

前端-浏览器面试题

笔记本： 前端知识点

创建时间： 2020/8/30 14:52

更新时间： 2020/9/6 19:13

作者： 1317468898@qq.com

URL: <https://www.cnblogs.com/fundebug/p/10454641.html>

1. GET 和 POST 区别

GET参数通过URL传递，POST放在Request body中

GET比POST更不安全，因为参数直接暴露在URL上，所以不能用来传递敏感信息

GET产生一个TCP数据包；POST产生两个TCP数据包（1、在网络环境好的情况下，发一次包的时间和发两次包的时间差别基本可以无视。而在网络环境差的情况下，两次包的TCP在验证数据包完整性上，有非常大的优点；2、并不是所有浏览器都会在POST中发送两次包，Firefox就只发送一次；）

GET请求参数会被完整保留在浏览器历史记录里，而POST中的参数不会被保留

GET请求在URL中传送的参数是有长度限制的，而POST没有

2. 主流浏览器及其内核

IE--trident; Chrome--webkit/blink; Firefox--gecko; Opera--presto; Safari--webkit

1. CDN缓存原理

广泛采用各种缓存服务器，将这些缓存服务器分布到用户访问相对集中的地区或网络中，在用户访问网站时，利用全局负载技术将用户的访问指向距离最近的工作正常的缓存服务器上，由缓存服务器直接响应用户请求。

1.1. 前端如何优化网站性能？

减少 HTTP 请求数量

在浏览器与服务器进行通信时，主要是通过 HTTP 进行通信。浏览器与服务器需要经过三次握手，每次握手需要花费大量时间。而且不同浏览器对资源文件并发请求数量有限（不同浏览器允许并发数），一旦 HTTP 请求数量达到一定数量，资源请求就存在等待状态，这是很致命的，因此减少 HTTP 的请求数量可以很大程度上对网站性能进行优化。

- CSS Sprites：国内俗称 CSS 精灵，这是将多张图片合并成一张图片达到减少 HTTP 请求的一种解决方案，可以通过 CSS background 属性来访问图片内容。这种方案同时还可以减少图片总字节数。
- 合并 CSS 和 JS 文件：现在前端有很多工程化打包工具，如：grunt、gulp、webpack等。为了减少 HTTP 请求数量，可以通过这些工具再发布前将多个 CSS 或者 多个 JS 合并成一个文件。
- 采用 lazyLoad：俗称懒加载，可以控制网页上的内容在一开始无需加载，不需要发请求，等到用户操作真正需要的时候立即加载出内容。这样就控制了网页资源一次性请求数量。

控制资源文件加载优先级

浏览器在加载 HTML 内容时，是将 HTML 内容从上至下依次解析，解析到 link 或者 script 标签就会加载 href 或者 src 对应链接内容，为了第一时间展示页面给用户，就需要将 CSS 提前加载，不要受 JS 加载影响。

一般情况下都是 CSS 在头部，JS 在底部。

利用浏览器缓存

浏览器缓存是将网络资源存储在本地，等待下次请求该资源时，如果资源已经存在就不需要到服务器重新请求该资源，直接在本地读取该资源。

减少重排 (Reflow)

基本原理：重排是 DOM 的变化影响到了元素的几何属性（宽和高），浏览器会重新计算元素的几何属性，会使渲染树中受到影响的部分失效，浏览器会验证 DOM 树上的所有其它结点的 visibility 属性，这也是 Reflow 低效的原因。如果 Reflow 的过于频繁，CPU 使用率就会急剧上升。

减少 Reflow，如果需要在 DOM 操作时添加样式，尽量使用 增加 class 属性，而不是通过 style 操作样式。

减少 DOM 操作

图片优化

CDN缓存

3、在项目中如何做图片优化

- CDN缓存
- 减小图片的大小，在线压缩图的网站如[TinyPNG:一个压缩PNG的神站](#)
- 选择适当的图片宽度尺寸（即响应式图片）
- 减少HTTP的网络资源请求，考虑CSS Sprites（背景精灵图/雪碧图）
-

（3）减少HTTP的网络资源请求 考虑CSSSprites(背景精灵图/雪碧图)

利用CSS的background-image, background-repeat, background-position的组合进行背景定位，background-position可以用数字精确的定位出背景图片的位置

（4）css和css3`制作简单的图标和动画（代替gif图片）

随着技术的发展，css3可以实现的效果越来越多，比如箭头图标，三角形，梯形等图标，或者一些阴影效果，渐变的效果，所以可以用css制作一些简单的图标，而且具有多变性

（5）SVG技术替换图片

SVG 是使用 XML 来描述二维图形和绘图程序的语言,支持透明，缩放，动画

（6）字体图库代替图标

随着技术的更新和浏览器的更新，字体图库去代替一些图标我认为还是极好的，使用字体图库你不仅可以改变大小，而且还可以改变颜色

（7）图片延迟加载（懒惰加载）（js/lazyload.js）

有个页面会很大很长很多的图片素材，这样全部加载就会变的很慢，因此需要修改一下加载方案，只加载窗口内的图片，我们大家浏览网站的时候会经常看到会有默认图，图片加载成功之后会替换默认图

(8) 图片没加载出来显示默认图片

<https://zhuanlan.zhihu.com/p/50959463>

[https://blog.csdn.net/lianwenxiu/article/details/93191678?](https://blog.csdn.net/lianwenxiu/article/details/93191678?utm_medium=distribute.pc_relevant_t0.none-task-blog-BlogCommendFromMachineLearnPai2-1.edu_weight&depth_1-utm_source=distribute.pc_relevant_t0.none-task-blog-BlogCommendFromMachineLearnPai2-1.edu_weight)

