

REACT



THE GOAL FOR THIS SESSION

1. Understand how *useRef* works
2. Understand how forms work with state (controlled)
3. Understand how forms work without state (uncontrolled)

AGENDA

1. Mini Case: Forms and controlled components
2. `useRef`
3. Forms
 - uncontrolled with `useRef`
 - controlled with `useState`
 - form validation with `useEffect`?

useRef

- <https://reactjs.org/docs/hooks-reference.html#userref>
- <https://beta.reactjs.org/learn/referencing-values-with-refs>
- The returned object will persist for the full lifetime of the component.
- A common use case is to access a child imperatively
- Or to store a value between renders

- If you pass a ref object to React with `<div ref={myRef}/>`, React will set its `.current` property to the corresponding DOM node whenever that node changes.
- We use it when we really need to go into the DOM
- Like for forms

useRef vs useState

useRef

Doesn't trigger re-render when you change it.

Mutable—you can modify and update current's value

useState

Triggers re-render when you change it.

“Immutable”—you must use the state setting function

```
1 import { useRef } from "react";
2
3 export default function Form() {
4   const form = useRef(null);
5
6   function onSubmit(e) {
7     if (form.current.checkValidity()) {
8       //...
9     }
10  }
11  return (
12    <form onSubmit={onSubmit} ref={form}>
13      <input type="submit" />
14    </form>
15  );
16 }
```



Using **useRef** as a "global variable"

If the component is rerendered, the id is persisted

[Link \(content removed from PDF\)](#)

FORMS

<https://reactjs.org/docs/forms.html>

What? class NameForm extends React.Component?

Working with forms is quite different in React

1. We store the initial value of each field in **state**
2. We set the **value** of each field to it's matching **state**
3. We attach an **onChange** listener to each field
4. Whenever the event fires, we update **state**
5. **onSubmit** we use state as the values (remember, we don't have direct access to the DOM)

```
1 import { useState } from "react";
2
3 export default function MyForm(props) {
4   const [name, setName] = useState("");
5   const [email, setEmail] = useState("");
6
7   const nameChanged = (e) => {
8     setName(e.target.value);
9   };
10  const emailChanged = (e) => {
11    setEmail(e.target.value);
12  };
13  const onSubmit = (e) => {
14    e.preventDefault();
15    postStuff({
16      name: name
```



**FORM
VALIDATION
W
USEEFFECT**

```
1 import { useState, useEffect, useRef } from "react";
2 export default function VanillaForm() {
3   const [name, setName] = useState("");
4   const [cardnumber, setCardnumber] = useState("");
5   const [isValid, setIsValid] = useState(false);
6   const form = useRef(null);
7
8   useEffect(() => {
9     // prettier-ignore
10    const isCreditCardValid =
11      cardnumber.replaceAll(" ", "").length === 16;
12    // prettier-ignore
13    setIsValid(
14      form.current.checkValidity() && isCreditCardValid
15    );
16  }, [name, cardnumber]).
```

+3/4