React专题

1. 什么是 React ？

React 是一个开源前端 JavaScript 库，用于构建用户界面，尤其是单页应用程序。它用于处理网页和移动应用程序的视图层。React 是由 Facebook 的软件工程师 Jordan Walke 创建的。在 2011 年 React 应用首次被部署到 Facebook 的信息流中，之后于 2012 年被应用到 Instagram 上。

1. React 的主要特点是什么?

考虑到真实的 DOM 操作成本很高，它使用 VirtualDOM 而不是真实的 DOM。

支持服务端渲染。

遵循单向数据流或数据绑定。

使用可复用/可组合的 UI 组件开发视图。

1. 什么是 JSX?

JSX 是 ECMAScript 一个类似 XML 的语法扩展。基本上，它只是为 React.createElement() 函数提供语法糖，从而让在我们在 JavaScript 中，使用类 HTML 模板的语法，进行页面描述。

在下面的示例中，<h1> 内的文本标签会作为 JavaScript 函数返回给渲染函数。

以上示例 render 方法中的 JSX 将会被转换为以下内容：

React.createElement("div", null, React.createElement(

"h1", null, 'Welcome to React world!'));

这里你可以访问 Babeljs 在线体验一下。

1. React 中的合成事件是什么?

SyntheticEvent 是对浏览器原生事件的跨浏览器包装。它的 API 与浏览器的原生事件相同，包括 stopPropagation() 和 preventDefault()，除了事件在所有浏览器中的工作方式相同。

1. 什么是 forward refs?

Ref forwarding 是一个特性，它允许一些组件获取接收到 ref 对象并将它进一步传递给子组件。

const ButtonElement = React.forwardRef((props, ref) => (

<button ref={ref} className="CustomButton">

{props.children}

</button>

));

// Create ref to the DOM button:

const ref = React.createRef();

<ButtonElement ref={ref}>{'Forward Ref'}</ButtonElement>

6、Shadow DOM 和 Virtual DOM 之间有什么区别？

Shadow DOM 是一种浏览器技术，它解决了构建网络应用的脆弱性问题。Shadow DOM 修复了 CSS 和 DOM。它在网络平台中引入作用域样式。 无需工具或命名约定，你即可使用原生 JavaScript 捆绑 CSS 和标记、隐藏实现详情以及编写独立的组件。Virtual DOM 是一个由 JavaScript 库在浏览器 API 之上实现的概念。

7、什么是 React Fiber？

Fiber 是 React v16 中新的 reconciliation 引擎，或核心算法的重新实现。React Fiber 的目标是提高对动画，布局，手势，暂停，中止或者重用任务的能力及为不同类型的更新分配优先级，及新的并发原语等领域的适用性。

8、createElement 和 cloneElement 有什么区别？

JSX 元素将被转换为 React.createElement() 函数来创建 React 元素，这些对象将用于表示 UI 对象。而 cloneElement 用于克隆元素并传递新的属性。

9、如何为高阶组件创建属性代理？

你可以使用属性代理模式向输入组件增加或编辑属性（props）：

function HOC(WrappedComponent) {

return class Test extends Component {

render() {

const newProps = {

title: 'New Header',

footer: false,

showFeatureX: false,

showFeatureY: true

};

return <WrappedComponent {...this.props} {...newProps} />

}

}

}

1. children 属性是什么？

Children 是一个属性（this.props.chldren），它允许你将组件作为数据传递给其他组件，就像你使用的任何其他组件一样。在组件的开始和结束标记之间放置的组件树将作为children属性传递给该组件。

React API 中有许多方法中提供了这个不透明数据结构的方法，包括：React.Children.map、React.Children.forEach、React.Children.count、React.Children.only、React.Children.toArray。

const MyDiv = React.createClass({

render: function() {

return <div>{this.props.children}</div>

}

})

ReactDOM.render(

<MyDiv>

<span>{'Hello'}</span>

<span>{'World'}</span>

</MyDiv>,

node

)

1. 怎样在 React 中写注释？

React/JSX 中的注释类似于 JavaScript 的多行注释，但是是用大括号括起来。

单行注释：

<div>

{/\* 单行注释（在原生 JavaScript 中，单行注释用双斜杠（//）表示） \*/}

{`Welcome ${user}, let's play React`}

</div>

多行注释：

<div>

{/\* 多行注释超过

一行 \*/}

{`Welcome ${user}, let's play React`}

</div>

1. 什么是调解？

当组件的props或state发生更改时，React 通过将新返回的元素与先前呈现的元素进行比较来确定是否需要实际的 DOM 更新。当它们不相等时，React 将更新 DOM 。此过程称为reconciliation。

1. 为什么有组件名称要首字母大写？

这是必要的，因为组件不是 DOM 元素，它们是构造函数。 此外，在 JSX 中，小写标记名称是指 HTML 元素，而不是组件。

1. React的局限性是什么？

React 只是一个视图库，而不是一个完整的框架。

对于 Web 开发初学者来说，有一个学习曲线。

将 React 集成到传统的 MVC 框架中需要一些额外的配置。

代码复杂性随着内联模板和 JSX 的增加而增加。

如果有太多的小组件可能增加项目的庞大和复杂。

1. 在 React 中事件有何不同？

React 事件处理程序是采用驼峰而不是小写来命名的。

使用 JSX，你将传递一个函数作为事件处理程序，而不是字符串。