

Basic Methods

Prefix	Method
imp→	import moduleName from 'module'
imn→	import 'module'
imd→	import { destructuredModule } from 'module'
ime→	import * as alias from 'module'
ima→	import { originalName as aliasName} from 'module'
exp→	export default moduleName
exd→	export { destructuredModule } from 'module'
exa→	export { originalName as aliasName} from 'module'
enf→	export const functionName = (params) => { }
edf→	export default (params) => { }
met→	methodName = (params) => { }
fre→	arrayName.forEach(element => { }
fof→	for(let itemName of objectName { }
fin→	for(let itemName in objectName { }
anfn→	(params) => { }
nfn→	const functionName = (params) => { }
dob→	const {propName} = objectToDescruct
dar→	const [propName] = arrayToDescruct
sti→	setInterval(() => { }, intervalTime
sto→	setTimeout(() => { }, delayTime
prom→	return new Promise((resolve, reject) => { }
cmmb→	comment block
cp→	const { } = this.props
cs→	const { } = this.state

React

Prefix	Method
imr→	import React from 'react'
imrd→	import ReactDOM from 'react-dom'
imrc→	import React, { Component } from 'react'

Prefix	Method
imrcp→	import React, { Component } from 'react' & import PropTypes from 'prop-types'
imrpc→	import React, { PureComponent } from 'react'
imrpcp→	import React, { PureComponent } from 'react' & import PropTypes from 'prop-types'
imrm→	import React, { memo } from 'react'
imrmp→	import React, { memo } from 'react' & import PropTypes from 'prop-types'
impt→	import PropTypes from 'prop-types'
imrr→	import { BrowserRouter as Router, Route, NavLink } from 'react-router-dom'
imbr→	import { BrowserRouter as Router } from 'react-router-dom'
imbrc→	import { Route, Switch, NavLink, Link } from react-router-dom'
imbr→	import { Route } from 'react-router-dom'
imbrs→	import { Switch } from 'react-router-dom'
imbrl→	import { Link } from 'react-router-dom'
imbrnl→	import { NavLink } from 'react-router-dom'
imrs→	import React, { useState } from 'react'
imrse→	import React, { useState, useEffect } from 'react'
redux→	import { connect } from 'react-redux'
rconst→	constructor(props) with this.state
rconc→	constructor(props, context) with this.state
est→	this.state = { }
cwm→	componentWillMount = () => { } DEPRECATED!!!
cdm→	componentDidMount = () => { }
cwr→	componentWillReceiveProps = (nextProps) => { } DEPRECATED!!!
scu→	shouldComponentUpdate = (nextProps, nextState) => { }
cwup→	componentWillUpdate = (nextProps, nextState) => { } DEPRECATED!!!
cdup→	componentDidUpdate = (prevProps, prevState) => { }
cwun→	componentWillUnmount = () => { }
gdsfp→	static getDerivedStateFromProps(nextProps, prevState) { }
gsbu→	getSnapshotBeforeUpdate = (prevProps, prevState) => { }
ren→	render() { return() }

Prefix	Method
sst→	this.setState({ })
ssf→	this.setState((state, props) => return { })
props→	this.props.propName
state→	this.state.stateName
rcontext→	const \${1:contextName} = React.createContext()
cref→	this.\${1:refName}Ref = React.createRef()
fref→	const ref = React.createRef()
bnd→	this.methodName = this.methodName.bind(this)

React Hooks

- All hooks from [official docs](#) are added with hook name prefix.

