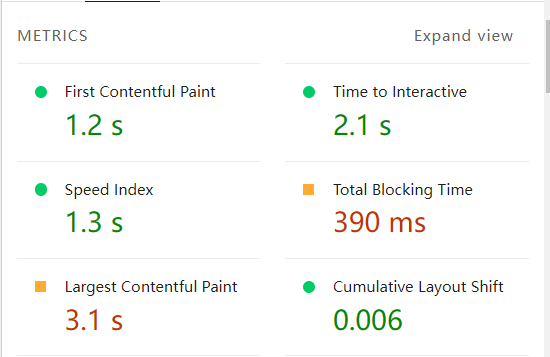
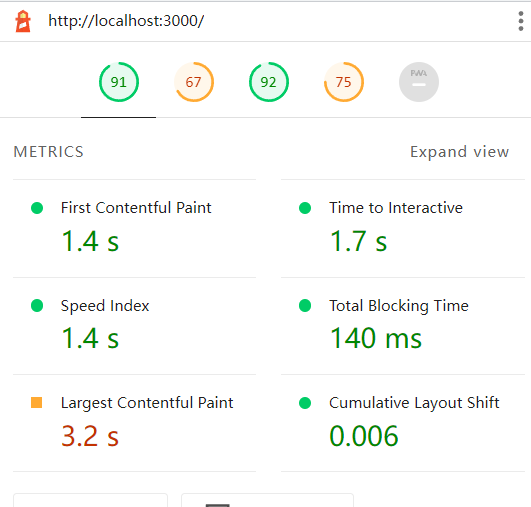
验证分数 谷歌浏览器Lighthouse mobile分数:

不用vite框架直接加载静态网页：



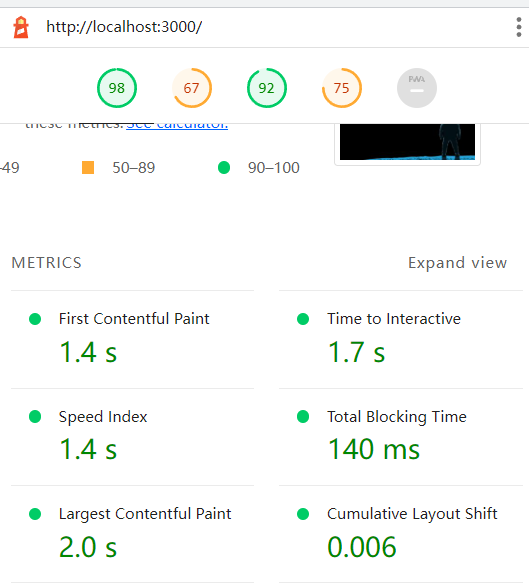
Vite框架初始状态：



部分指标略低于静态网页，猜测是由于内容较少，架构的冗余加载（封装层次更高）带来的负担大于vite框架的优势。

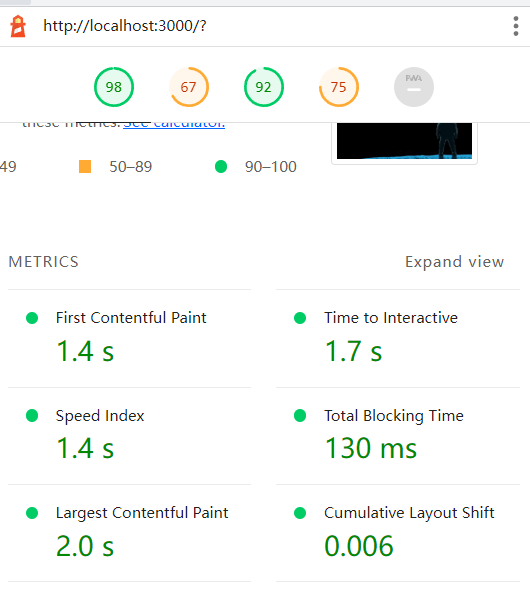
使用 tinypng压缩png图片：

将背景图丢入tinypng后



（网页总共就一张背景图片）

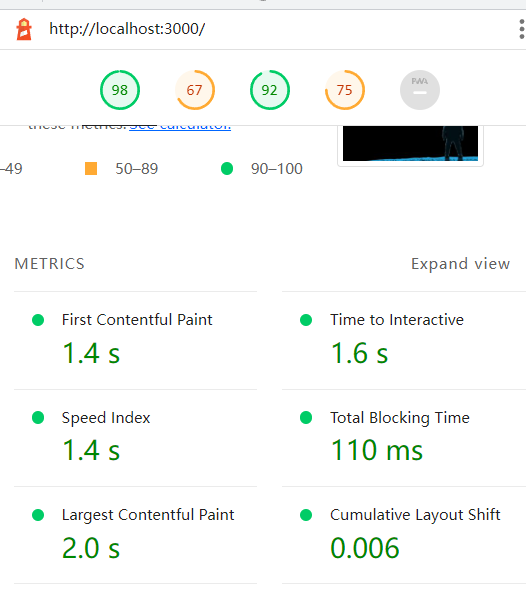
将tinypng转换后的图片转为webp格式：



可见和tinypng的性能差不多，应该是经过转换后的png图片大小和webp大小差不多，心性能主要取决于图片大小而不是webp格式。(而且好像同一个代码，每次跑都稍微有些差别。。所以参考价值也不太大)

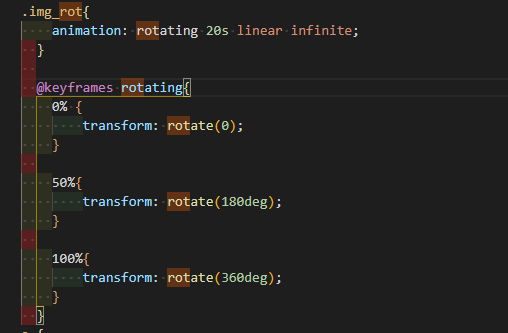
以下在webp格式下继续优化：

使用img的loading lazy:



TBT少了20ms，有点用

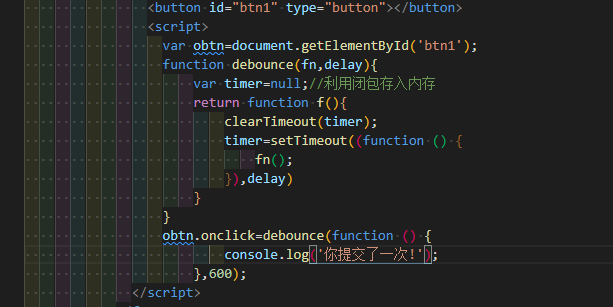
使用transform代替position:



由于之前没有使用旋转图标，性能得分已没有参考价值，没有跑。

使用防抖函数：

为了便于debug，将button设置为type="button"防止页面重绘



可见，类似于python的装饰器，闭包把要执行的函数套一层settimeout然后返回给onclick函数，这样每次在设置的600ms中点击都会出发onclick的函数，而每次点击都会把通过闭包在内存中的唯一变量timer cleartimeout并重新设置时间为600ms，即重新计时，从而减少服务器端的请求数。

总结：

实现了一些基本的性能优化方式，而本静态页面图片和内容有点少，预计当大型网页的时候这些优化手段的效果应该更明显。