[utm_medium=distribute.pc_relevant_t0.none-task-blog-BlogCommendFromMachineLearnPai2-1.edu_weight&depth_1-utm_source=distribute.pc_relevant_t0.none-task-blog-BlogCommendFromMachineLearnPai2-1.edu_weight](https://blog.csdn.net/lianwenxiu/article/details/93191678?utm_medium=distribute.pc_relevant_t0.none-task-blog-BlogCommendFromMachineLearnPai2-1.edu_weight&depth_1-utm_source=distribute.pc_relevant_t0.none-task-blog-BlogCommendFromMachineLearnPai2-1.edu_weight)

12.SVG

SVG动画优点:

- 矢量图形，不受像素影响——SVG的这个特性使得它在不同的平台或者媒体下表现良好，无论屏幕分辨率如何；
- SVG对动画的支持较好，其DOM结构可以被其特定语法或者Javascript控制，从而轻松的实现动画；
- Javascript可以完全控制SVGDom元素；
- SVG的结构是XML，其可访问性（盲文、声音朗读等）、可操作性、可编程性、可被CSS样式化完胜Canvas。另外，其支持ARIA属性，使其如虎添翼；

SVG动画缺点:

- DOM比正常的图形慢，而且如果其结点多而杂，就更慢；
- SVG画点报表什么的，还行；在网页游戏前，就束手无策了；当然可以结合Canvas+SVG实现；
- 不能动态的修改动画内容；
- 不能与HTML内容集成；
- 整个SVG作为一个动画；
- 浏览器兼容性问题，IE8-以及Android2.3默认浏览器是不支持SVG；

12. 网页从输入网址到渲染完成经历了哪些过程？

大致可以分为如下6步：

- DNS解析：发送到DNS服务器，并获取域名对应的web服务器对应的ip地址；
- TCP连接：与web服务器建立TCP连接；
- http请求：浏览器向web服务器发送http请求；
- 响应请求：web服务器响应请求，并返回指定url的数据（或错误信息，或重定向的新的url地址）；
- HTML词法解析：浏览器下载web服务器返回的数据及解析html源文件；
- 渲染页面：生成DOM树，解析css和js，渲染页面，直至显示完成；

13. 跨域

- JsonP：利用标签没有跨域限制的漏洞；
- cors：跨域资源共享，Access-control-Allow-Origin:*
- Nginx反向代理：proxyTable里设置开发机地址；

```
proxyTable: {  
  "/api": {  
    target: 'https://h5.zhaoyabuy.com/api/v1',  
    changeOrigin: true,  
    pathRewrite: {
```

```

    "/api": ""
  },
  // 解决跨域post请求不生效
  headers: {
    host: "https://h5.zhaoyabuy.com/api/v1",
    origin: "https://h5.uat.zhaoyabuy.com/api/v1"
  }
}
}

```

1.cookie

document.cookie = '名字=值';

document.cookie = 'username=cfangxu;domain=baike.baidu.com' 并且设置了生效域

2. localStorage

设置: localStorage.setItem('username','cfangxu');

获取: localStorage.getItem('username'), 也可以获取键名, localStorage.key(0) #获取第一个键名

删除: localStorage.removeItem('username'), 也可以一次性清除所有存储, localStorage.clear()

3.sessionStorage

其实跟localStorage差不多, 也是本地存储, 会话本地存储

用于本地存储一个会话 (session) 中的数据, 这些数据只有在同一个会话中的页面才能访问并且当会话结束后数据也随之销毁。因此sessionStorage不是一种持久化的本地存储, 仅仅是会话级别的存储。也就是说只要这个浏览器窗口没有关闭, 即使刷新页面或进入同源另一页面, 数据仍然存在。关闭窗口后, sessionStorage即被销毁, 或者在新窗口打开同源的另一个页面, sessionStorage也是没有的。

cookie、localStorage、sessionStorage区别

相同: 在本地 (浏览器端) 存储数据

不同:

localStorage只要在相同的协议、相同的主机名、相同的端口下, 就能读取/修改到同一份localStorage数据;

sessionStorage比localStorage更严苛一点, 除了协议、主机名、端口外, 还要求在同一窗口 (也就是浏览器的标签页) 下;

localStorage是永久存储, 除非手动删除;

sessionStorage当会话结束 (当前页面关闭的时候, 自动销毁);

cookie的数据会在每一次发送http请求的时候, 同时发送给服务器而localStorage、sessionStorage不会;

git 常用命令

1. git statu 查看改动文件
2. git add. 上传全部改动文件 git add ./pages/ // (单个文件路径) 上传单个文件
3. git commit -m '备注' 可省略2直接使用git commit -a 进入另一个页面, 将需要上传的文件前面的#删除, 输入:wq git commit 进入另一个页面, 按 i 进入插入模式, 在最上方输入注释, 按ESC退出插入模式, 输入:wq
4. git push origin '分支名' (origin master 没有分支时不需要) 上传到远程分支

5. `git pull origin '分支名'` (`origin master` 没有分支时不需要) 从远程分支拉取代码
6. `git branch -a` 查看所有分支, `git branch` 查看本地分支, `git branch -r` 查看远程分支, `git branch '分支名'` 创建分支, 依然停留本分支, `git checkout -b '分支名'`, 创建并切换分支
7. `git checkout '分支名'` 切换分支
8. `git merge '分支名'` 合并输入分支名的代码到现分支
9. `git clone '路径'` 下载
10. `git init` 在当前目录新建一个Git代码库