# 前端面试指南 — 性能优化篇面试题

# ****性能优化基本百分百会问到，特别是在笔试题最后一道，跟在面试中会提问，所以需要做好充分准备。****

目录

[前端面试指南 — 性能优化篇面试题 1](#_Toc4105)

[性能优化基本百分百会问到，特别是在笔试题最后一道，跟在面试中会提问，所以需要做好充分准备。 1](#_Toc27220)

[1、谈谈你对重构的理解。 1](#_Toc479)

[2、 如果一个页面上有大量的图片（大型电商网站），网页加载很慢，可以用哪些方法优化这些图片的加载，从而提升用户体验 2](#_Toc28069)

[3、谈谈性能优化问题。 2](#_Toc25103)

[4、移动端性能如何优化？ 3](#_Toc12496)

[5、如何对网站的文件进行优化？ 3](#_Toc1792)

[6、请说出几种缩短页面加载时间的方法。 3](#_Toc10695)

[7、哪些方法可以提升网站前端性能？ 4](#_Toc19296)

[8、你知道哪些优化性能的方法？ 4](#_Toc6904)

[9、列举你知道的Web性能优化方法。 5](#_Toc22645)

[10、平时你是如何对代码进行性能优化的？ 6](#_Toc32479)

[11、针对CSS，如何优化性能？ 6](#_Toc1008)

[12、针对HTML，如何优化性能？ 8](#_Toc3329)

[13、针对 JavaScript，如何优化性能？ 8](#_Toc25969)

[14、如何优化服务器端？ 9](#_Toc3047)

[15、如何优化服务器端的接口？ 9](#_Toc10528)

[16、如何优化脚本的执行？ 10](#_Toc32727)

[17、如何优化渲染？ 10](#_Toc4094)

[18、如何设置DNS缓存？ 11](#_Toc1109)

[19、什么时候会出现资源访问失败？ 12](#_Toc21474)

[20、jQuery性能优化如何做？ 12](#_Toc20148)

[21、哪些方法能提升移动端CSS3动画体验？ 13](#_Toc5182)

# 

# 1、谈谈你对重构的理解。

网站重构是指在不改变外部行为的前提下，简化结构、添加可读性，且在网站前端保持一 致的行为。也就是说，在不改变UI的情况下，对网站进行优化，在扩展的同时保持一致的 UI。对于传统的网站来说，重构通常包括以下方面。

1. 如果一个页面上有大量的图片（大型电商网站），网页加载很慢，可以用哪些方法优化这些图片的加载，从而提升用户体验**？**

对于图片懒加载，可以为页面添加一个滚动条事件，判断图片是否在可视区域内或者即将进入可视区域，优先加载。如果为幻灯片、相册文件等，可以使用图片预加载技术，对于当前展示图片的前一张图片和后一张图片优先下载。如果图片为CSS图片，可以使用 CSS Sprite、SVG sprite、 Icon font、Base64等技术。如果图片过大，可以使用特殊编码的图片，加载时会先加载一张压缩得特别小的缩略图，以提高用户体验。如果图片展示区域小于图片的真实大小，则应在服务器端根据业务需要先行进行图片压缩，图片压缩后，图片大小与展示的就一致了。

# 3、谈谈性能优化问题。

可以在以下层面优化性能。

**缓存利用：**缓存Ajax，使用CDN、外部 JavaScript和CSS文件缓存，添加 Expires头，在服务器端配置Etag，减少DNS查找等。

**请求数量：**合并样式和脚本，使用CSS图片精灵，初始首屏之外的图片资源按需加载，静态资源延迟加载。

**请求带宽：**压缩文件，开启GZIP。

**CSS代码：**避免使用CSS表达式、高级选择器、通配选择器。

**JavaScript代码：**用散列表来优化查找，少用全局变量，用 innerHTML代替DOM操作，减少DOM操作次数，优化 JavaScript性能，用 setTimeout避免页面失去响应，缓存DOM节点查找的结果，避免使用with（with会创建自己的作用域，增加作用域链的长度），多个变量声明合并。

