React and Redux: Lesson 2

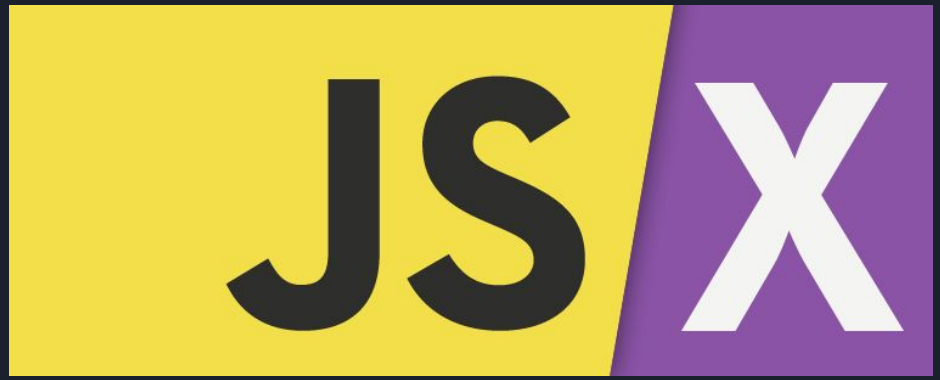
JSX, State, Lifecycle methods

William Abboud

* Covers React v15 - 16

JSX Deep Dive

- XML-like syntax to create UI elements
- Compiles down to React.createElement() functions
- Needs React in scope
- Capitalize custom components
- Pass JS expressions in props
- Pass strings into props directly
- Booleans attributes default to true
- Spread attributes
- Children
- Comments



