**1、什么是虚拟 DOM （vdom）**

* **起源**  
  虚拟dom最先是由facebook团队提出的，最先运用在react中，之后在vue2.0版本中引入了虚拟DOM的概念
* **本质是什么**
* Vnode的本质就是用**树型结构的JS对象**来描述真实的DOM结构的信息，这个树结构的JS对象包含了整个DOM结构的信息.。  
    
  可见左边的DOM结构，不论是标签名称还是标签的属性或标签的子集，都会对应在右边的树结构里。

**2、虚拟 DOM 的优缺点**

**优点：**

* 降低浏览器性能消耗  
  因为**Javascript的运算速度远大于DOM操作的执行速度**，因此，运用patching算法来计算出真正需要更新的节点，最大限度地减少DOM操作，从而提高性能。

**在vnode技术出现之前**，我们要改变页面展示的内容**只能通过遍历查询 dom 树**的方式找到需要修改的 dom，然后修改样式行为或者结构，来达到**更新 ui** 的目的。这种方式**相当消耗计算资源**，因为每次查询 dom 几乎都需要遍历整颗 dom树。

**在vnode技术出现之后**，我们建立一个**虚拟 dom 对象来对应真实的 dom 树**，那么**每次 dom 的更改就变成了 js 对象的属性的更改** ，这样一来就能查找 js 对象的属性变化要比查询 dom 树的 性能开销小。

* diff算法,减少回流和重绘  
  通过diff算法，优化遍历，对真实dom进行打补丁式的新增、修改、删除，实现局部更新，减少回流和重绘。

vnode优化性能核心思想，就是每次**更新 dom 都尽量避免刷新整个页面**，而是有针对性的 去**刷新那被更改的一部分** ，来释放掉被无效渲染占用的 gpu，cup性能。同时，也减少了大量的dom操作，减少了浏览器的回流和重绘。

* 无需手动操作 DOM  
  我们不再需要手动去操作 DOM，框架会根据虚拟 DOM 和 数据双向绑定，帮我们以可预期的方式更新视图，极大提高我们的开发效率；
* 跨平台  
  **虚拟 DOM 本质上是 JavaScript 对象**，而 DOM 与平台强相关，相比之下虚拟 DOM ，可以进行更方便地跨平台操作，例如：服务器渲染、weex 开发等等

**缺点**

* 首次显示要慢些:  
  首次渲染大量DOM时，由于多了一层虚拟DOM的计算, 会比innerHTML插入慢
* 无法进行极致优化：  
  虽然虚拟 DOM + 合理的优化，足以应对绝大部分应用的性能需求，但在一些性能要求极高的应用中 无法进行针对性的极致优化。

**3、angular、vue、react，更新vnode对象的原理**

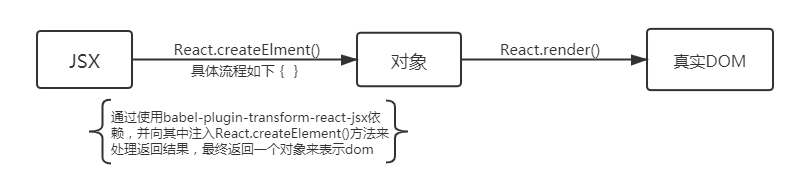
* **angular**  
  angular 采用的机制是 **脏值检测机制**， 所有使用了 ng 指令的data 和 {{}} 语法的data 都会被加入脏检测的队列，以监听数据的变化。
* **vue**  
  vue 采用的是**Object.defineProperty，重写 对象的setter ， getter方法**以实现监听 data 属性的变化**生成新的虚拟 dom**， **通过 h 函数**创建真实 dom 替换掉dom树上对应的旧 dom。
* **react**  
  react 也是**通过setState方法更改 data 以生成新的虚拟 dom** ，再**通过 diff 算法**来计算和生成需要替换的 dom 做到**局部更新**的。

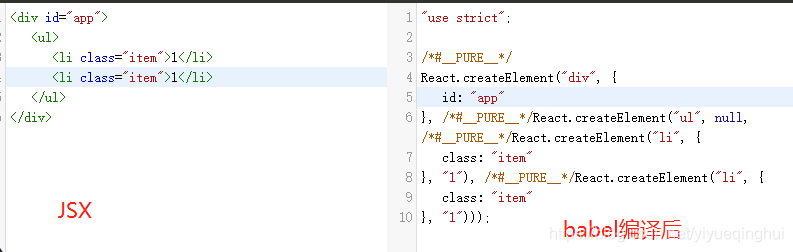
**4、简述虚拟 DOM 实现原理**

虚拟 DOM 的简单实现原理主要包括以下 3 部分：

1. 用 JavaScript 对象模拟真实 DOM 树，对真实 DOM 进行抽象；
2. 通过diff 算法 — 比较两棵虚拟 DOM 树的差异；
3. 根据差异，对真正的 DOM 树进行增、删、改。

**5、虚拟DOM的逻辑（React虚拟dom到真实dom发生了什么）**





**6，React16 更新了什么**

React16中引入了 Fiber 这个概念，从根本上解决了由于js的单线程运行，遇到计算量较大的情况，导致动画和交互卡顿的问题。

**Fiber**的核心是实现了一个**基于优先级和requestIdleCallback的循环任务调度算法** 。它可以把一个任务拆分成多个小任务，并且可随时中止或恢复任务，同时也可以根据优先级不同来选择优先执行任务

参考链接 https://www.jianshu.com/p/4db4b191de06