**HTML代码：**避免图片和 iFrame等src属性为空。src属性为空，会重新加载当前页面，影响速度和效率，尽量避免在HTML标签中写 Style属性

# 4、移动端性能如何优化？

尽量使用CSS3动画，开启硬件加速。

适当使用 touch事件代替 click事件。

避免使用CSS3渐变阴影效果。

可以用 transform:translateZ（0）来开启硬件加速。

不滥用 Float, Float在渲染时计算量比较大，尽量少使用。

不滥用Web字体，Web字体需要下载、解析、重绘当前页面，尽量少使用。

合理使用requestAnimation Frame动画代替 setTimeout。

合理使用CSS中的属性（CSS3 transitions、CSS3 3D transforms、 Opacity、 Canvas、 WebGL、Video）触发GPU渲染。过度使用会使手机耗电量増加。

# 

# 5、如何对网站的文件进行优化？

可以进行文件合并、文件压缩使文件最小化；可以使用CDN托管文件，让用户更快速地访问；可以使用多个域名来缓存静态文件。

# 6、请说出几种缩短页面加载时间的方法。

具体方法如下。

（1）优化图片

（2）选择图像存储格式（比如，GIF提供的颜色较少，可用在一些对颜色要求不高的地方）

（3）优化CSS（压缩、合并CSS）

（4）在网址后加斜杠

（5）为图片标明高度和宽度（如果浏览器没有找到这两个参数，它需要一边下载图片一边计算大小。如果图片很多，浏览器需要不断地调整页面。这不但影响速度，而且影响浏览体验。当浏览器知道高度和宽度参数后，即使图片暂时无法显示，页面上也会腾出图片的空位，然后继续加载后面的内容，从而优化加载时间，提升浏览体验）。

# 

# 7、哪些方法可以提升网站前端性能？

精灵图合并，减少HTTP请求；压缩HTML、CSS、JavaScript文件；使用CDN托管静态文件；使用 localstorage缓存和 mainfest应用缓存。

# 8、你知道哪些优化性能的方法？

具体方法如下。

（1）减少HTTP请求次数，控制CSS Sprite、JavaScript与CSS源码、图片的大小，使用网页Gzip、CDN托管、data缓存、图片服务器

（2）通过前端模板 JavaScript和数据，减少由于HTML标签导致的带宽浪费，在前端用变量保存Ajax请求结果，每次操作本地变量时，不用请求，减少请求次数。

（3）用 innerHTML代替DOM操作，减少DOM操作次数，优化 JavaScript性能。

（4）当需要设置的样式很多时，设置 className而不是直接操作 Style。

（5）少用全局变量，缓存DOM节点查找的结果，减少I/O读取操作

（6）避免使用CSS表达式，它又称动态属性，

（7）预加载图片，将样式表放在顶部，将脚本放在底部，加上时间戳。

（8）避免在页面的主体布局中使用表，表要在其中的内容完全下载之后才会显示出来，显示的速度比DIV+CSS布局慢。

# 9、列举你知道的Web性能优化方法。

具体优化方法如下。

（1）压缩源码和图片（ JavaScript采用混淆压缩，CSS进行普通压缩，JPG图片根据具体质量压缩为50%~70%，把PNG图片从24色压缩成8色以去掉一些PNG格式信息等）

（2）选择合适的图片格式（颜色数多用JPG格式，而很少使用PNG格式，如果能通过服务器端判断浏览器支持WebP就用WebP或SVG格式）。

（3）合并静态资源（减少HTTP请求）

（4）把多个CSS合并为一个CSS，把图片组合成雪碧图。

（5）开启服务器端的Gzip压缩（对文本资源非常有效）。

（6）使用CDN（对公开库共享缓存）。

（7）延长静态资源缓存时间。

（8）把CSS放在页面头部把 JavaScript代码放在页面底部（这样避免阻塞页面渲染而使页面出现长时间的空白）

# 10、平时你是如何对代码进行性能优化的？

利用性能分析工具监测性能，包括静态 Analyze工具和运行时的 Profile工具（在Xcode工具栏中依次单击 Product→ Profile项可以启动）。比如测试程序的运行时间，当单击 Time Profiler项时，应用程序开始运行，这就能获取到运行整个应用程序所消耗时间的分布和百分比。为了保证数据分析在同一使用场景下的真实性，一定要使用真机，因为此时模拟器在Mac上运行，而Mac上的CPU往往比iOS设备要快。

