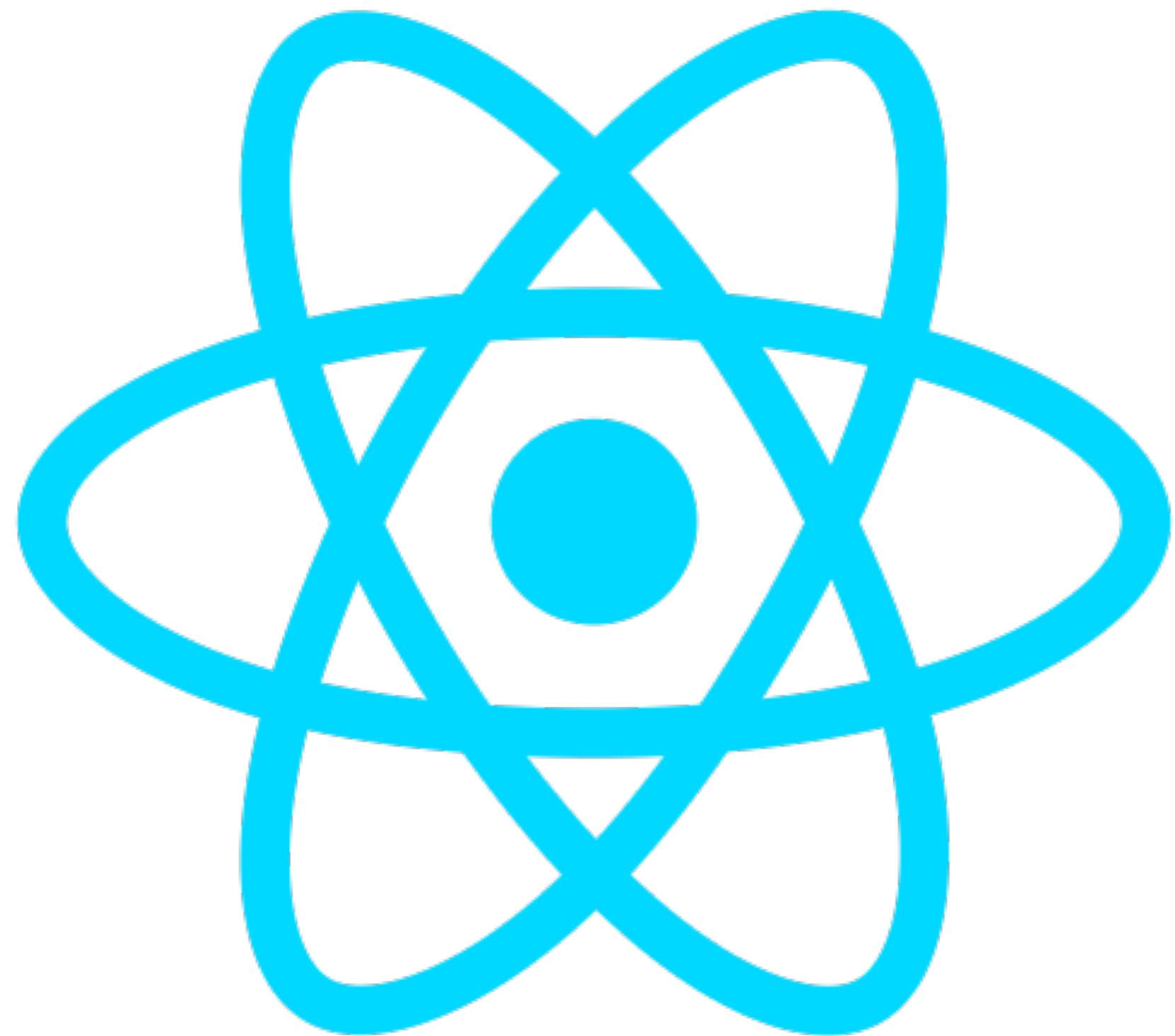


Hooks!

Putting the fun in functional

Why I Love Hooks



- Two different ways to build components always felt like a design flaw
- Excellent way of cross-cutting concerns
- Often eliminate the need for higher-order components and render props
- Closer to the mental model of JavaScript's prototypal inheritance
- Easy to get started, powerful to build on

That Said...

- Classes are not deprecated
- Classes are not inferior in any objective way
- There's no reason to refactor working code if it's stable and easy to reason about.

So... what are hooks, really?

Hooks are just closure
really.

A simple example of closure

Incrementing a private value

```
1  function getCounter(initialValue=0) {
2    let value = initialValue
3
4    function count() {
5      value += 1
6      return value
7    }
8
9    return count
10 }
11
12 const count = getCounter()
13 count()
14 // => 0
15 count()
16 // => 1
17 count()
18 // => 2
19 console.log(value)
20 // => Reference error: value is undefined
```

Two Rules of Hooks

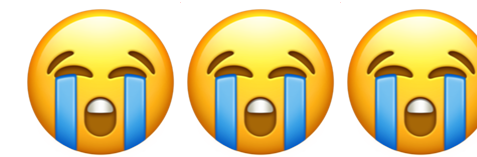
Rule 1:

Only call Hooks at the top level.

Don't call them inside loops, conditions, or nested functions.



```
1 const MyComponent = (props) => {  
2   const [ val, setVal ] = useState(42)  
3  
4   return <p>The Value: {val}</p>  
5 }
```



```
1 const MyComponent = (props) => {  
2   if (props.someCondition) {  
3     useEffect(() => { /* ... */ }, [])  
4   }  
5  
6   return <p>Oh no</p>  
7 }
```


Rule 2:

Only call Hooks from React function components
or your own custom hooks



```
1 const MyComponent = (props) => {  
2   const [ val, setVal ] = useState(42)  
3  
4   return <p>The Value: {val}</p>  
5 }
```



```
1 function someArbitraryFunction(value) {  
2   const [ persisted, setValue ] = useState(value)  
3  
4   // do some arbitrary stuff  
5  
6   return someArbitraryThing  
7 }
```

Writing Custom Hooks

**“A custom Hook is a JavaScript function whose name starts with
‘use’ and that may call other Hooks”**

React Docs: <https://reactjs.org/docs/hooks-custom.html>

[Demo]