

React

INTRODUCTION

React is an open-source **JavaScript library** for creating user interfaces that aims to address challenges encountered in developing **single-page applications**.



Agenda

- **Not a framework**
- **Components**
- **Virtual DOM**
- **One-way data flow**

NOT A FRAMEWORK

said some random dude

frame·work

/'frām,wɜrk/ 

noun

an essential supporting structure of a building, vehicle, or object.

"a conservatory in a delicate framework of iron"

synonyms: [frame](#), [substructure](#), [infrastructure](#), [structure](#), [skeleton](#), [chassis](#), [shell](#),
[body](#), [bodywork](#); [More](#)

- a basic structure underlying a system, concept, or text.

"the theoretical framework of political sociology"

synonyms: [structure](#), [shape](#), [fabric](#), [order](#), [scheme](#), [system](#), [organization](#),
[construction](#), [configuration](#), [composition](#), warp and woof;
informal [makeup](#)
"the framework of society"



Translations, word origin, and more definitions

JUST THE “V”

Not the MVC

A glass of whiskey with ice and a coconut with two straws. The glass is on the left, filled with a golden liquid and large ice cubes. The coconut is on the right, with a light blue and a yellow straw inserted into its top. The background is white.

AGNOSTIC

*...Ruby, Rails, Java,
.NET, Node...*

COMPONENTS

COMPONENT ORIENTED

- **Self contained, concern specific building blocks**
- **Divide and conquer strategy**
- **Reusable and composable**

COMPONENT ORIENTED

```
5 export default class Account extends React.Component {
6   render () {
7     return (
8       <div className="row">
9         <SubHeader value={this.props.account.name} />
10        <div className="large-12 columns">
11          {this._createTeams()}
12        </div>
13      </div>
14    );
15  }
16
17  _toTeamComponent(team, index) {
18    return (
19      <div className="row" key={index}>
20        <Team name={team.name} spots={team.spots} periods={this.props.periods} startsOn={team.s
21      </div>
22    )
23  }
24
```

**It looks like HTML
It tastes like HTML
But it's something else**



It looks like HTML
It tastes like HTML
But it's something else



The screenshot shows the CodePen web editor interface. At the top, there's a navigation bar with the CodePen logo and buttons for 'Save', 'Fork', 'Info', and 'Share'. Below this, the editor is divided into two main sections. The left section is the code editor, which is currently showing a JavaScript file named 'JS'. It contains two code snippets. The first snippet, on lines 1-4, uses the `React.render()` method to render a simple `<div>!` into the `document.body`. The second snippet, on lines 6-9, uses `React.render()` with `React.createElement()` to create a `div` element with the text `¡Hola, Mundo con jsx!` and renders it into the `document.body`. The right section of the editor is the preview area, which displays the rendered output of the code. It shows the text `¡Hola, Munc`, which is a truncated version of the text in the second code snippet. The interface has a dark theme.

```
1 React.render(  
2   <div>!Hola, Mundo!</div>,  
3   document.body  
4 );  
5  
6 React.render(  
7   React.createElement('div', {}, '¡Hola, Mundo con jsx!'),  
8   document.body  
9 );  
10
```

¡Hola, Munc

THE POWER OF JAVASCRIPT

*Map, reduce, filter,
sort... not {{#each}}*

AUTOMATIC REFRESH

lifecycle


```

export default class OpenRoles extends React.Component {
  static propTypes = {
    openRoles: React.PropTypes.array.isRequired,
    todayDate: React.PropTypes.string.isRequired,
    teamName: React.PropTypes.string
  };

  constructor(props) {
    super(props);
    this.state = {
      open: false
    };
  }

  render() {
    return (
      <div className="open-roles-details-container">
        <a onClick={this._toggleOpenRolesDetailsVisibility.bind(this)}>
          <i className={this.state.open ? 'fa fa-chevron-left' : 'fa
            fa-chevron-right' } ></i>
        </a>
        {this._renderDetails()}
      </div>
    );
  }

  componentWillMount() {
    if (this.props.teamName == undefined) {
      this.setState({ open: true });
    }
  }
}

