

1. 什么是DOM？

DOM (document object model) 指的是文档对象模型，它指的是把文档当做一个对象来对待，这个对象主要定义了处理网页内容的方法和接口。DOM 使我们可以像操作一个对象一样操作一个网页。对象通常具有属性 (Attribute) 和方法 (Method)，DOM对象也具有属性和方法，如可以通过 `document.getElementById(nodeld)` 获取网页中的某个节点，通过`document.charset` 获取当前页面的字符集。

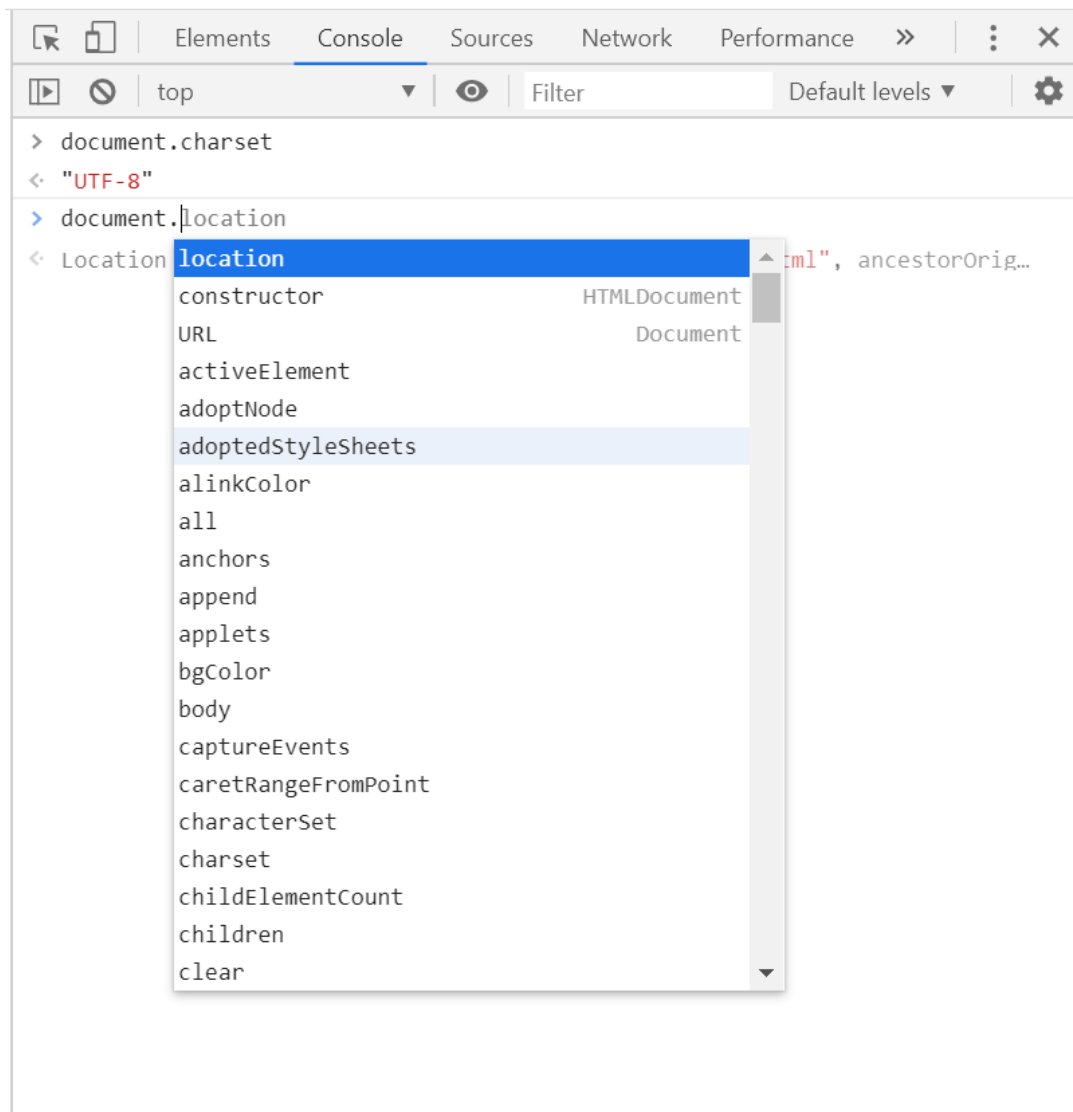


图1 Chrome 浏览器控制台中可以查看document 对象的属性和方法

2. 什么是虚拟 DOM？

在没有虚拟DOM 的时候，我们只能通过 JavaScript 手动操作 DOM 修改节点，更新页面。在页面较为简单的时候这种方式还可以接受，当应用程序变复杂之后，用JavaScript 手动操作 DOM 就越来越复杂。

