React

1. React 中 keys 的作⽤是什么？

标识在同层级元素的唯一性。

追踪元素被修改添加或者移除的辅助标识。

Diff算法中通过key来判断是新建的还是被移动的。减少不必要的元素重新渲染。

1. React事件机制：

将浏览器的原生事件封装成合成事件，

抹平浏览器的兼容性差异

事件委托，事件都代理到了document上，减少内存开销，同一个事件类型，最终只会保留一个。

存储事件委托，相同类型事件存在一个对象中，回调函数采用键值对。

1. 生命周期

Constructor：初始化

componentWillMount：组件挂载到DOM前调用

componentDidMount:：组件挂载到DOM后调用

componentWillReceiveProps(nextProps)：

shouldComponentUpdate(nextProps, nextState)：一些组件更新发生前的工作

componentWillUpdate(nextProps, nextState)

componentDidUpdate(prevProps, prevState) 组件更新后被调用

1. 为什么虚拟dom会提⾼性能
2. 节省开销
3. 缓存利用diff减少dom操作
4. Redux数据流

用户通过dispatch触发action，action将被送到一个叫Reducer的方法里面去，Reducer将会更新Store，Store中包含state。

1. redux，主要解决什么问题

组件通信和组件共享状态

1. redux和全局对象区别

redux的state改变会重新渲染视图

1. ref（受控组件和非受控组件）forwardref
2. PureComponent:会对传过来的props数据进行浅比较，props发生变化则重新渲染，不发生变化则不重新渲染
3. Context
4. Jsx 是一个 JavaScript 的语法扩展
5. State合并，异步同步

13．Hook使用

14. immutable reducer纯函数会返回一个新对象，immutable修改对象后会返回一个新对象，并复用之前的数据结构不浪费内存

15. diff先复用移动，新建和删除

HTML

1. Doctype作用：告诉浏览器该文件的类型。让浏览器解析器知道应该用哪个规范来解析。

严格模式下：页面排版及 JS 解析是以该浏览器支持的最高标准来执行

混杂模式：不严格按照标准执行，主要用来兼容旧的浏览器，向后兼容

DOCTYPE 不存在或格式不正确会导致文档以混杂模式呈现。

1. Src与href

Src指向外部资源的位置，指向的内容将会嵌入到文档中当前标签所在 位置

会停下来对后续文档的处理，直到src的内容加载完毕

href指向网络资源所在位置，建立和当前元素（锚点） 或当前文档（链接）之间的链接，在加载它的时候，不会停止对当前文档的处理，浏览器会继续往下走

CSS

1. BFC:触发条件

根元素

float不为none，

position不为static和relative，

overflow不为visible。

Display为inline-block table-cell、flex、table-caption或者inline-flex

特性：

同一个BFC容器中，相邻容器上下的margin会重叠。

计算BFC高度时浮动元素也参于计算（清除浮动）

BFC区域不会与浮动容器发生重叠。

垂直方向依次排列。

1. 居中的方式
2. 选择器优先级
3. 清除浮动
4. Link和@import区别

link是HTML标签，@import是css语法

兼容问题。

加载页面时，link引入的CSS被同时加载，@import引入的CSS将在页面加载完毕后加载；

1. Css动画
2. Css3新特性

2d变换，动画，过度，圆角，盒子阴影，flex，border-box，属性选择器，结构为类选择器（nth-child）

1. 回流和重绘：

回流一定导致重绘，重绘不一定导致回流。

重绘：只改变外观不影响布局

回流：改变尺寸大小，display隐藏，影响布局。

1. Img垂直水平居中

父容器text-align:center;

            vertical-align:middle;

            display:table-cell;