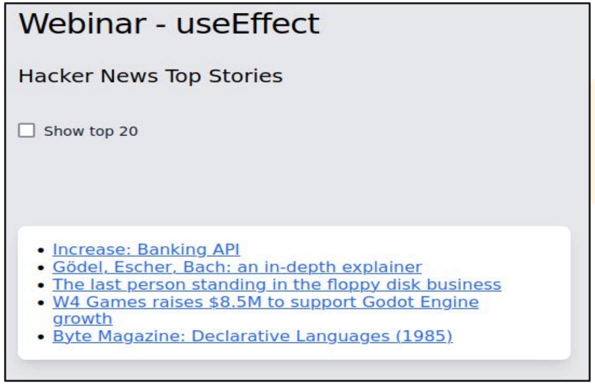


```
import { useState } from "react";

const randomD6 = () => Math.floor(Math.random()* 6)+1;
const App = () => {
  const [roll, setRoll] = useState(randomD6());
  return (
    <>
      <p>You rolled a {roll} </p>
      <button onClick={() => setRoll(randomD6())}> Roll Die </button>
      <button onClick={() => setRoll((prev) => 7 - prev)}> Flip Dice </button>
    </>
  );
};
```

Demo - useState

Hooks Base

useEffect



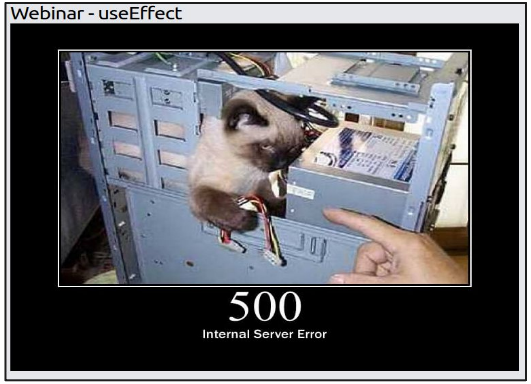
Webinar - useEffect

Hacker News Top Stories

□ Show top 20

- Increase: Banking API
- Gödel, Escher, Bach: an in-depth explainer
- The last person standing in the floppy disk business
- W4 Games raises $8.5M to support Godot Engine growth
- Byte Magazine: Declarative Languages (1985)

