* 部署
  + 使用npm start方式启动
  + 使用node.js部署
    - 执行npx react-app build
    - 在node中将”/”请求sendfile 至build 中的index.html
* 安装
  + 使用npx create-react-app {app名称} 构建单页应用模板
  + 或在html中静态添加script
  + npm的降级或则升级
    - 升级npm install –g npm
    - 降级npm install –g npm@版本号
  + 更新错误时清除npm缓存
    - npm cache clean --force
* 使用babel全新的编译环境
  + 需将script类型由 text/javascript 更改为 text/babel
  + 在babel中如需要改变样式表则需要使用className而不是class
  + 引用变量使用“{变量名}”大括号
* React功能
  + 通过代码建立对象
    - const element = createElement(‘类型’,{属性:’值’},标签内容);
    - const element = {type:’类型’,props:{属性:’值’,children:’标签内容’}};
  + 已经建立的组件是不能改变内容的，要改变内容必须使用React重新渲染或是使用状态的方式对动态改变的属性使用state保留
    - ReactDOM.render(element,getElementById(‘标签名称’)); 不推荐
    - 在constructor中保留state 推荐
      * this.state = {date:new Date()};
    - 设置state应使用setState({date:Date})方法，不能直接设置state
      * 不要把一些重要的数值放在state内部，state是异步执行的，某些数据在计算时如果存在上一次的参考数据，结果可能会不准确
      * 如果state中含有其他数据，也可以使用setState方法，state中的其他数据不会受到影响
  + 在属性中组合字符串
    - someprop={`btn ${variable}`}
  + Component 组件的事件
    - DOM中的事件命名是一致的，如onClick
    - 在component的构建器工作时绑定事件，以及在方法中的bind(因为要使用回调实现功能，所以必须加bind)
      * this.handleClick = this.handleClick.bind(this);
  + Component 组件的生命周期
    - componentDidMount 组件加载后执行的方法
    - componentWillUnmout 组件卸载后执行的方法