一：加载优化

1. 合并css、js文件
2. 合并小图片，使用雪碧图
3. 缓存一切可以缓存的资源
4. 使用外联样式css、js
5. 压缩html、css、js
6. 使用首屏加载
7. 使用按需加载
8. 减少cookie
9. 避免重定向
10. 异步加载第三方资源

二：图片优化

1. 使用智图 <http://zhitu.tencent.com/>
2. 使用css、SVG代替图片
3. 使用srcset
4. Webp优于JPG PNG8优于GIF
5. 首次加载不大于1014kb
6. 图片不宽与640px

三：脚本优化

1. 减少重绘和回流
2. 缓存DOM选择和计算
3. 缓存列表.length
4. 尽量使用事件代理，避免事件绑定
5. 尽量使用ID选择器

四：css优化

1. css写在头部js写在脚步
2. 避免iFrame和img的空src
3. 尽量避免重设图片的大小
4. 尽量避免写在HTML标签中的style
5. 避免css表达式
6. 移除空的css规则
7. 正确使用display属性
8. 不滥用float
9. 不滥用web字体
10. 不声明过多的font-size
11. 值为0时不需要任何单位
12. 标准化各种浏览器前缀

五：渲染优化

1. 减少DOM节点
2. 尽量使用css动画
3. 合理使用requestAnimationErame动画代替setTimeout
4. 适当使用canvas动画