Vanilla JS → React Cheat Sheet

Selecting & Updating the UI

```
Vanilla:
const el = document.getElementById('msg');
el.textContent = 'Hi';

React:
const [msg, setMsg] = useState('');
setMsg('Hi'); // JSX shows {msg}
```

Event Listeners

```
Vanilla:
btn.addEventListener('click', () => doThing());

React:
<button onClick={doThing}>Click</button>
```

Styles & Classes

```
Vanilla:
btn.style.backgroundColor = 'red';
btn.classList.add('active');

React:
<button style={{ backgroundColor: 'red' }} className={`btn ${active ? 'active' : ''}`} />
```

Conditional Rendering

```
Vanilla:
el.style.display = show ? 'block' : 'none';

React:
{show && <Panel />}
```

Loops / Lists

```
Vanilla:
items.forEach(i => container.appendChild(makeRow(i)));

React:
{items.map(i => <Row key={i.id} item={i} />)}
```

Reading/Writing Form Values

```
Vanilla:
const val = input.value;
```

```
input.value = 'next';

React:
const [val, setVal] = useState('');
<input value={val} onChange={e => setVal(e.target.value)} />
```

Disable / Enable

```
Vanilla:
btn.disabled = isDisabled;

React:
<button disabled={isDisabled}>Save</button>
```

Lifecycle / Effects

```
Vanilla:
document.addEventListener('DOMContentLoaded', init);

React:
useEffect(() => { init(); }, []);
```

Keyboard Events

```
Vanilla:
document.addEventListener('keydown', e => { if (e.key === 'r') reset(); });

React:
useEffect(() => {
  const onKey = e => e.key === 'r' && reset();
  window.addEventListener('keydown', onKey);
  return () => window.removeEventListener('keydown', onKey);
}, [reset]);
```

Timers

```
Vanilla:
const id = setInterval(tick, 1000);
clearInterval(id);

React:
useEffect(() => {
  const id = setInterval(tick, 1000);
  return () => clearInterval(id);
}, []);
```

Fetching Data

```
Vanilla:
fetch('/api').then(r => r.json()).then(setUI);
React:
```

```
useEffect(() => {
  (async () => {
    const res = await fetch('/api');
    setData(await res.json());
  })();
}, []);
```

Show/Hide & Toggle

```
Vanilla:
box.classList.toggle('open');

React:
setOpen(prev => !prev);
{open && <Box />}
```

Increment/Computed Updates

```
Vanilla:
count = count + 1;

React:
setCount(c => c + 1);
```

Direct DOM Access

```
Vanilla:
const input = document.getElementById('name');
input.focus();

React:
const ref = useRef(null);
<input ref={ref} />
useEffect(() => ref.current?.focus(), []);
```

Prevent Default

```
Vanilla:
form.addEventListener('submit', e => { e.preventDefault(); save(); });

React:
<form onSubmit={e => { e.preventDefault(); save(); }}>
```

Cleanup on Unmount

```
Vanilla:
// manually remove listeners/timers when needed

React:
useEffect(() => {
  const id = setInterval(tick, 1000);
```

```
  return () => clearInterval(id);
}, []);
```