← DEVHINTS.IO

React.js cheatsheet

```
React is a JavaScript library for building user interfaces. This guide targets React v15 to v16.
                                                        Import multiple exports
Components
                                                                                                                 Properties
                                                                                                                   <Video fullscreen={true} autoplay={false} />
                                                           import React, {Component} from 'react'
  import React from 'react'
  import ReactDOM from 'react-dom'
                                                           import ReactDOM from 'react-dom'
                                                                                                                   render () {
                                                          class Hello extends Component {
                                                                                                                     this.props.fullscreen
  class Hello extends React.Component {
                                                                                                                     const { fullscreen, autoplay } = this.props
    render () {
      return <div className='message-box'>
        Hello {this.props.name}
      </div>
                                                        States
                                                                                                                   Use this.props to access properties passed to the
                                                                                                                   component.
                                                           constructor(props) {
                                                                                                                   See: Properties
  const el = document.body
                                                             super(props)
  ReactDOM.render(<Hello name='John' />, el)
                                                            this.state = { username: undefined }
                                                                                                                 Nesting
  Use the React.js jsfiddle to start hacking. (or the
  unofficial jsbin)
                                                           this.setState({ username: 'rstacruz' })
                                                                                                                   class Info extends Component {
                                                                                                                     render () {
                                                           render () {
                                                                                                                       const { avatar, username } = this.props
Children
                                                             this state username
                                                            const { username } = this.state
                                                                                                                        return <div>
                                                             . . .
  <AlertBox>
                                                                                                                          <UserAvatar src={avatar} />
    <h1>You have pending notifications</h1>
                                                                                                                          <UserProfile username={username} />
  </AlertBox>
                                                                                                                        </div>
                                                           Use states (this.state) to manage dynamic data.
  class AlertBox extends Component {
                                                           With Babel you can use proposal-class-fields and get rid
    render () {
                                                           of constructor
      return <div className='alert-box'>
                                                                                                                   As of React v16.2.0, fragments can be used to return
        {this.props.children}
                                                                                                                   multiple children without adding extra wrapping nodes
      </div>
                                                           class Hello extends Component {
                                                                                                                   to the DOM.
                                                            state = { username: undefined };
                                                                                                                   import React, {
                                                                                                                     Component,
  Children are passed as the children property.
                                                                                                                     Fragment
                                                                                                                   } from 'react'
                                                           See: States
                                                                                                                   class Info extends Component {
                                                                                                                     render () {
                                                                                                                       const { avatar, username } = this.props
                                                                                                                        return (
                                                                                                                          <Fragment>
                                                                                                                            <UserAvatar src={avatar} />
                                                                                                                            <UserProfile username={username} />
                                                                                                                          </Fragment>
                                                                                                                   Nest components to separate concerns.
                                                                                                                   See: Composing Components
```

Hello.defaultProps = { color: 'blue'

Defaults

Setting default props

```
See: defaultProps

See: defaultProps

Set the default state in the constructor ().

And without constructor using Babel with proposal-class-fields.

class Hello extends Component {
    state = { visible: true }
    }
}

See: Setting the default state

# Other components

Functional components

Pure components

Component API
```

import React, {PureComponent} from 'react'

class MessageBox extends PureComponent {

Performance-optimized version of React.Component.

Doesn't rerender if props/state hasn't changed.

. . .

After rendering (DOM available) #

Before DOM removal #

Setting default state

class Hello extends Component {

this.forceUpdate()

this state

this.props

this.setState({ ... })

this.setState(state => { ... })

Render

Operate on the DOM here

constructor (props) {

Functional components have no state. Also, their props are passed as the first parameter to a function.

Hello {name}

</div>

function MyComponent ({ name }) {

return <div className='message-box'>

```
See: Function and Class Components
                                                                 See: Pure components
                                                                                                                             These methods and properties are available for
                                                                                                                             Component instances.
                                                                                                                             See: Component API
# Lifecycle
    Mounting
                                                                                             Updating
                                                                     Before rendering #
                                                                                                                                                       Use setState() here, but
                                                                                               componentDidUpdate (prevProps, prevState, snapshot)
      constructor (props)
                                                                                                                                                    remember to compare props
                                                                        Don't use this #
      componentWillMount()
                                                                                                                                                   Skips render() if returns false
                                                                                               shouldComponentUpdate (newProps, newState)
      render()
                                                                              Render #
```

render()

componentDidUpdate (prevProps, prevState)

onChange (event) {

onChange={event => this.onChange(event)} />

this.setState({ value: event.target.value })

ReactDOMServer.renderToString(<Component />)

There are more, but these are most common.

See: React top-level API

See: Dangerously set innerHTML

Inner HTML

Lists

ReactDOMServer.renderToStaticMarkup(<Component />)

function markdownify() { return "..."; }

<div dangerouslySetInnerHTML={{__html: markdownify()}} />

See: Error handling in React 16

ReactDOM.hydrate(<App />, el)

of ReactDOMServer.

