**When should I use arrow functions with React?**

In a JavaScript method, the value of this has nothing to do with the class on which it was *defined*. Instead, it depends on the object that it was called *upon*.

In terms of . syntax, this means that if you call obj.someMethod(), the value of this within someMethod will be obj — regardless of where someMethod was originally defined! If you just call someMethod() without using a ., this will be null.

There are two important exceptions:

* You can force a specific value of this with a function’s call, apply and bind methods
* If the function is an => arrow function, this is hard-wired to refer to the value of this at the location where the function was defined

## Arrows prevent this bugs

Arrow functions don’t redefine the value of this within their function body. This makes it a lot easier to predict their behavior when passed as callbacks, and prevents bugs caused by use of this within callbacks.

**You won’t introduce any bugs by using too many arrow functions.** You probably will introduce bugs but not using enough.

So to wrap things up, here are the three rules of arrow functions:

1. If your environment supports arrow methods, you can use them for *all* methods.
2. Use arrow functions within render. It’s ok. I promise.
3. If performance becomes an issue, check whether arrow functions are causing PureComponent or shouldComponentUpdate to make unnecessary updates.