

# Introduction à ReactJS



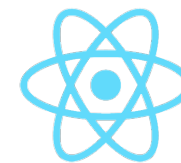
François Agneray

# Présentation

## Permet de faciliter la création d'application web monopage

Les caractéristiques :

- ReactJS permet de fabriquer des composants web
- Un composant ReactJS génère du code HTML à chaque changement d'état
- ReactJS ne gère que la partie interface de l'application web (Vue)
- ReactJS peut être utilisé avec une autres bibliothèque ou un framework (AngularJS)



React



Instagram

## Les composants ReactJS sont décrits via le langage JSX

Un premier exemple de composant simple :

- Notre composant définit une méthode **render()**
- Notre composant accepte un paramètre d'entrée **name** accessible via **this.props**
- Notre composant est monté sur le nœud dont l'id est app

```
var HelloMessage = React.createClass({
  render: function() {
    return <div>Hello {this.props.name}</div>;
  }
});

ReactDOM.render(<HelloMessage name="John" />, app);
```

## Les scripts JSX doivent être compilés en JavaScript

```
var HelloMessage = React.createClass({
  render: function() {
    return <div>Hello {this.props.name}</div>;
  }
});

ReactDOM.render(<HelloMessage name="John" />, app);
```

*BABEL*

```
var HelloMessage = React.createClass({
  displayName: "HelloMessage",
  render: function render() {
    return React.createElement("div", null, "Hello ", this.props.name);
  }
});

ReactDOM.render(React.createElement(HelloMessage, { name: "John" }), app);
```

# HTML

Pour fonctionner la page HTML doit :

- Charger la bibliothèque **ReactJS**
- Charger la bibliothèque **Babel Browser**
- Spécifier le type **text/babel** pour les fichiers **JSX**

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Hello React !</title>
    <link rel="stylesheet" href="css/app.css" />
  </head>
  <body>
    <div id="app"></div>

    <script src="bower_components/react/react.min.js"></script>
    <script src="bower_components/react/react-dom.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/babel-
core/5.8.23/browser.min.js"></script>
    <script type="text/babel" src="js/helloworld.jsx"></script>
  </body>
</html>
```

# Composant avec état

## Un composant ReactJS peut garder un état interne

```
var Timer = React.createClass({
  getInitialState: function() {
    return {secondsElapsed: 0};
  },
  tick: function() {
    this.setState({secondsElapsed: this.state.secondsElapsed + 1});
  },
  componentDidMount: function() {
    this.interval = setInterval(this.tick, 1000);
  },
  componentWillUnmount: function() {
    clearInterval(this.interval);
  },
  render: function() {
    return (
      <div>Secondes : {this.state.secondsElapsed}</div>
    );
  }
});

ReactDOM.render(<Timer />, app);
```

Secondes : 0



Secondes : 1



Secondes : 2



...

# Composant complexe

```
var Board = React.createClass({
  render: function() {
    var className = "board";
    if (this.props.selected) {
      className += " selected";
    }
    return (
      <div className={className}>
        {this.props.index + 1}
      </div>
    );
  }
});
```



```
.board {
  border: 5px solid #ccc;
  float: left;
  font: 700 24px Helvetica;
  margin-right: 20px;
  padding: 20px;
}
```

# Composant complexe

```
var BoardSwitcher = React.createClass({
  getInitialState: function() {
    return {selectedIndex: 0}
  },

  onToggleClick: function() {
    this.setState({
      selectedIndex: (this.state.selectedIndex + 1) % this.props.numBoards
    })
  },

  render: function() {
    var boards = [];
    for (var ii = 0; ii < this.props.numBoards; ii++) {
      var isSelected = (ii === this.state.selectedIndex);
      boards.push(<Board index={ii} selected={isSelected} />);
    }

