Snippets

| **Snippet** | **Renders** |
| --- | --- |
| imr | Import React |
| imrc | Import React / Component |
| imrd | Import ReactDOM |
| imrs | Import React / useState |
| imrse | Import React / useState useEffect |
| impt | Import PropTypes |
| impc | Import React / PureComponent |
| cc | Class Component |
| ccc | Class Component With Constructor |
| cpc | Class Pure Component |
| ffc | Function Component |
| sfc | Stateless Function Component (Arrow function) |
| cdm | componentDidMount |
| uef | useEffect Hook |
| cwm | componentWillMount |
| cwrp | componentWillReceiveProps |
| gds | getDerivedStateFromProps |
| scu | shouldComponentUpdate |
| cwu | componentWillUpdate |
| cdu | componentDidUpdate |
| cwun | componentWillUnmount |
| cdc | componentDidCatch |
| gsbu | getSnapshotBeforeUpdate |
| ss | setState |
| ssf | Functional setState |
| usf | Declare a new state variable using State Hook |
| ren | render |
| rprop | Render Prop |
| hoc | Higher Order Component |
| cp | Context Provider |
| cpf | Class Property Function |

Full Expansions

imr - Import React

import \* as React from "react";

imrc - Import React, Component

import \* as React from "react";  
import { Component } from "react";

imrd - Import ReactDOM

import ReactDOM from "react-dom";

imrs - Import React, useState

import \* as React from "react";  
import { useState } from "react";

imrse - Import React, useState, useEffect

import \* as React from "react";  
import { useState, useEffect } from "react";

impt - Import PropTypes

import PropTypes from "prop-types";

impc - Import PureComponent

import \* as React from "react";  
import { PureComponent } from "react";

cc - Class Component

class | extends React.Component {  
 render() {  
 return <div>|</div>  
 }  
}  
  
export default |;

ccc - Class Component With Constructor

class | extends Component {  
 constructor(props) {  
 super(props);  
 this.state = { | };  
 }  
 render() {  
 return ( | );  
 }  
}  
  
export default |;

cpc - Class Pure Component

class | extends PureComponent {  
 state = { | },  
 render() {  
 return ( | );  
 }  
}  
  
export default |;

ffc - Function Component

function (|) {  
 return ( | );  
}  
  
export default |;

sfc - Stateless Function Component (Arrow function)

const | = props => {  
 return ( | );  
};  
  
export default |;

cdm - componentDidMount

componentDidMount() {  
 |  
}

uef - useEffect Hook

useEffect(() => {  
 |  
}, []);

cwm - componentWillMount

//WARNING! To be deprecated in React v17. Use componentDidMount instead.  
componentWillMount() {  
 |  
}

cwrp - componentWillReceiveProps

//WARNING! To be deprecated in React v17. Use new lifecycle static getDerivedStateFromProps instead.  
componentWillReceiveProps(nextProps) {  
 |  
}

gds - getDerivedStateFromProps

static getDerivedStateFromProps(nextProps, prevState) {  
 |  
}

scu - shouldComponentUpdate

shouldComponentUpdate(nextProps, nextState) {  
 |  
}

cwu - componentWillUpdate

//WARNING! To be deprecated in React v17. Use componentDidUpdate instead.  
componentWillUpdate(nextProps, nextState) {  
 |  
}

cdu - componentDidUpdate

componentDidUpdate(prevProps, prevState) {  
 |  
}

cwun - componentWillUnmount

componentWillUnmount() {  
 |  
}

cdc - componentDidCatch

componentDidCatch(error, info) {  
 |  
}

gsbu - getSnapshotBeforeUpdate

getSnapshotBeforeUpdate(prevProps, prevState) {  
 |  
}

ss - setState

this.setState({ | : | });

ssf - Functional setState

this.setState(prevState => {  
 return { | : prevState.| }  
});

usf - Declare a new state variable using State Hook

const [|, set|] = useState();

