

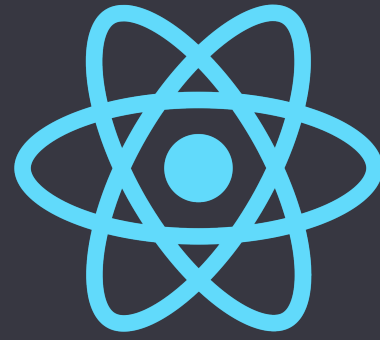
Reactjs

JS LIBRARY FOR BUILDING USER INTERFACES



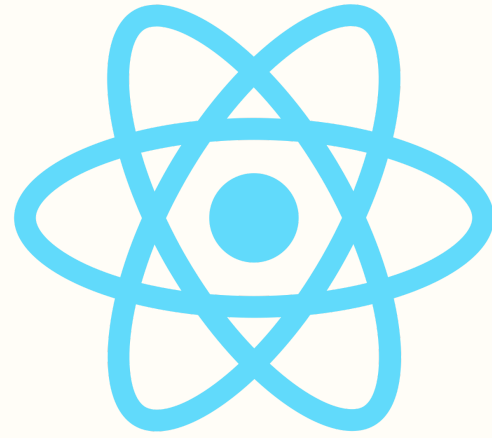
Akash Saxena

CO FOUNDER AND CTO, GETSETGO FITNESS



**Cool! But what
exactly will we be
learning in this
training?**

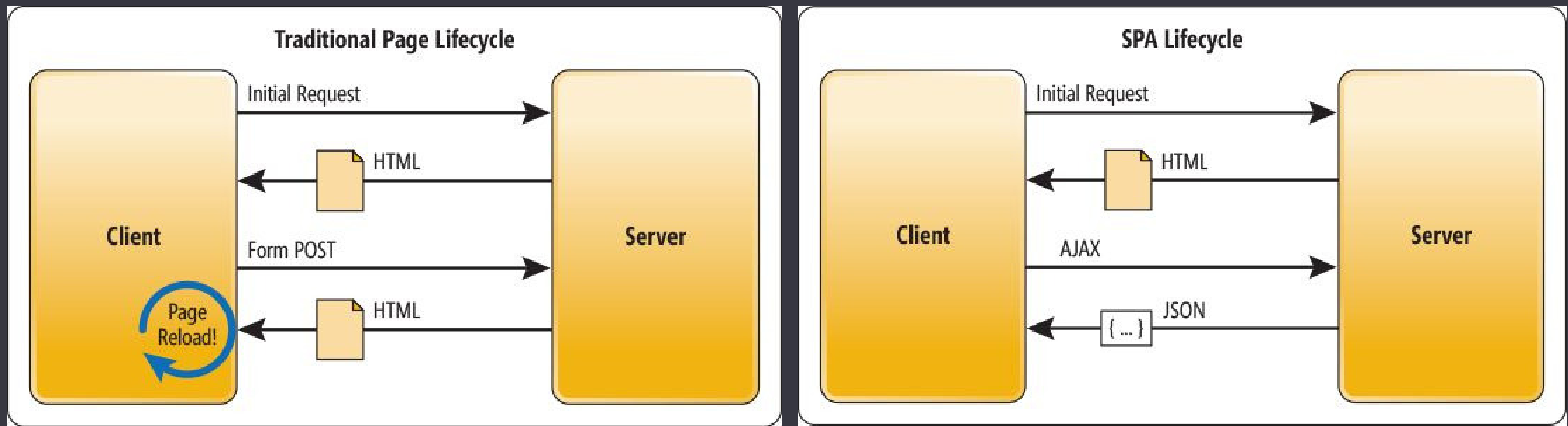
LET'S LOOK AT THE COURSE OUTLINE



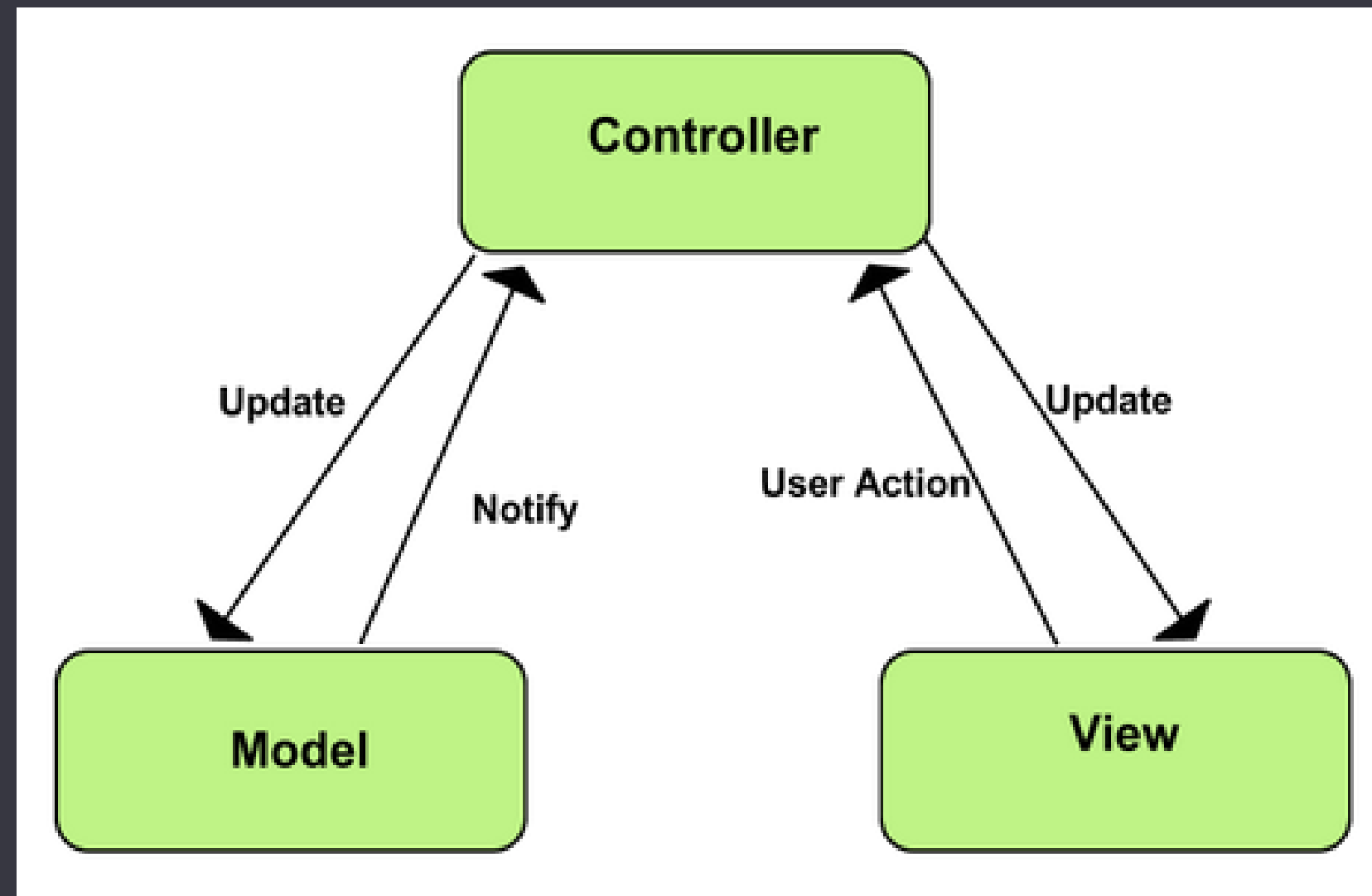
What is React?

