

26 命题难点深入挖掘—重绘与重排

更新时间：2020-09-03 09:43:00



人生的价值，并不是用时间，而是用深度去衡量的。——列夫·托尔斯泰

导读

学到这里，如果你能够确认以上内容已经充分掌握，那么相信你已经有实力去应对大多数浏览器相关的面试题了。

本节和下节内容，各位根据自己的实际情况来安排学习优先级：若是目前还处于知识点扫盲阶段，建议同学直接跳过本章去读后续专题、确保优先完成知识盲区的清除工作；若是对自己的知识网络已经有充分的把握，同时对大厂、对好团队抱有绝对的信念，那么不妨耐下心来把下面的内容读完。

接下来我们要聊的是渲染知识板块中的面试“老大难”三兄弟——重排、重绘与图层。

重排与重绘——渲染性能问题重要抓手

通过前面的学习，我们已经知道，只有经过HTML解析->CSS解析->渲染树构建->布局->绘制这样一个漫长的过程，页面才能够得以呈现在我们面前，不过这还不是故事的全部——在页面初始化完成后，我们可能会通过CSS、JS来对页面中的元素进行修改，这些修改会重新触发页面的一部分生命周期，进而带来性能上的开销。

重走页面生命周期的这个过程，有两种主要的形式——重排与重绘。

什么是重排？

当我们的操作引发了 **DOM** 几何尺寸的变化（比如修改元素的宽、高或隐藏元素等）时，浏览器需要重新计算元素的几何属性（其他元素的几何属性和位置也会因此受到影响），然后再将计算的结果绘制出来。这个过程就是重排（也叫回流）。

简而言之，重排多数情况下是由对元素几何属性的修改引发的。

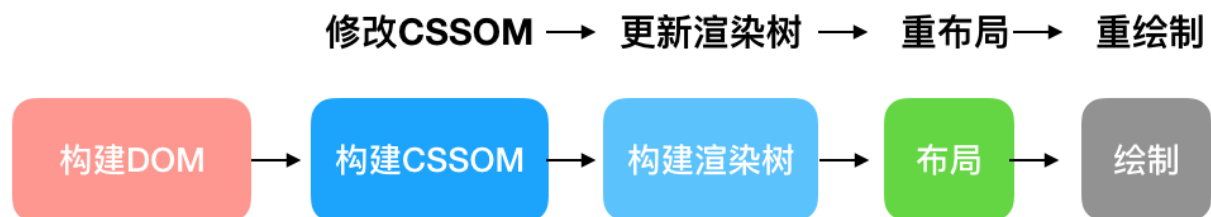
比如说这种操作：

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>重排demo</title>
  <style>
    #target {
      width: 100px;
      height: 100px;
    }
  </style>
</head>
<body>
  <div id="target">
    <span id="targetText">我是一个小测试</span>
  </div>
</body>
</html>
```

```
var targetDom = document.getElementById('target');
targetDom.style.width = '200px';
```

这样的一个修改宽度的动作，不仅影响了盒模型本身的“占地面积”、还会整个页面中其它元素的布局形式。浏览器不得不重新针对布局信息进行计算，这就是典型的重排过程。

当重排发生时，“重新来过”的流程如下图所示：



面试官问：还有什么动作会触发重排？

注意，重排多数情况下是由对元素几何属性的修改引发的，但不总是由此引发的。这个知识点很多同学都不知道，也是面试官区分新手和老手的关键。大家谨记，以下操作也会触发重排：

- 改变 **DOM** 树的结构

这里主要指的是节点的增减、移动等操作。

- 获取一些特定属性的值（重要）

如offsetTop、offsetLeft、offsetWidth、offsetHeight、scrollTop、scrollLeft、scrollWidth、scrollHeight、clientTop、clientLeft、clientWidth、clientHeight 等属性（挑几个背下来，答题的时候不要哑了）。

这些属性有一个共性，就是需要通过**即时计算**得到。因此浏览器为了获取这些值，也会进行回流。

除此之外，当我们调用了 `getComputedStyle` 方法，或者 IE 里的 `currentStyle` 时，也会触发回流。原理是一样的，都为求一个“即时性”和“准确性”。

什么是重绘？

当我们对 DOM 的修改导致了样式的变化、却并未影响其几何属性（比如修改了颜色或背景色）时，浏览器不需重新计算元素的几何属性、直接为该元素绘制新的样式（跳过了上图所示的回流环节）。这个过程叫做重绘。

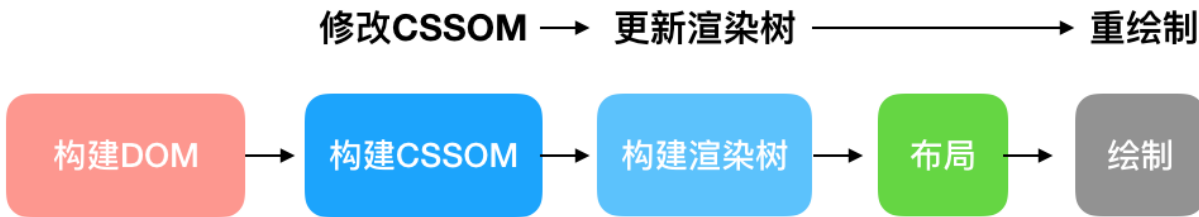
简而言之，重绘是由对元素绘制属性的修改引发的。

举个例子，咱们仍然沿用上文的demo，假如咱们修改了 `targetText` 的颜色：

```
var targetDom = document.getElementById('targetText');
targetDom.style.color = 'red';
```

浏览器要想改变文字的颜色，只需要去变更像素点的色值即可，不涉及任何布局计算。因此，这是一个典型的重绘过程。

当重绘发生时，“重新来过”的流程如下图所示：



命题思路点拨

“针对页面渲染流程中的性能优化，你有什么思路？”——当面试官抛出这个问题时，不管你对性能优化掌握到什么程度、对渲染原理是否自信，你盲答一波重排与重绘，至少说明你不是一个彻头彻尾的小白。

对于重排和重绘，考官除了喜欢考察它们本身的含义、触发的机制外，还热衷于对两者进行对比。重排与重绘有哪些相同点、哪些不同点，相信经过我们上面的讲解，你已经心知肚明。这个问题，希望每一位同学课下都整理出一份属于自己的答案，以备不时之需。

}