# 1.http请求中get和post方法的区别？

* get请求的参数是放在URL里面的，而post请求的参数是放在请求体里面的；
* get请求可以被浏览器缓存，而post请求是不能被浏览器缓存的；
* get请求参数由于是放在URL里面的，而URL的长度是受限的，最大为2k(2048个字符)，而post是没有限制的；
* get请求参数由于放在URL里面的，所以安全性较差，post请求是放在body体里面的，安全性较高；
* get请求可以通过浏览器直接访问，支持刷新和后退，post请求是不能被浏览器直接访问的，刷新后需要重新传送；

# 2.cookie和session的区别？

* cookie是存在在客户端，session是存放在服务端的；
* cookie是不安全的，别人可以分析你本地的cookie，考虑安全性最好选用session;
* session在一定的时间之内是保存在服务器上的，不过当访问越来越多的时候会占用服务器的资源性能，考虑到减轻服务器性能最好选用cookie;
* 单个cookie保存数据是有限的，一般不超过4k，很多站点限制最多保存20个cookie，而session是没有这种数据量的限制的，可以保存更加复杂的类型；
* 两者的生命周期有着明显的区别，cookie的生命周期是累计的，session的生命周期是间隔的；

# 3.发现一个bug怎么定位它是服务端还是客户端的问题呢？

常用方法：抓包分析和日志分析

* 使用抓包工具：例如fiddler或者charles模拟客户端向服务端发送一个请求，分析返回的数据对不对，如果服务端返回的数据是对的，就是客户端的问题，如果返回的数据不对，就是服务端的问题；
* 采用日志分析：查看客户端的日志或服务端的日志，进一步判断哪里出现了问题；

# 4.什么是协程？

简单来说，协程是一种基于线程之上，但又比线程更加轻量级的存在。对于系统内核来说，协程具有不可见的特性，所以这种由 程序员自己写程序来管理的轻量级线程又常被称作 "用户空间线程"。

# 5.协程比多线程好在哪呢？

* 线程的控制权在操作系统手中，而协程的控制权掌握在用户自己手中，因此利用协程可以减少程序运行时的上下文切换，有效提高程序运行效率；
* 建立线程时，系统默认分配给线程的栈大小是 1 M，而协程更轻量，接近 1 K 。因此可以在相同的内存中开启更多的协程；
* 由于协程的本质不是多线程而是单线程，所以不需要多线程的锁机制。因为只有一个线程，也不存在同时写变量而引起的冲突。在协程中控制共享资源不需要加锁，只需要判断状态即可。所以协程的执行效率比多线程高很多，同时也有效避免了多线程中的竞争关系；