ReactJS basically is an open-source JavaScript library which is used for building user interfaces specifically for single page applications.

Oh, SPA, ar'rite. What's that?



Sweet! But how do we implement SPA and what's MV?

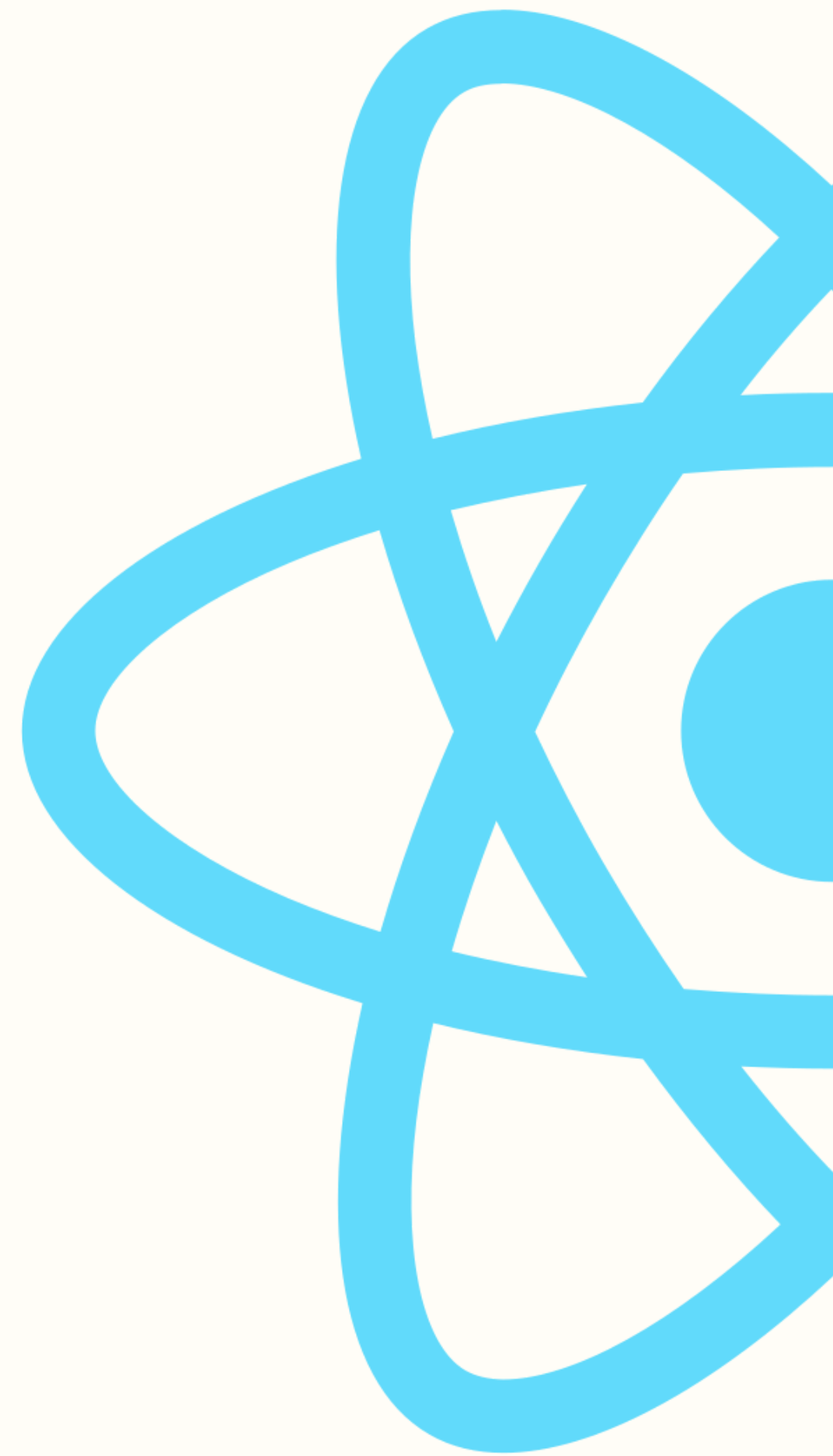


