Cheatography

React Reusability Cheat Sheet

by hackingbeauty (hackingbeauty) via cheatography.com/37001/cs/11658/

</> Component Type #1 - Class-based

```
class MyComponent extends Component {
  static propTypes    = {}
  static defaul tProps = {}
    con str uct or( props) {
        sup er( props);
        thi s.state = { // state keys go here }
   }
    ren der() { }
}
```

Define a component this way when you need to maintain internal state.

</> Component Type #2 - Functional

```
function MyComponent(props) {
    return ( <di v>< /di v>);
}
```

Define a component this way when you just need to render UI (no internal state required).
Synonymous with "Presentational Component".

Lifecycle Hooks

`constr uctor()`	Called before component is mounted
`render()`	Never call setState() here
`compon ent Wil lMo unt()`	Before rendering (no DOM yet)
`compon ent Did Mount()`	After rendering
`compon ent Wil lUp date()`	Can't use `setState()` here
`compon ent Did Upd ate()`	Operate on the DOM here
`compon ent Wil lRe cei veP - rops()`	Use `setState()` here
`should Com pon ent Upd ate( )`	Skips `render()` if returns false

Available PropTypes

`PropTy pes.array`	Check if prop is an array
`PropTy pes.string`	Check if prop is a string
`PropTy pes.bool`	Check if prop is a boolean (true/false)
`PropTy pes.number`	Check if prop is a number
`PropTy pes.object`	Check if prop is an object
`PropTy pes.node`	Check if prop is anything that can be rendered in a React component
`PropTy pes.func`	Check if prop is a function
`PropTy pes.el emen t`	Check if prop is a React element

Available PropTypes (cont)

`PropTy pes.on eOf (['M', 'F''])`	Check if prop matches one of these values
`PropTypes.oneOfType([ PropTypes.string, PropTypes.bool ])`	Check if prop matches one of these types
`PropTy pes.in sta nce Of( Class)`	Check if prop is an instance of a class
`PropTy pes.ar ray Of( Array)`	Check if prop is an array of some specific type
`PropTy pes.ob jec tOf (Pr opT ype s. n umber)`	Check if prop is an object with property values of a specific type
`PropTy pes.sh ape({ potato: PropTypes.object, turkey: PropTypes.string })`	Check if prop object has a particular "sha- pe"
`PropTy pes.any`	Check if anything at all is passed

Reusability Checklist (ES6)

Step #1 -Define your component and import PropTypes from the 'prop-types' package

```
 import React, { Component } from 'react';
   import PropTypes                        from 'prop
types';

   class MyComp onent extends Component {
       static propTypes      =  {}
       static defaul tProps = {}

       con str uct or( props) {
           sup er( props);
       }

       ren der() {}
   }
```

Step #2 - Define the props & what type they are

```
...
 static propTypes: {
     cla ssName   : PropTy pes.string
     isC losed    :  PropTy pes.bool
     email           : PropTy pes.string
 }
...
```

Step #3 - Determine which props are required or optional

```
...
 static propTypes: {
     cla ssName   : PropTy pes.string
     isC losed    :  PropTy pes.bool
     email           : PropTy pes.st rin g.i sRe qu
ed
 }
...
```

Step #4 - Define the defaults (if applicable)

```
...
static defaul tProps: {
     classN ame   : "btn"
     isClos ed   : true
}
...
```

By **hackingbeauty**
(hackingbeauty)

Published 7th May, 2017.
Last updated 7th May, 2017.

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

React Reusability Cheat Sheet

by hackingbeauty (hackingbeauty) via cheatography.com/37001/cs/11658/

</> You can custom validate a prop

```
...
static propTypes: {
    email: ((props, propName) => {
        const regex = /^\w+@  [a - z A-  Z_]  +? \.
[ a - zA  -Z]  {2 , 3}$/;
        if (!rege  x.t  e st  (pr  op s [ pr  -
opN  ame])) {
         return new Error(  'I n valid email!');
        }
   })
}
...
```

Other APIs

`setSta te( upd ater, [callb a ck])`	Component will re-render with new state
`forceU pdate()`	Calls `render()`, skips `should Com pon ent - Upd ate()`

Next Steps

Knowing how to create reusable components is step 1.

Then you need to learn how to architect professional apps using React & Redux.

Learn how at www.singlepageapplication.com

By **hackingbeauty**
(hackingbeauty)

cheatography.com/hackingbeauty/
hackingbeauty.com

Published 7th May, 2017.
Last updated 7th May, 2017.

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