# Web应用安全发展与介绍

Web安全，也可以叫做Web应用安全。互联网本来是安全的，自从有了研究安全的人之后，互联网就变的不安全了。

Web应用经历了开始、1.0以及现在3.0概念的出现，不断的发展：

20世纪60年代IBM的GML（通用标记语言）以及发展到后来的SGML（标准通用标记语言）

20世纪90年代，HTML的出现

浏览器的出现与发展

2004之后，XMLHttpRequest的出现将Web推向2.0时代

而现在，开始出现Web3.0的概念

Web安全跟随着Web应用的发展也不断发展着：

Web 1.0时代，更多被关注的时服务器端的脚本的安全问题，如SQL注入等

Web 2.0时代，2005年Samy蠕虫的爆发震惊了世界，Web安全主战场由服务器端转换到浏览器

SQL注入和XSS的出现发别是Web安全史上的两个里程碑

Web安全的本质是信任问题

由于信任，正常处理用户恶意的输入导致问题的产生

非预期的输入

安全是木桶原理，短的那块板决定的木桶世纪能装多少水，同样的，假设把99%的问题都处理了，那么1%的余留会是造成安全问题的那个短板

# HTTP协议与会话管理

当我们访问一个网址的时候，这中间发生了什么？

输入网址

浏览器查找域名的IP地址

浏览器给Web服务器发送一个HTTP请求

服务端处理请求

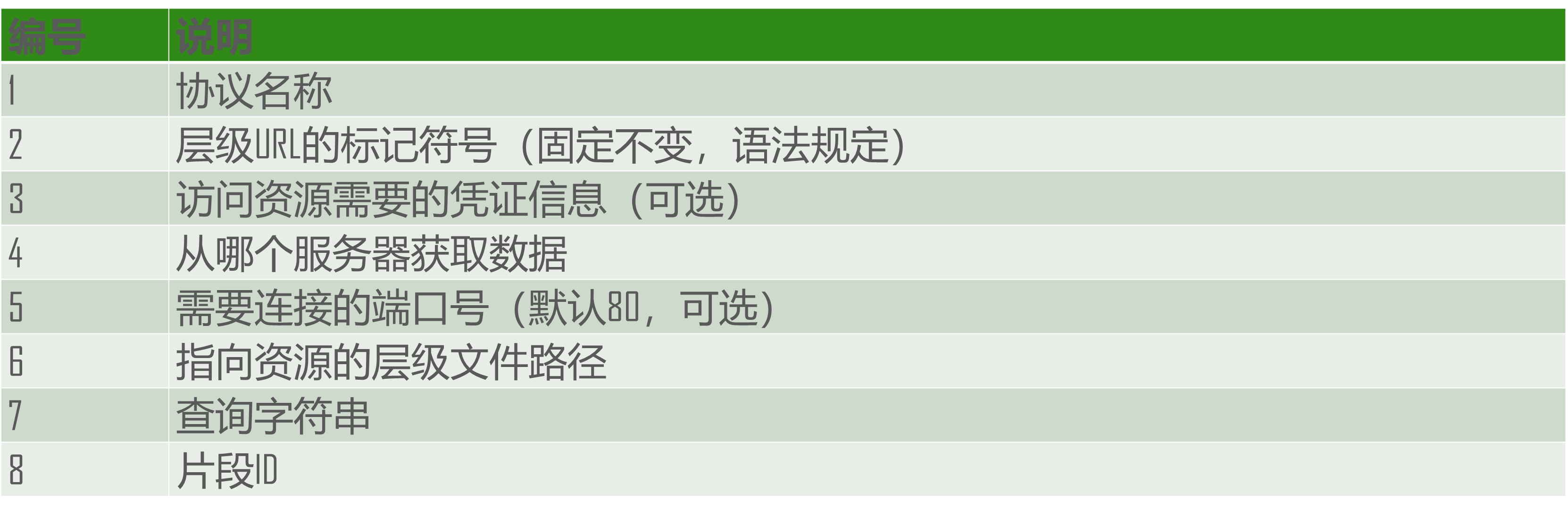
服务端发回一个HTTP响应

浏览器渲染显示HTML

## URL

让我们来看一个URL（统一资源定位器）





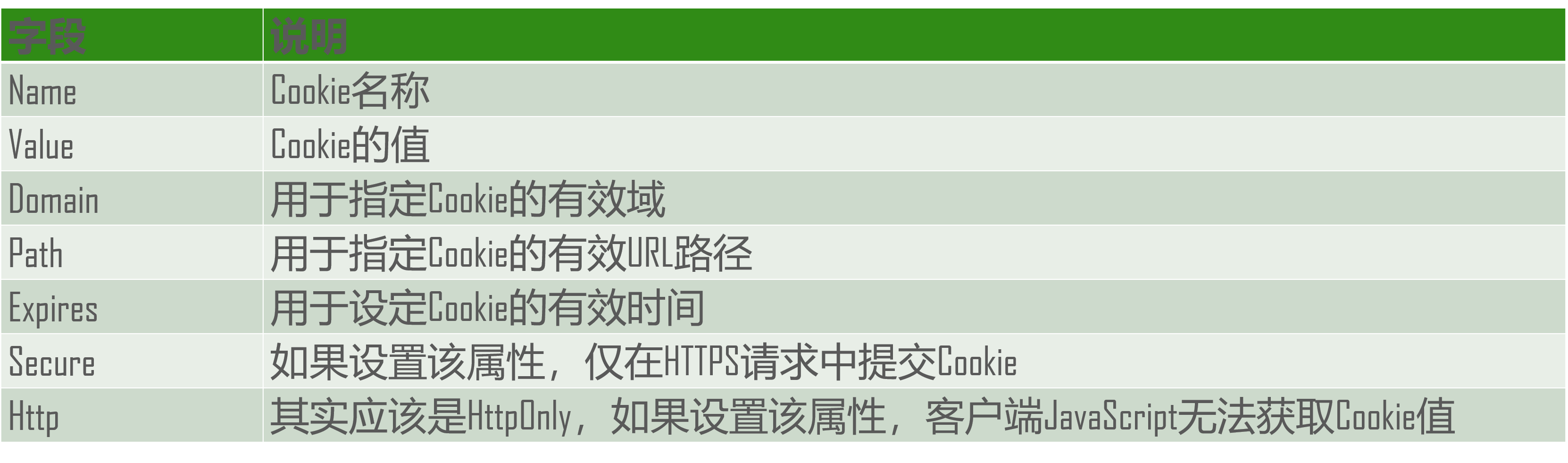
## 发送和返回

接下来我们来看下，请求一个网址的时候到底发送和返回了什么？

### Cookie

浏览器调试界面的Resource/Cookie中可以查看：