**That sounds way
too cool! How does
it really work?**

OBSERVER PATTERN

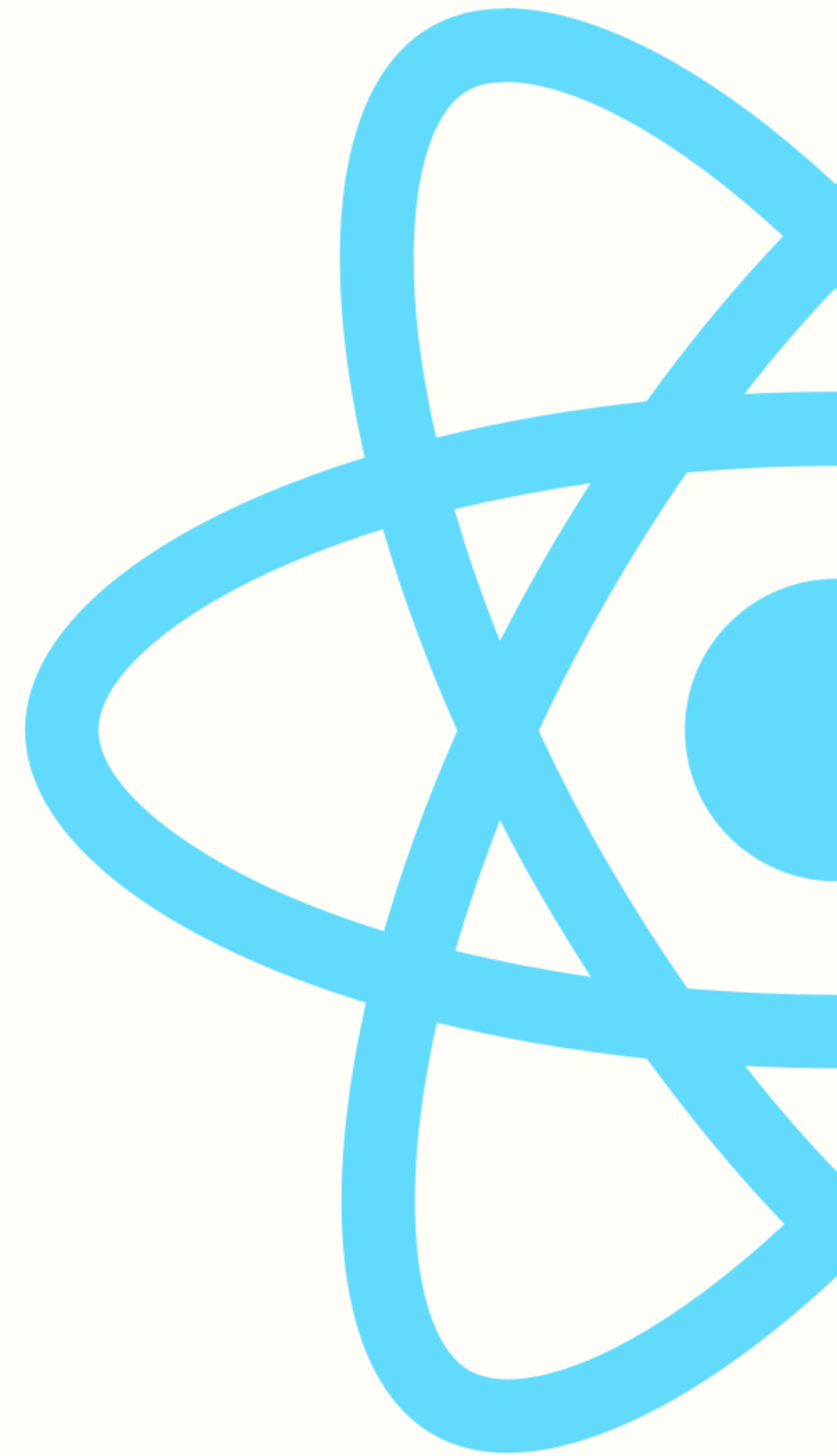
We get it, but why react?

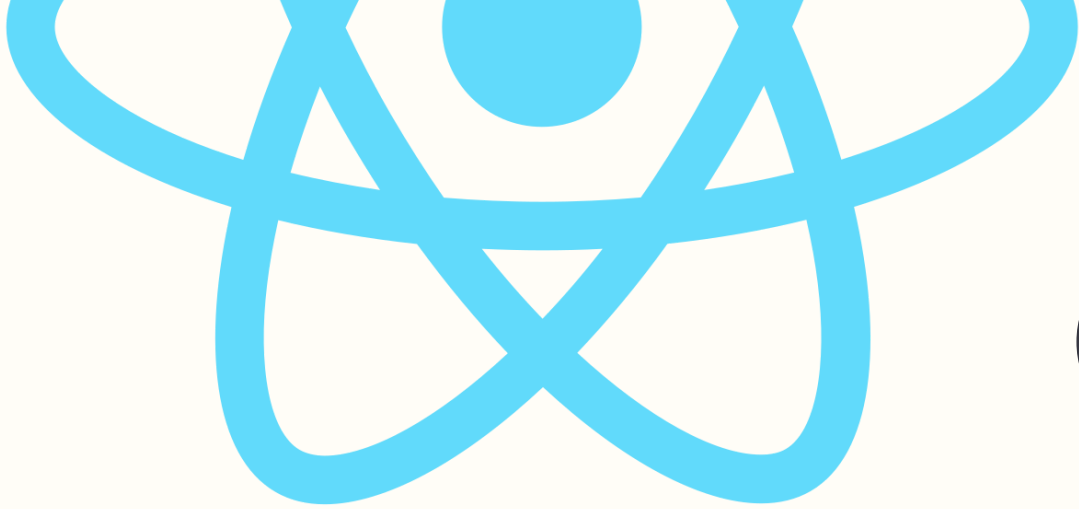
Good question! Let's talk
Angular, JQuery, Backbone and
React



Love react because ...

