HANDLING EVENTS IN REACT

Event handlers in React are functions you write to respond to user interactions. From form submissions to mouse movements, event handlers help you manage how your application responds to user input.

WHAT ARE SYNTHETIC EVENTS?

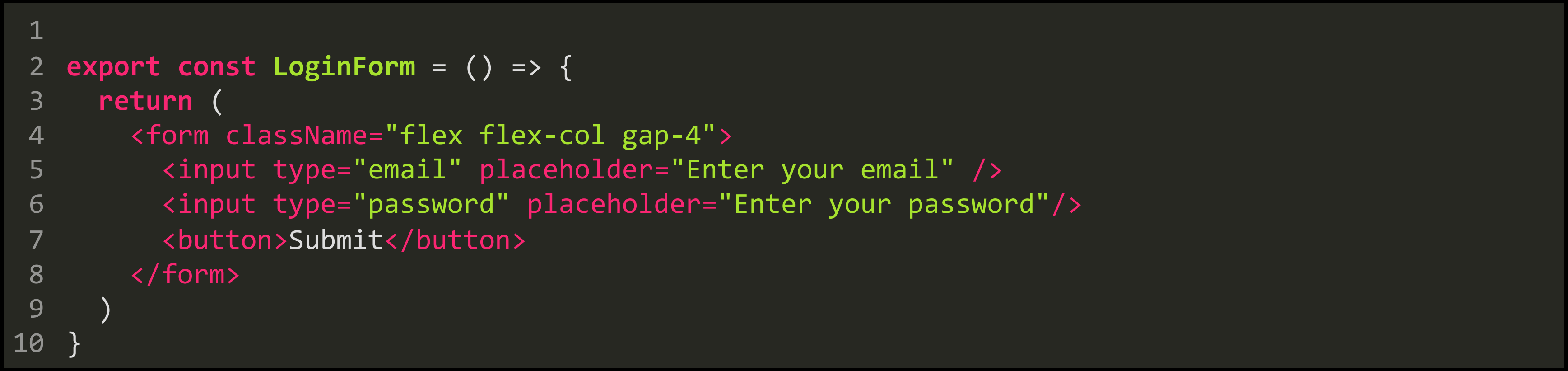
WHAT ARE SYNTHETIC EVENTS?

- React wraps the browser's native events into what we call "Synthetic events"

WHAT ARE SYNTHETIC EVENTS?

- React wraps the browser's native events into what we call "Synthetic events"
- This provides consistency and ensures that the event handlers work the same way across browsers.

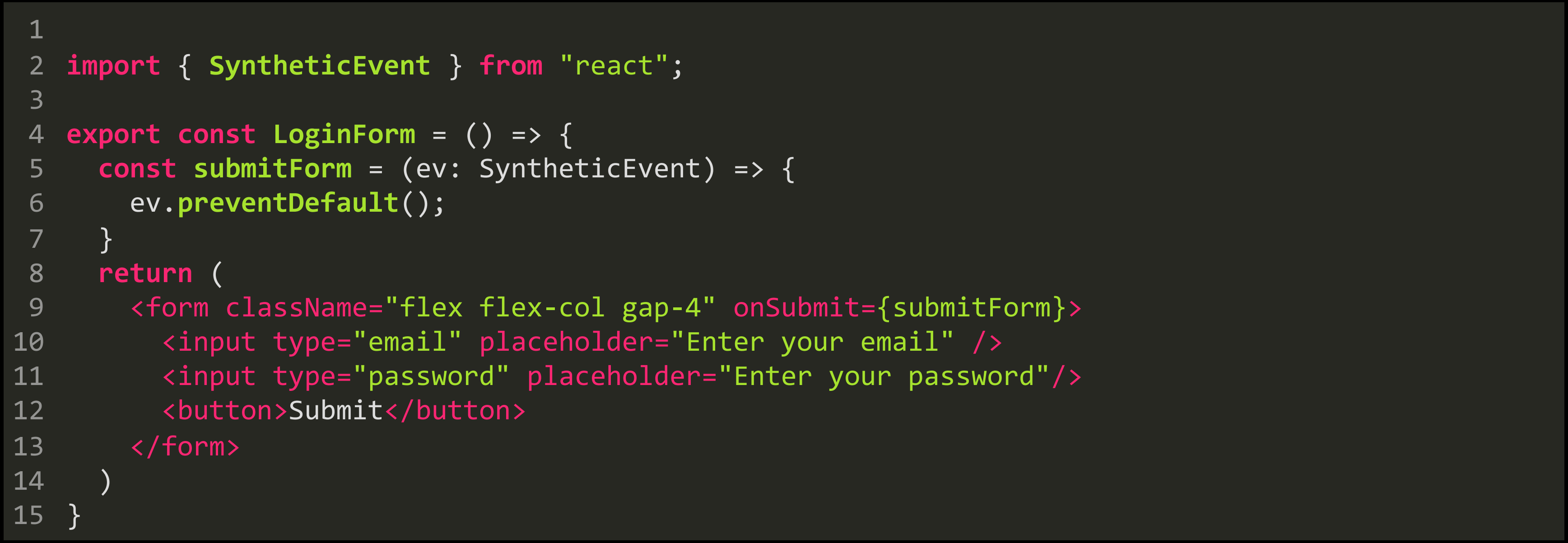
EXAMPLE OF SYNTHETIC EVENT



```

export const LoginForm = () => {
  return (
    <form className="flex flex-col gap-4">
      <input type="email" placeholder="Enter your email" />
      <input type="password" placeholder="Enter your password"/>
      <button>Submit</button>
    </form>
  )
}
```