# 11、针对CSS，如何优化性能？

（1）正确使用 display属性， display属性会影响页面的渲染，因此要注意以下几方面。display:inline后不应该再使用 width、 height、 margin、 padding和float 。display:inline- block后不应该再使用 float。display:block后不应该再使用 vertical-align。display:table-\*后不应该再使用 margin或者float。

（2）不滥用 float。

（3）不声明过多的font-size。

（4）当值为0时不需要单位。

（5）标准化各种浏览器前缀，并注意以下几方面。

浏览器无前缀应放在最后。

CSS动画只用（ -webkit-无前缀）两种即可。

其他前缀包括 -webkit-、-moz-、-ms-、无前缀（ Opera浏览器改用 blink内核，所以-0-被淘汰）

（6）避免让选择符看起来像是正则表达式。高级选择器不容易读懂，执行时间也长。

（7）尽量使用id、 class选择器设置样式（避免使用 style属性设置行内样式）

（8）尽量使用CSS3动画。

（9）减少重绘和回流。

# 12、针对HTML，如何优化性能？

具体方法如下。

（1）对于资源加载，按需加载和异步加载

（2）首次加载的资源不超过1024KB，即越小越好。

（3）压缩HTML、CSS、 JavaScript文件。

（4）减少DOM节点。

（5）避免空src（空src在部分浏览器中会导致无效请求）。

（6）避免30\*、40\*、50\*请求错误

（7）添加 Favicon.ico，如果没有设置图标ico，则默认的图标会导致发送一个404或者500请求。

# 13、针对 JavaScript，如何优化性能？

具体方法如下。

（1）缓存DOM的选择和计算。

（2）尽量使用事件委托模式，避免批量绑定事件。

（3）使用 touchstart、 touchend代替 click。

（4）合理使用 requestAnimationFrame动画代替 setTimeOut。

（5）适当使用 canvas动画。

（6）尽量避免在高频事件（如 TouchMove、 Scroll事件）中修改视图，这会导致多次渲染。

# 14、如何优化服务器端？

（1）启用Gzip压缩。

（2）延长资源缓存时间，合理设置资源的过期时间，对于一些长期不更新的静态资源过期时间设置得长一些。

（3）减少 cookie头信息的大小，头信息越大，资源传输速度越慢。

（4）图片或者CSS、 JavaScript文件均可使用CDN来加速。

# 15、如何优化服务器端的接口？

具体方法如下。

（1）接口合并：如果一个页面需要请求两部分以上的数据接口，则建议合并成一个以减少HTTP请求数。

（2）减少数据量：去掉接口返回的数据中不需要的数据。

（3）缓存数据：首次加载请求后，缓存数据；对于非首次请求，优先使用上次请求的数据，这样可以提升非首次请求的响应速度。

# 16、如何优化脚本的执行？

脚本处理不当会阻塞页面加载、渲染，因此在使用时需注意。

（1）把CSS写在页面头部，把 JavaScript程序写在页面尾部或异步操作中。

（2）避免图片和 iFrame等的空src，空src会重新加载当前页面，影响速度和效率。

（3）尽量避免重设图片大小。重设图片大小是指在页面、CSS、 JavaScript文件等中多次重置图片大小，多次重设图片大小会引发图片的多次重绘，影响性能

（4）图片尽量避免使用 DataURL。DataURL图片没有使用图片的压缩算法，文件会变大，并且要在解码后再渲染，加载慢，耗时长。

# 17、如何优化渲染？

具体方法如下。通过HTML设置 Viewport元标签， Viewport可以加速页面的渲染，如以下代码所示。

<meta name="viewport" content="width=device=width,initial-scale=1">

（2）减少DOM节点数量，DOM节点太多会影响页面的渲染，应尽量减少DOM节点数量。

（3）尽量使用CSS3动画，合理使用 requestAnimationFrame动画代替 setTimeout，适当使用 canvas动画（5个元素以内使用CSS动画，5个元素以上使用 canvas动画（iOS 8中可使用 webGL））。

（4）对于高频事件优化 Touchmove, Scroll事件可导致多次渲染。使用 requestAnimationFrame监听帧变化，以便在正确的时间进行渲染，增加响应变化的时间间隔，减少重绘次数。使用节流模式（基于操作节流，或者基于时间节流），减少触发次数。

（5）提升GPU的速度，用CSS中的属性（CSS3 transitions、CSS3 3D transforms、 Opacity、 Canvas、 WebGL、Video）来触发GPU渲染.

