

# 1.图片压缩对比

通过jpg压缩后

15.jpg	378.7 KB	Finished	173.3 KB	download	-54%
16.jpg	450.8 KB	Finished	220.9 KB	download	-51%

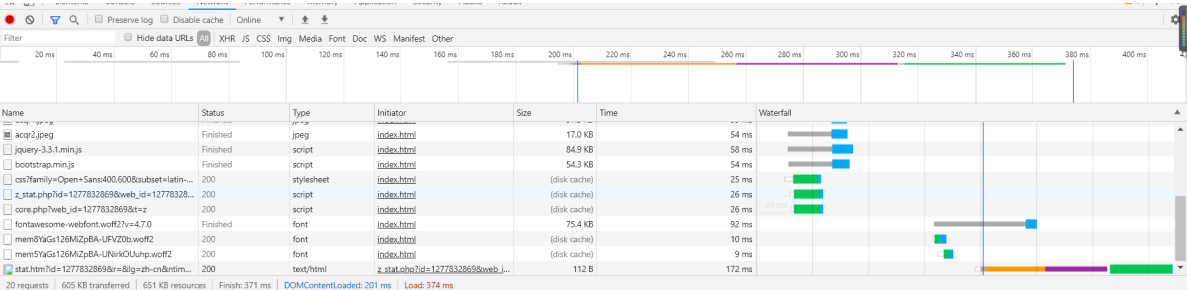
通过png压缩后

15.png	378.7 KB	Finished	173.3 KB	download	-54%
16.png	450.8 KB	Finished	220.9 KB	download	-51%

结果看来是相同的..

# 2.某公司页面压缩优化

在压缩以前先来看看最先开始未经压缩的效果

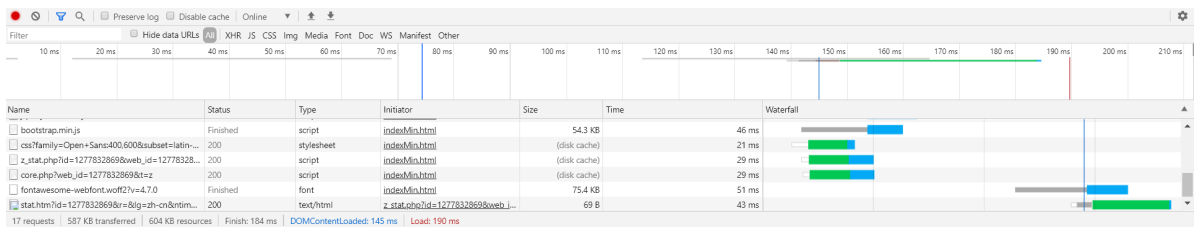


应该可以很清楚的看到**DOMContentLoaded**数值为201ms,而**Load**数值为 374 ms

下面我们来看看文字压缩后的

首先是只对css进行压缩后的,下图是我对style.css进行的压缩





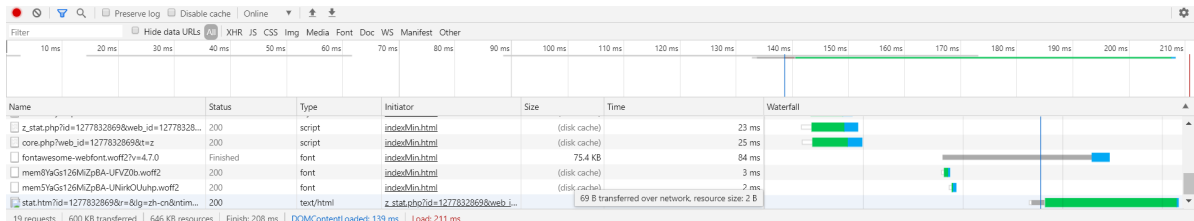
应该可以很清楚的看到**DOMContentLoaded**数值为145ms,而**Load**数值为 190 ms ,load速度和最初相比将近节省了一半的时间,速度得到了大幅度的提升。

速度提升的原因就是字符的减小吧,文件大小也明显的下降。

	index.html	2020/4/3 15:18	Chrome HTML D...	12 KB
	indexMin.html	2020/4/3 15:10	Chrome HTML D...	8 KB
	style.css	2020/3/27 9:32	层叠样式表文档	15 KB
	style.min.css	2020/4/3 14:59	层叠样式表文档	13 KB

下面进行http请求的优化

首先先做的就是**合并文字资源，减少下载频率**

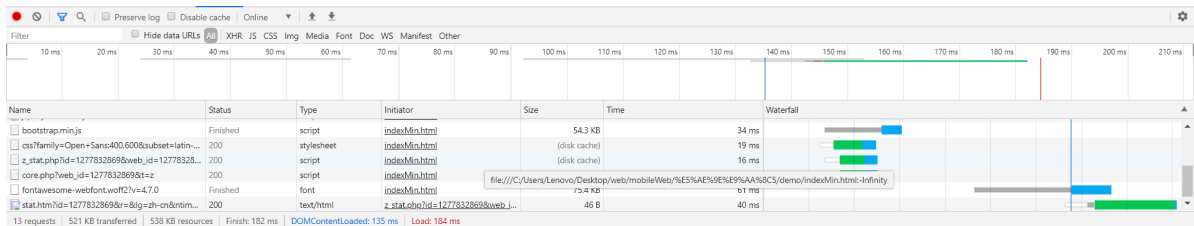


应该可以很清楚的看到**DOMContentLoaded**数值为139ms,而**Load**数值为 211 ms ,减少了发出的HTTP请求的数量,又快了一些。

接下来就是**图片合并**将图片制作为精灵图。效果如下图:



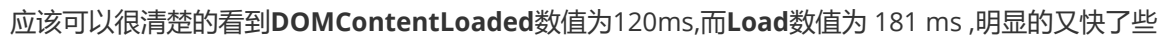
再通过css修改



应该可以很清楚的看到**DOMContentLoaded**数值为135ms,而**Load**数值为 184 ms ,明显的又快了些

速度提升的原因就是请求数量的减小,请求数量的减少,大大缩短了时间,上次实验报告中也有提到,文件资源下载的过程中很多时间都是Stalled等待时间,减少请求次数等待时间也大大的缩短了很多。

话不多说放结果



Over.