EXAMPLE OF SYNTHETIC EVENT

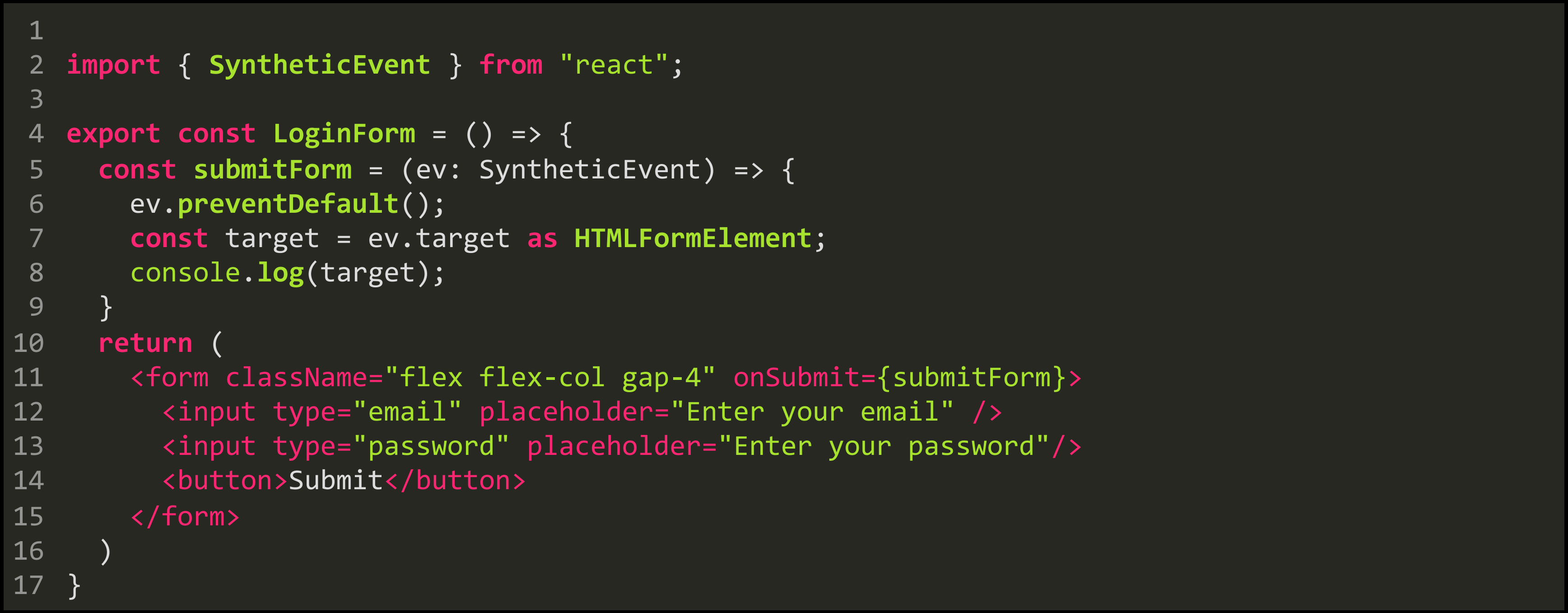


```

import { SyntheticEvent } from "react";

export const LoginForm = () => {
  const submitForm = (ev: SyntheticEvent) => {
    ev.preventDefault();
  }
  return (
    <form className="flex flex-col gap-4" onSubmit={submitForm}>
      <input type="email" placeholder="Enter your email" />
      <input type="password" placeholder="Enter your password"/>
      <button>Submit</button>
    </form>
  )
}
```