### Session





Session与Cookie的区别

Cookie的数据保存在客户端浏览器，Session保存在服务器

服务端保存状态机制需要在客户端做标记，所以Session可能借助Cookie机制

Cookie通常用于客户端保存用户的登录状态

# Web应用的组成与网页的渲染

## HTML

超文本标记语言

由标签（Tag）组成的层级结构，标签参数的格式为“名称=值”数据对，则文本节点（构成了文档实际内容）穿插其中。



## CSS（层叠样式表）

HTML页面引用示例：

<link href="css/style.css" rel="stylesheet">

或

<style>body {width}</style>

## 浏览器解析顺序



## 浏览器解码顺序



## DOM树

DOM，文档对象模型(Document Object Model)

JavaScript可以操作DOM树

这是一个网页的DOM结构：



这是一个通过JavaScript代码操作DOM树的示例代码：



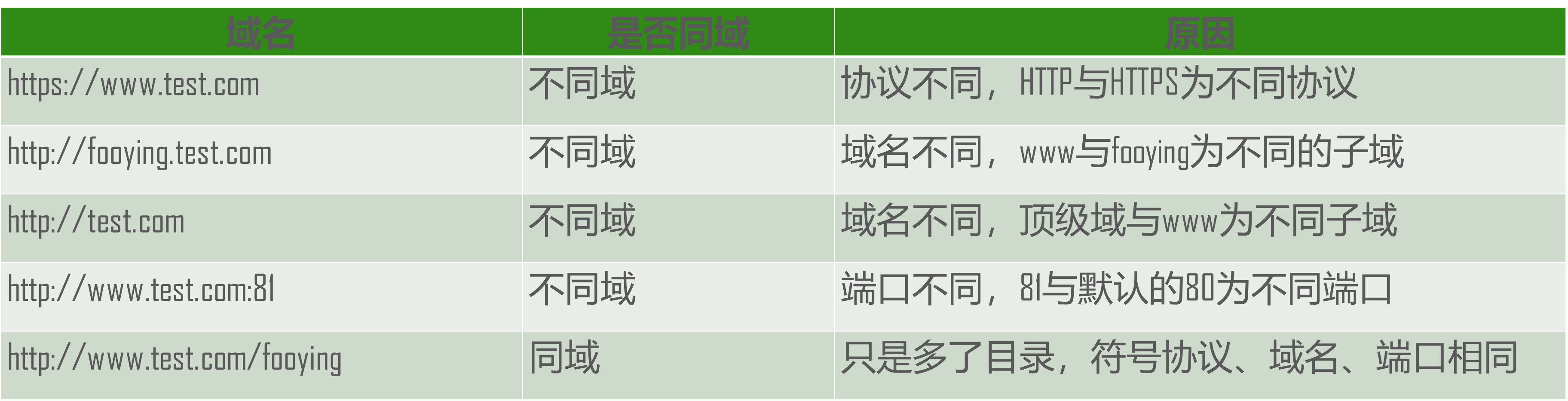
# 浏览器特性与安全策略

## 同源策略

同源策略规定：不同域的客户端脚本在没明确授权的情况下，不能读写对方的资源。

## 同域与不同域

以下为与www.test.com同域情况



## 授权

HTTP响应头返回

Access-Control-Allow-Origin:http://www.fooying.com

## 沙盒框架（Sandboxed frame）

是对常规<iframe>表现行为的扩展，它能让顶级页面对其嵌入的子页面及这些子页面的子资源设置一些额外的限制

通过设置<iframe>的参数实现限制