- > It is declarative in nature
- > It is component based
- > It does not affect the rest of your technology stack



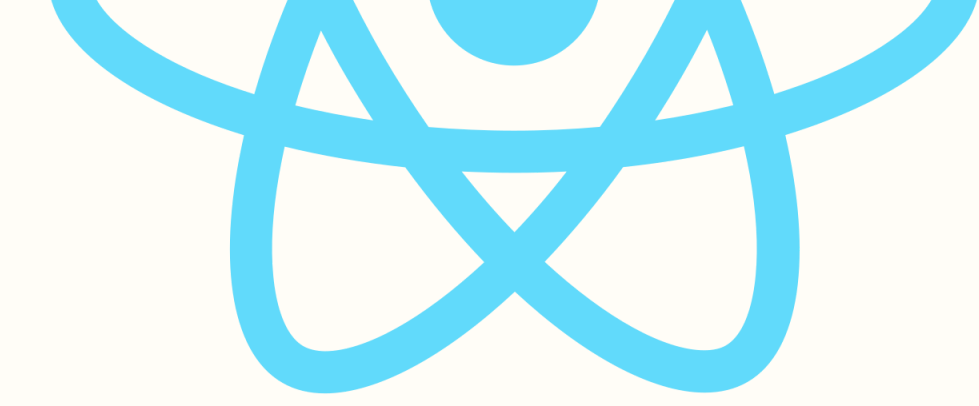


Okay, but can we first look at important JS concepts?

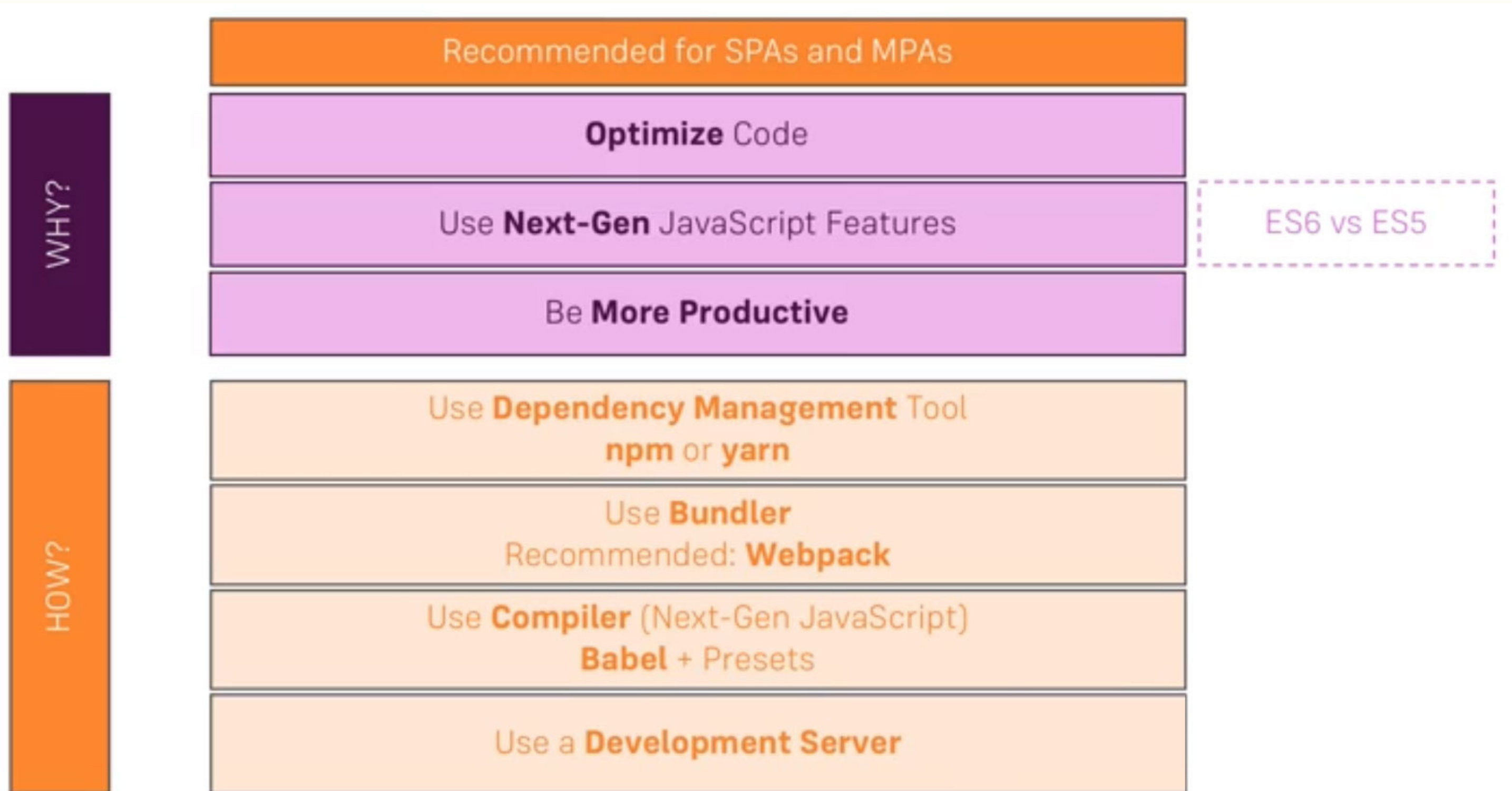
- First Class JS functions
- Object literals
- Prototypical inheritance
- Pass value by reference
- IIFE and potential issue with hoisting
- 'This'
- Writing a custom event emitter to understand observer pattern