EXAMPLE OF SYNTHETIC EVENT

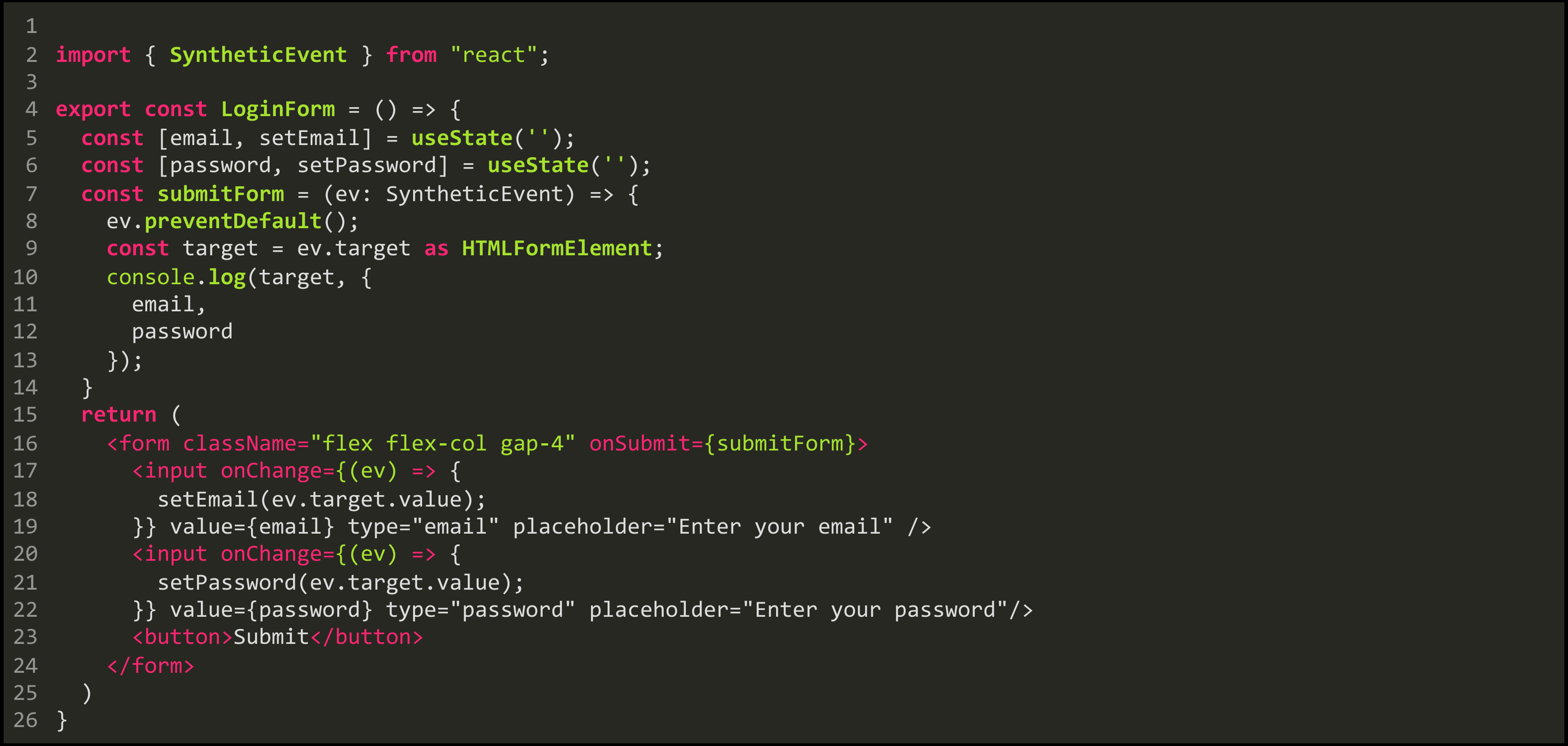


```

import { SyntheticEvent } from "react";

export const LoginForm = () => {
  const submitForm = (ev: SyntheticEvent) => {
    ev.preventDefault();
    const target = ev.target as HTMLFormElement;
    console.log(target);
  }
  return (
    <form className="flex flex-col gap-4" onSubmit={submitForm}>
      <input type="email" placeholder="Enter your email" />
      <input type="password" placeholder="Enter your password"/>
      <button>Submit</button>
    </form>
  )
}
```

