04 React 世界中那些重要的概念

更新时间：2020-08-11 15:13:13

> 读一本好书，就是和许多高尚的人谈话。——歌德

前言

上一章提到 React 通过 ReactDOM.render 函数将应用程序首次渲染到屏幕，ReactDOM.render 函数中的两个重要的参数分别是 React 元素（element）和容器（container）。**事实上，开发者写的应用程序一般由众多「组件」组成，为什么在渲染时会以「元素」的形式传染到渲染函数中呢？**要想弄清楚这个问题，我们需要对 React 世界中的组件与元素进行深入了解。

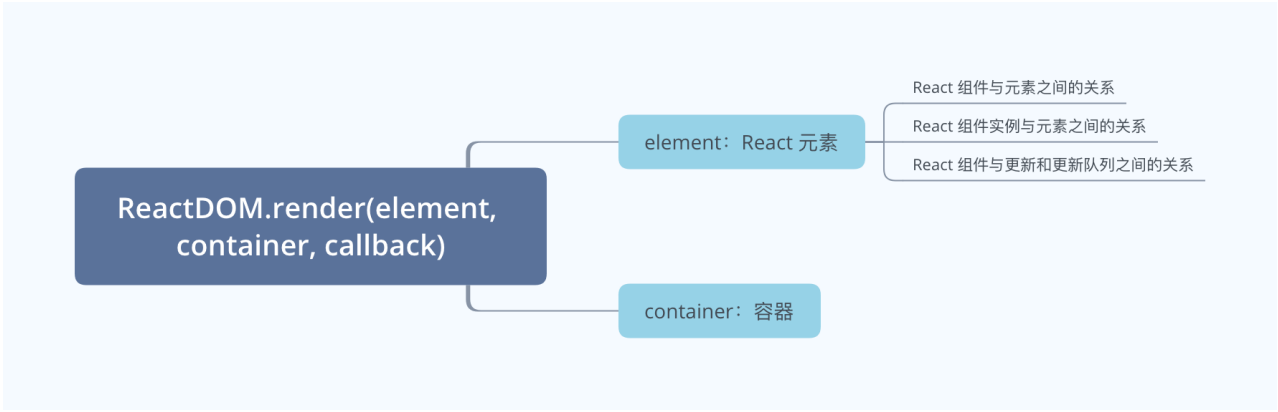

图 2.1.1 ReactDOM.render 函数的两个重要参数

通过执行 ReactDOM.render 函数 React 将应用程序首次渲染到屏幕。在这个渲染流程中，**React 需要先后经历组件转换元素、解析更新与更新队列、调用生命周期函数等操作**，在这些操作的背后离不开基础理论体系的支撑。

那么，在 React 世界中有哪些重要的基础概念呢？

重要的基础概念

- **组件**：可以是一个函数或一个类。
- **生命周期函数**：在应用程序运行的特定阶段执行对应的函数。
- **元素**：一个普通 JS 对象，由组件转化而来，在运行时再转化为 React fiber 对象。
- **组件实例**：类组件实例化后的对象，主要作用是返回组件元素、响应事件等。
- **更新（update）**：一个 JS 对象，包含过期时间（expirationTime）和更新内容（payload）等。
- **更新队列（updateQueue）**：一个 JS 对象，是更新（update）的集合，链表结构。
- **React Fiber**：React v16 版本开始引入的架构。

这些重要的基础概念之间有什么关系呢？ 见图 2.1.2。

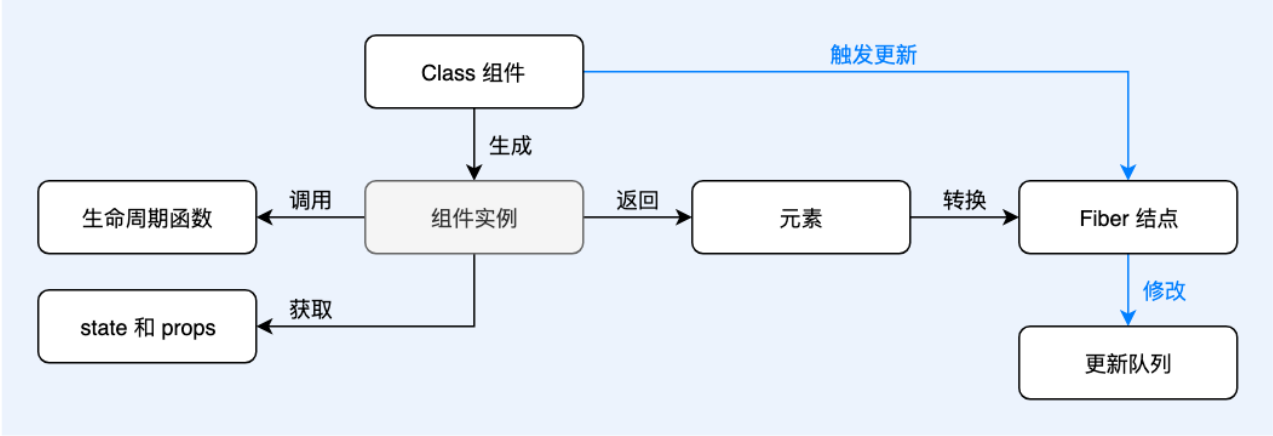

图 2.1.2 React 世界中重要基础概念之间的关系

React 组件一般由开发者定义，应用程序运行时 React 为 class 类型的组件创建组件实例。在应用程序渲染的不同阶段 React 会执行组件实例上面对应的生命周期函数以及获取其 state 和 props。此外，React 通过组件实例调用组件的 render 函数获取到该组件内部的元素。在 render 阶段 React 将元素逐个转换为对应类型的 Fiber 结点（最终形成 workInProgress 树）。

当组件内部有 this.setState(...) 操作时，React 首先根据 this 对象找到对应的 Fiber 结点，然后将更新加入到当前 Fiber 结点的更新队列。

> 注：上面组件生成组件实例，组件实例返回元素等逻辑在后面会进行详细介绍。

你真正理解 React 组件吗？下一节将会详细介绍 React 组件的设计思想。

}