**学习目标**

* 学会使用 create-react-app 创建 React 应用（创造她！）
* 学会通过 Props 传递参数（控制她！）
* 学会如何响应 Events（接受）
* 学会提取公用组件（精简）
* 学会利用 npm（美化）

**create-react-app介绍**

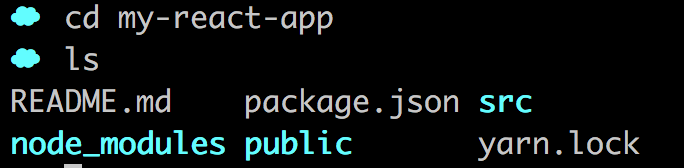
create-react-app 是由 Facebook 提供的一套开箱即用的 React 开发解决方案，提供了一键创建项目、开发、打包的一站式功能，集成了 React + Webpack 的功能。

**下载安装运行**

1. 下载安装Node.js
2. 打开终端，进入工作目录，输入指令：



1. 等待运行完成，运行以下指令进入目录



1. 在目录中运行以下指令，开启项目：



**Props**

React 中，每个组件均可以接收一些属性，即 Props。这些 Props 在渲染时被传递给组件，在组件中可以将 Props 的值与渲染进行绑定。绑定之后，Props 发生改变时，渲染出来的页面也会随着发生变更。

**Events**

React 中，事件和 DOM 事件非常相似。当一些事件发生时（比如点击按钮、按下鼠标、输入字符、元素加载完成等），会形成一个事件对象，并向上冒泡。在组件中，你可以通过捕获事件并绑定 Handler 的形式，处理这个事件。当然，不管是否对事件添加处理函数，该事件对象在发生时都会产生。

**公共组件**

在实际开发过程中，往往将具有一些统一逻辑 / 样式的组件提取出来，形成公共组件。在应用时，根据 Props 来控制其特别的地方。比如显示值、颜色甚至行为等等。当形成了一群公共组件时，便可以进一步封装成公共组件库，如：Ant Design、Material UI 等。

何时应该提取、如何封装、如何设计接口是体现一个工程师水平的最佳时刻，而这些操作原则，总结起来，就是：看情况，渐进的。

Props 既可以传递数值，也可以传递函数。

一般来说，函数传递使用在事件回调场景中。

不仅父组件可以通过 Props 改变子组件，子组件也可以通过事件回调，影响父组件。



**npm**

npm 是JavaScript 世界的包管理工具，并且是 Node.js 平台的默认包管理工具。通过 npm 可以安装、共享、分发代码，管理项目依赖关系。

**安装**

使用 yarn 安装 number-to-words