EXAMPLE OF SYNTHETIC EVENT



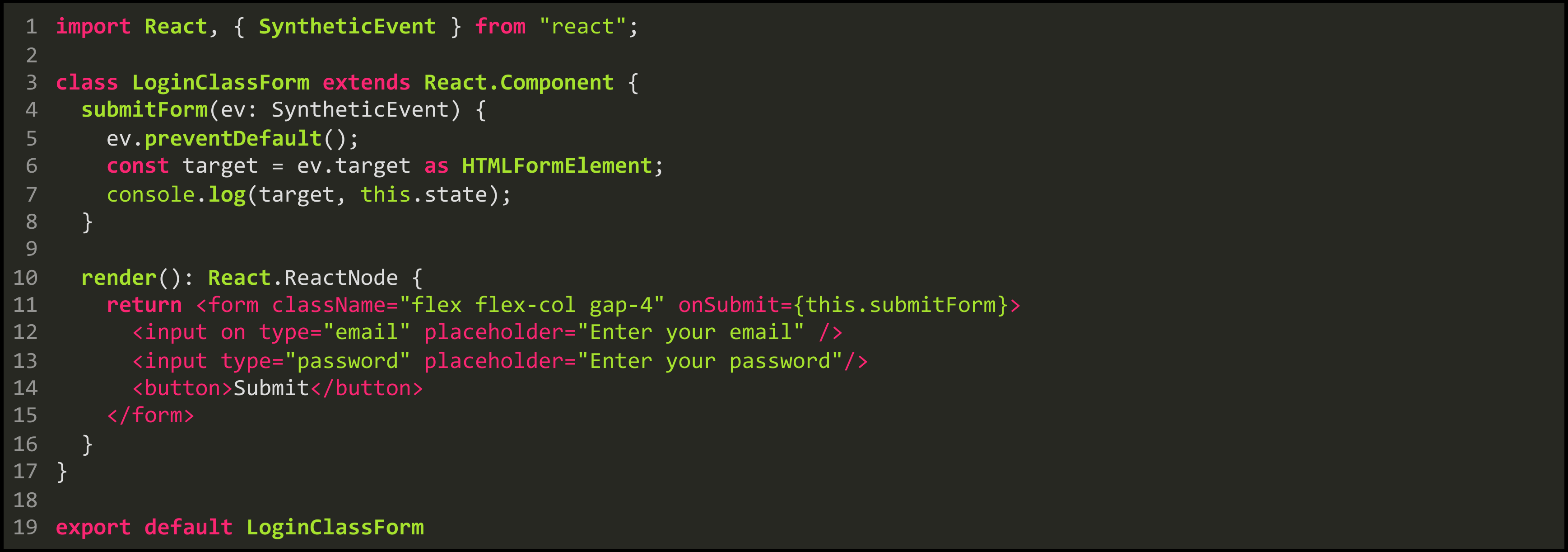
```
import { SyntheticEvent } from "react";

export const LoginForm = () => {
  const [email, setEmail] = useState('');
  const [password, setPassword] = useState('');
  const submitForm = (ev: SyntheticEvent) => {
    ev.preventDefault();
    const target = ev.target as HTMLFormElement;
    console.log(target, {
      email,
      password
    });
  }
  return (
    <form className="flex flex-col gap-4" onSubmit={submitForm}>
      <input onChange={(ev) => {
        setEmail(ev.target.value);
      }} value={email} type="email" placeholder="Enter your email" />
      <input onChange={(ev) => {
        setPassword(ev.target.value);
      }} value={password} type="password" placeholder="Enter your password"/>
      <button>Submit</button>
    </form>
  )
}
```

ONE SMALL PITFALL WHEN BINDING EVENTS

When we want to use `this` inside our event handlers, for example, in case of Class components, we have to bind the event in a different way