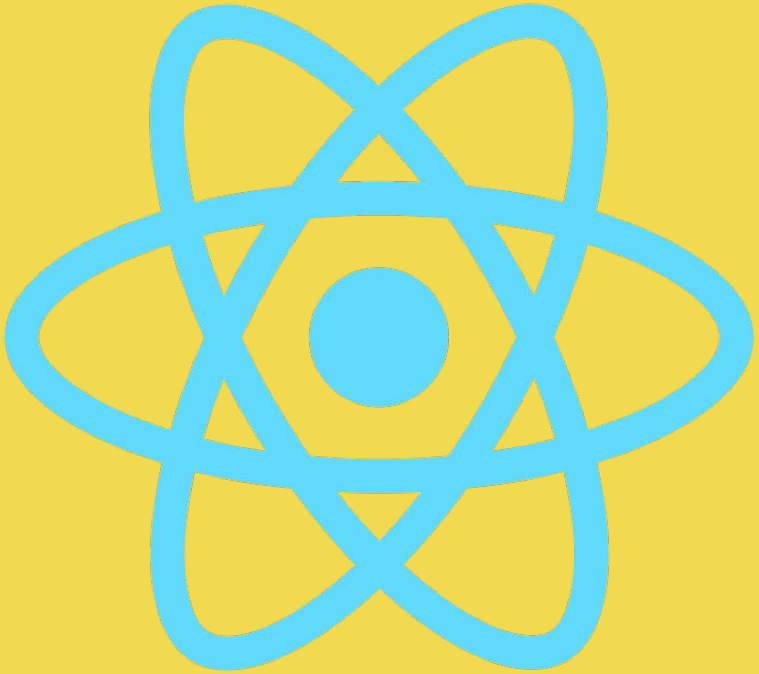
Displaying JSON data

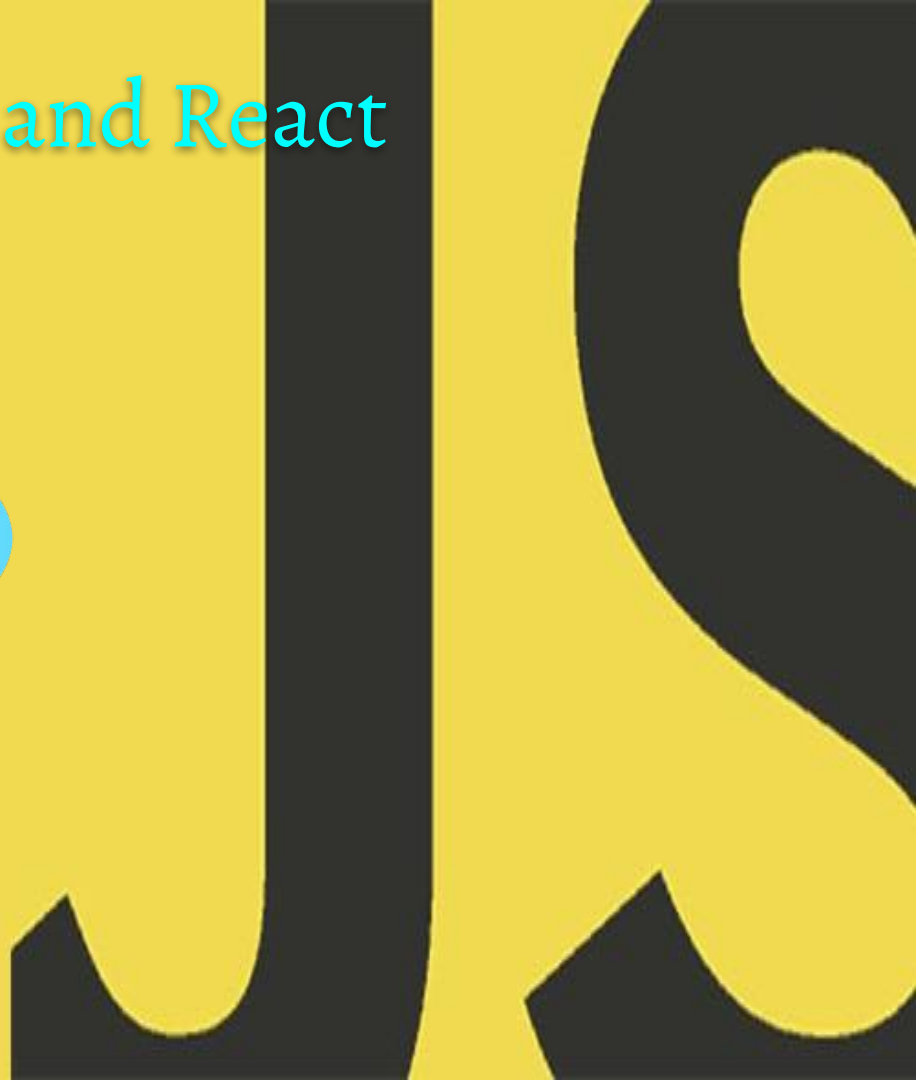
```json
[
    {
        "artist": "Eminem",
        "real-name": "Marshall Bruce Mathers III",
        "aliases": ["Ken Kaniff", "Marshall Mathers", "Slim Shady"],
        "albums": [
            {
                "name": "Infinite",
                "year": "1996"
            },
            {
                "name": "The Slim Shady LP",
                "year": "1999"
            },
            {
                "name": "The Marshall Mathers LP",
                "year": "2000"
            },
            {
                "name": "The Eminem Show",
                "year": "2002"
            },
            {
                "name": "Encore",
                "year": "2004"
            }
        ]
    }
]
```