# 18、如何设置DNS缓存？

在浏览器地址栏中输入URL以后，浏览器首先要查询域名（ hostname）对应服务器的IP地址，一般需要耗费20~120ms的时间。DNS查询完成之前，浏览器无法识别服务器IP，因此不下载任何数据。基于性能考虑，ISP运营商、局域网路由、操作系统、客户端（浏览器）均会有相应的DNS缓存机制。

1. 正IE缓存30min，可以通过注册表中 DnsCacheTimeout项设置。
2. Firefox混存1 min，通过 network.dnsCacheExpiration配置。
3. 在 Chrome中通过依次单击“设置”→“选项”→“高级选项”，并勾选“用预提取DNS提高网页载入速度”选项来配置缓存时间。

# 19、什么时候会出现资源访问失败？

开发过程中，发现很多开发者没有设置图标，而服务器端根目录也没有存放默认的 Favicon.ico，从而导致请求404出现。通常在App的 webview里打开 Favicon.ico，不会加载这个 Favicon.ico，但是很多页面能够分享。如果用户在浏览器中打开 Favicon. ico，就会调取失败，一般尽量保证该图标默认存在，文件尽可能小，并设置一个较长的缓存过期时间。另外，应及时清理缓存过期导致岀现请求失败的资源。

# 20、jQuery性能优化如何做？

优化方法如下。

（1）使用最新版本的 jQuery类库。JQuery类库每一个新的版本都会对上一个版本进行Bug修复和一些优化，同时也会包含一些创新，所以建议使用最新版本的 jQuery类库提高性能。不过需要注意的是，在更换版本之后，不要忘记测试代码，毕竟有时候不是完全向后兼容的。

（2）使用合适的选择器。jQuery提供非常丰富的选择器，选择器是开发人员最常使用的功能，但是使用不同选择器也会带来性能问题。建议使用简凖选择器，如i选择器、类选择器，不要将i选择器嵌套等。

（3）以数组方式使用 jQuery对象。使用 jQuery选择器获取的结果是一个 jQuery对象。然而， jQuery类库会让你感觉正在使用一个定义了索引和长度的数组。在性能方面，建议使用简单的for或者 while循环来处理，而不是$. each()，这样能使代码更快。

（4）每一个 JavaScript事件（例如 click、 mouseover等）都会冒泡到父级节点。当需要给多个元素绑定相同的回调函数时，建议使用事件委托模式。

（5）使用join( )来拼接字符串。使用 join( )拼接长字符串，而不要使用“+”拼接字符串，这有助于性能优化，特别是处理长字符串的时候。

（6）合理利用HTML5中的data属性。HTML5中的data属性有助于插入数据，特别是前、后端的数据交换；jQuery的 data( )方法能够有效地利用HTML5的属性来自动获取数据。

# 

# 21、哪些方法能提升移动端CSS3动画体验？

（1）尽可能多地利用硬件能力，如使用3D变形来开启GPU加速，例如以下代码。

-webkit-transform: translate 3d(0, 0, 0);

-moz-transform : translate3d(0,0, 0);

-ms-transform : translate 3d(0,0,0);

transform: translate3d(0,0,0)；

一个元素通过 translate3d右移500X的动画流畅度会明显优于使用left属性实现的动画移动，原因是CSS动画属性会触发整个页面重排、重绘、重组。paint通常是最耗性能的，尽可能避免使用触发 paint的CSS动画属性。如果动画执行过程中有闪烁（通常发生在动画开始的时候），可以通过如下方式处理。

-webkit-backface-visibility:hidden;

-moz-backface-visibility:hidden；

-ms-backface-visibility:hidden ;

backface-visibility:hidden；

-webkit-perspective：1000；

-moz-perspective：1000；

-ms-perspective：1000；

perspective：1000;

（2）尽可能少使用box- shadows和 gradients，它们往往严重影响页面的性能，尤其是在一个元素中同时都使用时。（3）尽可能让动画元素脱离文档流，以减少重排，如以下代码所示。

position:fixed；

position:absolute;