EVENT BINDING IN CLASS COMPONENTS



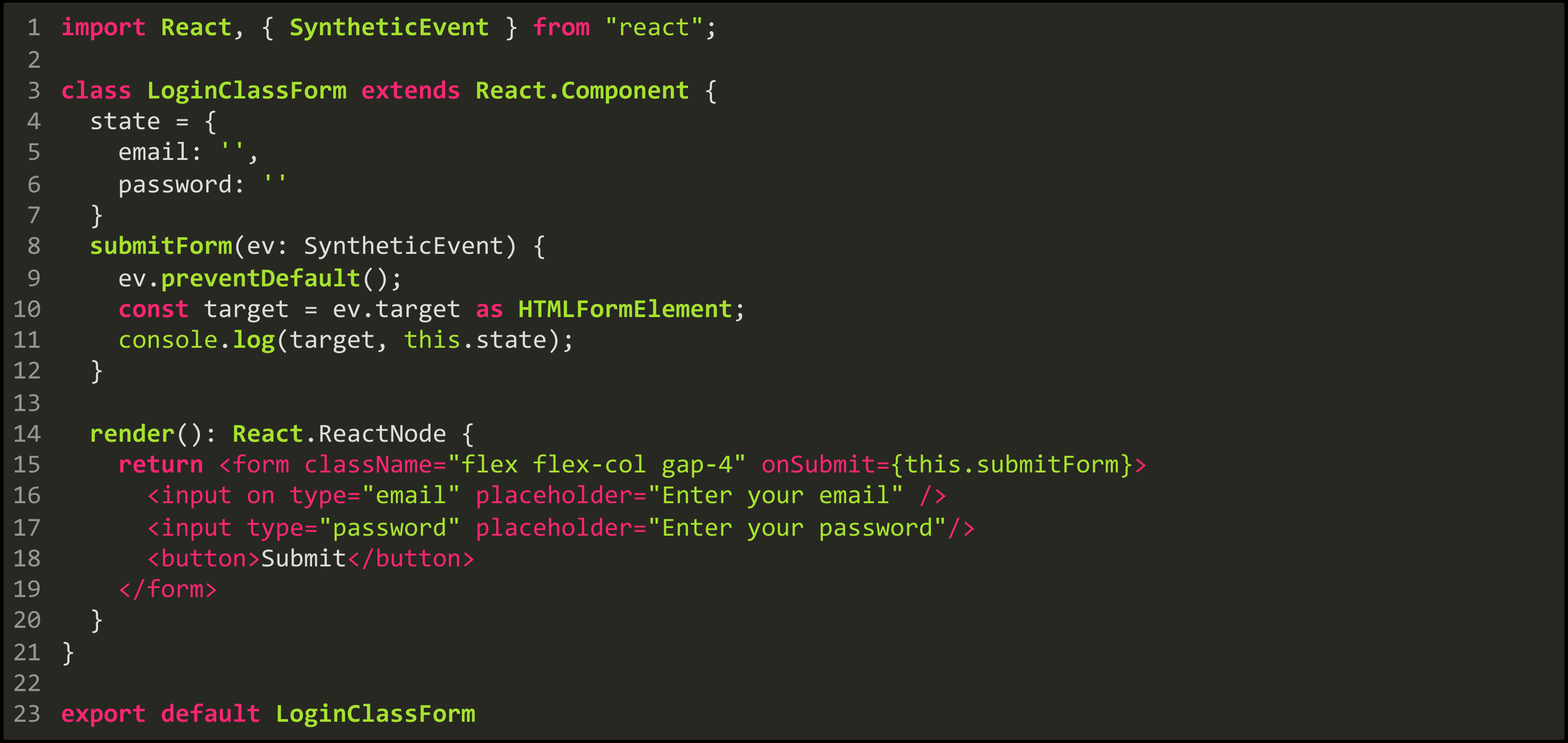
```
import React, { SyntheticEvent } from "react";

class LoginClassForm extends React.Component {
  submitForm(ev: SyntheticEvent) {
    ev.preventDefault();
    const target = ev.target as HTMLFormElement;
    console.log(target, this.state);
  }

  render(): React.ReactNode {
    return <form className="flex flex-col gap-4" onSubmit={this.submitForm}>
      <input on type="email" placeholder="Enter your email" />
      <input type="password" placeholder="Enter your password"/>
      <button>Submit</button>
    </form>
  }
}

export default LoginClassForm
```

