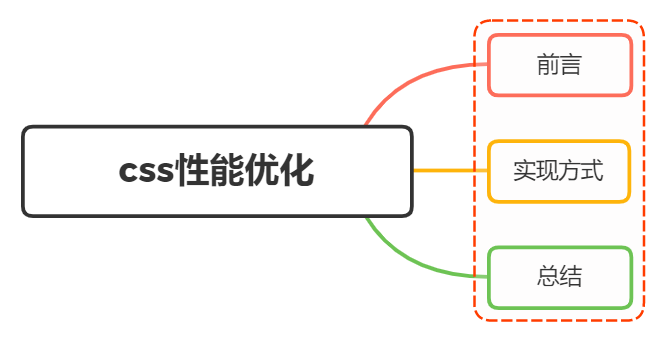
# 面试官：如果要做优化，CSS提高性能的方法有哪些？



## 一、前言

每一个网页都离不开css，但是很多人又认为，css主要是用来完成页面布局的，像一些细节或者优化，就不需要怎么考虑，实际上这种想法是不正确的

作为页面渲染和内容展现的重要环节，css影响着用户对整个网站的第一体验

因此，在整个产品研发过程中，css性能优化同样需要贯穿全程

## 二、实现方式

实现方式有很多种，主要有如下：

* 内联首屏关键CSS
* 异步加载CSS
* 资源压缩
* 合理使用选择器
* 减少使用昂贵的属性
* 不要使用@import

### 内联首屏关键CSS

在打开一个页面，页面首要内容出现在屏幕的时间影响着用户的体验，而通过内联css关键代码能够使浏览器在下载完html后就能立刻渲染

而如果外部引用css代码，在解析html结构过程中遇到外部css文件，才会开始下载css代码，再渲染

所以，CSS内联使用使渲染时间提前

注意：但是较大的css代码并不合适内联（初始拥塞窗口、没有缓存），而其余代码则采取外部引用方式

### 异步加载CSS

在CSS文件请求、下载、解析完成之前，CSS会阻塞渲染，浏览器将不会渲染任何已处理的内容

前面加载内联代码后，后面的外部引用css则没必要阻塞浏览器渲染。这时候就可以采取异步加载的方案，主要有如下：

* 使用javascript将link标签插到head标签最后

// 创建link标签  
const myCSS = document.createElement( "link" );  
myCSS.rel = "stylesheet";  
myCSS.href = "mystyles.css";  
// 插入到header的最后位置  
document.head.insertBefore( myCSS, document.head.childNodes[ document.head.childNodes.length - 1 ].nextSibling );

* 设置link标签media属性为noexis，浏览器会认为当前样式表不适用当前类型，会在不阻塞页面渲染的情况下再进行下载。加载完成后，将media的值设为screen或all，从而让浏览器开始解析CSS

<link rel="stylesheet" href="mystyles.css" media="noexist" onload="this.media='all'">

* 通过rel属性将link元素标记为alternate可选样式表，也能实现浏览器异步加载。同样别忘了加载完成之后，将rel设回stylesheet

<link rel="alternate stylesheet" href="mystyles.css" onload="this.rel='stylesheet'">

### 资源压缩

利用webpack、gulp/grunt、rollup等模块化工具，将css代码进行压缩，使文件变小，大大降低了浏览器的加载时间

### 合理使用选择器

css匹配的规则是从右往左开始匹配，例如#markdown .content h3匹配规则如下：

* 先找到h3标签元素
* 然后去除祖先不是.content的元素
* 最后去除祖先不是#markdown的元素

如果嵌套的层级更多，页面中的元素更多，那么匹配所要花费的时间代价自然更高

所以我们在编写选择器的时候，可以遵循以下规则：

* 不要嵌套使用过多复杂选择器，最好不要三层以上
* 使用id选择器就没必要再进行嵌套
* 通配符和属性选择器效率最低，避免使用

### 减少使用昂贵的属性

在页面发生重绘的时候，昂贵属性如box-shadow/border-radius/filter/透明度/:nth-child等，会降低浏览器的渲染性能

### 不要使用@import

css样式文件有两种引入方式，一种是link元素，另一种是@import

@import会影响浏览器的并行下载，使得页面在加载时增加额外的延迟，增添了额外的往返耗时

而且多个@import可能会导致下载顺序紊乱

比如一个css文件index.css包含了以下内容：@import url("reset.css")

那么浏览器就必须先把index.css下载、解析和执行后，才下载、解析和执行第二个文件reset.css

### 其他

* 减少重排操作，以及减少不必要的重绘
* 了解哪些属性可以继承而来，避免对这些属性重复编写
* cssSprite，合成所有icon图片，用宽高加上backgroud-position的背景图方式显现出我们要的icon图，减少了http请求
* 把小的icon图片转成base64编码
* CSS3动画或者过渡尽量使用transform和opacity来实现动画，不要使用left和top属性

## 三、总结

css实现性能的方式可以从选择器嵌套、属性特性、减少http这三面考虑，同时还要注意css代码的加载顺序

## 参考文献

* https://www.zhihu.com/question/19886806
* https://juejin.cn/post/6844903649605320711#heading-1
* https://vue3js.cn/interview/