ES7 React VSCode Shortcuts Keyboard Shortcuts

by jaredcobb via cheatography.com/135652/cs/28165/

Basic Methods

```
imp→ import moduleName from 'module'
imn→ import 'module'
imd→ import { destru ctu red Module } from 'module'
ime→ import * as alias from 'module'
ima→ import { origin alName as aliasName} from
'module'
exp→ export default moduleName
exd→ export { destru ctu red Module } from 'module'
exa→ export { origin alName as aliasName} from
'module'
enf→ export const functi onName = (params) => { }
edf→ export default (params) => { }
met→ methodName = (params) => { }
fre→ arrayN ame.fo rEa ch( element => { }
fof→ for(let itemName of objectName { }
fin→ for(let itemName in objectName { }
anfn→ (params) => { }
nfn→ const functi onName = (params) => { }
dob→ const {propName} = object ToD escruct
dar→ const [propName] = arrayT oDe scruct
sti→ setInt erv al(() => { }, interv alTime
sto→ setTim eout(() => { }, delayTime
prom→ return new Promis e(( res olve, reject) => {
}
cmmb→ comment block
```

PropTypes

```
pta→ PropTypes.array
ptar→ PropTy pes.ar ray.is Req uired
ptb→ PropTy pes.bool
ptbr→ PropTy pes.bo ol.i sR equired
ptf→ PropTy pes.func
ptfr→ PropTy pes.fu nc.i sR equired
ptn→ PropTy pes.number
ptnr→ PropTy pes.nu mbe r.i sRe quired
pto→ PropTy pes.object
ptor→ PropTy pes.ob jec t.i sRe quired
pts→ PropTy pes.string
ptsr→ PropTy pes.st rin g.i sRe quired
ptnd→ PropTy pes.node
ptndr→ PropTy pes.no de.i sR equired
ptel→ PropTy pes.el ement
ptelr→ PropTy pes.el eme nt.i sR equired
pti→ PropTy pes.in sta nce Of( name)
```

PropTypes (cont)

```
> 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 = {}
```

Components

```
rfc
import React from 'react'
export default function $1() {
    return <di v>$ 0</ div>
}
rfcp
import React from 'react'
import PropTypes from 'prop- types'
function $1(props) {
    return <di v>$ 0</ div>
}
$1.pro pTypes = {}
export default $1
rfce
import React from 'react'
function $1() {
    return <di v>$ 0</ div>
}
export default $1
```

React

```
imr→ import React from 'react'
imrd→ import ReactDOM from 'react -dom'
imrc→ import React, { Component } from 'react'
imrcp→ import React, { Component } from 'react' &
import PropTypes from 'prop- types'
imrpc→ import React, { PureCo mponent } from
'react'
```


Sponsored by **CrosswordCheats.com**
Learn to solve cryptic crosswords!
http://crosswordcheats.com

React (cont)

> 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'
imbrr→ 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'
cdm→ componentDidMount = () => { }
scu→ shouldComponentUpdate = (nextProps, nextState) => { }
cdup→ componentDidUpdate = (prevProps, prevState) => { }
cwun→ componentWillUnmount = () => { }
gdsfp→ static getDerivedStateFromProps(nextProps, prevState) { }
gsbu→ getSnapshotBeforeUpdate = (prevProps, prevState) => { }

Redux

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

Console

```
clg→ console.log(object)
clo→ consol e.l og( 'ob ject', object)
ctm→ consol e.t ime ('t imeId')
cte→ consol e.t ime End ('t imeId')
cas→ consol e.a sse rt( exp res sio n,o bject)
ccl→ consol e.c lear()
cco→ consol e.c oun t(l abel)
cdi→ consol e.dir
cer→ consol e.e rro r(o bject)
cgr→ consol e.g rou p(l abel)
cge→ consol e.g rou pEnd()
ctr→ consol e.t rac e(o bject)
```

Console (cont)

> cwa→ console.warn
cin→ console.info

Components

```
rfcredux
import React, { Component } from 'react'
import { connect } from 'react -redux'
export const FileName = () => {
    return <di v>$ 4</ div>
}
const mapSta teT oProps = (state) => ({})
const mapDis pat chT oProps = {}
export default connec t(m apS tat eTo Props,
mapDis pat chT oPr ops )(F ile Name)
rfreduxp
import React, { Component } from 'react'
import PropTypes from 'prop- types'
import { connect } from 'react -redux'
export const FileName = () => {
    return <di v>$ 4</ div>
}
FileNa me.p ro pTypes = {
   $2: $3,
}
const mapSta teT oProps = (state) => ({})
const mapDis pat chT oProps = {}
export default connec t(m apS tat eTo Props,
mapDis pat chT oPr ops )(F ile Name)
reduxmap
const mapSta teT oProps = (state) => ({})
const mapDis pat chT oProps = {}
```


Sponsored by **CrosswordCheats.com**
Learn to solve cryptic crosswords!
http://crosswordcheats.com