https://codepen.io/BuccaneerDev/pen/oNdLpMN



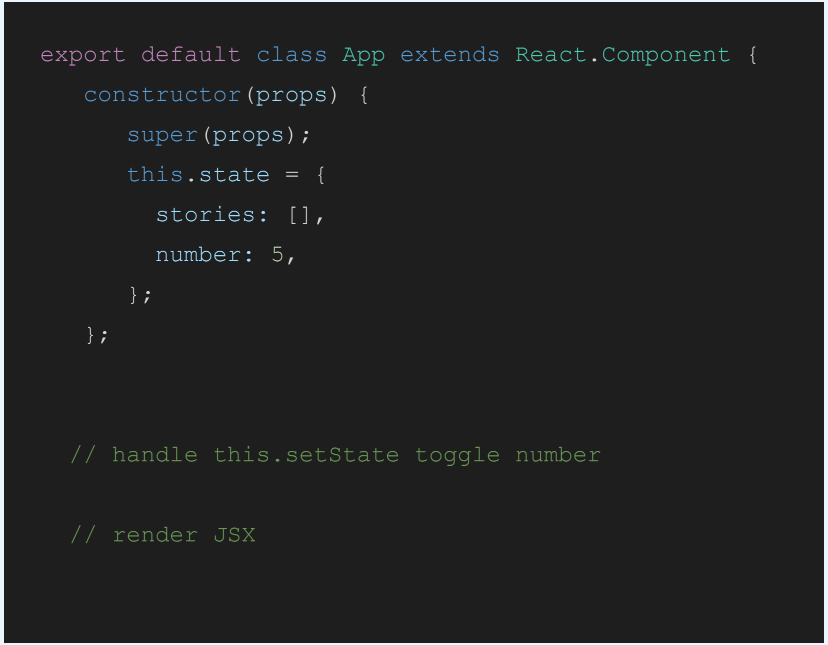
Webinar - useEffect

500

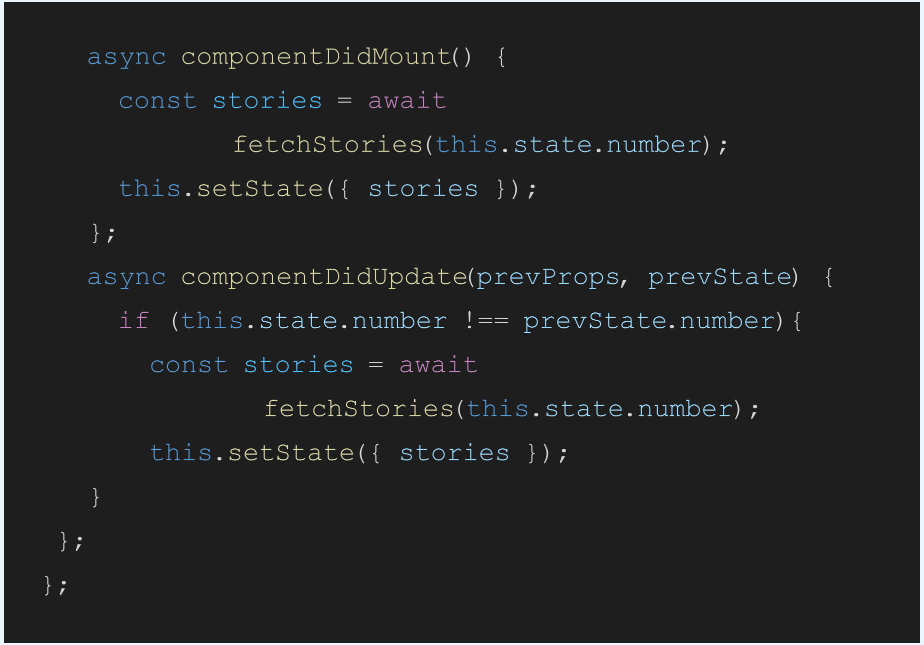
Internal Server Error

https://codepen.io/BuccaneerDev/pen/yLjaVEr

useEffect - il problema

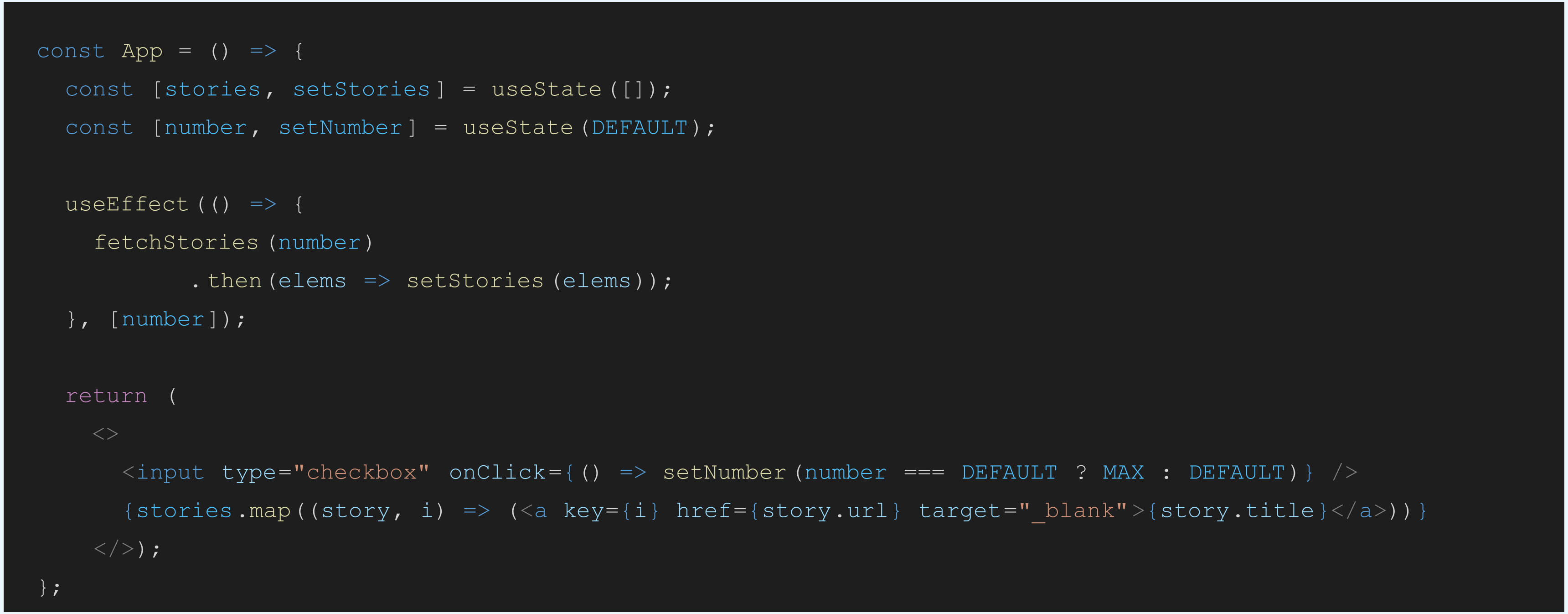


```
export default class App extends React.Component {
   constructor(props) {
      super(props);
      this.state = {
        stories: [],
        number: 5,
      };
   };


  // handle this.setState toggle number

  // render JSX
```



```
   async componentDidMount() {
     const stories = await
             fetchStories(this.state.number);
     this.setState({ stories });
   };
   async componentDidUpdate(prevProps, prevState) {
     if (this.state.number !== prevState.number){
       const stories = await
               fetchStories(this.state.number);
       this.setState({ stories });
     }
  };
};
```

