前端性能优化是所有前端必须面对的，甚至决定前端价值所在。以下就是我当前对前端性能优化的理解。

首先，必须明确一点，前端性能优化是个无底洞，永远没有终点。万事万物都有自然规则在内，前端也是，比如扫地，一开始，我只要起几下，就能将绝大多数垃圾扫进去，可剩下一点点灰尘，却需要多出数倍的花销。当然做很多事情都如是，前端性能优化也是如此。

首先，有很多常用的性能优化方案，比如，代码压缩，dom操作优化（dom操作是非常好性能的，甚至是一般操作的5至6倍，所以它的优化至关重要。具体可以查看《高性能javascript》）。雪碧图，图片压缩等等，这些都是非常常规的，很多公司也将它纳入开发规范中作为开发必须要遵守的一部分，这很棒，这就像是我上面说的，扫地时刚开始的那几下，它的性价比不言而喻，所以，如果你还有相应没遵守，那么最好马上就去做。

上面我提到“性价比”，没错，你必须明确，你在为项目，为公司服务，一开始你就必须明确。公司必须讲产出和回报，一个脱离公司的程序员恐怕是很多公司不愿意看到的。那么剩余那点“垃圾”怎么办？很多网站访问量并没有淘宝那种量级，做到上面提及的大部分，其实对公司来说早已足够了。如果继续优化，成本太大，其实对公司来说并没有多少实际意义，毕竟前端优化虽然重要，但http协议才是web性能瓶颈之一。加上如今v8引擎优化有目共睹。但并不意味着好了，我做完了，如果项目有空余，那么你完全可以继续优化下去，如果时间不充裕，那么告诉自己你已经做到最好了。那么接下来有哪些方面的优化切入点呢？就像一开始说的，性能优化没有终点，它也太大了。接下来就说说自己的一些小经验吧。

架构！这点很难说，牵扯太多，还需要前端对整个架构有部分决定权还有了解，需要和后端协调，这是公司架构师的主要事情，如果你觉得有些地方不好，可以提出来，当然实际上还是取决于公司特点了。

HTML and CSS ，对于前端来说页面是所有工作的基石了，（如果是node工程师，就当我没说），首先，你必须了解html，css整个渲染的流程有大致了解，举个例子，双飞翼布局就是为了让中间重要内容优先展示而采用的一种不错的方案，你得明白，html是从上到下依次解析的，和呈现给页面的外观无关。还有，检查一下是否有无甚必要的多余标签。CSS样式也不要出现过长的嵌套。

尤其讲一下，页面的重绘重排，webkit渲染引擎优化并没有如v8引擎一样激进，尤其在移动端，我们往往会在页面上添加许多动态元素。产品就会抱怨有些地方不流畅闪烁等等，简单说下，页面排列就像排队，如果你在最后面插入一个人，没问题，整个队伍不会有什么大的影响，如果你往头部插入一个人，（当然不提倡插队），那么整个队伍都会往后退出一个人的距离，那么这时候，渲染引擎不得不对整个页面进行渲染重绘，如果操作很多，那么结果显而易见。具体可以参考另一篇重绘重排。另外，页面渲染很多时候依靠cpu，如果cpu消耗过大，可以让部分变化剧烈的部分交给gpu来进行选人，这就是gpu加速，具体可以查看淘宝的前端无线性能优化。（有点难懂，需要对页面原理有扎实的了解。）

JS,对于它的优化，太多，也太难说了，没有哪个前端说我把js优化完全做到了。况且这不像有些东西是一蹴而就的，它贯穿你的整个js学习历程。《高性能javascipt》，很多性能优化都提到了，甚至开始衡量当中的算法复杂度。对我我自己来说，如今已慢慢养成自己的一套js规范和风格，不敢保证说尽善尽美，但好的规范养成最起码能让你即使无功也无过。而且，性能就是一切么？



很多人选择上面的命名方式，以前同事对我说上面的性能更好，减少重复声明的开销，但如今我依然选择下面的命名方式，没错，上面的性能更好，但下面的声明更稳定，易与修改，而上面一不小心很容易会将某个值声明成了全局变量。我觉得，性能优化固然重要，但不能摆在第一位，性能就是一切。需要权衡得失。

懒加载，很多时候，请求的接口就在那里，作为前端也不可能去指摘后端接口问题，可怎么管理接口就是前端的艺术了。懒加载，比如常见的图片懒加载，用户只需要看到当前一屏的展示效果，所以不需要一开始就加载全部内容让网页长时间挂起。阿里的首屏优化一篇，对淘宝的加载有非常详细的讲解，真的非常不错。懒加载的最高境界就像新闻里的一篇报道一样，同一屋檐下，同时生活2个人然而能做到不被发现，需要很多机巧的心思和用户操作有详细的了解。

未完待续~