http协议。

ISP(Internet Service Provider)，互联网服务提供商，即向广大用户综合提供互联网接入业务、信息业务和增值业务的电信运营商。

# 浏览器输入url按回车背后经历了哪些？

1.在PC浏览器的地址栏输入一串URL，然后按Enter键这个页面渲染出来，这个过程中都发生了什么事?

1、首先，在浏览器地址栏中输入url，先解析url，检测url地址是否合法

2、浏览器先查看浏览器缓存-系统缓存-路由器缓存，如果缓存中有，会直接在屏幕中显示页面内容。若没有，则跳到第三步操作。

浏览器缓存：浏览器会记录DNS一段时间，因此，只是第一个地方解析DNS请求；

操作系统缓存：如果在浏览器缓存中不包含这个记录，则会使系统调用操作系统，获取操作系统的记录(保存最近的DNS查询缓存)；

路由器缓存：如果上述两个步骤均不能成功获取DNS记录，继续搜索路由器缓存；

ISP缓存：若上述均失败，继续向ISP搜索。

3、在发送http请求前，需要[域名解析](https://cloud.tencent.com/product/cns?from=10680)(DNS解析)，解析获取相应的IP地址。

4、浏览器向服务器发起tcp连接，与浏览器建立tcp三次握手。

5、握手成功后，浏览器向服务器发送http请求，请求数据包。

6、服务器处理收到的请求，将数据返回至浏览器

7、浏览器收到HTTP响应

8、浏览器解码响应，如果响应可以缓存，则存入缓存。

9、 浏览器发送请求获取嵌入在HTML中的资源（html，css，javascript，图片，音乐······），对于未知类型，会弹出对话框。

10、 浏览器发送异步请求。

11、页面全部渲染结束。

{简单概括成六步如下：

1.域名解析

域名解析检查顺序为：浏览器自身DNS缓存---》OS自身的DNS缓存--》读取host文件--》本地域名服务器--》权限域名服务器--》根域名服务器。如果有且没有过期，则结束本次域名解析。域名解析成功之后，进行后续操作

2.tcp3次握手建立连接

3.建立连接后，发起http请求

4.服务器端响应http请求，浏览器得到到http请求的内容；

5.浏览器解析html代码，并请求html代码中的资源

6.浏览器对页面进行渲染，展现在用户面前。

}

# GET和POST的区别

2.get和post请求区别，这个是被问烂的题了

首先这个题看似简单，实际上是个送命题！如果你百度搜到的标准答案可能是这样的（本标准答案参考自w3schools）：

* GET在浏览器回退时是无害的，而POST会再次提交请求。
* GET产生的URL地址可以被Bookmark，而POST不可以。
* GET请求会被浏览器主动cache，而POST不会，除非手动设置。
* GET请求只能进行url编码，而POST支持多种编码方式。
* 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么有。
* 对参数的数据类型，GET只接受ASCII字符，而POST没有限制。
* 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参数通过URL传递，POST放在Request body中。

如果我告诉你，你死记硬背的这些所谓“标准答案”不是面试官想要的，你肯定不服，首先从安全性讲，get和post都一样，没啥所谓的哪个更安全 get请求参数在url地址上，直接暴露，post请求的参数放body部分，按F12也直接暴露了，所以没啥安全性可言

“GET参数通过URL传递，POST放在Request body中”这个其实也不准，post请求也可以没body，也可以在url传递呢？

如果我告诉你get请求和post请求本质上没区别，你肯定不信！ GET和POST有一个重大区别，简单的说： GET产生一个TCP数据包；POST产生两个TCP数据包。 长的说： 对于GET方式的请求，浏览器会把http header和data一并发送出去，服务器响应200（返回数据）； 而对于POST，浏览器先发送header，服务器响应100 continue，浏览器再发送data，服务器响应200 ok（返回数据）。

# cookies机制和session机制的区别

1. cookies机制和session机制的区别，这个也是经常会问的

* cookies数据保存在客户端，session数据保存在服务器端；
* cookies可以减轻服务器压力，但是不安全，容易进行cookies欺骗；
* session较安全，但占用服务器资源

# HTTP状态码

4.cookies机制和session机制的区别？这个是最基本的了，这个得熟练掌握，如果这个状态码都分不清，基本功就很弱了，印象分会大打折扣！

* 200 请求已成功，请求所希望的响应头或数据体将随此响应返回。
* 201 请求已经被实现，而且有一个新的资源已经依据请求的需要而建立，且其 URI 已经随Location 头信息返回
* 202 服务器已接受请求，但尚未处理
* 301 （永久移动） 请求的网页已永久移动到新位置。 服务器返回此响应（对 GET 或 HEAD 请求的响应）时，会自动将请求者转到新位置。
* 302 （临时移动） 服务器目前从不同位置的网页响应请求，但请求者应继续使用原有位置来进行以后的请求。--即重定向。这种情况下，服务器返回的头部信息中会包含一个 Location 字段，内容是重定向到的url。
* 303 （查看其他位置） 请求者应当对不同的位置使用单独的 GET 请求来检索响应时，服务器返回此代码。
* 304 （未修改） 自从上次请求后，请求的网页未修改过。 服务器返回此响应时，不会返回网页内容。
* 305 （使用代理） 请求者只能使用代理访问请求的网页。 如果服务器返回此响应，还表示请求者应使用代理。
* 307 （临时重定向） 服务器目前从不同位置的网页响应请求，但请求者应继续使用原有位置来进行以后的请求。
* 400：请求没有进入到后台服务里（一般都是前端的锅），参数错误
* 401 当前请求需要用户验证。如果当前请求已经包含了 Authorization 证书，那么401响应代表着服务器验证已经拒绝了那些证书
* 403 服务器已经理解请求，但是拒绝执行它。与401响应不同的是，身份验证并不能提供任何帮助，而且这个请求也不应该被重复提交
* 404 请求失败，请求所希望得到的资源未被在服务器上发现
* 500 服务器遇到了一个未曾预料的状况，导致了它无法完成对请求的处理。一般来说，这个问题都会在服务器的程序码出错时出现。
* 501 服务器不支持当前请求所需要的某个功能。当服务器无法识别请求的方法，并且无法支持其对任何资源的请求。
* 502 作为网关或者代理工作的服务器尝试执行请求时，从上游服务器接收到无效的响应。
* 503 由于临时的服务器维护或者过载，服务器当前无法处理请求。这个状况是临时的，并且将在一段时间以后恢复。

