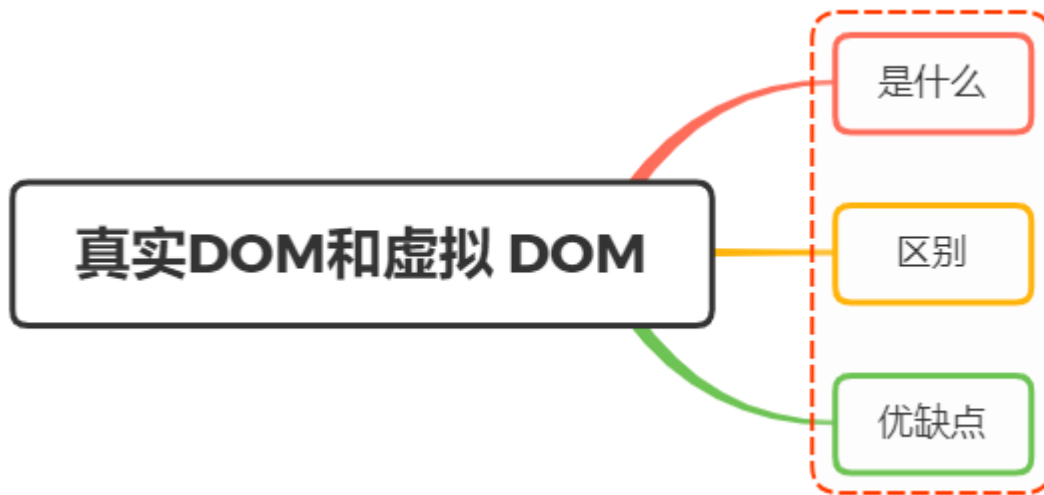


面试官：说说 Real DOM和 Virtual DOM 的区别？优缺点？



一、是什么

Real DOM · 真实DOM，意思为文档对象模型，是一个结构化文本的抽象，在页面渲染出的每一个结点都是一个真实DOM结构，如下：

```
<div id="root">
  <h1>Hello World</h1>
</div>
```

Virtual Dom · 本质上是以 JavaScript 对象形式存在的对 DOM 的描述

创建虚拟DOM目的就是为了更好将虚拟的节点渲染到页面视图中，虚拟DOM对象的节点与真实DOM的属性一一照应

在React中，JSX是其一大特性，可以让你在JS中通过使用XML的方式去直接声明界面的DOM结构

```
const vDom = <h1>Hello World</h1> // 创建h1标签，右边千万不能加引号
const root = document.getElementById('root') // 找到<div id="root"></div>节点
ReactDOM.render(vDom, root) // 把创建的h1标签渲染到root节点上
```

上述中，ReactDOM.render()用于将你创建好的虚拟DOM节点插入到某个真实节点上，并渲染到页面上

JSX实际是一种语法糖，在使用过程中会被babel进行编译转化成JS代码，上述VDOM转化为如下：

```
const vDom = React.createElement(
  'h1',
  { className: 'hClass', id: 'hId' },
  'hello world'
)
```

可以看到，**JSX**就是为了简化直接调用`React.createElement()`方法：

- 第一个参数是标签名，例如h1、span、table...
- 第二个参数是个对象，里面存着标签的一些属性，例如id、class等

第三个参数是节点中的文本

通过`console.log(VDOM)`，则能够得到虚拟VDOM消息

```
▼ Object ⓘ
  $$typeof: Symbol(react.element)
  key: null
  ▶ props: {children: "Hello World"}
  ref: null
  type: "h1"
  _owner: null
  ▶ _store: {validated: false}
  _self: undefined
  ▶ _source: {fileName: "E:\\Users\\u
  ▶ __proto__: Object
```

所以可以得到，**JSX**通过**babel**的方式转化成`React.createElement`执行，返回值是一个对象，也就是虚拟DOM

二、区别

两者的区别如下：

- 虚拟DOM不会进行排版与重绘操作，而真实DOM会频繁重排与重绘
- 虚拟DOM的总损耗是“虚拟DOM增删改+真实DOM差异增删改+排版与重绘”，真实DOM的总损耗是“真实DOM完全增删改+排版与重绘”

拿[以前文章](#)举过的例子：

传统的原生**api**或**jQuery**去操作DOM时，浏览器会从构建DOM树开始从头到尾执行一遍流程

当你在一次操作时，需要更新10个DOM节点，浏览器没这么智能，收到第一个更新DOM请求后，并不知道后续还有9次更新操作，因此会马上执行流程，最终执行10次流程

而通过**VNode**，同样更新10个DOM节点，虚拟DOM不会立即操作DOM，而是将这10次更新的**diff**内容保存到本地的一个**js**对象中，最终将这个**js**对象一次性**attach**到DOM树上，避免大量的无谓计算

三、优缺点

真实DOM的优势：

- 易用

缺点：

- 效率低，解析速度慢，内存占用量过高

- 性能差：频繁操作真实DOM，易于导致重绘与回流

使用虚拟DOM的优势如下：

- 简单方便：如果使用手动操作真实DOM来完成页面，繁琐又容易出错，在大规模应用下维护起来也很困难
- 性能方面：使用Virtual DOM，能够有效避免真实DOM数频繁更新，减少多次引起重绘与回流，提高性能
- 跨平台：React借助虚拟DOM，带来了跨平台的能力，一套代码多端运行

缺点：

- 在一些性能要求极高的应用中虚拟 DOM 无法进行针对性的极致优化
- 首次渲染大量DOM时，由于多了一层虚拟DOM的计算，速度比正常稍慢

参考文献

- <https://juejin.cn/post/6844904052971536391>
- <https://www.html.cn/qa/other/22832.html>