Prefix	Method
rch→	import { \$1 } from 'react-native'

React Native

Prefix	Method
imrn→	import { \$1 } from 'react-native'
rnstyle→	const styles = StyleSheet.create({})

Redux

Prefix	Method
rxaction→	redux action template
rxconst→	export const \$1 = '\$1'
rxreducer→	redux reducer template
rxselect→	redux selector template
rxslice→	redux slice template

PropTypes

Prefix	Method
pta→	PropTypes.array

Prefix	Method
ptar→	PropTypes.array.isRequired
ptb→	PropTypes.bool
ptbr→	PropTypes.bool.isRequired
ptf→	PropTypes.func
ptfr→	PropTypes.func.isRequired
ptn→	PropTypes.number
ptnr→	PropTypes.number.isRequired
pto→	PropTypes.object
ptor→	PropTypes.object.isRequired
pts→	PropTypes.string
ptsr→	PropTypes.string.isRequired
ptnd→	PropTypes.node
ptndr→	PropTypes.node.isRequired
ptel→	PropTypes.element
ptelr→	PropTypes.element.isRequired
pti→	PropTypes.instanceOf(name)
ptir→	PropTypes.instanceOf(name).isRequired
pte→	PropTypes.oneOf([name])
pter→	PropTypes.oneOf([name]).isRequired
ptet→	PropTypes.oneOfType([name])
ptetr→	PropTypes.oneOfType([name]).isRequired
ptao→	PropTypes.arrayOf(name)
ptaor→	PropTypes.arrayOf(name).isRequired
ptoo→	PropTypes.objectOf(name)
ptoor→	PropTypes.objectOf(name).isRequired
ptsh→	PropTypes.shape({ })
ptshr→	PropTypes.shape({ }).isRequired
ptany→	PropTypes.any
ptypes→	static propTypes = {}

GraphQL

Prefix	Method
graphql→	import { compose, graphql } from react-apollo'
expgraphql->	export default compose(graphql(\$1, { name: \$2 }))(\$3)

Console

Prefix	Method
clg→	console.log(object)
clo→	console.log(`object`, object)
ctm→	console.time(`timeId`)
cte→	console.timeEnd(`timeId`)
cas→	console.assert(expression,object)
ccl→	console.clear()
cco→	console.count(label)
cdi→	console.dir
cer→	console.error(object)
cgr→	console.group(label)
cge→	console.groupEnd()
ctr→	console.trace(object)
cwa→	console.warn
cin→	console.info

React Components

Note for React 17 use _ prefix. Currently workaround for future options.

rcc

```
import React, { Component } from 'react'

export default class FileName extends Component {
  render() {
    return <div>$2</div>
  }
}
```

rce


```
import React, { Component } from 'react'
import { connect } from 'react-redux'

export const FileName = () => {
  return <div>$4</div>
}

const mapStateToProps = (state) => ({}))

const mapDispatchToProps = {}

export default connect(mapStateToProps, mapDispatchToProps)(FileName)
```

rfreduxp

```
import React, { Component } from 'react'
import PropTypes from 'prop-types'
import { connect } from 'react-redux'

export const FileName = () => {
  return <div>$4</div>
}

FileName.propTypes = {
  $2: $3,
}

const mapStateToProps = (state) => ({}))

const mapDispatchToProps = {}

export default connect(mapStateToProps, mapDispatchToProps)(FileName)
```

reduxmap

```
const mapStateToProps = (state) => ({}))

const mapDispatchToProps = {}
```

React Native Components

rnc

```
import React, { Component } from 'react'
import { Text, View } from 'react-native'

export default class FileName extends Component {
  render() {
    return (
      <View>
        <Text> $2 </Text>
      </View>
    )
  }
}
```



```

import React, { Component } from 'react'
import { View, Text } from 'react-native'
import PropTypes from 'prop-types'
import { connect } from 'react-redux'

export class FileName extends Component {
  static propTypes = {
    $2: $3,
  }

  render() {
    return (
      <View>
        <Text> $2 </Text>
      </View>
    )
  }
}

const mapStateToProps = (state) => ({}))

const mapDispatchToProps = {}

export default connect(mapStateToProps, mapDispatchToProps)(FileName)

```

Others

cmmmb

```

/**
|-----
| $1
|-----
*/

```

desc

```

describe('$1', () => {
  $2
})

```

test

```

test('should $1', () => {
  $2
})

```

tit