*Hit Tab to apply CamelCase to function. e.g. [count, setCount]*

ren - render

render() {  
 return (  
 |  
 );  
}

rprop - Render Prop

class | extends Component {  
 state = { | },  
 render() {  
 return this.props.render({  
 |: this.state.|  
 });  
 }  
}  
  
export default |;

hoc - Higher Order Component

function | (|) {  
 return class extends Component {  
 constructor(props) {  
 super(props);  
 }  
  
 render() {  
 return < | {...this.props} />;  
 }  
 };  
}

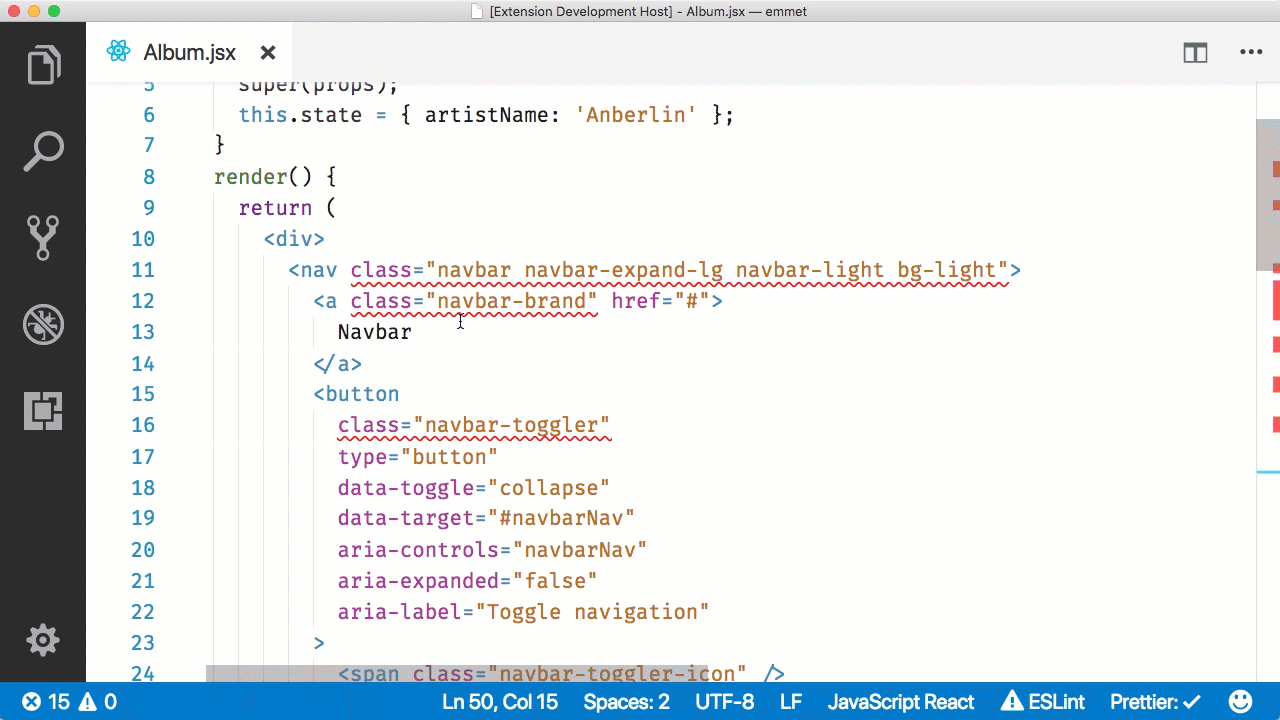
cpf - Class Property Function

| = (e) => {  
 |  
 }

Commands

React: class to className

Changes all occurences of class in JSX to className. This transform is safe to run multiple times on any document. No text needs to be selected as the command is executed on the entire document.



Open Settings: add Emmet Abbreviation

"emmet.includeLanguages": {

"javascript": "javascriptreact"

}