MyCo.propTypes = {

user: PropTypes.shape({

name: PropTypes.string,

age: PropTypes.number

Use ReactDOM. hydrate instead of using

const el = document.getElementById('app')

ReactDOM.render if you're rendering over the output

Hydration

componentWillUnmount()

componentDidMount()

```
Called when parents change properties and .setState(). These are not called for initial
                                                                 Catch errors (16+) #
      componentDidCatch()
                                                                                           renders.
                                                                                          See: Component specs
      Set initial the state on constructor(). Add DOM event handlers, timers (etc) on
      componentDidMount(), then remove them on componentWillUnmount().
# DOM nodes
    References
                                                                                        DOM Events
      class MyComponent extends Component {
                                                                                          class MyComponent extends Component {
                                                                                             render () {
        render () {
          return <div>
                                                                                               <input type="text"</pre>
            <input ref={el => this.input = el} />
                                                                                                   value={this.state.value}
```

componentDidMount () { this.input.focus()

</div>

```
# Other features

Transferring props

Top-level API

React.createClass({ ... })
React.isValidElement(c)

ReactDOM.render(<Component />, domnode, [callback])
ReactDOM.ummountComponentAtNode(domnode)
```

const style = { height: 10 }

return <div style={style}></div>

return <div style={{ margin: 0, padding: 0 }}></div>

JSX patterns

Style shorthand

See: Inline styles

See Transferring props

Propagates src="..." down to the sub-component.

```
Conditionals
                                                                                     class TodoList extends Component {
                                                                                       render () {
                                                                                        const { items } = this.props
      <Fragment>
        {showMyComponent
                                                                                         return 
         ? <MyComponent />
                                                                                           {items.map(item =>
         : <OtherComponent />}
                                                                                            <TodoItem item={item} key={item.key} />)}
      </Fragment>
                                                                                         Short-circuit evaluation
                                                                                     Always supply a key property.
      <Fragment>
       {showPopup && <Popup />}
      </Fragment>
# New features
   Returning multiple elements
                                                        Returning strings
                                                                                                             Errors
                                                          render() {
                                                                                                               class MyComponent extends Component {
     You can return multiple elements as arrays or
                                                            return 'Look ma, no spans!';
     fragments.
                                                                                                                 componentDidCatch (error, info) {
                                                                                                                   this.setState({ error })
      Arrays
                                                          You can return just a string.
     render () {
       // Don't forget the keys!
                                                          See: Fragments and strings
        return [
                                                                                                               Catch errors via componentDidCatch. (React 16+)
         First item,
```

Portals

render () {

the DOM.

return React.createPortal(

document.getElementById('menu')

This renders this.props.children into any location in

this.props.children,

</Fragment>

Fragments

key="B">Second item

First item

Second item

```
See: Portals
                                                                                                                  See: Hydrate
      See: Fragments and strings
# Property validation
                                                                                                                 Required types
    PropTypes
                                                          Basic types
      import PropTypes from 'prop-types'
                                                                                                                  MyCo.propTypes = {
                                                            {\tt MyComponent.propTypes} \ = \ \{
                                                                                                                    name: PropTypes.string.isRequired
                                                                          PropTypes.string,
                                                              email:
                                                                          PropTypes.number,
                                                              seats:
      See: Typechecking with PropTypes
                                                                          PropTypes.func,
                                                              callback:
                                                                          PropTypes.bool,
                                                              isClosed:
                                                                          PropTypes.any
                                                              any:
                                            Anything
      any
                                                                                                                 Elements
      Basic
                                                                                                                  MyCo.propTypes = {
      string
                                                                                                                    // React element
                                                          Enumerables (oneOf)
                                                                                                                    element: PropTypes.element,
      number
                                                                                                                    // num, string, element, or an array of those
                                                            MyCo.propTypes = {
                                             Function
      func
                                                              direction: PropTypes.oneOf([
                                                                                                                     node: PropTypes.node
                                                                'left', 'right'
      bool
                                          True or false
      Enum
      oneOf(any)
                                          Enum types
                                                                                                                Arrays and objects
                                                          Custom validation
      oneOfType(type array)
                                               Union
                                                                                                                  MyCo.propTypes = {
      Array
                                                                                                                    list: PropTypes.array,
                                                            MyCo.propTypes = {
                                                                                                                    ages: PropTypes.arrayOf(PropTypes.number),
                                                              customProp: (props, key, componentName) => {
      array
                                                                                                                     user: PropTypes.object,
                                                                if (!/matchme/.test(props[key])) {
                                                                                                                     user: PropTypes.objectOf(PropTypes.number),
                                                                  return new Error('Validation failed!')
      array0f(...)
                                                                                                                    message: PropTypes.instanceOf(Message)
      Object
```

Object object objectOf(...) Object with values of a certain type

instanceOf(...)

Instance of a class

```
shape(...)

Elements

element React element

node DOM node

Required

(···).isRequired Required

**Also see*

React website (react;s.org)

React cheatsheet (reactcheatsheet.com)

Awesome React (github.com)

React v0.14 cheatsheet Legacy version
```

• React v0.14

Other React cheatsheets

Top cheatsheets