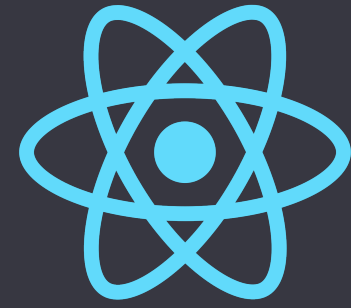
**Damn! That was a
lot of learning. But
doesn't react
prefer es6?**

You got me! Let's look at es6



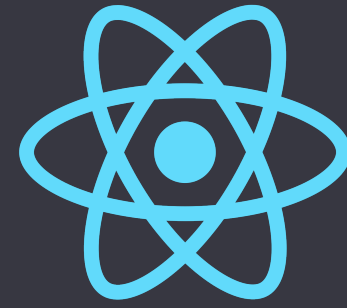
Using a build workflow





**Sounds cool! So,
what are we
building?**

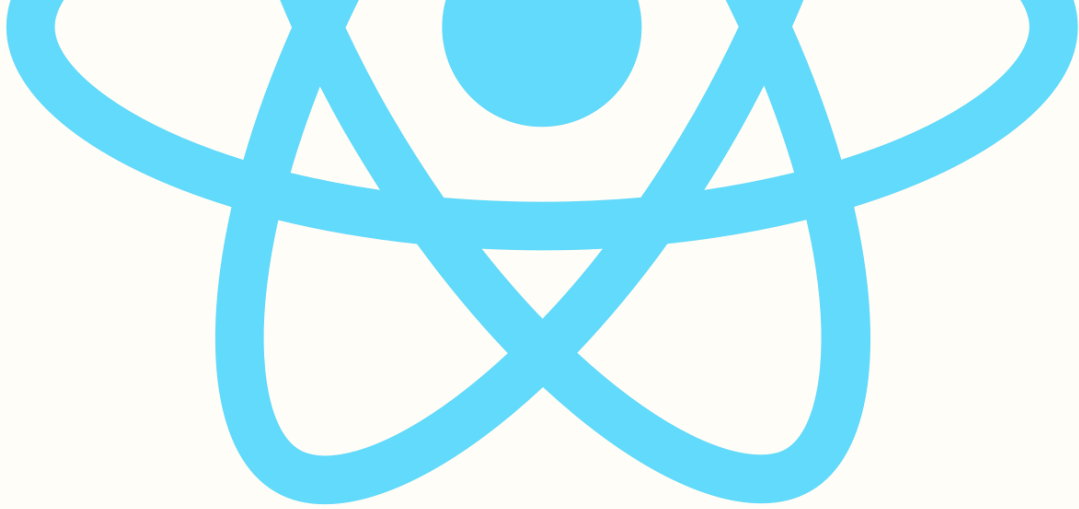
LET'S SEE ...



**Well, well, well. I
can't wait to start.
What next?**

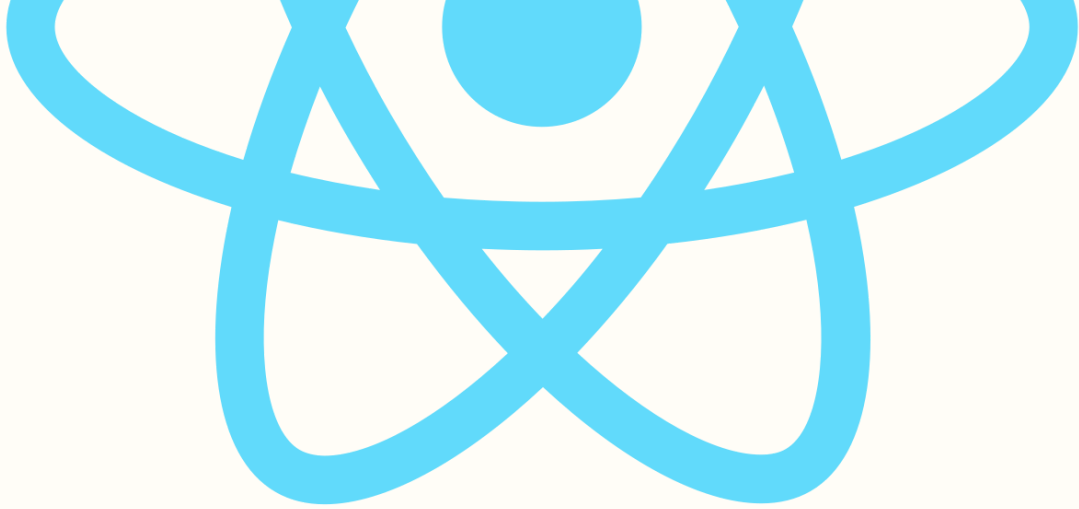
**LET'S LOOK AT AVAILABLE
BOILERPLATES AND CREATE-REACT-APP**

`create-react-app <app name> --scripts-version 1.1.5`



Boilerplate - Check! Let's discuss JSX

- Let's fire up the app and check some cool stuff
- This code looks like a cross between JS and html. Is this JSX? Let's try out a few expressions
- What does it translate to? Let's check out the babel transform
- JSX restrictions: class



Let's re-write a piece of code!