在应用程序越来越复杂的情况下，为了避免 通过 JavaScript 手动操作 DOM，我们需要一种能够维护视图 (view) 和数据 (model) 一致的解决方案。

对于这个问题的一种解决方案就是 MVVM，MVVM 是一种软件架构设计模式，通过分离关注点来组织代码结构，优化开发效率。MVVM 中的 VM 指的是 ViewModel，通过双向数据绑定将 View 和 Model 的同步更新自动化。当 Model 发生变化时，ViewModel 自动更新，ViewModel 更新后，View 也会更新。一个例子就是 Vue，Vue 通过数据劫持和发布订阅者模式来实现了双向数据绑定。

而另一种解决方案就是 虚拟 DOM，Virtual DOM 是一个基本的 JavaScript 对象，通过 Virtual DOM 我们能够创建出真实的 DOM。首先要创建一个 JavaScript 对象作为虚拟 DOM，当数据 (Model) 发生变化时，我们先修改虚拟 DOM，然后比较虚拟 DOM 树和 真实 DOM 树的差异，将两棵树差异记录在差异对象中，最后一步，根据差异对象修改 真实 DOM 树。

使用 虚拟 DOM 的方式维护 视图 和 数据的同步有以下好处：

- 实现数据和视图分离，增加可维护性。
- 集中化 DOM 操作，用最小的代价更新 DOM，提高效率。
- 打开了函数式 UI 编程的大门
- 可以渲染到 DOM 以外的端，比如 ReactNative

为什么 虚拟 DOM 这种 方式能够提高效率呢？直接使用 修改 innerHTML 不是更直接吗？

从修改 DOM 到页面更新的过程中，有许多因素影响其中的效率。如果不采用虚拟 DOM 的方式，而是直接更新 innerHTML，在页面中有微小变化的时候计算开销就包含渲染整个页面的 HTML 和重新创建所有 DOM。在有 虚拟 DOM 的情况下，只需要渲染虚拟 DOM，Diff 两棵 DOM 树的差异和 必要的 DOM 更新。其中渲染虚拟 DOM 和 Diff 算法都是 JavaScript 完成的，而 JavaScript 计算要比 DOM 操作快的多。这就是虚拟 DOM 的长处了，无论数据变化的量是大是小，每次重绘的性能都是可以接受的。

3. 虚拟 DOM 的 Diff 算法？

DOM 树其实就是一棵 n 叉树，比较两棵树的差异其实就比较 两棵 n 叉树的差异。由于在前端的业务场景中很少有跨层级的 DOM 操作，所以我们只需要比较同一层的 DOM 树节点之间的差异就够了。首先对两棵树进行一次深度优先遍历（类似于二叉树的先序遍历）给每个节点一个唯一的标记（如 0,1,2,3...）。在遍历的过程中，每遍历到一个节点就比较该节点和虚拟 DOM 树中对应节点，如果有差异的话就记录到另一个差异对象中。差异对象的内容包含四种，分别是：1) 替换节点，2) 增加/删除节点，3) 修改节点属性，4) 改变文本内容。对应四类 DOM 操作方法，分别是：1) replaceChild, 2) appendChild, removeChild, 3) setAttribute, removeAttribute, 4).textContent。

4. 循环渲染时设置 key 的作用？

这涉及到列表的重绘优化机制，也就是提示框架如何有效地复用实例和 DOM 元素。比如说数据库中的一个对象在变化前后的两次请求中被请求到前端，这个对象的 id 是没有变化的，也就是说这是这两个对象其实是同一份数据，那么这份数据在前端对应的实例和 DOM 元素都是可以复用的，只需要变更变动了的部分。

虚拟 DOM 算法在对列表元素进行比较的时候，需要一个属性来比较两个列表的 DOM 结构之间的差异，而列表元素的 tagName 都是相同的 (li)，而这个属性就是我们在循环渲染时对子节点设置的 key。有了这个 key 才能进行两个列表的 DOM 结构的比较，才能复用旧的 DOM 中列表元素。

5. react 中 shouldComponentUpdate 应该怎么用？

在 react 中，修改了数据，调用 `setState` 方法后，组件的 `render` 方法会自动调用，而且父组件会嵌套渲染子组件，哪怕子组件中并没有数据修改。而 `shouldComponentUpdate` 可以用来避免这种多余的操作，也就是说可以提升小量数据更新时的性能。`shouldComponentUpdate` 是一个组件的方法，可以拦截组件渲染。常见的用法是在 `shouldComponentUpdate` 中比较该组件的 `props` 和 `state` 是否有变化，若没有变化返回 `false` 避免不必要的渲染。

参考文献：

- [1] <https://segmentfault.com/a/1190000018549047>
- [2] <https://segmentfault.com/a/1190000016494335>
- [3] <https://www.zhihu.com/question/31809713>
- [4] <https://github.com/y8n/blog/issues/5>