**The question is, why is immutability so important? What is wrong in mutating objects? Doesn't it make things simple?**

Basically, it comes down to the fact that immutability increases predictability, performance (indirectly) and allows for mutation tracking.

**Predictability**

Mutation hides change, which create (unexpected) side effects, which can cause nasty bugs. When you enforce immutability, you can keep your application architecture and mental model simple, which makes it easier to reason about your application.

**Performance**

Even though adding values to an immutable Object means that a new instance needs to be created where existing values need to be copied and new values need to be added to the new Object which cost memory, immutable Objects can make use of structural sharing to reduce memory overhead.

All updates return new values, but internally structures are shared to drastically reduce memory usage (and GC thrashing). This means that if you append to a vector with 1000 elements, it does not actually create a new vector 1001-elements long. Most likely, internally only a few small objects are allocated.

You can read more about this [here](http://jlongster.com/Using-Immutable-Data-Structures-in-JavaScript#Immutable.js).

**Mutation Tracking**

Besides reduced memory usage, immutability allows you to optimize your application by making use of reference- and value equality. This makes it really easy to see if anything has changed. For example, a state change in a react component. You can use shouldComponentUpdate to check if the state is identical by comparing state Objects and prevent unnecessary rendering. You can read more about this [here](http://jlongster.com/Using-Immutable-Data-Structures-in-JavaScript#Reference---Value-Eq).

Additional resources:

* [The Dao of Immutability](https://medium.com/javascript-scene/the-dao-of-immutability-9f91a70c88cd#.9g51h5stk)
* [Immutable Data Structures and JavaScript](http://jlongster.com/Using-Immutable-Data-Structures-in-JavaScript#Immutable.js)
* [Immutability in JavaScript](http://www.sitepoint.com/immutability-javascript/)