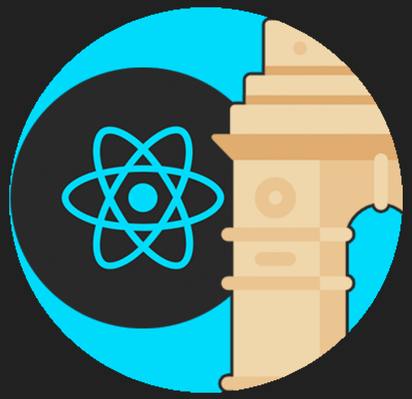
Intro to React Components

Arun Michael Dsouza

http://arunmichaeldsouza.com/



React Delhi/NCR

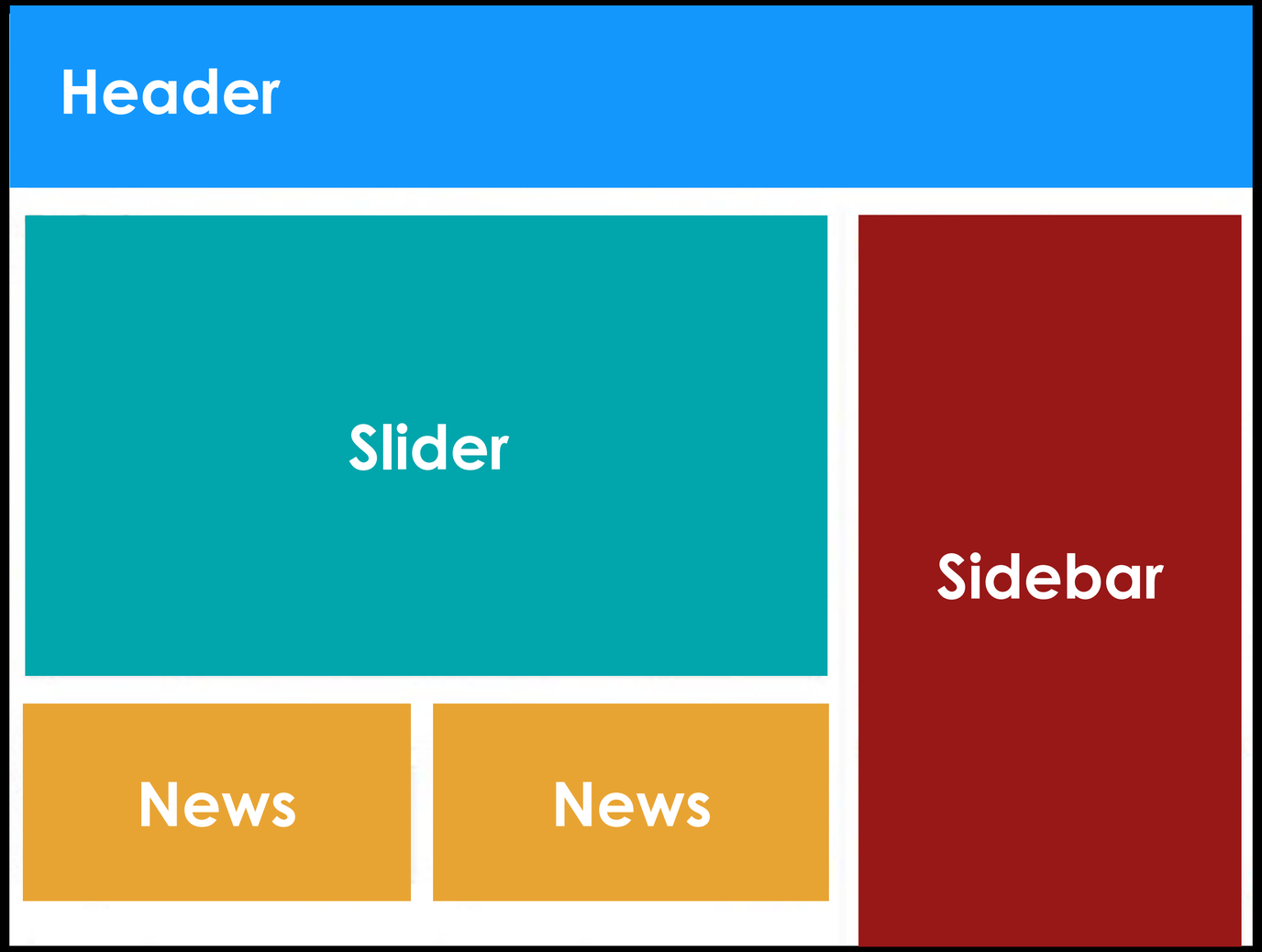
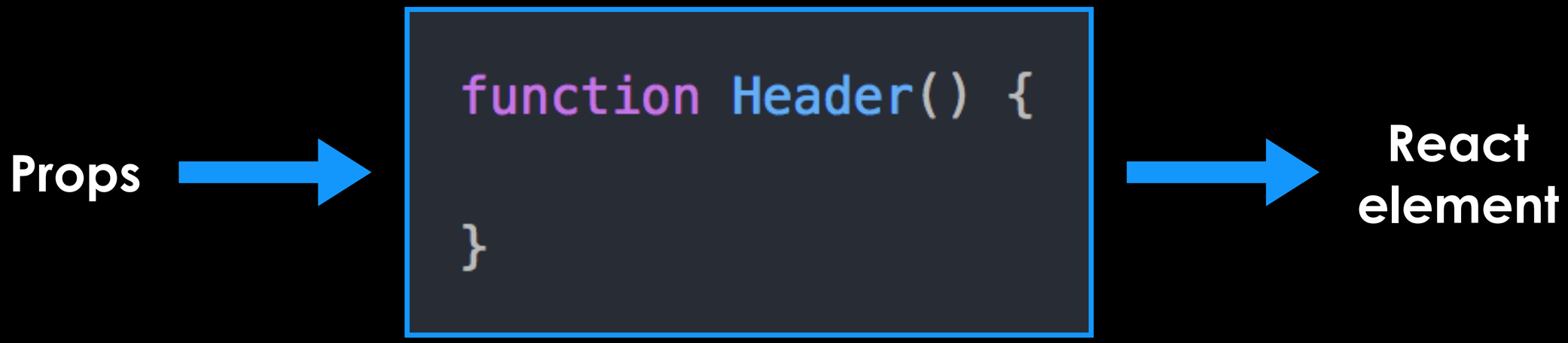
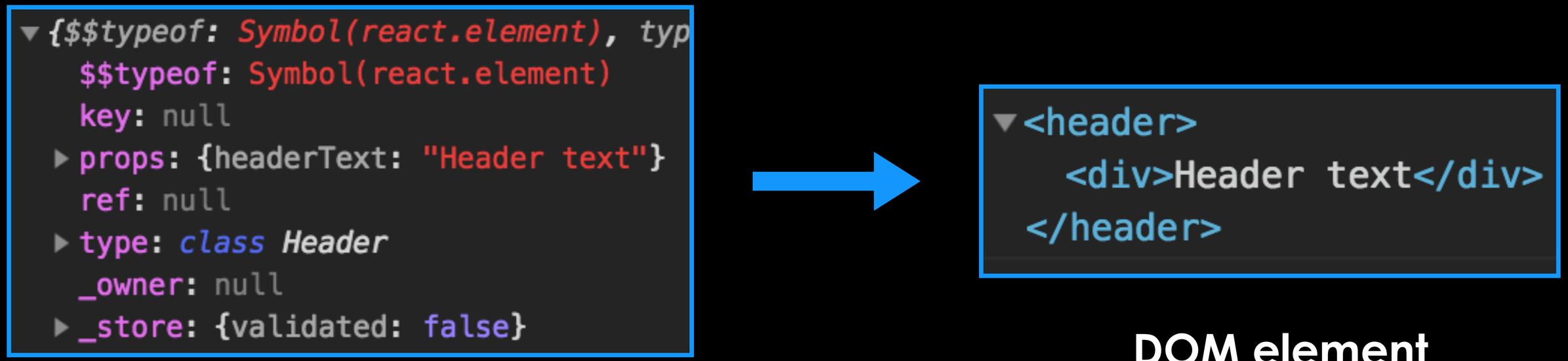

