**BASICS RULE**

* Only include one component per file  
  - However, multiple [Stateless, or Pure, Components](https://facebook.github.io/react/docs/reusable-components.html#stateless-functions) are allowed per file.
* Do not use React.Component or React.createClass() for creating react class component. Instead, do like “import React, {Component} from ‘react’

Declaring Class component

// bad

const Listing = React.createClass({  
 // ..  
 render() {

return <div>{this.state.hello}</div>;

}

});

// good

import React, {Component} from ‘react’;

class Listing extends Component {

//...

render() {

return <div>{this.state.hello}</div>;  
 }  
}

**Naming**

* **File Name**: should be in PascalCase
* **File Extension**: As per Facebook-React Community, to declare a js file as jsx, you need to write at the starting portion of the file import React from ‘react’ to be recognised from react as jsx.
* **Reference Naming**: Use Pascal Case for declaring React Component and Camel Case for instance.

**Component Name**

* Use filename as the component name. Example: Button.js as class Button
* For the root component of the application, use index.js as filename and use the directory name as the component name

**Props Naming**

* Always use camelCase with props name
* Avoid declaring the props when they are not in use

**Alignment**

Follow this alignment style of for component:

// Type - 1

<MyComponent  
 className=”class-name”

type=”text”

/>

// Type - 2

<MyComponent

className=”class-name”

type=”button”

>

Submit

</MyComponent>

**Quotes**

* Always use double quotes for jsx attributes but for simple string use single quotes

**Indentation**

* Indent code 4 type space

**Parentheses**

* Start writing class and function with open braces in same line
  + Example:  
     // Function  
     Hello () {

console.log(‘hello !’);

}  
  
// Class  
Class Hello extends Component {

render() {

return <div>Hello</div>;  
 }  
}

* Give one space after class / function declaration.

**Tags**

* Always self close tag that has no children  
  - example:  
   <Foo />
* If a component has multi-line properties, close it’s tag in new line
  + Example:  
     <Foo  
     className=”hello”  
     type=”button”  
     />

**Methods**

* Use ES6 arrow function to close over local variable  
  - example:  
   class hello extends Component {  
   render() {  
   return let myName = () => ‘kris’;  
   }  
   }
* Bind event handler in the constructor instead on declaring inside the render function. To avoid performance issue, function binding should be done in constructor  
  - example:  
  class extends React.Component {  
   constructor(props) {  
   super(props);  
    
   this.onClickDiv = this.onClickDiv.bind(this);  
   }  
    
   onClickDiv() {  
   // do stuff  
   }  
    
   render() {  
   return <div onClick={this.onClickDiv} />;  
   }  
  }