

目录	
第1章 优化的概念	
1 开篇词：你的前端性能还能再抢救一下	
2 解读雅虎35条军规（上）	
3 解读雅虎35条军规（下）	
4 你要不要看这些优化指标？	
第2章 性能工具介绍	
5 性能优化百宝箱（上）	
6 性能优化百宝箱（下）	
第3章 网络部分	
7 聊聊 DNS Prefetch	
8 Webpack 性能优化两三事	
9 图片加载优化（上）	
10 图片加载优化（下）	
第4章 缓存部分	
11 十八般缓存	
12 CDN 缓存	
13 本地缓存（Web Storage）	
14 浏览器缓存（上）	
15 浏览器缓存（下）	
第5章 渲染部分	
16 渲染原理与性能优化	
17 如何应对首屏“一片空白”（上）	
18 如何应对首屏“一片空白”（下）	
19 不容小觑的 DOM 性能优化	

# 19 不容小觑的 DOM 性能优化

更新时间：2019-08-30 10:07:32



“学习要注意到细处，不是粗枝大叶的，这样可以逐步学习、摸索，找到客观规律。”  
—— 徐特立

提起 DOM 很多人都认为这是一个不怎么重要的概念，与他相关的很多东西也都选择性忽略掉了，但其实与 DOM 有关的性能优化在前端开发工作当中起着非常重要的作用。DOM 是将 Web 页面与脚本或者其他编程语言连接起来的桥梁。这里的语言当然就是我们天天在使用的 JavaScript。

DOM 也是 Web 页面实现面向对象编程的中介。我们的网页都是由 HTML 元素构成页面的基本结构，每一个 HTML 元素都是一个 DOM 节点，一个一个的 DOM 节点就构成了我们平时所说的 DOM 树，正所谓万事万物皆为对象，所以我们的每个 DOM 节点，甚至是整个 DOM 树都可以看成对象，我们通过使用 JavaScript 就可以操作这些 DOM 节点，实现丰富的交互功能，这也符合面向对象的编程思想。

过去我们没有 React、Vue 这些框架的时候，我们实现复杂的交互，需要获取对应元素，然后在其上进行各种操作，但其实操作 DOM 是非常耗费性能的，仅仅是一个节点它下面的属性就非常多，这里我们以 div 为例打印，如下图：



目录	一些优化工作势在必行。
第1章 优化的概念	关于DOM的优化工作
1 开篇词：你的前端性能还能再抢救一下	在日常开发当中，不管我们是否使用框架，书写 HTML 是不可避免的，虽然 React 是 JSX 语法，但是其本质也是HTML。我们在书写 HTML 标签的时候，一定要注意语义化，这样不仅有利于浏览器解析时间，而且对搜索引擎的抓取也是十分友好的；尽量减少无用的标签，我们在做网页布局的时候，一定要注意不要嵌套多个无用的 div，对一下没有用的div要及时删除；涉及到一些样式能用 CSS 实现的，尽量不要使用相关的 HTML 元素，如：我们需要添加水平线就不要使用 <code>&lt;hr /&gt;</code> ，我们可以直接使用 CSS 的 border 属性就可实现。
2 解读雅虎35条军规（上）	
3 解读雅虎35条军规（下）	
4 你要不要看这些优化指标？	<div>Tips:如果你使用的 React 框架，遇到需要必须要一个父元素包裹的情况，我们可以用 <code>&lt;Fragment&gt;</code> 标签来实现，它不会在 DOM 树种创建对应的节点，是个不错的选择。</div>
第2章 性能工具介绍	我们常常会有如下场景，我们开发的整个网页的导航栏，通常都是使用 <code>&lt;li&gt;</code> 标签来代表每个具体的菜单，然后一般都需要为每个菜单都添加点击事件，如果不仔细思考，很容易会写出如下代码：
5 性能优化百宝箱（上）	<pre>let container = document.getElementById('container'); let wrapper = document.createDocumentFragment for(let i = 0; i &lt; data.length; i++){   let li = document.createElement('li');   li.innerText = data[i];   li.addEventListener('click', function(){     // 触发click后要做的的事情   },false)   wrapper.appendChild(li); } container.appendChild(wrapper);</pre>
6 性能优化百宝箱（下）	这里我们假设 data 数据有一万条，那么就需要执行一万次监听事件，这对性能的影响是非常大的，所以这里我们完全可以使用事件代理机制来解决这个问题，如下：
第3章 网络部分	<pre>let container = document.getElementById('container'); let wrapper = document.createDocumentFragment for(let i = 0; i &lt; data.length; i++){   let li = document.createElement('li');   li.innerText = data[i]; } wrapper.appendChild(li); container.appendChild(wrapper); container.addEventListener('click', function(e){   target = e.target    e.srcElement;   if(target.tagName.toLowerCase() == 'li'){     // 触发click后要做的的事情   } }, false)</pre>
7 聊聊 DNS Prefetch	
8 Webpack 性能优化两三事	
9 图片加载优化（上）	
10 图片加载优化（下）	
第4章 缓存部分	
11 十八般缓存	
12 CDN 缓存	
13 本地缓存（Web Storage）	
14 浏览器缓存（上）	
15 浏览器缓存（下）	
第5章 渲染部分	
16 渲染原理与性能优化	
17 如何应对首屏“一片空白”（上）	
18 如何应对首屏“一片空白”（下）	