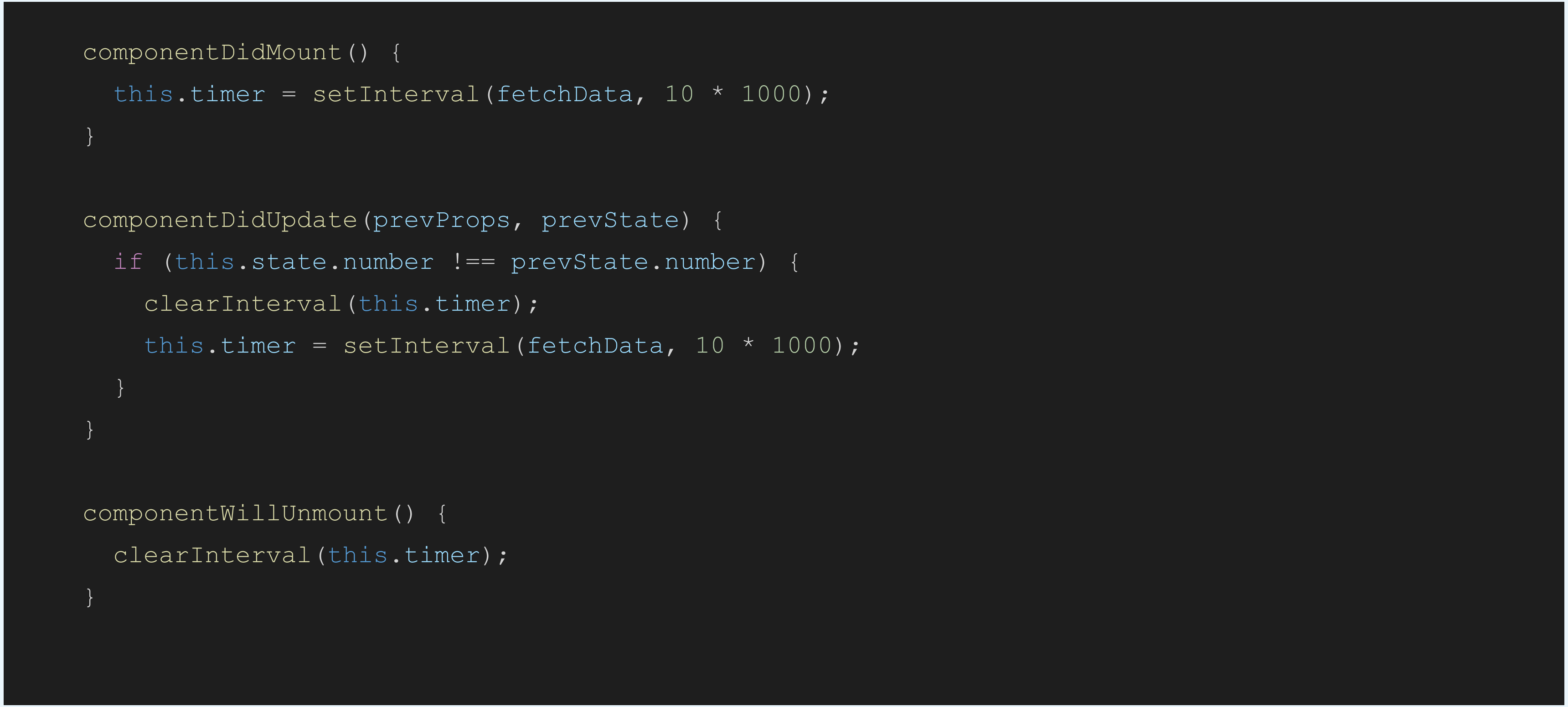
useEffect - la soluzione



```
const App = () => {
  const [stories, setStories] = useState([]);
  const [number, setNumber] = useState(DEFAULT);

  useEffect(() => {
    fetchStories(number)
           .then(elems => setStories(elems));
  }, [number]);

  return (
    <>
      <input type="checkbox" onClick={() => setNumber(number === DEFAULT ? MAX : DEFAULT)} />
      {stories.map((story, i) => (<a key={i} href={story.url} target="_blank">{story.title}</a>))}
    </>);
};
```

