JS性能优化和安全

**1页面加载**

题目

1从输入URL 到得到HTML的详细过程

答:

1 浏览器根据DNS服务器得到域名的IP地址

2 向这个IP的服务器发送HTTP请求

3服务器收到、处理并返回HTTP请求

4浏览器得到返回内容

2 window.onload和DOMContentLoaded的区别是

答：

1 window.onload

页面的全部资源加载完成才会执行，包括图片 视频等

2DOMContentLoaded

DOM渲染完即可执行，些时图片、-视频还没有加载完

知识点

加载资源的形式

加载一个资源的过程

浏览器渲染页面的过程

解答

**加载资源的的形式**

1输入url（或跳 转页面）加载html

2加载html中的静态资源

**加载一个资源的过程**

1浏览器根据DNS服务器得到域名的IP地址

2向这个IP的服务器发送HTTP请求

3服务器收到、处理并返回HTTP请求

4浏览器得到返回内容

**浏览器渲染页面的过程**

1根据HTMl结构生成DOM Tree

2根据css生成CSS DOM

3将DOM 和CSSDOM整合形成Render Tree

4根据render Tree 开始渲染和展示

5遇到script时会执行并阻塞渲染（为什么JS会阻塞渲染因为JS有权力改变DOM Tree）

**JS性能优化**

这本身就是一个综合性的问题

只关注核心点

1多使用内存、缓存或者其它方法

2减少CPU计算、减少网络负载

如何处理

1加载页面和静态资源

2页面渲染

**加载资源优化**

1静态资源的压缩合并 如（a.js,b.js,c.js 压缩成一个commin.min.js）

2静态资源缓存

3使用CDN让资源加载更快

4使用SSR（server side render）后端渲染，数据直接输出到HTML中

**渲染优化**

1 CSS放前面,JS放后面

2懒加载（图片懒加载、下拉加载更多）

3减少DOM查询，对DOM查询做缓存

4减少DOM操作，多个操作尽量合并在一起执行

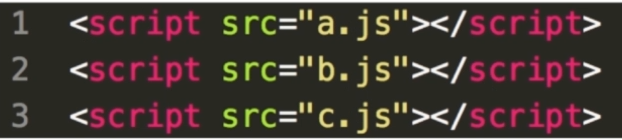
5事件节流

6尽量执行操作（如DOMContentLoaded）

示例

**资源合并**

如下图所示





**缓存**

通过链接名称控制缓存

<script src=”abs\_1.js”></script>

只有内容改变的时候，链接名称才会改变

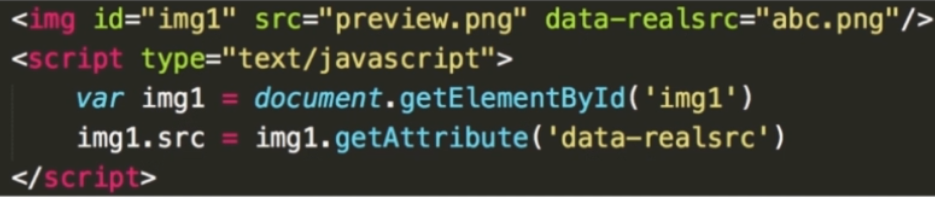
<script src=”src\_2.js”></script>

使用SSR后端渲染

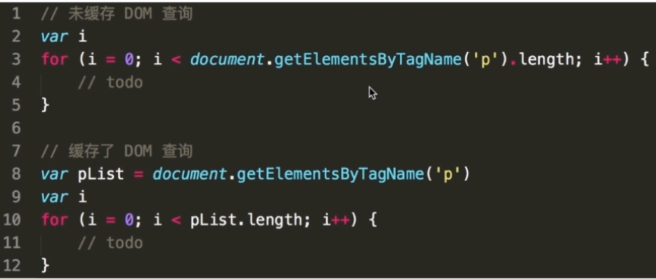
现在Vue React 提出了这样的概念

其实JSP PHP ASP 都属于后端渲染

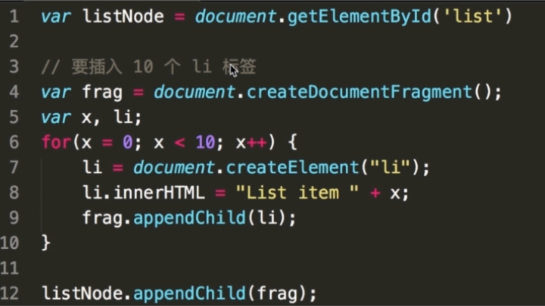
懒加载



**缓存DOM查询**



**合并DOM插入**



**事件节流**

使用setTimeout延时触发change事件



**尽早操作**



**安全性**

综合性的问题：场景的前端究全问题有哪些

XSS跨站请求攻击

XSRF跨站请求伪造

XSS

再博客写一篇文章，同时偷偷插入一段<Script>

攻击代码中，获取cookie，发送自己的服务器

发布博客，有人查看博内容

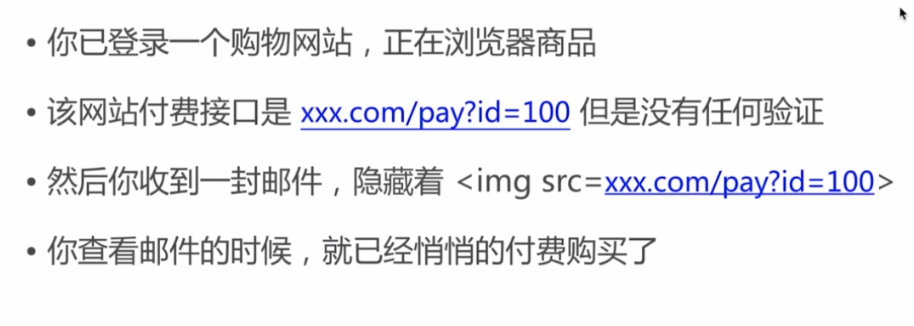
会把查看者的cookie发送到攻击者的服务器

怎么预防

前端替换关键字，例如替换<为&lt; >为&gt;（前端替换会响影性能）

后端替换

XSRF



增加验理论上流程，输入指纹，密码、短信验证码

技巧

简历

简洁明了，重点突出项目经历和解决方案

把个个博客放在简历中，并且定期维户更新博客

把个人的开源项止放在简历中，并维护开源项目