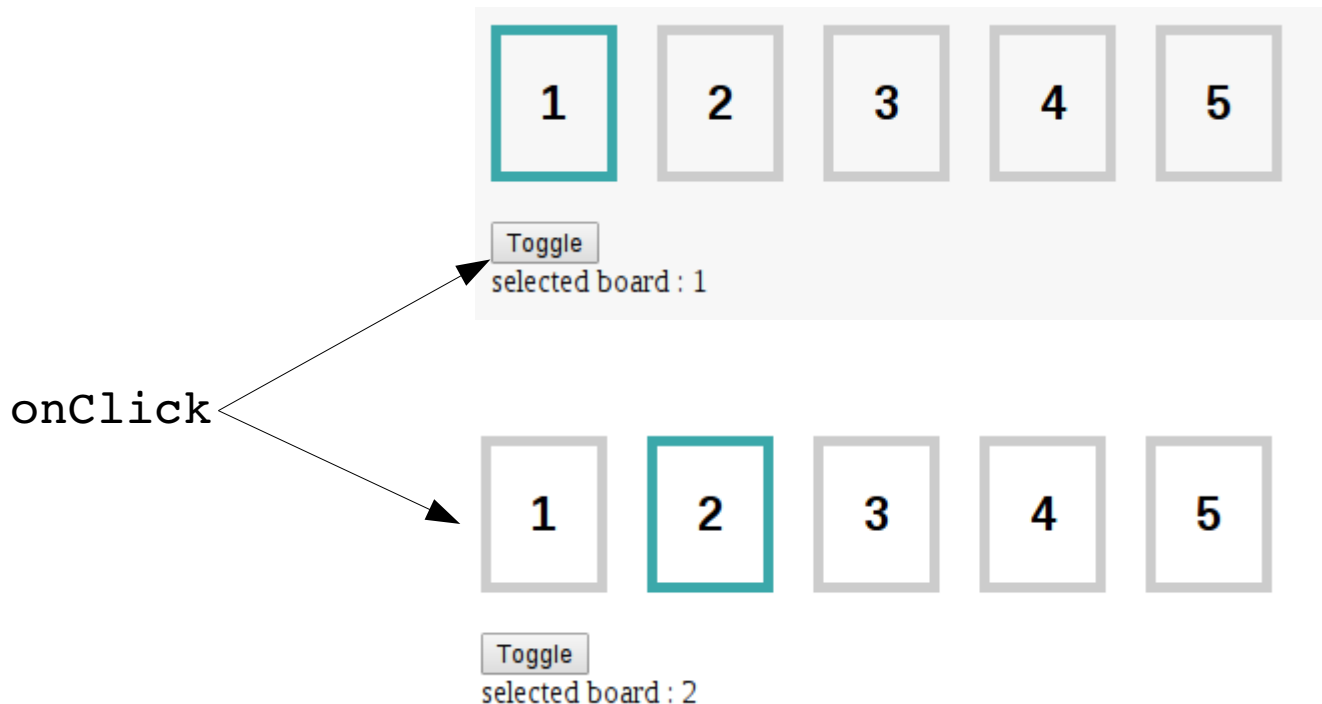
    return (
      <div>
        <div className="boards">{boards}</div>
        <button onClick={this.onToggleClick}>Toggle</button>
        <div>selected board : {this.state.selectedIndex + 1}</div>
      </div>
    );
  }
});
```



# Composant complexe

```
React.render(  
  <BoardSwitcher numBoards={5} />,  
  document.body  
);
```