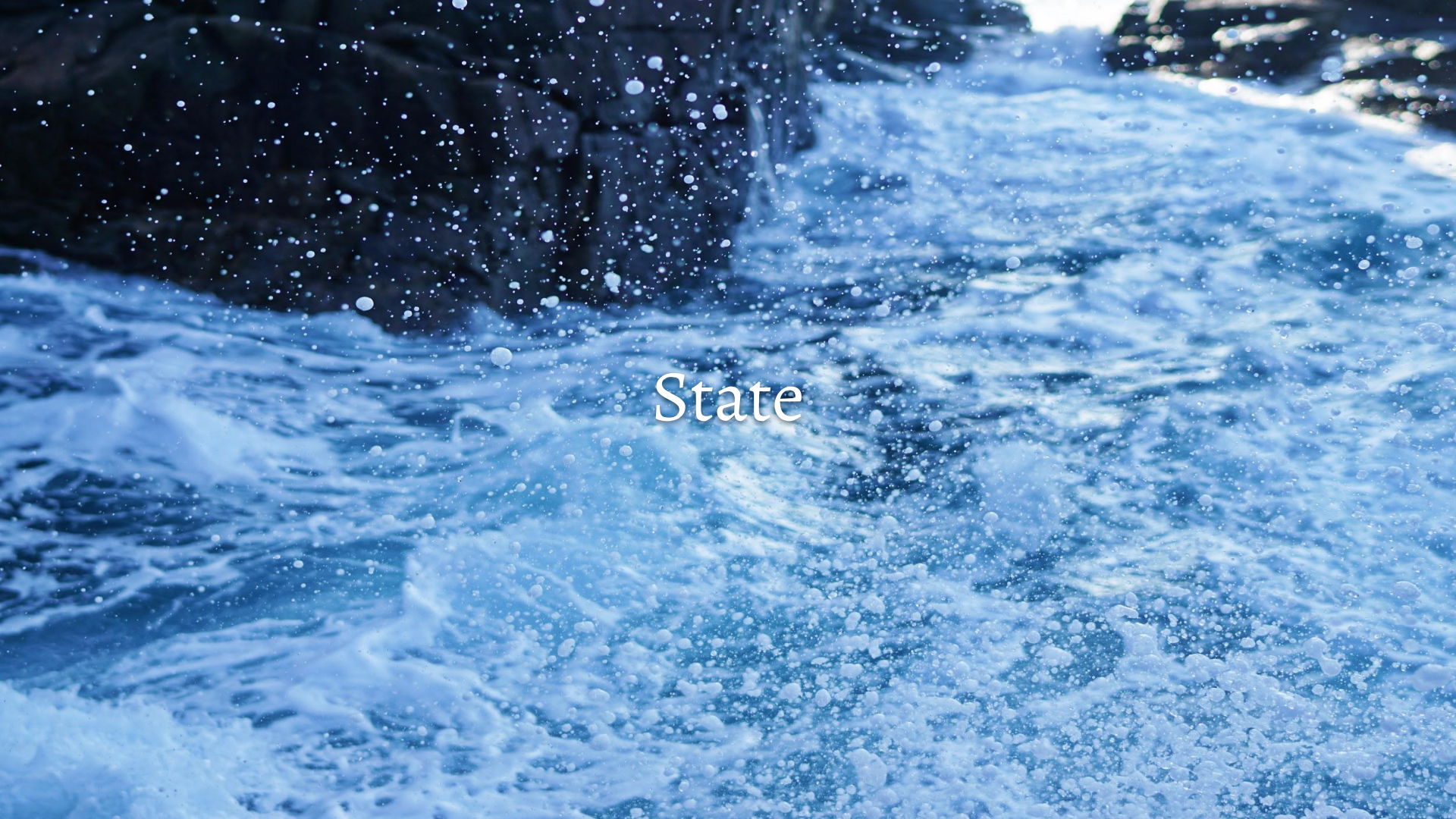
```jsx
import React from 'react';

export default function ArtistCard(props) {
    return (
        <div>
            <h2>{ props.artist }</h2>
            <img src={ props.photo } />
            <br />
            { props.aliases.map(alias => <span>{alias}</span>) }
            <br />
            { props.albums.map(album => <span>{album.name}</span>) }
        </div>
    );
}
```

ES6 Classes and React






State

PropTypes

- Used for documentation
- Evaluates props runtime
- Can remove the need to use typeof operator
- Can be used as API documentation

https://reactjs.org/docs/typechecking-with-proptypes.html

Composition

- Composition offers the ability to create hierarchies of components
- Increases reusability
- Easy to test
- Easy to isolate bugs or errors

Component Lifecycle

Photo by Sebastian Schädler on Unsplash

Initialization phase

1. GetDefaultProps - called only once to return default props
2. GetInitialState - called only once to return an initial state object
3. ComponentWillMount - called before render, useless
4. Render - returns your markup for rendering, has to be pure
5. ComponentDidMount - called last, most frequently used to do AJAX calls, and interacting with the DOM

Update phase - Props

1. ComponentWillReceiveProps - used to set state based on upcoming props
2. ShouldComponentUpdate - determines renders on props/state update
3. ComponentWillUpdate - used to prepare for update (no setState) almost useless
4. ComponentDidUpdate - basically just like componentDidMount but for after update widely used to change the DOM

Update phase - State

1. ShouldComponentUpdate - determines renders on props/state update
2. ComponentWillUpdate - used to prepare for update (no setState) almost useless
3. ComponentDidUpdate - basically just like componentDidMount but for after update widely used to change the DOM

Unmounting

1. ComponentWillUnmount - called before the component is removed from the DOM, used to clear out any side effects such as leftover timeouts, event listeners etc.