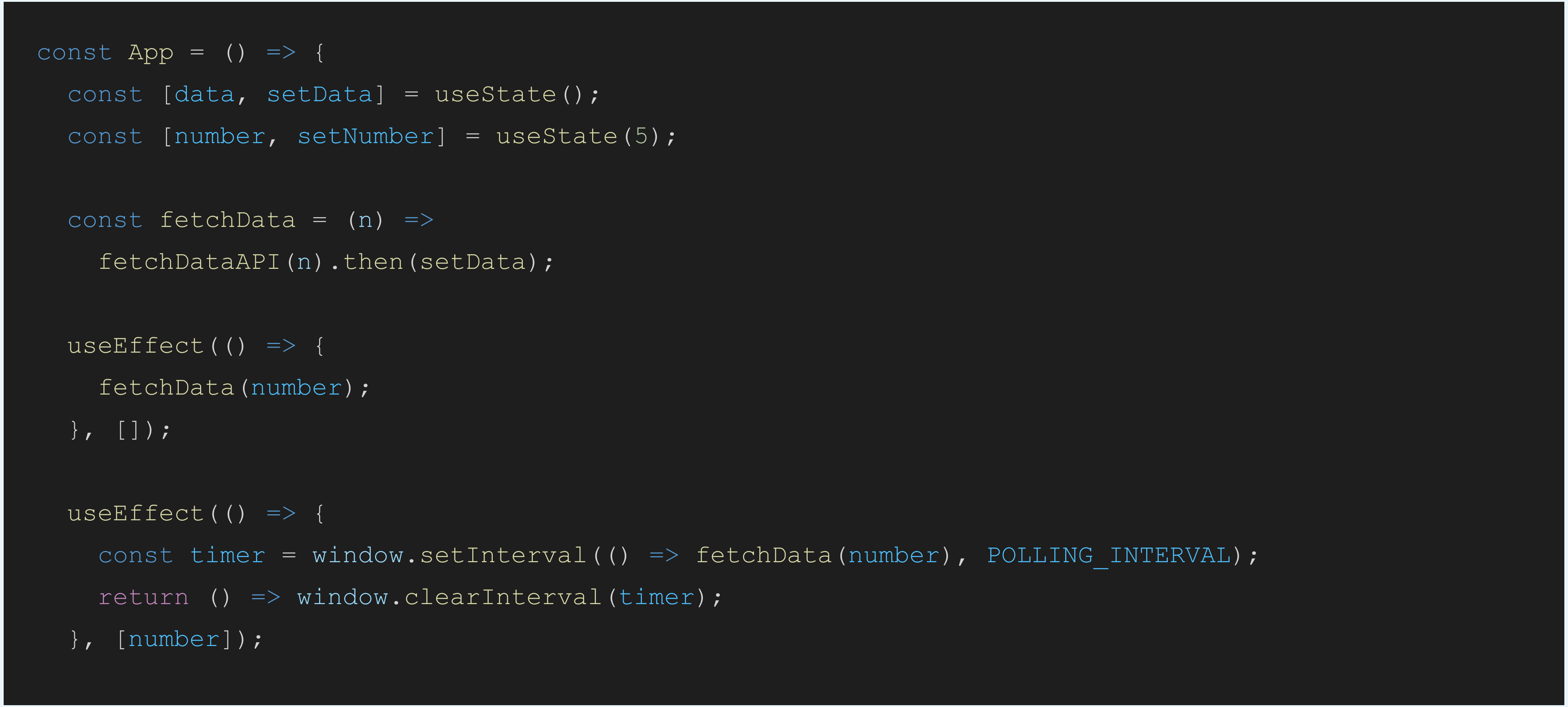
Demo - useEffect

useEffect - il problema



```
componentDidMount() {
  this.timer = setInterval(fetchData, 10 * 1000);
}

componentDidUpdate(prevProps, prevState) {
  if (this.state.number !== prevState.number) {
    clearInterval(this.timer);
    this.timer = setInterval(fetchData, 10 * 1000);
  }
}

componentWillUnmount() {
  clearInterval(this.timer);
}
```

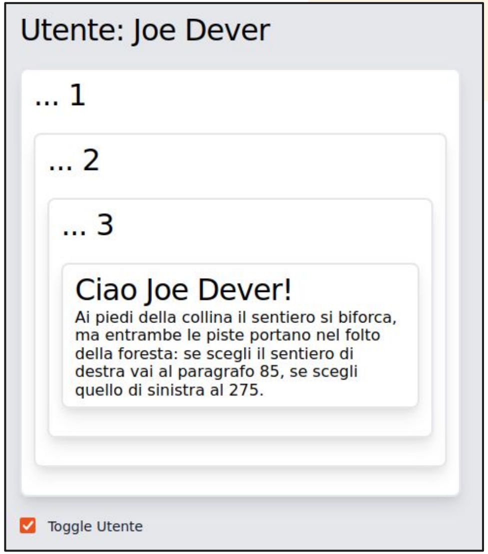
useEffect - la soluzione



```
const App = () => {
  const [data, setData] = useState();
  const [number, setNumber] = useState(5);

  const fetchData = (n) =>
    fetchDataAPI(n).then(setData);

  useEffect(() => {
    fetchData(number);
  }, []);

  useEffect(() => {
    const timer = window.setInterval(() => fetchData(number), POLLING_INTERVAL);
    return () => window.clearInterval(timer);
  }, [number]);
```