- Why don't we get rid of the code that is already written and try understanding it better?
- ReactDOM vs React
- App vs <App/>
- Render targets

Let's start organizing the code

- Let's look at components
- Let's write a function component
- Why use components?
- Outputting dynamic content
- Working with props
- Understanding children property
- Understanding state
- Props and state
- Handling events with methods - different ways for methods
- Manipulating the state
- Using hooks to manage state
- Stateless vs stateful components

Course project

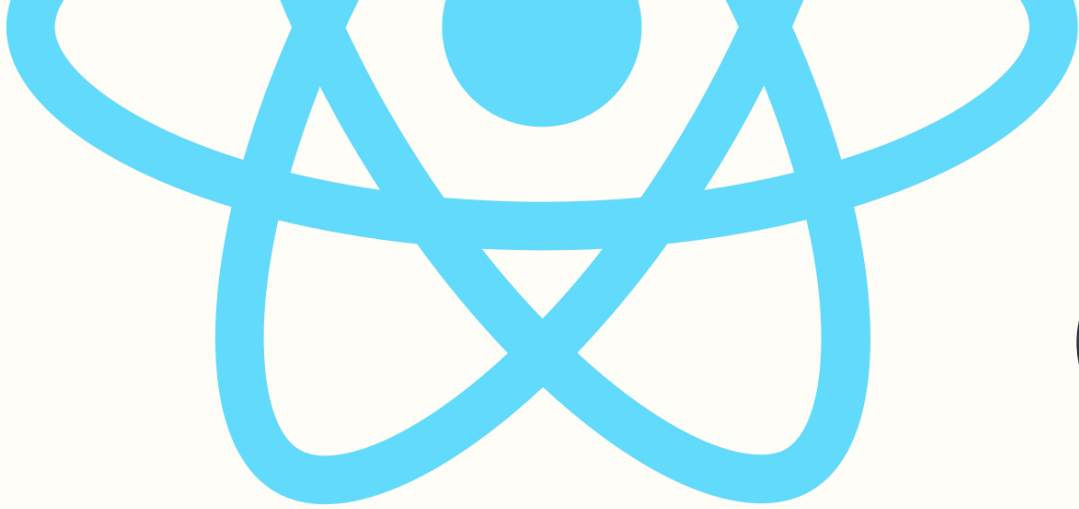
Let's start organizing the code

- Adding two-way binding
- Rendering content conditionally
- Rendering conditional content the JS way
- Outputting lists
- Lists and state
- Updating state immutably with spread operator
- Lists and keys
- Setting styles dynamically
- Setting class dynamically
- Using radium as HOC

Course project

Debugging a react app

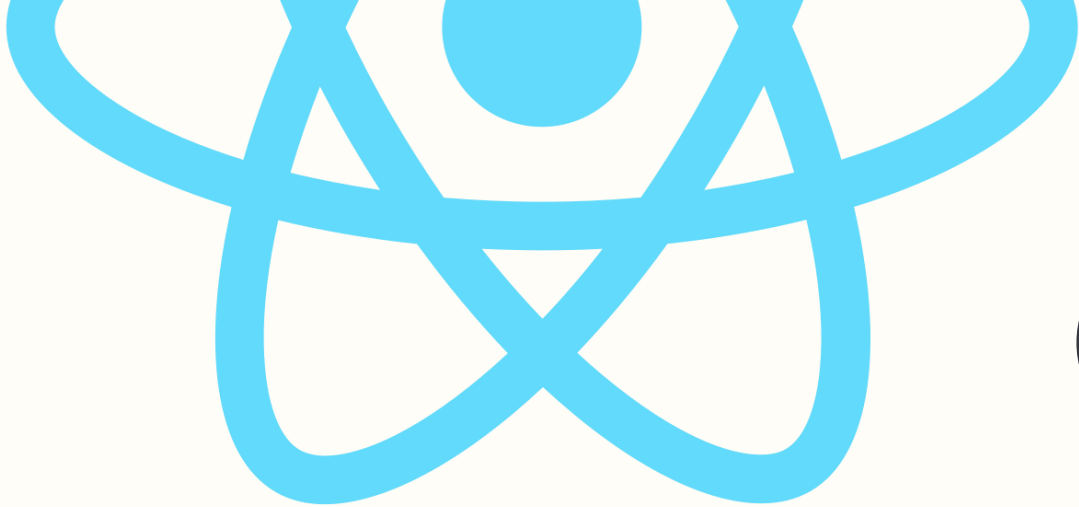
- Understanding error messages
- Finding logical errors using dev tools
- React developer tools
- Using error boundaries



Component lifecycle

Creation

- `constructor()`
- `componentWillMount()`
- `componentDidMount()`
- `render()`



Creation

Default es6 class feature

Constructor

Call super(props)

