refs

Refs provide a way to access DOM nodes or React elements created in the render method.

When to Use Refs

- Managing focus, text selection, or media playback.
- Triggering imperative animations.
- Integrating with third-party DOM libraries.

Creating Refs

Refs are created using *React.createRef()* and attached to React elements via the ***ref*** attribute.

Refs are commonly assigned to an instance property when a component is constructed so they can be referenced throughout the component.

```
 // Create a ref to store the DOM element
 this.myRef = React.createRef();


render() {
     // Attaching created ref to react element
    return <div ref={this.myRef} />;
  }
```

Accessing Refs

When a *ref* is passed to an element in render, a reference to the node becomes accessible at the *current* attribute of the *ref*.

```
const node = this.myRef.current;
```

React will assign the current property with the DOM element when the component mounts, and assign it back to null when it unmounts.

The value of the ref differs depending on the type of the node:

- When the *ref* attribute is used on an HTML element, the ref created in the constructor with *React.createRef()* receives the underlying DOM element as its current property.
- When the *ref* attribute is used on a custom class component, the ref object receives the mounted instance of the component as its current.
- You may not use the *ref* attribute on function components because they don’t have instances.

refs

- Adding a Ref to a DOM Element
- Adding a Ref to a Class Component

callback refs

React also supports another way to set refs called “callback refs”, which gives more fine-grain control over when refs are set and unset.

Instead of passing a *ref* attribute created by *createRef()*, you pass a function. The function receives the React component instance or HTML DOM element as its argument, which can be stored and accessed elsewhere.