

Basic Methods		Others		React (cont)	
imp→	import moduleName from 'module'	cmmb→	comment block	cwup→	componentWillUpdate = (nextProps, nextState) => { } DEPRECATED!!!
imn→	import 'module'	React		cdup→	componentDidUpdate = (prevProps, prevState) => { }
imd→	import { destructuredModule } from 'module'	imr→	import React from 'react'	cwun→	componentWillUnmount = () => { }
ime→	import * as alias from 'module'	imrd→	import ReactDOM from 'react-dom'	gdsfp→	static getDerivedStateFromProps(nextProps, prevState) { }
ima→	import { originalName as aliasName } from 'module'	imrc→	import React, { Component } from 'react'	gsbu→	getSnapshotBeforeUpdate = (prevProps, prevState) => { }
exp→	export default moduleName	imrcp→	import React, { Component } from 'react' & import PropTypes from 'prop-types'	ren→	render() { return () }
exd→	export { destructuredModule } from 'module'	imrpc→	import React, { PureComponent } from 'react'	sst→	this.setState({ })
exa→	export { originalName as aliasName } from 'module'	imrpcp→	import React, { PureComponent } from 'react' & import PropTypes from 'prop-types'	ssf→	this.setState((state, props) => return { })
enf→	export const functionName = (params) => { }	imrm→	import React, { memo } from 'react'	props→	this.props.propName
edf→	export default (params) => { }	imrmp→	import React, { memo } from 'react' & import PropTypes from 'prop-types'	state→	this.state.stateName
met→	methodName = (params) => { }	impt→	import PropTypes from 'prop-types'	rcontext→	const \${1:contextName} = React.createContext()
fre→	arrayName.forEach(element => { }	imrr→	import { BrowserRouter as Router, Route, Link } from 'react-router-dom'	cref→	this.\${1:refName}Ref = React.createRef()
fof→	for(let itemName of objectName { }	redux→	import { connect } from 'react-redux'	fref→	const ref = React.createRef()
fin→	for(let itemName in objectName { }	rconst→	constructor(props) with this.state	bnd→	this.methodName = this.methodName.bind(this)
anfn→	(params) => { }	rconc→	constructor(props, context) with this.state	Redux	
nfn→	const functionName = (params) => { }	est→	this.state = { }	rxaction→	redux action template
dob→	const {propName} = objectToDe- escruct	cwm→	componentWillMount = () => { } DEPRECATED!!!	rxconst→	export const \$1 = '\$1'
dar→	const [propName] = arrayToDe- scruct	cdm→	componentDidMount = () => { }	rxreducer→	redux reducer template
sti→	setInterval(() => { }, intervalTime	cwr→	componentWillReceiveProps = (nextProps) => { } DEPREC- ATED!!!	rxselect→	redux selector template
sto→	setTimeout(() => { }, delayTime	scu→	shouldComponentUpdate = (nextProps, nextState) => { }	Console	
prom→	return new Promise((resolve, reject) => { }			clg→	console.log(object)
cmmb→	comment block			clo→	console.log("object", object)
cp→	const { } = this.props			ctm→	console.time("timelid")
cs→	const { } = this.state			cte→	console.timeEnd("timelid")
				cas→	console.assert(expression,object)
				ccl→	console.clear()
				cco→	console.count(label)
				cdi→	console.dir
				cer→	console.error(object)

Console (cont)

cgr→	console.group(label)
cge→	console.groupEnd()
ctr→	console.trace(object)
cwa→	console.warn
cin→	console.info

Component Templates

rfc	react functional component
rfce	react functional component with last line export
rfcp	react functional component with PropTypes and last line export
rcc	react class component
rce	react class component with last line export
rccp	react class component with PropTypes and ES7 module system
rcep	react class component with PropTypes and last line export
rcredux	redux connected class component with PropTypes
reduxmap	mapping to props constants
rpc	react pure component
rpcp	react pure component with PropTypes
rpce	react pure component with PropTypes and last line export
hoc	react higher order component with PropTypes
hocredux	redux connected higher order component with PropTypes
rafc	react arrow function component



By **frankieali4**
cheatography.com/frankieali4/

Published 19th February, 2019.
Last updated 19th February, 2019.
Page 2 of 2.

Sponsored by **Readable.com**
Measure your website readability!
<https://readable.com>