

General Rules

- Only one React component per file
- Always use JSX syntax

Naming Conventions

- Use **PascalCase** for filenames
Ex. RecipesList.js
- Use **PascalCase** for React components
- Use **camelCase** for React component instances
Ex. const recipesList = <RecipesList />;
- Use the filename as the component name (unless root component)
 - Ex. RecipesList.js should have reference name RecipesList
 - Root component should be named index.jsx

Declarations

- Name components by reference instead of by displayName
 - Ex.

```
export default function RecipesList(props) {  
  return()  
}
```

Props

- Use **camelCase** for prop names
- Use **PascalCase** if the prop value is a React component
- Leave the value of a prop blank when it is explicitly true
 - Ex.

<u>Incorrect</u>	<u>Correct</u>
<pre><RecipesList showRecipeList={true} /></pre>	<pre><RecipeList showRecipeList /></pre>

- Do not use array index as the key prop if the order of the items in the array changes

React Hooks

- Only call hooks at the top level, never inside loops, conditions, or nested functions
- Only call hooks from React functions

Parentheses

- Always use parentheses around JSX tags that take up multiple lines

Alignment

```
    {signupPageToggle && (  
      <SignupPage  
        signupPageToggle={signupPageToggle}  
        setSignupPageToggle={setSignupPageToggle}  
        setSearchToggle={setSearchToggle}  
      />  
    )}  
  
    {loginPageToggle && (  
      <LoginPage  
        setUser={setUser}  
        setLoginPageToggle={setLoginPageToggle}  
        setLoggedIn={setLoggedIn}  
        setSearchToggle={setSearchToggle}  
      />  
    )}  
  </div>  
  
  {showSearchbar() && (  
    <Searchbar  
      setIngredients={setIngredients}  
      ingredients={ingredients}  
      searchTagList={searchTagList}  
      setSearchTagList={setSearchTagList}  
      setSearchToggle={setSearchToggle}  
      setRecipes={setRecipes}  
      getRecipe={getRecipe}  
    />  
  )}
```