**Component Life Cycle**

* constructor

The constructor for a React component is called before it is mounted. When implementing the constructor for a React.Component subclass, you should call super(props) before any other statement. Otherwise, this.props will be undefined in the constructor, which can lead to bugs.

* render()
* componentDidMount()

componentDidMount() is invoked immediately after a component is mounted.

* shouldComponentUpdate(nextProps, nextState)

shouldComponentUpdate() is invoked before rendering when new props or state are being received. Defaults to true. Use shouldComponentUpdate() to let React know if a component’s output is not affected by the current change in state or props.

* render()
* componentDidUpdate(prevProps, prevState)

componentDidUpdate() is invoked immediately after updating occurs.

* componentWillUnmonut()

componentWillUnmount() is invoked immediately before a component is unmounted and destroyed.

**16.3**

Many of these issues are exacerbated by a subset of the component lifecycles (componentWillMount, componentWillReceiveProps, and componentWillUpdate). These also happen to be the lifecycles that cause the most confusion within the React community. For these reasons, we are going to deprecate those methods in favor of better alternatives.

* [getDerivedStateFromProps](https://reactjs.org/docs/react-component.html#static-getderivedstatefromprops) is being added as a safer alternative to the legacy componentWillReceiveProps. (Note that [in most cases you don’t need either of them](https://reactjs.org/blog/2018/06/07/you-probably-dont-need-derived-state.html).)
* [getSnapshotBeforeUpdate](https://reactjs.org/docs/react-component.html#getsnapshotbeforeupdate) is being added to support safely reading properties from e.g. the DOM before updates are made.

**16.8**

目前，只有getSnapshotBeforeUpdate（）和componentDidCatch（）方法没有等效的Hooks API，并且这些生命周期相对不常见。 如果你愿意，你应该能够在你正在编写的大部分新代码中使用Hook。

# nodejs里的repl

## 什么是REPL?

REPL（Read-Eval-Print Loop） 中文的话有翻译成“交互式解释器”或“交互式编程环境”的。不过我觉得不用翻译，直接REPL就好了，这样的术语，翻译成中文后，读者更难理解。下面是对 REPL 的解释：

A Read-Eval-Print-Loop (REPL) is available both as a standalone program and easily includable in other programs. REPL provides a way to interactively run JavaScript and see the results. It can be used for debugging, testing, or just trying things out.

交互式解释器（REPL）既可以作为一个独立的程序运行，也可以很容易地包含在其他程序中作为整体程序的一部分使用。REPL为运行JavaScript脚本与查看运行结果提供了一种交互方式，通常REPL交互方式可以用于调试、测试以及试验某种想法。

基本上所有的脚本语言有REPL的。