

✓ Congratulations! You passed!

[Go to next item](#)Grade received **100%** To pass 80% or higher

1. When setting a new value for the password state, that is represented by an object with two properties, `value` and `isTouched`, what's the correct call to the `setPassword` state setter inside the `onChange` event handler? Select all that apply.

1 / 1 point

- ☒ `setPassword({ ...password, value: e.target.value });`
- ☐ `setPassword({ isTouched: false, value: e.target.value });`
- ☐ `setPassword({ value: e.target.value });`

✓ Correct

That's correct, `isTouched` is preserved, since it's been merged at the beginning in the state setter, before setting the new value.

2. What's the correct event prop you should use to determine when an input has been interacted with at least once? Select all that apply.

1 / 1 point

- ☐ `onFocus`
- ☒ `onBlur`
- ☐ `onChange`

✓ Correct

That's correct, `onBlur` fires when the user leaves the focus from a particular input and it's the best place to set the interaction state to true and provide actionable feedback in the UI if needed.

3. How do you prevent the default behavior of the `form` HTML tag in React when a submission event occurs?

1 / 1 point

- ☐ By returning `false` from the `onSubmit` function prop that the `form` tag provides.
- ☐ By calling `preventDefault` on the event object inside any `onChange` handler from an `input` tag.
- ☒ By calling `preventDefault` on the event object inside the `onSubmit` function prop from the `form` tag.

✓ Correct

That's correct, that's the proper way to prevent the default behavior and the right place to perform the call.