Demo - cleanup

useContext



Utente: Joe Dever

... 1

... 2

... 3

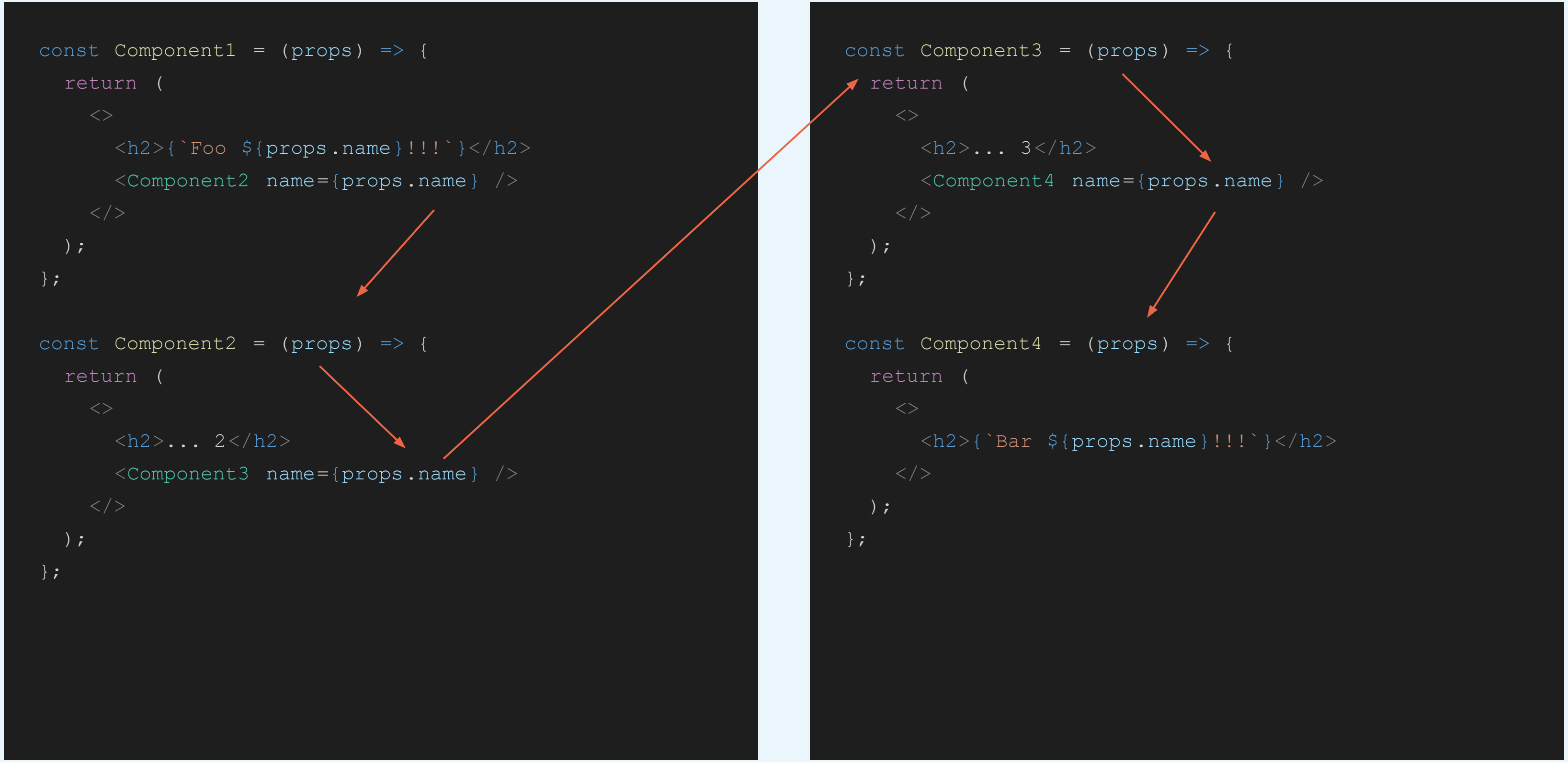
Ciao Joe Dever!

Ai piedi della collina il sentiero si biforca, ma entrambe le piste portano nel folto della foresta: se scegli il sentiero di destra vai al paragrafo 85, se scegli quello di sinistra al 275.