```
.boards {  
  margin: 20px 0;  
  overflow: hidden;  
}  
.board.selected {  
  border-color: #3BA8AA;  
}
```



# Sudoku

1	7					4	3	
		6				7	9	
								1
2					9		8	
4	1		2	7	8		6	9
	6		5					2
9								
	8	3				2		
	4	2					5	8

New Game

00:00:11

<http://andrey.nering.com.br/sudoku/>

NETFLIX **Browse** ▼ **KIDS** DVD Search Timothy ▼

New Releases


NETFLIX **DAREDEVIL** NETFLIX **Grace and Frankie** NETFLIX **HOUSE of CARDS** **BOB'S BURGERS**


## Marvel's Daredevil

★★★★★ 2015 TV-MA 1 Season

**Resume:**  
Season 1, Episode 1: "Into the Ring"  
— 51 of 53m

Murdock's vigilante crime fighting and his new law practice find equally dangerous challenges in a murder case tied to a corporate crime syndicate.

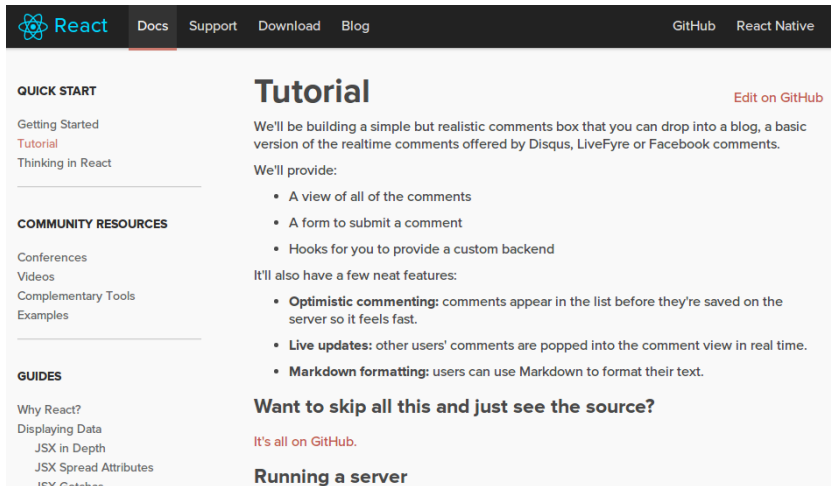
 TV star Vincent D'Onofrio plays Daredevil's nemesis Wilson Fisk, a.k.a. Kingpin.

 MY LIST

OVERVIEW EPISODES MORE LIKE THIS DETAILS

Resume

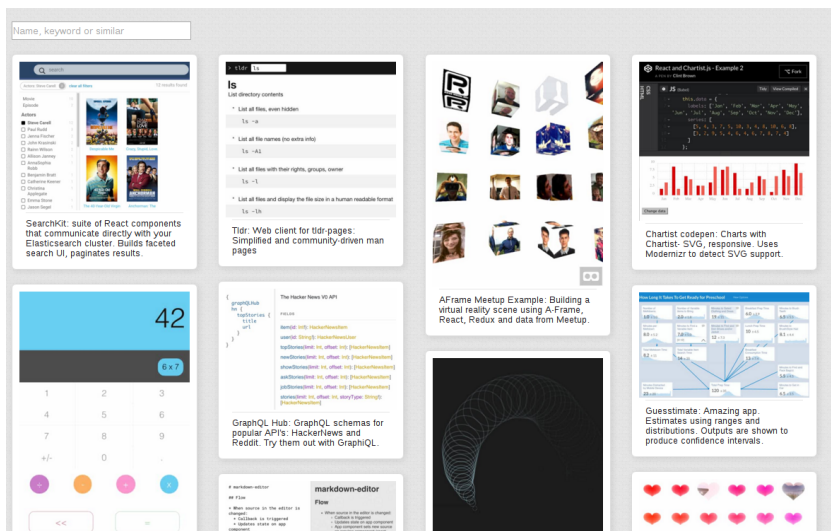
# Conclusion



The screenshot shows the React documentation website. The top navigation bar includes 'React', 'Docs', 'Support', 'Download', 'Blog', 'GitHub', and 'React Native'. The main content area is titled 'Tutorial' and includes a 'QUICK START' section with links for 'Getting Started', 'Tutorial', and 'Thinking in React'. Below this is a 'COMMUNITY RESOURCES' section with links for 'Conferences', 'Videos', 'Complementary Tools', and 'Examples'. A 'GUIDES' section lists 'Why React?', 'Displaying Data', 'JSX in Depth', 'JSX Spread Attributes', and 'JSX Github'. The 'Tutorial' section itself contains a 'Want to skip all this and just see the source?' link and a 'Running a server' section.

La documentation et des tutoriaux sont disponibles sur :

<https://facebook.github.io>



The screenshot shows the React Examples website. It features a search bar at the top with the text 'Name, keyword or similar'. Below the search bar, there are several example cards. The first card is 'SearchKit: suite of React components that communicates directly with your Elasticsearch cluster. Builds faceted search UI, paginates results.' The second card is 'Is: List directory contents' with a list of file names and sizes. The third card is 'Tldr: Web client for tldr-pages: Simplified and community-driven man pages'. The fourth card is 'AFrame Meetup Example: Building a virtual reality scene using A-Frame, React, Redux and data from Meetup.' The fifth card is 'GraphQL Hub: GraphQL schemas for popular APIs: Hackernews and Reddit. Try them out with GraphQL.' The sixth card is 'Guestimate: Amazing app. Estimates using ranges and distributions. Outputs are shown to produce confidence intervals.' The seventh card is 'markdown-editor' with a preview of a document.

Des exemples d'application sont disponibles sur :

<https://react.rocks>

<http://www.overreact.io/>