Set up state

Do not cause side effects

Not used too often

ComponentWillMount

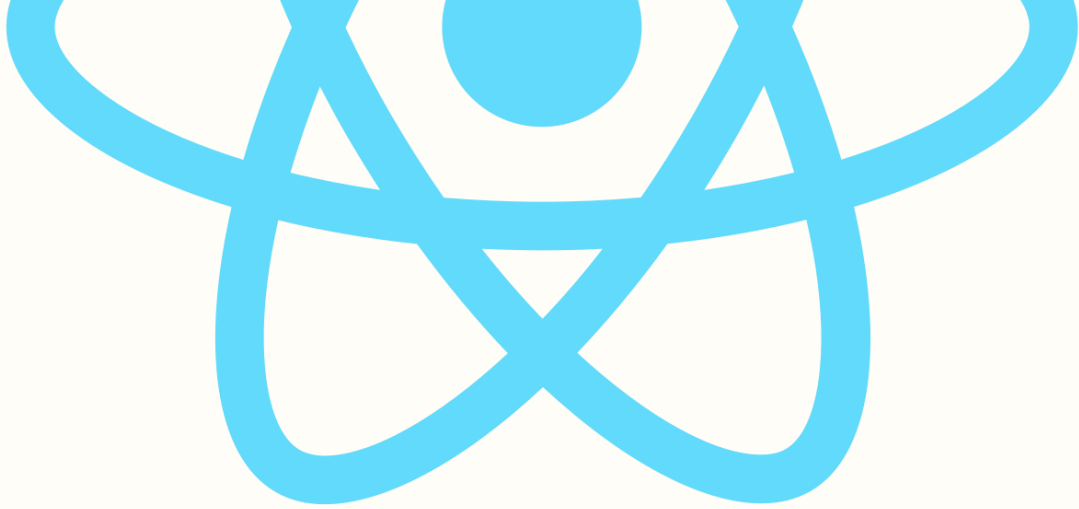
Render

Prepare and structure your JSX
code

Render child component

ComponentDidMount

cause side effects
Do not update state



Update

`componentWillReceiveProps (nextProps)`

Sync state to props

May cancel updating
process

`shouldComponentUpdate (nextProps, nextState)`

Decide to continue or not

`componentWillUpdate (nextProps, nextState)`

Sync state to props

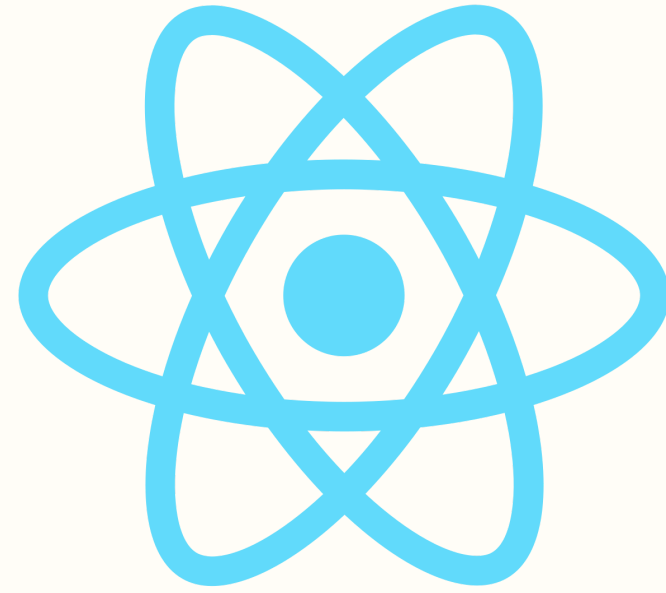
Render

Prepare and structure your JSX
code

Update child component

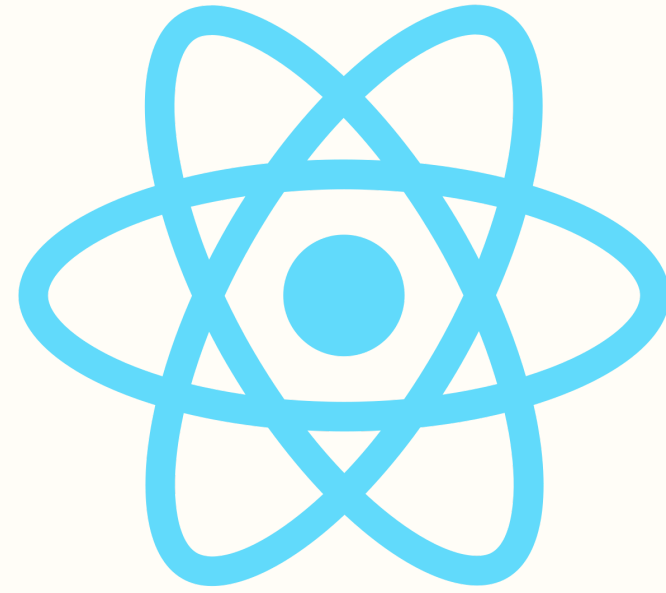
`ComponentDidUpdate`

cause side effects
Do not update state



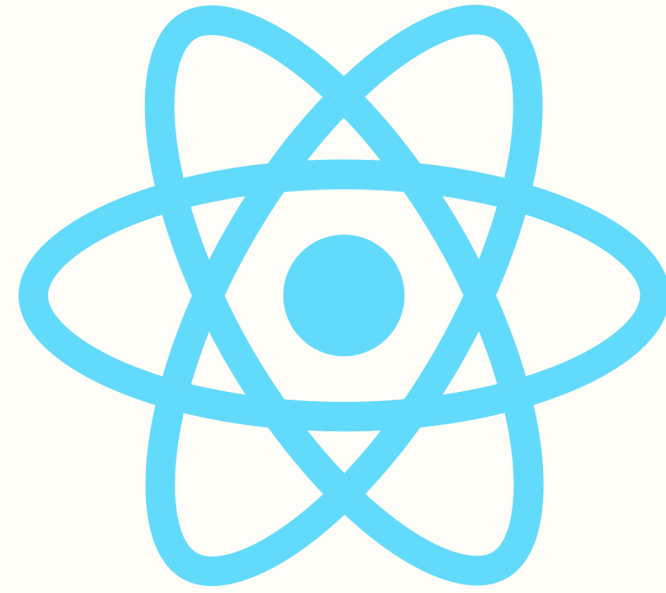
Understanding

React's DOM updation strategy



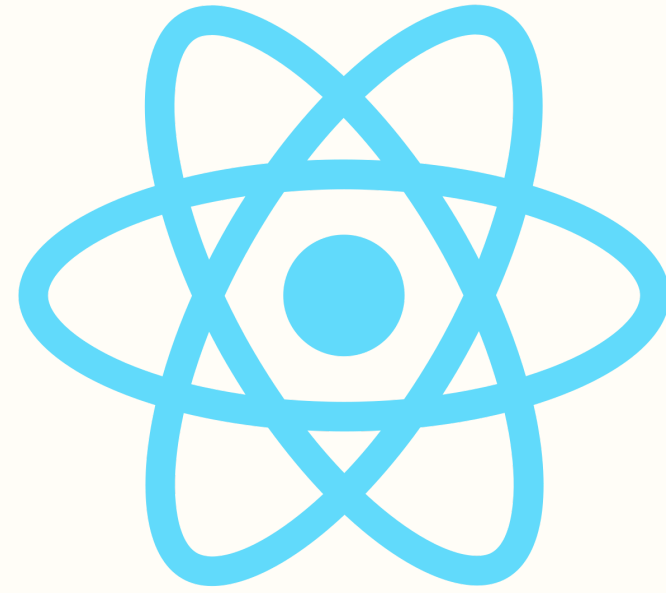
Let's look at

Higher Order Component



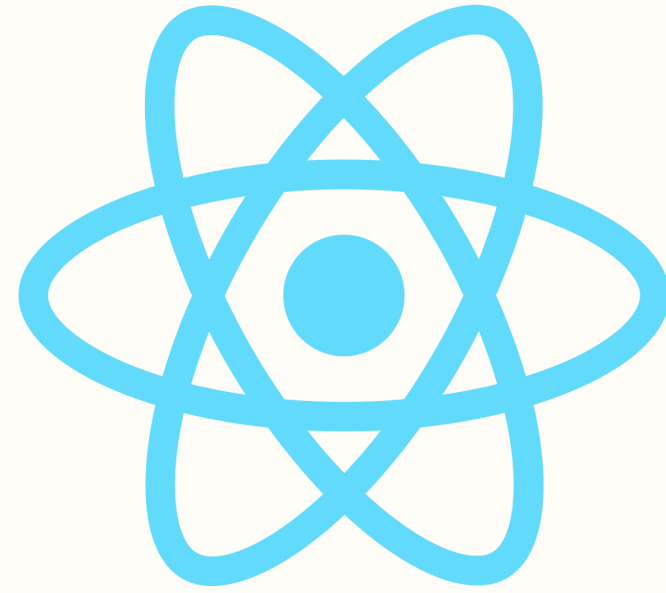
Ref - the tiny task handler

