

React Setup

Add Script	<code><script src="foo.com/react.js" crossorigin /></code>
Add targeting div	<code><div id="react" /></code>
Get root Dom	<code>root = document.getElementById("root");</code>
Bootstrap React	<code>ReactDOM.render(<App />, rootElement);</code>

React Hooks

```
const [state, setState] = useState(initialState);

useEffect(() => { ...logic, return cleanup }, [deps]);

useMemo(() => computeExpensiveValue(a, b), [a, b]);
```

Context

```
ThemeContext = React.createContext(themes.light);
<ThemeContext.Provider value={themes.dark}>
```

```
theme = useContext(ThemeContext);
background: theme.background
```

useRef

```
const inputEl = useRef(null);
<input ref={inputEl} />
inputEl.current.focus();
```

async useEffect

```
useFetch = (param) => {
  const [state, setState] = useState([]);
  useEffect(() => {
    async function fetchData() {
      const response = await fetch('url');
      const data = await response.json();
      setState((prev) => prev.concat(data.results));
    }
    fetchData();
  }, [param]);
  return { state };
};
```

JS Arrays

Insert at Start	<code>unshift()</code>
Insert at End	<code>push()</code>
Insert at Location	<code>splice(start, 0, item2, itemN)</code>
Remove First	<code>shift()</code>
Remove Last	<code>pop()</code>
Remove at Location	<code>splice(start, deleteCount)</code>
Get parts	<code>slice(start, end)</code>

Algos

DFS	<code>getNodes(root)</code> <code>[...getNodes(root.left), getNodes(root.right)]</code>
BFS	<code>while(queue.length)</code> <code>for(queue.length)</code> <code>node = queue.shift()</code> <code>queue.push(node.left/right)</code>

Big-O