☑ Toggle Utente

https://codepen.io/BuccaneerDev/pen/yLjJjOz

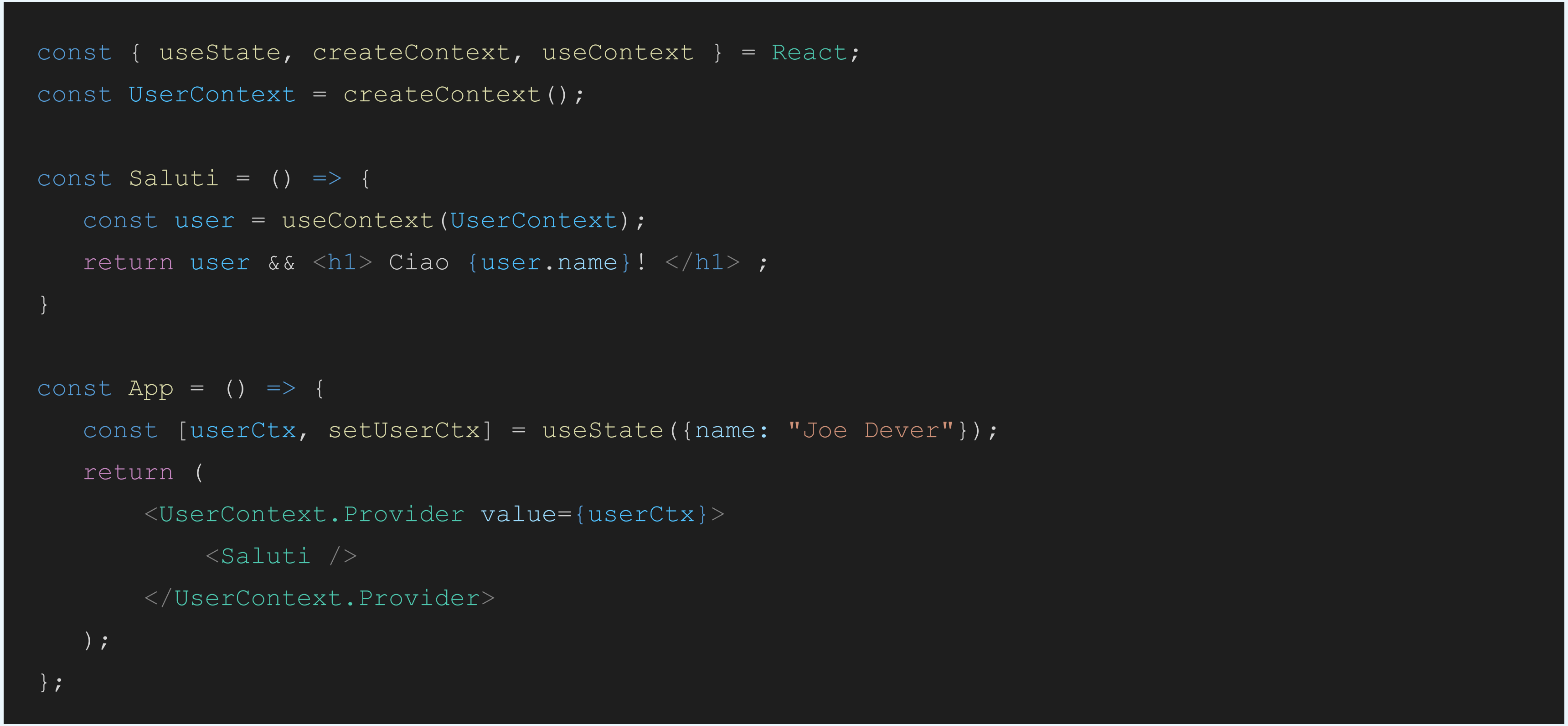
useContext - il problema



```
const Component1 = (props) => {
  return (
    <>
      <h2>{`Foo ${props.name}!!!`}</h2>
      <Component2 name={props.name} />
    </>
  );
};

const Component2 = (props) => {
  return (
    <>
      <h2>... 2</h2>
      <Component3 name={props.name} />
    </>
  );
};
```

```
const Component3 = (props) => {
  return (
    <>
      <h2>... 3</h2>
      <Component4 name={props.name} />
    </>
  );
};

const Component4 = (props) => {
  return (
    <>
      <h2>{`Bar ${props.name}!!!`}</h2>
    </>
  );
};
```

