

HTML页面的重绘 (repaint) 和重流 (reflow)

重流 (Reflow) 是指布局引擎为frame计算图形的过程。frame是一个矩形，拥有宽高和相对父容器的偏移。frame用来显示盒模型 (content model)，但一个content model可能会显示为多个frame，比如换行的文本每行都会显示为一个frame。

重绘 (Repaint) 发生在元素的可见性发生变化时，比如背景色、前景色等。因此回流必然会引起重绘。

重流的原因：

- 初始化 (Initial)。DOM载入后的第一次回流，将会遍历所有frame。
- 渐进 (Incremental)。当一个frame发生渐进回流时，意味着它前面的元素都没有变，而是它里面的元素变了。这会引起自底向上的作用。
- 改变大小 (Resize)。元素的容器边界发生变化时，此时元素内部状态没变。在计算自顶向下的布局约束的同时，可以复用内部状态。
- 样式改变 (StyleChange)。整个frame树都应得到遍历。
- Dirty。当一个容器已经缓存了多个子元素的Incremental回流时，该容器出于Dirty的状态。

如果你是Web开发者，可能更关注的是哪些具体原因会引起浏览器的回流，下面罗列一下：

1. 调整窗口大小
2. 改变字体大小
3. 样式表变动
4. 元素内容变化，尤其是输入控件
5. CSS伪类激活
6. DOM操作
7. offsetWidth, width, clientWidth, scrollTop/scrollHeight的计算，会使浏览器将渐进回流队列Flush，立即执行回流。

最佳实践

对我们Web开发者最有用的还是如何去做，才能减少页面回流。先来个例子：



```
var s = document.body.style;

s.padding = "2px"; // 回流+重绘
s.border = "1px solid red"; // 再一次 回流+重绘
```

```
s.color = "blue"; // 再一次重绘
s.backgroundColor = "#ccc"; // 再一次 重绘

s.fontSize = "14px"; // 再一次 回流+重绘

// 添加node, 再一次 回流+重绘
document.body.appendChild(document.createTextNode('abc!'));
```



可以看到每次DOM元素的样式操作都会引发重绘，如果涉及布局还会引发回流。
避免大量页面回流的手段也有很多，其本质都是尽量减少引起回流和重绘的DOM操作：

1. 避免逐项更改样式。最好一次性更改style属性，或者将样式列表定义为class并一次性更改class属性。
2. 避免循环操作DOM。创建一个documentFragment或div，在它上面应用所有DOM操作，最后再把它添加到window.document。
3. 避免循环读取offsetLeft等属性。在循环之前把它们存起来。
4. 绝对定位具有复杂动画的元素。绝对定位使它脱离文档流，否则会引起父元素及后续元素大量的回流。