Header
Slider
Sidebar
News
News

Image source: www.hawaii.edu


Props
function Header() {
}
React element


{$$typeof: Symbol(react.element), typ
$$typeof: Symbol(react.element)
key: null
props: {headerText: "Header text"}
ref: null
type: class Header
_owner: null
_store: {validated: false}
<header>
Header text
</header>
React element
DOM element

```
function Header(props) {
    return (
        <header>
            <div>{props.headerText}</div>
        </header>
    );
};
```

JSX

bit.ly/2ttPzcV

```
<Header headerText="Header text" />
```

Functional components

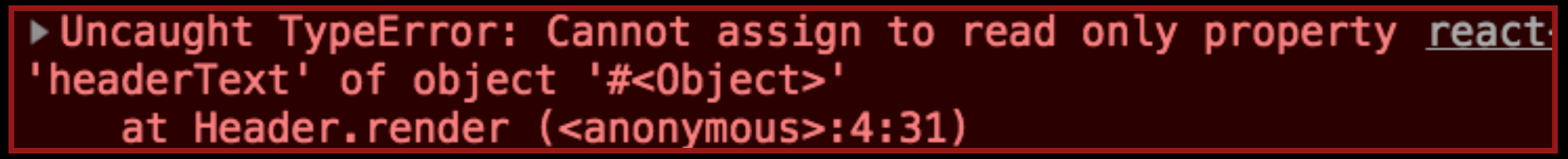
```
function Header(props) {
    return (
        <header>
            <div>{props.headerText}</div>
        </header>
    );
};
```

Class components

```
class Header extends React.Component {
    render() {
        return (
            <header>
                <div>{this.props.headerText}</div>
            </header>
        );
    }
};
```

```
class Header extends React.Component {
    render() {
        this.props.headerText = 'New header text';
        return (
            <header>
                <div>{this.props.headerText}</div>
            </header>
        );
    }
}
```

Props are read-only! 😕



All React components must act like pure functions with respect to their props.

bit.ly/2ywNi6n

Use state to update a component.

```
class Header extends React.Component {
    constructor(props) {
        super(props);

        this.state = {
            headerText: props.headerText // Set initial state
        };
    }

    changeHeaderText() {
        this.setState({ headerText: 'New header text' }); // Update state
    }

    render() {
        return (
            <header>
                <div>{this.state.headerText}</div>
                <button onClick={this.changeHeaderText.bind(this)}>
                    Change header text
                </button>
            </header>
        );
    }
}
```

Dumb/stateless components

```
function Loader(props) {
    let { type, style } = props,
        src = null;

    switch (type) {
        case 'green':
            src = 'green-loader.gif';
            break;
        case 'blue':
        default:
            src = 'blue-loader.gif';
            break;
    }

    return (
        <img src={src} style={style} />
    );
};
```

Smart/statefull components

```
class ListMenu extends React.Component {
    constructor(props) {
        super(props);

        this.state = { hidden: true };
    }

    toggleMenuVisibility() {
        this.setState((prevstate) => {
            return { hidden: !prevstate.hidden };
        });
    }

    render() {
        return (
            <div>
                {
                    !this.state.hidden ? (<ul><li>Item 1</li><li>Item 2</li><li>Item
                    3</li></ul>) : null
                }
                <button onClick={this.toggleMenuVisibility.bind(this)}>
                    Toggle Menu
                </button>
            </div>
        );
    }
}
```

- Abstracting complexity
- Maintainability
- Reusability
- Ease of testing

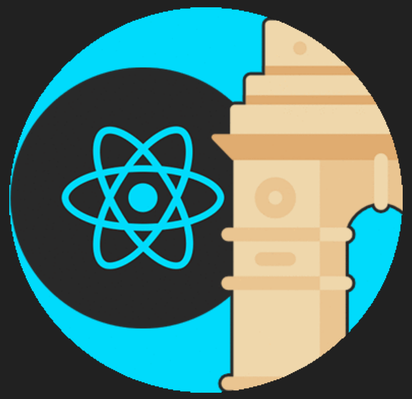
Quick links

- https://reactjs.org/docs/components-and-props.html
- https://www.youtube.com/watch?v=0BHoz0ptv7k
- https://reactjs.org/docs/thinking-in-react.html

Thank you!

Arun Michael Dsouza

http://arunmichaeldsouza.com/



React Delhi/NCR