**什么是cookie？**

基于http协议的，浏览器与服务器进行会话时的一种本地'存储'技术

因为http协议属于无状态、短连接、断开式，为了解决（无状态）问题

采用cookie + session的模式，而session属于服务器的存储

cookie属于客户端存储

一次请求+一次响应形成一次会话

**cookie的特点:**

1.基于http协议

2.它会随着请求携带到服务器

3.只能存储4k左右

4.能跨域(设置domain),默认不可以

5.容易被伪造,不安全,会造成 xss攻击 (站点伪造)

解决方案：后端设置签名

**如何使用cookie**

基于http协议 ，但是火狐的Firefox可以在file上存储cookie

**语法：**

cookie由键值对形式的文本组成：name=value。

完整格式为：

document.cookie = "name=value;[expires=date];[path=路径];[domain=域名];[secure]"

其中中括号[]表示该值是可选。

name=value: 为你要保存的键值对(必选)

expires=date: 表示cookie的失效时间, 默认是session模式，关闭浏览器就会失效(可选)

path=路径: 访问路径, 默认为当前文件所在目录(可选)

存在根路径下的cookie 根路径下的任何路径都能取到值

不同分支路径下的cookie不能互相访问

domain=域名: 访问域名, 限制在该域名下访问(可选)

举例： 在百度的页面上 存储了“你好”的cookie 默认无法发送到qq的页面

secure: 安全设置, 如果设置了则必须使用https协议才可获取cookie(可选)

**cookie的增删查改**

**1.设置和获取cookie**

//设置cookie和获取cookie

document.cookie = "user="+"张三"; //设置

console.log(document.cookie); //获取

//URI编码后设置cookie, 和URI解码后获取cookie

document.cookie = "user2=" + encodeURIComponent("张三");

console.log(decodeURIComponent(document.cookie));

**sessionStorage**

它能存储5M大小的内容

也只能存字符串

关闭浏览器就会销毁

**JSON**

JSON是一个特定格式的字符串，几乎所有语言都能识别

能够转换成每个语言的对象，主要用于数据交互

javascript object notation

JSON与js对象格式相同但是属性和值必须加引号

**JSON分类：**

对象

数组

**JSON中可以使用的值：**

数字

字符串

布尔值

null

对象

数组

**JSON.parse()**

JSON ---> js对象

使用JSON对象作为参数，返回js对象

**JSON.stringify()**

js对象 ---> JSON

使用js对象作为参数，返回JSON

兼容IE7及以下使用eval来代替不能使用的JSON.parse(),或者引入JSON2库

**eval()**

如果执行的字符串中有{}，则会被当做代码块，如果不希望被当做代码块被解析，在{}前后使用()

由于效率与安全问题，尽量不要使用

左右查询

对等号左右两边查询即左右查询，右查询没有找到时会直接报错，左查询没有找到时会在全局创建