EVENT BINDING IN CLASS COMPONENTS



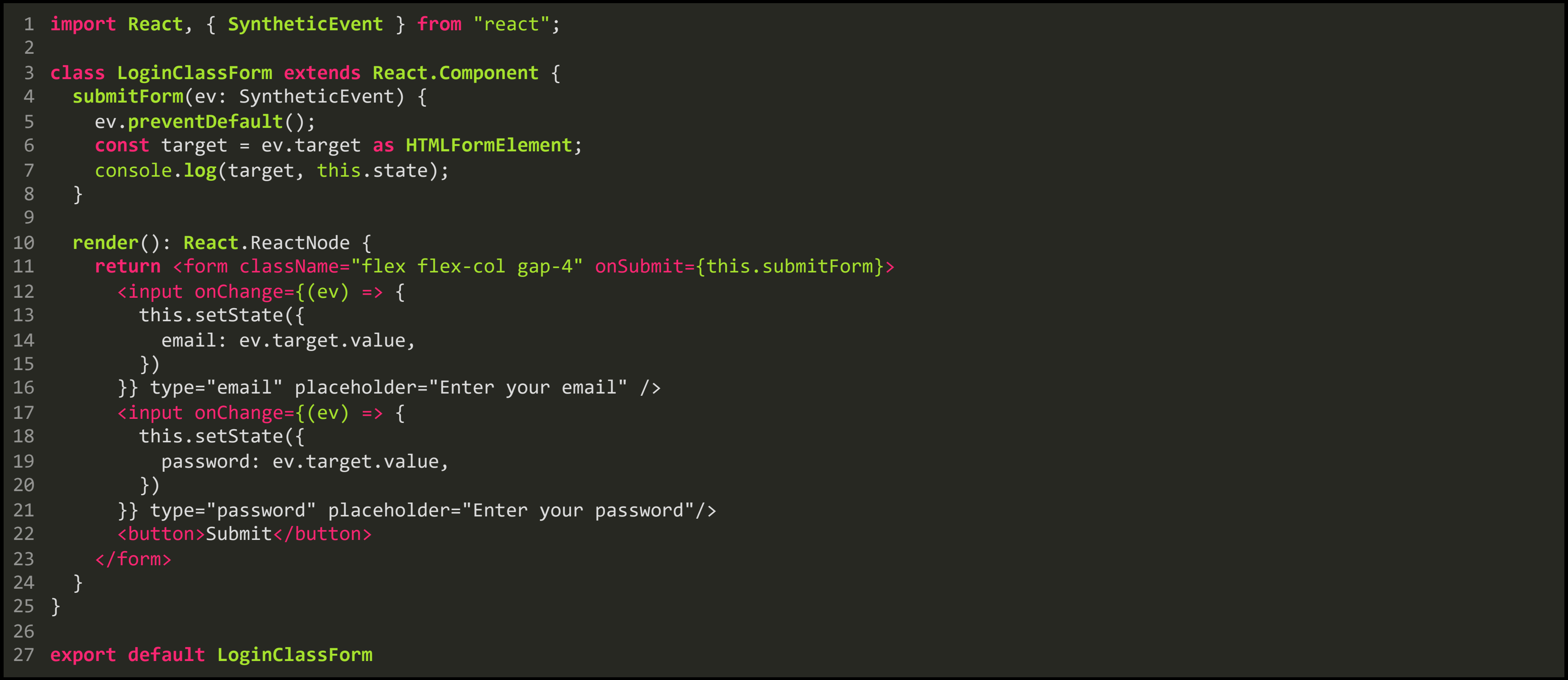
```tsx
import React, { SyntheticEvent } from "react";

class LoginClassForm extends React.Component {
  state = {
    email: '',
    password: ''
  }
  submitForm(ev: SyntheticEvent) {
    ev.preventDefault();
    const target = ev.target as HTMLFormElement;
    console.log(target, this.state);
  }

  render(): React.ReactNode {
    return <form className="flex flex-col gap-4" onSubmit={this.submitForm}>
      <input on type="email" placeholder="Enter your email" />
      <input type="password" placeholder="Enter your password"/>
      <button>Submit</button>
    </form>
  }
}

export default LoginClassForm
```

