

css 页面重绘和回流(重排)以及优化

一、html页面的呈现流程

1. 浏览器把获取到的HTML代码解析成1个DOM树，HTML中的每个tag都是DOM树中的1个节点，根节点就是我们常用的document对象。DOM树里包含了所有HTML标签，包括display:none隐藏，还有用JS动态添加的元素等。
2. 浏览器把所有样式(用户定义的CSS和用户代理)解析成样式结构体，在解析的过程中会去掉浏览器不能识别的样式，比如IE会去掉-moz开头的样式，而FF会去掉_开头的样式。
3. DOM Tree 和样式结构体组合后构建render tree, render tree类似于DOM tree，但区别很大，render tree能识别样式，render tree中每个NODE都有自己的style，而且 **render tree不包含隐藏的节点** (比如display:none的节点，还有head节点)，因为这些节点不会用于呈现，而且不会影响呈现的，所以就不会包含到 render tree中。注意 visibility:hidden隐藏的元素还是会包含到 render tree中的，因为visibility:hidden 会影响布局(layout)，会占有空间。根据CSS2的标准，render tree中的每个节点都称为Box (Box dimensions)，理解页面元素为一个具有填充、边距、边框和位置的盒子。
4. 一旦render tree构建完毕后，浏览器就可以根据render tree来绘制页面了。

二、回流与重绘

1. 当render tree中的一部分(或全部)因为元素的规模尺寸，布局，隐藏等改变而需要重新构建。这就称为回流(reflow)。每个页面至少需要一次回流，就是在页面第一次加载的时候。在回流的时候，浏览器会使渲染树中受到影响的部分失效，并重新构造这部分渲染树，完成回流后，浏览器会重新绘制受影响的部分到屏幕中，该过程成为重绘。

2. 当render tree中的一些元素需要更新属性，而这些属性只是影响元素的外观，风格，而不会影响布局的，比如background-color。则就叫称为重绘。

注意：回流必将引起重绘，而重绘不一定会引起回流。

回流何时发生：

当页面布局和几何属性改变时就需要回流。下述情况会发生浏览器回流：

- 1、添加或者删除可见的DOM元素；
- 2、元素位置改变；
- 3、元素尺寸改变——边距、填充、边框、宽度和高度
- 4、内容改变——比如文本改变或者图片大小改变而引起的计算值宽度和高度改变；

- 5、页面渲染初始化；
- 6、浏览器窗口尺寸改变——resize事件发生时；

三、优化（减少回流、重绘）

浏览器本身的优化策略：浏览器会维护1个队列，把所有会引起回流、重绘的操作放入这个队列，等队列中的操作**到了一定的数量或者到了一定的时间间隔**，浏览器就会flush队列，进行一个批处理。这样就会让多次的回流、重绘变成一次回流重绘。但有时候我们写的一些代码可能会强制浏览器提前flush队列，这样浏览器的优化可能就起不到作用了。当你请求向浏览器请求一些 style信息的时候，就会让浏览器flush队列。

减少对render tree的操作（合并多次多DOM和样式的修改），并减少对一些style信息的请求，尽量利用好浏览器的优化策略

方法：

1. 将多次改变样式属性的操作合并成一次操作。
2. 将需要多次重排的元素，position属性设为absolute或fixed，这样此元素就脱离了文档流，它的变化不会影响到其他元素。例如有动画效果的元素就最好设置为绝对定位。
3. 在内存中多次操作节点，完成后再添加到文档中去。例如要异步获取表格数据，渲染到页面。可以先取得数据后在内存中构建整个表格的html片段，再一次性添加到文档中去，而不是循环添加每一行。
4. 由于display属性为none的元素不在渲染树中，对隐藏的元素操作不会引发其他元素的重排。如果要对一个元素进行复杂的操作时，可以先隐藏它，操作完成后再显示。这样只在隐藏和显示时触发2次重排。
5. 在需要经常取那些引起浏览器重排的属性值时，要缓存到变量。