useContext - la soluzione



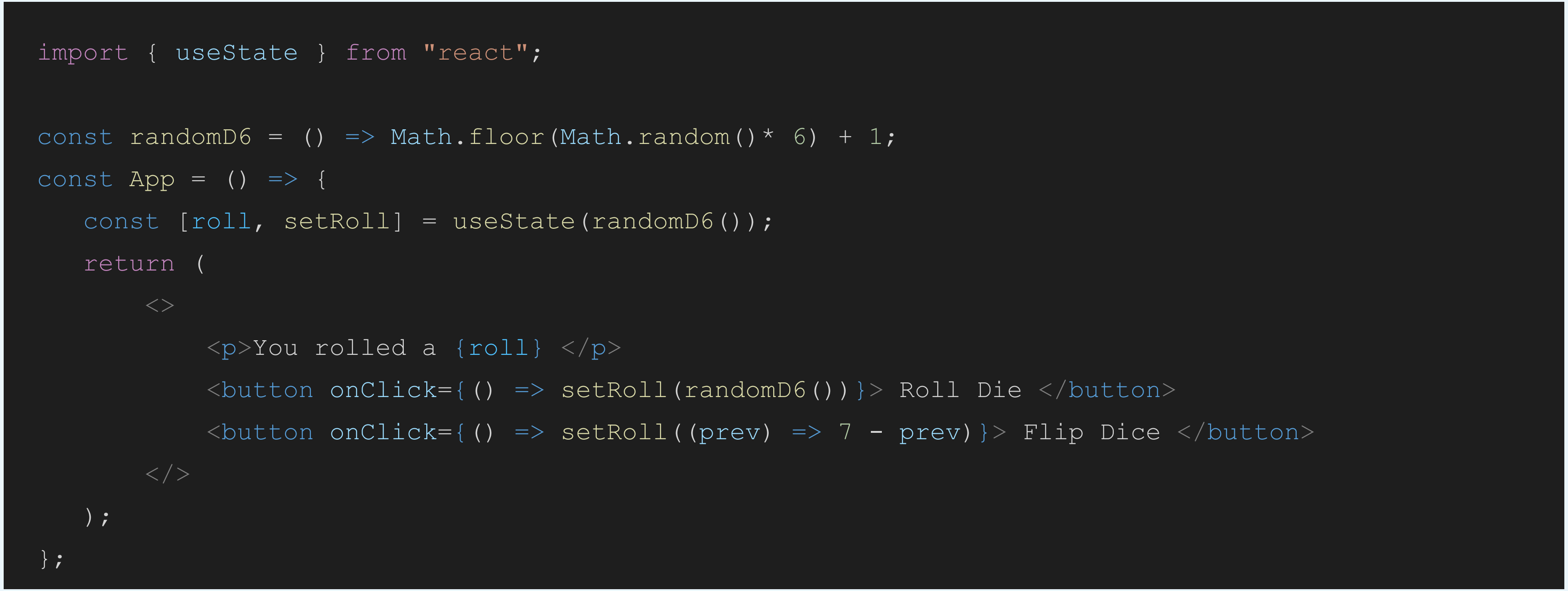
```
const { useState, createContext, useContext } = React;
const UserContext = createContext();

const Saluti = () => {
   const user = useContext(UserContext);
   return user && <h1> Ciao {user.name}! </h1> ;
}

const App = () => {
   const [userCtx, setUserCtx] = useState({name: "Joe Dever"});
   return (
       <UserContext.Provider value={userCtx}>
           <Saluti />
       </UserContext.Provider>
   );
};
```

Demo - useContext

Ricapitoliamo

- Regole degli Hooks
- useState
- useEffect
- useContext

useState - la soluzione



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

const randomD6 = () => Math.floor(Math.random()* 6) + 1;
const App = () => {
   const [roll, setRoll] = useState(randomD6());
   return (
       <>
           <p>You rolled a {roll} </p>
           <button onClick={() => setRoll(randomD6())}> Roll Die </button>
           <button onClick={() => setRoll((prev) => 7 - prev)}> Flip Dice </button>
       </>
   );
};
```

LINK UTILI

- React Docs
  https://reactjs.org/docs/getting-started.html

- Regole degli Hooks
  https://reactjs.org/docs/hooks-rules.html

- You Don't Know JS Yet
  https://github.com/getify/You-Dont-Know-JS

Salvatore Ventrone

ventrosky@develer.com

Ventrosky

develer

www.develer.com