```

VIRTUAL DOM AND REACTIVE RENDER

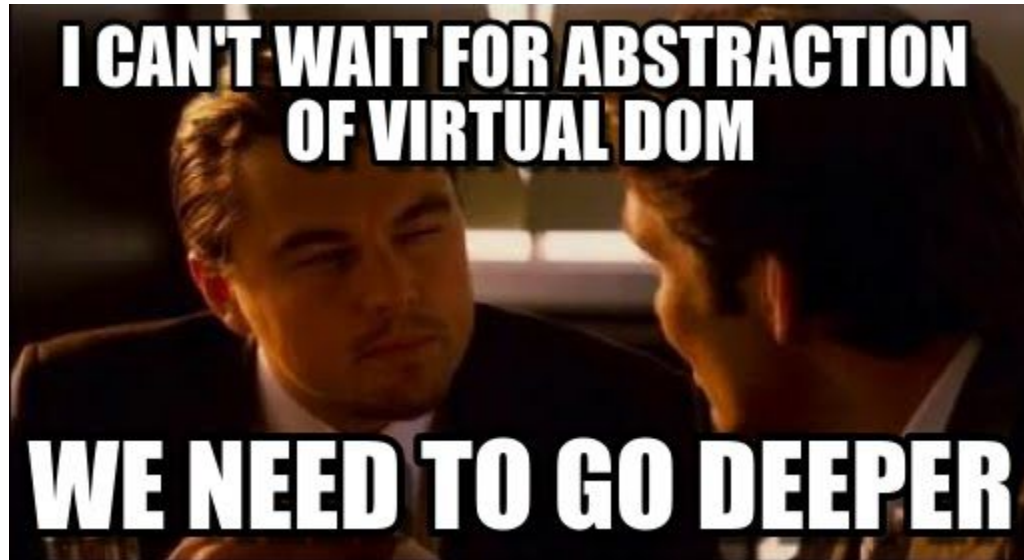
DOM CHANGING OVER TIME

...is the root of all evil...

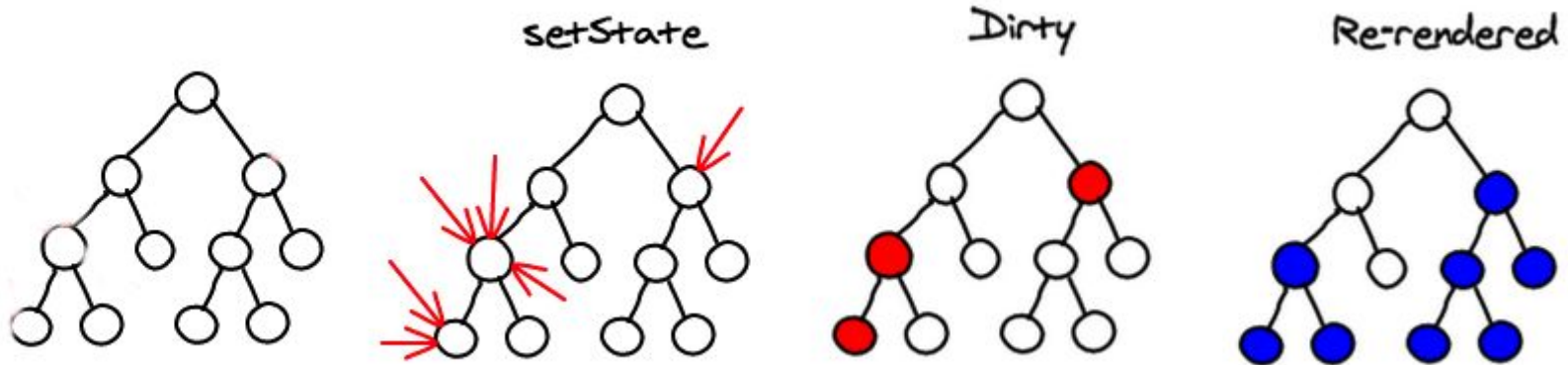
DOM manipulation is slow, but React
keeps it at the **minimum**

DOM ABSTRACTION with **VIRTUAL DOM**

Performance, server-side rendering



React builds a tree **representation of the DOM** and calculates what should change -
in memory



ONE-WAY DATA BINDING

LOW COUPLING

PERFORMANCE & DEBUGGING

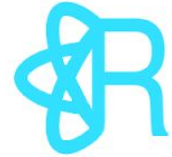
WHAT WE'VE MISSED

TOOLKIT

ES6, module bundlers, Redux



webpack
MODULE BUNDLER



QUESTIONS AND FEEDBACK?
