DEVHINTS.IO

Edit

React.js cheatsheet

React is a JavaScript library for building user interfaces. This guide targets React v15 to v16.

Components

```
import React from 'react'
import ReactDOM from 'react-dom'
```

```
class Hello extends React.Component {
  render () {
    return <div className='message-box'>
      Hello {this.props.name}
    </div>
  }
}
```

```
const el = document.body
ReactDOM.render(<Hello name='John' />, el)
```

Use the React.js jsfiddle to start hacking. (or the unofficial jsbin)

Import multiple exports

```
import React, {Component} from 're
import ReactDOM from 'react-dom'
```

```
class Hello extends Component {
  ...
}
```

States

```
constructor(props) {
  super(props)
  this.state = { username: undefin
}
```

```
this.setState({ username: 'rstacru
```

```
render () {
```

Children

```
<AlertBox>

</AlertBox>
```

```
class AlertBox extends Component {
  render () {
    return <div className='alert-box'>
    </div>
  }
}
```

Children are passed as the `children` property.

```
};
```

Defaults

Setting default props

```

  color: 'blue'
}
```

See: defaultProps

Setting

```
class H
  const
    sup

  }
}
```

Set the

And witl

```
class H

}
```

See: Set

Other components

Functional components

```
  return <div className='message-box'>
    Hello {name}
  </div>
}
```

Functional components have no state. Also, their `props` are pass function.

See: Function and Class Components

Pure components

```
import React, {PureComponent} from

  ...
}
```

Performance-optimized version of `React`

See: Pure components

Lifecycle

Mounting

`constructor` (props)	Before render
`componentWillMount()`	Don't use
`render()`	Ren
`componentDidMount()`	After rendering (DOM availa
`componentWillUnmount()`	Before DOM remo
`componentDidCatch()`	Catch errors (1

Set initial the state on `constructor()`. Add DOM event handlers, timers (etc) on `componentDidMount()`, then remove them on `componentWillUnmount()`.

Updatin

`compone`

`shouldC`

`render(`

`compone`

Called w

See: Cor

Hooks (New)

State Hook

```
import React, { useState } from 'react';

function Example() {
  // Declare a new state variable, which we'll call "count"


  return (
    <div>
      <p>You clicked {count} times</p>

        Click me
      </button>
    </div>
  );
}
```

Hooks are a new addition in React 16.8.

See: Hooks at a Glance

Building your own hooks

Define FriendStatus

```
import React, { useState, useEffect } from 'react';

function FriendStatus(props) {
  const [isOnline, setIsOnline] = useState(null);

  useEffect(() => {
    function handleStatusChange(status) {
      setIsOnline(status.isOnline);
    }





  }, [props.friend.id]);

  if (isOnline === null) {
    return 'Loading...';
  }
```

Declarin

```
import

functic
  // De
  const
  const
  const
  // ..
}
```

Effect h

```
import

functic
  const
```

```
ur
di
  <
  <

  <
/c
```

're

one

fau

Hooks A

```
  return isOnline ? 'Online' : 'Offline';
}
```

Effects may also optionally specify how to “clean up” after them by returning a function.

Use FriendStatus

```
function FriendStatus(props) {


  if (isOnline === null) {
    return 'Loading...';
  }
  return isOnline ? 'Online' : 'Offline';
}
```

See: Building Your Own Hooks

Also see

Basic Hoo

useStat

useEffe

useCont

Full deta

Additiona

useRedu

useCall

useMemc

useRef(i

useImpe

useLayc

useDebu

Full deta

DOM nodes

References

```
class MyComponent extends Component {
  render () {
    return <div>

    </div>
  }

  componentDidMount () {
```

DOM Ev

```
class M
  rende
    <ir


  }

  onCha
  }
}
```

Allows access to DOM nodes.

See: Refs and the DOM

Other features

Transferring props

```
<VideoPlayer src="video.mp4" />
```

```
class VideoPlayer extends Component {
  render () {

  }
}
```

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

See Transferring props

JSX patterns

Style shorthand

```
const style = { height: 10 }
return <div style={style}></div>
```

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

See: Inline styles

Conditionals

```
  }
}
```

Pass fun

See: Eve

Top-leve

```
React.c
React.i
```

```
ReactDC
ReactDC
```

```
ReactDC
ReactDC
```

There ar

See: Rea

Inner HT

```
functic
<div da
```

See: Dar

Lists

```
class T
```

```
<Fragment>
  {showPopup && <Popup />}
  ...
</Fragment>
```

```
nde
cor

et
 {

</u
  }
}
```

Always s

New features

Returning multiple elements

You can return multiple elements as arrays or fragments.

Arrays

```
render () {
  // Don't forget the keys!




}
```

Fragments

```
render () {
  return (
```

See: Fragments and strings

Returning strings

```
render() {

}
```

You can return just a string.

See: Fragments and strings

Portals

```
render () {




}
```

This renders `this.props.children` into a

See: Portals

Property validation

PropTypes

```
import PropTypes from 'prop-types'
```

See: Typechecking with PropTypes

any	
Basic	
string	
number	
func	
bool	
Enum	
oneOf(any)	Enum types
oneOfType(type array)	Union
Array	
array	
arrayOf(…)	
Object	
object	
objectOf(…)	Object with values of a certain type
instanceOf(…)	Instance of a class
shape(…)	
Elements	
element	React element
node	DOM node

Basic types

```
MyComponent.propTypes = {
  email:      PropTypes.string,
  seats:      PropTypes.number,
  callback:   PropTypes.func,
  isClosed:   PropTypes.bool,
  any:        PropTypes.any
}
```

Enumerables (oneOf)

```
MyCo.propTypes = {
  direction: PropTypes.oneOf([
    'left', 'right'
  ])
}
```

Custom validation

```
MyCo.propTypes = {
  customProp: (props, key, compone
    if (!/matchme/.test(props[key]
      return new Error('Validation
    }
  }
}
```

Required	
`(···).isRequired`	Required

Also see

- React website (reactjs.org)
- React cheatsheet (reactcheatsheet.com)
- Awesome React (github.com)
- React v0.14 cheatsheet Legacy version

Search 358+ cheatsheets

Over 358 curated cheatsheets, by developers for developers.

Devhints home

Other React cheatsheets

Redux
cheatsheet

Enzyme
cheatsheet

Top cheatsheets

Elixir
cheatsheet

ES2015+
cheatsheet

Enzyme v2
cheatsheet

Awesome Redux
cheatsheet

Vimdiff
cheatsheet

Vim
cheatsheet

Flux architecture
cheatsheet

React-router
cheatsheet

Vim scripting
cheatsheet

Vue.js
cheatsheet