

A large, faint, light blue watermark of the React logo is centered in the background. It consists of three interlocking loops forming a stylized atomic or molecular structure.

Introduction **React.js**

Webworkers Cologne, 27th January 2016, Christoph Jerolimov

Agenda

1. Motivation & Concepts
2. Components & JSX
3. Virtuell DOM
4. Demo

React

A JAVASCRIPT LIBRARY FOR BUILDING USER INTERFACES

[Get Started](#)[Download React v0.14.3](#)

JUST THE UI

Lots of people use React as the V in MVC. Since React makes no assumptions about the rest of your technology stack, it's easy to try it out on a small feature in an existing project.

VIRTUAL DOM

React abstracts away the DOM from you, giving a simpler programming model and better performance. React can also render on the server using Node, and it can power native apps using [React Native](#).

DATA FLOW

React implements one-way reactive data flow which reduces boilerplate and is easier to reason about than traditional data binding.

A Simple Component

React components implement a `render()` method that takes input data and returns what to display. This example uses an XML-like syntax called JSX. Input data that is passed into the component can be accessed by `render()` via `this.props`.

JSX is optional and not required to use React. Try clicking on "Compiled JS" to see the raw JavaScript code produced by the JSX compiler.

2007



2009



2012



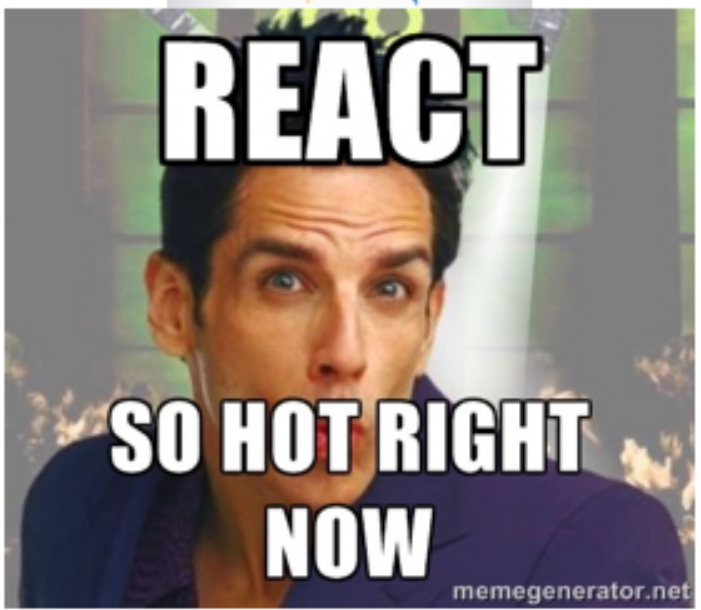
2013



2014



2015



Motivation

Web**apps** are becoming increasingly complex

Goal: Simplify UI-Code

Urban Exploration

[Follow](#)[Share](#)

3071
followers

🐦 23 shares

[Urban Exploration](#)[Japan](#)

Hashima: the Japanese island that time forgot

The Independent • 10h ago

The boat roiled dramatically, coming closer to horizontal than anyone would have liked, left and right, left, left, deeper left. Then, when we reached our destination, its port side bashed

violently against the concrete jetty. After several white-knuckle, rollercoaster-ride attempts at docking, our captain abandoned the idea, and we turned back. Gunkanjima would...



🐦 1 share

[Urban Exploration](#)[Water](#)

The Ruins of the Austin Dam

Abandoned Playgrounds • yesterday

The Austin Dam ruins are down a dirt road off of Jones Run, off of 105



Urban Exploration



Share

3072

followers

23 shares

Urban Exploration ✓

Japan

Hashima: the Japanese island that time forgot

The Independent • 10h ago

The boat roiled dramatically, coming closer to horizontal than anyone would have liked, left and right, left, left, deeper left. Then, when we reached our destination, its port side bashed

violently against the concrete jetty. After several white-knuckle, rollercoaster-ride attempts at docking, our captain abandoned the idea, and we turned back. Gunkanjima would...



1 share

Urban Exploration ✓

Water

The Ruins of the Austin Dam

Abandoned Playgrounds • yesterday

The Austin Dam ruins are down a dirt road off of Jones Run, off of 105



Quiz

How to update all relevant elements?

Idea

Just re-render the complete page!

Solution

Declarative UI

(Unidirectional data flow)

A view-only library

(the view in MVC, but MVC is not required)

Automatically updates the DOM

(the browser is just *one possible* rendering engine)

Components

Every Element is/extends a `React.Component`

Internal private *state*

VS

External immutable *props*

Must implement at least the `render()`-method

Optional methods to handle the lifecycle/updates
(`componentWillMount` ... `componentWillUnmount`)

Simple JSX example

JSX is a JS superset and supports sub-components (and DOM elements) inline:

```
class HelloWorld extends React.Component {  
  render() {  
    return <span>Hello World</span>;  
  }  
}
```

JSX usage

```
// Easy to render test data
```

```
const followButtonEnabled = <FollowButton enabled={ false } />  
const followButtonDisabled = <FollowButton enabled={ true } />
```

```
// Use "any source" as props in production
```

```
const followButton = <FollowButton  
    enabled={ currentUser.tags.contains(currentTag) }  
    onClick={ () => { toggleTag(currentTag); } } />
```


JSX state example

```
class Blink extends React.Component {
  constructor(props) {
    super(props);
    this.state = { visible: true }
  }
  componentWillMount() {
    this.interval = setInterval(() => {
      this.setState({ visible: !this.state.visible });
    }, 1000);
  }
  componentWillUnmount() {
    this.clearInterval(this.interval);
  }
  render() {
    const style = { opacity: this.state.visible ? 1 : 0 };
    return <span style={ style }>{ this.props.children }</span>
  }
}
```

Virtual DOM

DOM manipulations are slow.

Render method generates a VDOM ("JSON")
and calculates a diff to reduce DOM manipulations.

$$\text{render} (UI\text{-}State_n) \Rightarrow VDOM_n$$
$$\text{diff} (VDOM_{n-1} , VDOM_n) \Rightarrow DOM \text{ updates...}$$

Demo

Questions?

Thank you