← 慕课专栏

≡ 你不知道的前端性能优化技巧 / 19 不容小觑的 DOM 性能优化

目录

第1章 优化的概念

1 开篇词：你的前端性能还能再抢救一下

2 解读雅虎35条军规（上）

3 解读雅虎35条军规（下）

4 你要不要看这些优化指标？

第2章 性能工具介绍

5 性能优化百宝箱（上）

6 性能优化百宝箱（下）

第3章 网络部分

7 聊聊 DNS Prefetch

8 Webpack 性能优化两三事

9 图片加载优化（上）

10 图片加载优化（下）

第4章 缓存部分

11 十八般缓存

12 CDN 缓存

13 本地缓存（Web Storage）

14 浏览器缓存（上）

15 浏览器缓存（下）

第5章 渲染部分

16 渲染原理与性能优化

17 如何应对首屏“一片空白”（上）

18 如何应对首屏“一片空白”（下）

DocumentFragments 是DOM节点。它们不是主DOM树的一部分。通常的用例是创建文档片段，将元素附加到文档片段，然后将文档片段附加到DOM树。在DOM树中，文档片段被其所有的子元素所代替。

因为文档片段存在于**内存中**，并不在DOM树中，所以将子元素插入到文档片段时不会引起页面**回流**（对元素位置和几何上的计算）。因此，使用文档片段通常会带来更好的性能。

可见这个方法真是处理批量标签的一大利器，使用它可以节省下很多时间，提高 DOM 性能。

一定要将循环当中没必要的 DOM 操作放在外面，同理这里我们还是先看代码，如下：

```
for(let i = 0; i < data.length; i++){
  let li = document.createElement('li');
  li.innerText = data[i];
  document.getElementById('container').appendChild(li);
}
```

在上面代码的循环当中，我们每次都去获取 container 元素，但是其实它从头到尾都没有变化，我们完全可以把它放在外面处理，其实这种问题并不难避免，我们在开发当中多留心即可。

DOM 性能优化还有一个重要的方面，就是重绘与回流，这个我们放到了下个小节，所以关于重绘与回流我们可以把它们也归结为 DOM 性能优化部分。

小结

上面我们介绍了一些 DOM 性能优化的方法，其实随着前端的发展，我们的开发工作也慢慢从写原生 JavaScript 逐步过渡到使用各种前端框架，比如现下的三种框架：React，Vue，Angular，框架的引入使我们在开发当中无需在操作 DOM，而把操作这种复杂的工作交给框架去做，我们只需要关注状态即可，这是前端发展具有里程碑意义的一步，虽然说操作 DOM 在今后会越来越来少，但是了解上面这些 DOM 的优化方法，对我们使用框架开发也是具有非常重要的意义。

← 18 如何应对首屏“一片空白”（下）

20 重绘与回流的爱恨相杀 →

精选留言 0

欢迎在这里发表留言，作者筛选后可公开显示

!

目前暂无任何讨论

19 不容小觑的 DOM 性能优化 目录

www.imooc.com/read/41/article/632

3/4

← 慕课专栏	☰ 你不知道的前端性能优化技巧 / 19 不容小觑的 DOM 性能优化
目录	
第1章 优化的概念	千学不如一看，千看不如一练
1 开篇词：你的前端性能还能再抢救一下	
2 解读雅虎35条军规（上）	
3 解读雅虎35条军规（下）	
4 你要不要看这些优化指标？	
第2章 性能工具介绍	
5 性能优化百宝箱（上）	
6 性能优化百宝箱（下）	
第3章 网络部分	
7 聊聊 DNS Prefetch	
8 Webpack 性能优化两三事	
9 图片加载优化（上）	
10 图片加载优化（下）	
第4章 缓存部分	
11 十八般缓存	
12 CDN 缓存	
13 本地缓存（Web Storage）	
14 浏览器缓存（上）	
15 浏览器缓存（下）	
第5章 渲染部分	
16 渲染原理与性能优化	
17 如何应对首屏“一片空白”（上）	
18 如何应对首屏“一片空白”（下）	