# 概述

UI渲染

# 一般渲染架构

## Graphics API

将image,drawRect,drawCircle映射到底层的点与颜色

## 组件

运用软件工程学，将UI抽象成渲染组件，文本，图片，按钮等，然后运用layout的办法组合组件。渲染设置为组件编程，而不是Graphics API编程。

## 脚本

Javascript+html,lua等脚本系统简化组件的控制方法，加快开发效率

# 浏览器渲染架构

## 架构



这张图应该可以很好理解，归纳为四个步骤：

1、解析HTML以构建DOM树：渲染引擎开始解析HTML文档，转换树中的html标签或js生成的标签到DOM节点，它被称为 -- 内容树。

2、构建渲染树：解析CSS（包括外部CSS文件和样式元素以及js生成的样式），根据CSS选择器计算出节点的样式，创建另一个树 —- 渲染树。

3、布局渲染树: 从根节点递归调用，计算每一个元素的大小、位置等，给每个节点所应该出现在屏幕上的精确坐标。

4、绘制渲染树: 遍历渲染树，每个节点将使用UI后端层来绘制。

## 渲染顺序



浏览器执行的几个步骤：

restyle/reflow/repaint 触发条件：

了解完以上信息，考虑以下条件：



把 JavaScript 逻辑、复杂的 DOM 与样式合成，并完成渲染；

HTTP 请求下载多媒体；

在一个线程里；

移动上的 ARM 架构；

以上操作能在每帧 16ms 里完成，想想都觉得是一件 TMD「不可思议」的事情。于是各种各样的奇葩优化出现了。

## 渲染触发



# 成本

Reflow 的成本比 Repaint 的成本高得多的多。DOM Tree 里的每个结点都会有 reflow 方法，一个结点的reflow 很有可能导致子结点，甚至父点以及同级结点的 reflow。在一些高性能的电脑上也许还没什么，但是如果 reflow 发生在手机上，那么这个过程是非常痛苦和耗电的。  
所以，下面这些动作有很大可能会是成本比较高的。

1. 当你增加、删除、修改 DOM 结点时，会导致 Reflow 或 Repaint。
2. 当你移动 DOM 的位置，或是搞个动画的时候。
3. 当你修改 CSS 样式的时候。
4. 当你 Resize 窗口的时候（移动端没有这个问题），或是滚动的时候。
5. 当你修改网页的默认字体时。

注：display:none 会触发 reflow，而 visibility:hidden 只会触发 repaint，因为没有发现位置变化。

# 优化

Reflow是不可避免的，只能将Reflow对性能的影响减到最小,给出下面几条建议：

1. 不要一条一条地修改 DOM 的样式。与其这样，还不如预先定义好 css 的 class，然后修改 DOM 的className：
2. // 不好的写法
3. var left = 10,
4. top = 10;
5. el.style.left = left + "px";
6. el.style.top = top + "px";
7. // 推荐写法
8. el.className += " theclassname";
9. 把 DOM 离线后修改。如：  
   a> 使用 documentFragment 对象在内存里操作 DOM。  
   b> 先把 DOM 给 display:none (有一次 repaint)，然后你想怎么改就怎么改。比如修改 100 次，然后再把他显示出来。  
   c> clone 一个 DOM 节点到内存里，然后想怎么改就怎么改，改完后，和在线的那个的交换一下。
10. 不要把 DOM 节点的属性值放在一个循环里当成循环里的变量。不然这会导致大量地读写这个结点的属性。
11. 尽可能的修改层级比较低的 DOM节点。当然，改变层级比较底的 DOM节点有可能会造成大面积的reflow，但是也可能影响范围很小。
12. 为动画的 HTML 元件使用 fixed 或 absoult 的 position，那么修改他们的 CSS 是会大大减小reflow 。
13. 千万不要使用 table 布局。因为可能很小的一个小改动会造成整个 table 的重新布局。

认识了这些是不是对浏览器的原理有更大兴趣了？OK,后面会更新关于浏览器原理的文章，或者你们可以先去搜搜别人的，因为我觉得理解浏览器的原理确实是很重要，可以帮助我们写出性能更好的website。

# 作业

## 需求

以html5的canvas API实现一个浏览器框架。

指定读入网络的一个xml文件，xml文件中只包含两种元素。

<table>与<span>元素

table元素只有column与row属性

span元素只有fontSize,color属性

table元素需要满足多层嵌套。

## 注意

利用canvas api读入文件，分析xml，然后渲染到屏幕中。

注意，只能使用canvas api进行绘图，不能使用html中自带的布局系统来绘图。

实现时注意参考浏览器的reflow,repaint与composition三个步骤来实现。

## 截止

下周五检查。