

# React: A Sneak Peak

Jai Santhosh

UI Engineer @ Flipkart

[@jaisanth](#)

Meet **React**.

Not yet another JS framework!

Meet **React**.

But just a library for creating user  
interfaces.

# Meet React.

Renders your UI and responds to events.

Meet React.

a.k.a. The “v” in **M V C**

Build components,  
not templates

Build components,  
not templates

Re-render, don't mutate

Build components,  
not templates

Re-render, don't mutate

Fast Virtual DOM



# Build components, not templates

- Combine DOM generation and display logic.

# Build components, not templates

- Combine DOM generation and display logic.
- Transform data into interface components.

# Build components, not templates

- Combine DOM generation and display logic.
- Transform data into interface components.
- Data can come from anywhere.

# Build components, not templates

# Build components, not templates

```
1  /** @jsx React.DOM */
2  var HelloMessage = React.createClass({
3    render: function() {
4      return <div>{'Hello ' + this.props.name}</div>;
5    }
6  });
7
8  React.renderComponent(<HelloMessage name="John" />, mountNode);
```

# JSX

Allows you to write JS & XML

# JSX

Allows you to write JS & XML  
Together!

# JSX

Don't need to use it as such



# JSX

```
React.DOM.div({}, 'Hello ' + this.props.name)
```

## Re-render, don't mutate!

- Every component has “state” and “props”.

## Re-render, don't mutate!

- Every component has “state” and “props”.
- Acts as an “Immutable DOM”.

## Re-render, don't mutate!

- Every component has “state” and “props”.
- Acts as an “Immutable DOM”.
- Re-render the whole app once the state changes.

Re-render, don't mutate!

Data is guaranteed  
to update!

Re-render, don't mutate!

Just like the 90's  
Full-page refresh!!

```
var Clicker = React.createClass({
  getInitialState: function() {
    return {count: 0};
  },
  handleClick: function() {
    this.setState({count: this.state.count + 1});
  },
  render: function() {
    return React.DOM.a(
      {onClick: this.handleClick},
      'You have clicked ' + this.state.count + ' times'
    );
  }
});

React.renderComponent(Clicker(), mountNode);
```

## Virtual DOM

Makes re-rendering on every change fast.



# On every update...

- **React** builds a new virtual DOM subtree

# On every update...

- **React** builds a new virtual DOM subtree
- ... **diffs** it with the old one

### On every update...

- React builds a new virtual DOM subtree
- ...diffs it with the old one
- ...computes the minimal set of DOM mutations and puts them in a queue

### On every update...

- **React** builds a new virtual DOM subtree
- ... **diffs** it with the old one
- ... computes the **minimal set of DOM mutations** and puts them in a queue
- ... and **batch exexutes** all updates

## none to first

- Create Node: `<div  
className="first"><span>A Span</span></div>`

## first to second

- Replace attribute: `className="first"` by `className="second"`
- Replace node: `<span>A Span</span>` by `<p>A Paragraph</p>`

## second to none

- Remove Node: `<div  
className="second"><p>A Paragraph</p></div>`

Virtual DOM is simple and fast

<http://reactjs.org/>  
[#reactjs](#) on Freenode IRC  
[reactjs](#) on Google Groups

# Other helpful links:

- <http://swannodette.github.io/2013/12/17/the-future-of-javascript-mvcs/>
- <https://news.ycombinator.com/item?id=6936975>
- <https://news.ycombinator.com/item?id=6937921>
- <http://calendar.perfplanet.com/2013/diff/>



Thank You!!