EVENT BINDING IN CLASS COMPONENTS



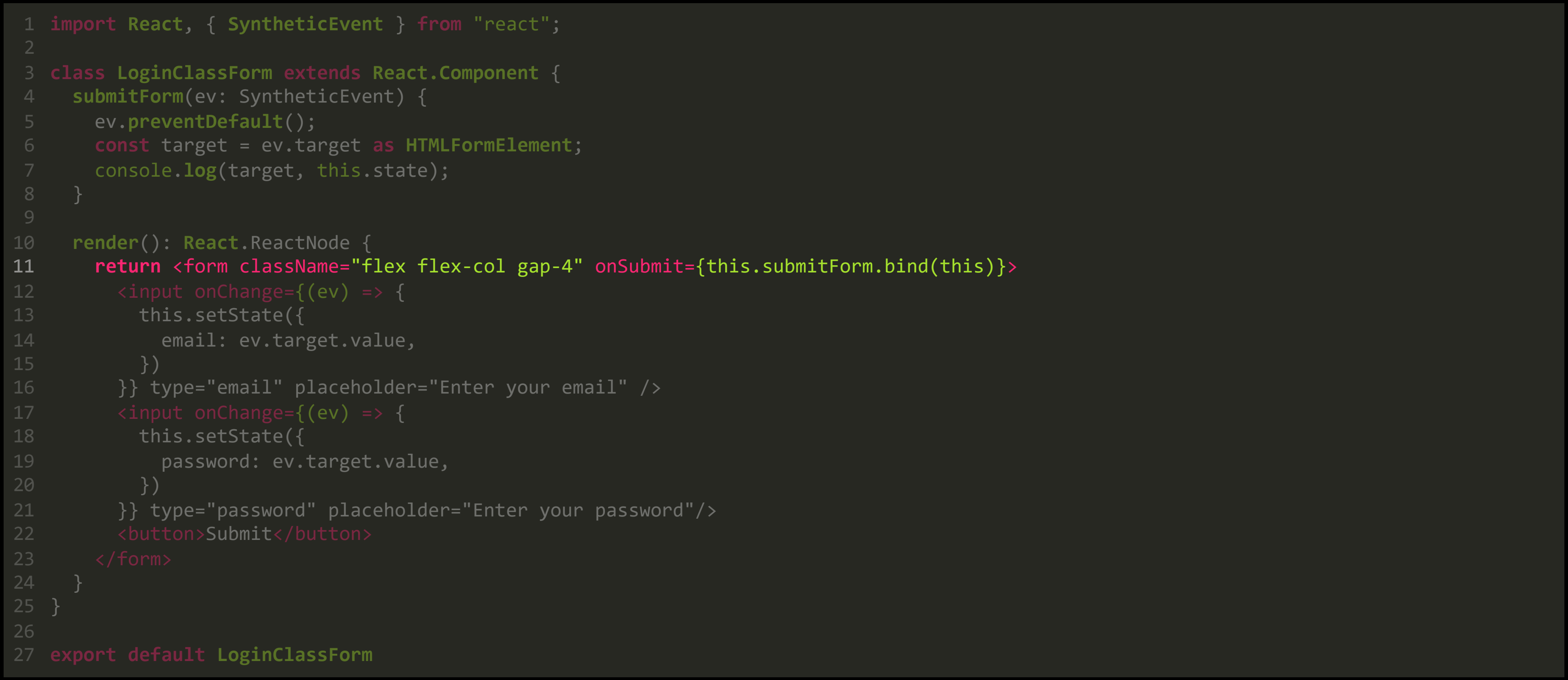
```
import React, { SyntheticEvent } from "react";

class LoginClassForm extends React.Component {
  submitForm(ev: SyntheticEvent) {
    ev.preventDefault();
    const target = ev.target as HTMLFormElement;
    console.log(target, this.state);
  }

  render(): React.ReactNode {
    return <form className="flex flex-col gap-4" onSubmit={this.submitForm}>
      <input onChange={(ev) => {
        this.setState({
          email: ev.target.value,
        })
      }} type="email" placeholder="Enter your email" />
      <input onChange={(ev) => {
        this.setState({
          password: ev.target.value,
        })
      }} type="password" placeholder="Enter your password"/>
      <button>Submit</button>
    </form>
  }
}

export default LoginClassForm
```

EVENT BINDING IN CLASS COMPONENTS



```tsx
import React, { SyntheticEvent } from "react";

class LoginClassForm extends React.Component {
  submitForm(ev: SyntheticEvent) {
    ev.preventDefault();
    const target = ev.target as HTMLFormElement;
    console.log(target, this.state);
  }

  render(): React.ReactNode {
    return <form className="flex flex-col gap-4" onSubmit={this.submitForm.bind(this)}>
      <input onChange={(ev) => {
        this.setState({
          email: ev.target.value,
        })
      }} type="email" placeholder="Enter your email" />
      <input onChange={(ev) => {
        this.setState({
          password: ev.target.value,
        })
      }} type="password" placeholder="Enter your password"/>
      <button>Submit</button>
    </form>
  }
}

export default LoginClassForm
```

