List & Keys

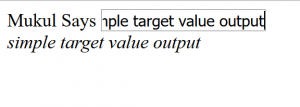
* References

Refs are a function provided by React to access the DOM element and the React element that you might have created on your own. They are used in cases where we want to change the value of a child component, without making use of props and all. They also provide us with good functionality as we can use callbacks with them.

**Example:**

|  |
| --- |
| // without refs  class App extends React.Component {    constructor(){     super();     this.state = { sayings: ""};     }    update(e){     this.setState({ sayings: e.target.value});    }    render(){     return (      <div>       Mukul Says <input type="text" onChange = {this.update.bind(this)}/>      <br/>      <em>{this.state.sayings}</em>     </div>    );   }  }  ReactDOM.render(< App />, document.getElementById('root')); |

**Output:**



* When to use Refs

There are a few good use cases for refs:

* Managing focus, text selection, or media playback.
* Triggering imperative animations.
* Integrating with third-party DOM libraries.
* When not to use Refs
* Should not be used with functional components because they don’t have instances.
* Not to be used on things that can be done declaratively.
* How to Create 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.

class MyComponent extends React.Component {

constructor(props) {

super(props);

this.myRef = React.createRef(); }

render() {

return <div ref={this.myRef} />; }

}

* How to access 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;

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.