Let's look at an example



React context

`react.createContext` => multi level context
passing



componentWillMount - Not good

Let's try - static
`getDerivedStateFromProps(nextProps,
prevState)`

Multiple Pages in a SPA?

Single Page
(HTML File)

/

/blog

/blog/post/1

/users

/account

/orders

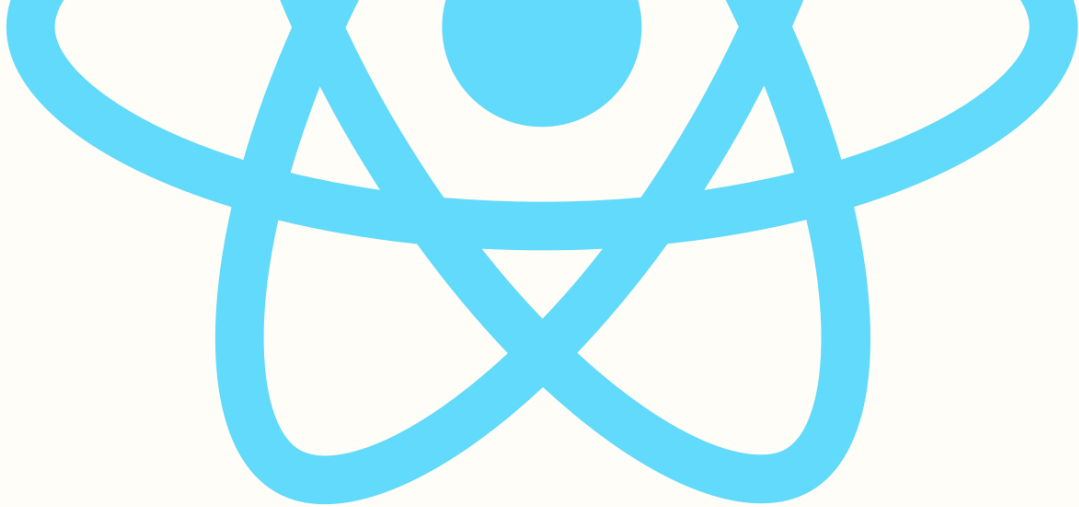
Not real Files but re-rendered Single Page!

Router Package

Parses URL / Path

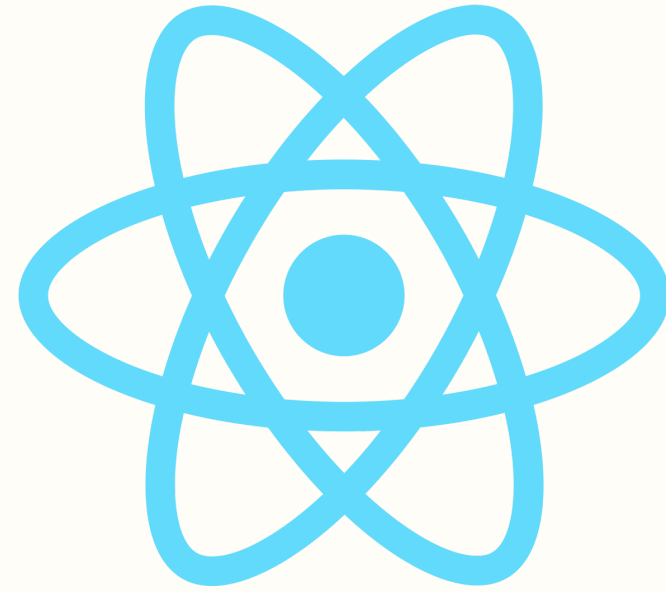
Read Config

Render / Load
appropriate JSX /
Component



React router 4

- Routing with render
- Routing with components
- Switching between pages
- Using link instead of href
- Props using routes
- HOC - withRouter
- Absolute vs relative route
- Styling active router
 - _NavLink with exact
- Route params
- Using switch to load a single route
- Navigating programmatically
- Nested routing
- Redirect component
- Navigating guard
- Handling 404
- Lazy loading routes



Routing and server