PASSING PARAMETERS TO EVENT HANDLERS

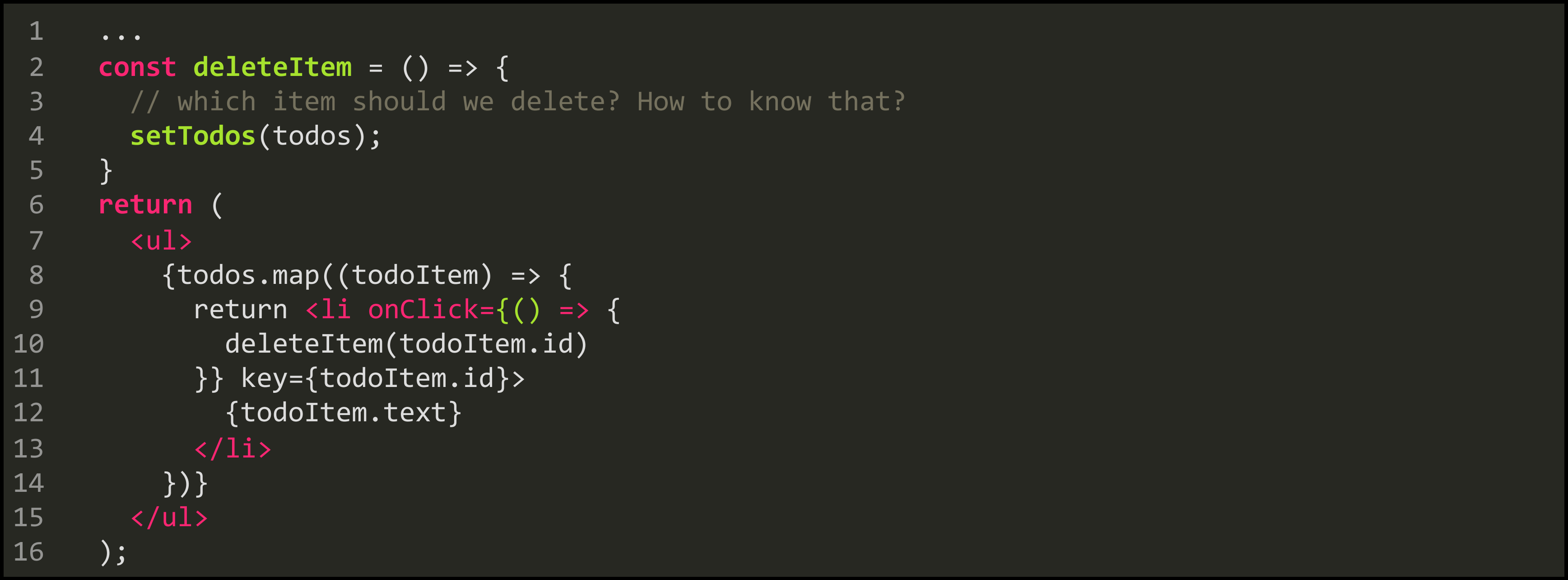
PASSING PARAMETERS TO EVENT HANDLERS

Sometimes we want to also pass parameters to our handler functions, how do we do that?

EXAMPLE: DELETE AN ITEM

```
...
const deleteItem = () => {
  // which item should we delete? How to know that?
  setTodos(todos);
}
return (
  <ul>
    {todos.map((todoItem) => {
      return <li onClick={deleteItem} key={todoItem.id}>
        {todoItem.text}
      </li>
    })}
  </ul>
);
```

EXAMPLE: DELETE AN ITEM



```
...
const deleteItem = () => {
  // which item should we delete? How to know that?
  setTodos(todos);
}
return (
  <ul>
    {todos.map((todoItem) => {
      return <li onClick={() => {
        deleteItem(todoItem.id)
      }} key={todoItem.id}>
        {todoItem.text}
      </li>
    })}
  </ul>
);
```

EXAMPLE: DELETE AN ITEM

```
...
const deleteItem = (itemId: number) => {
  // which item should we delete? How to know that?
  setTodos(todos.filter(itemEl => {
    return itemEl.id !== itemId
  }));
}
return (
  <ul>
    {todos.map((todoItem) => {
      return <li onClick={() => {
        deleteItem(todoItem.id)
      }} key={todoItem.id}>
        {todoItem.text}
      </li>
    })}
  </ul>
);
```