# http协议请求方式

5.http协议有哪几种请求方式？ GET, POST 和 HEAD方、OPTIONS, PUT, DELETE, TRACE 和 CONNECT 方法。

# http和https区别

6.http和https区别？

HTTP协议传输的数据都是未加密的，也就是明文的，因此使用HTTP协议传输隐私信息非常不安全，为了保证这些隐私数据能加密传输，于是网景公司设计了SSL（Secure Sockets Layer）协议用于对HTTP协议传输的数据进行加密，从而就诞生了HTTPS。简单来说，HTTPS协议是由SSL+HTTP协议构建的可进行加密传输、身份认证的网络协议，要比http协议安全。

HTTPS和HTTP的区别主要如下：

总的来说： HTTPS=SSL+HTTP

1、https协议需要到ca申请证书，一般免费证书较少，因而需要一定费用。

2、http是超文本传输协议，信息是明文传输，https则是具有安全性的ssl加密传输协议。

3、http和https使用的是完全不同的连接方式，用的端口也不一样，前者是80，后者是443。 （这个只是默认端口不一样，实际上端口是可以改的）

4、http的连接很简单，是无状态的；（什么是Http协议无状态协议？怎么解决Http协议无状态协议？

* 无状态协议对于事务处理没有记忆能力。缺少状态意味着如果后续处理需要前面的信息
* 无状态协议解决办法： 通过1、Cookie 2、通过Session会话保存。

）

HTTPS协议是由SSL+HTTP协议构建的可进行加密传输、身份认证的网络协议，比http协议安全。

# 报文

HTTP请求报文与响应报文格式

请求报文包含三部分： a、请求行：包含请求方法、URI、HTTP版本信息

b、请求头部（head）字段

c、请求内容实体(body)

响应报文包含三部分： a、状态行：包含HTTP版本、状态码、状态码的原因短语

b、响应头部（head）字段

c、响应内容(body)实体

# post请求body

8.常见的 POST 提交数据方式

application/x-www-form-urlencoded multipart/form-data application/json text/xml

# DNS

9.什么是DNS？ 域名解析服务。将主机名转换为IP地址。如将http://www.cnblogs.com/主机名转换为IP地址：211.137.51.78

Http协议中Http1.0与1.1区别？

* 在http1.0中，当建立连接后，客户端发送一个请求，服务器端返回一个信息后就关闭连接，当浏览器下次请求的时候又要建立连接，显然这种不断建立连接的方式，会造成很多问题。
* 在http1.1中，引入了持续连接的概念，通过这种连接，浏览器可以建立一个连接之后，发送请求并得到返回信息，然后继续发送请求再次等到返回信息，也就是说客户端可以连续发送多个请求，而不用等待每一个响应的到来。

HTTP（HyperText Transfer Protocol）协议是基于TCP的应用层协议，它不关心数据传输的细节，主要是用来规定客户端和服务端的数据传输格式，最初是用来向客户端传输HTML页面的内容。默认端口是80。

<https://www.cnblogs.com/wxisme/p/6212797.html>

tcp协议：<https://www.jellythink.com/archives/240>

TCP是面向连接的，无论哪一方向另一方发送数据之前，都必须先在双方之间建立一条连接。在TCP/IP协议中，TCP协议提供可靠的连接服务，连接是通过三次握手进行初始化的。三次握手的目的是同步连接双方的序列号和确认号并交换 TCP窗口大小信息。

三次握手过程：防止了服务器端的一直等待而浪费资源。

第一次握手：建立连接。客户端发送连接请求报文段，将SYN位置为1，Sequence Number为x；然后，客户端进入SYN\_SEND状态，等待服务器的确认；

1. 第二次握手：服务器收到SYN报文段。服务器收到客户端的SYN报文段，需要对这个SYN报文段进行确认，设置Acknowledgment Number为x+1(Sequence Number+1)；同时，自己自己还要发送SYN请求信息，将SYN位置为1，Sequence Number为y；服务器端将上述所有信息放到一个报文段（即SYN+ACK报文段）中，一并发送给客户端，此时服务器进入SYN\_RECV状态；
2. 第三次握手：客户端收到服务器的SYN+ACK报文段。然后将Acknowledgment Number设置为y+1，向服务器发送ACK报文段，这个报文段发送完毕以后，客户端和服务器端都进入ESTABLISHED状态，完成TCP三次握手。

完成了三次握手，客户端和服务器端就可以开始传送数据。以上就是TCP三次握手的总体介绍。

# 那四次分手呢？

当客户端和服务器通过三次握手建立了TCP连接以后，当数据传送完毕，肯定是要断开TCP连接的啊。那对于TCP的断开连接，这里就有了神秘的“四次分手”。

1. 第一次分手：主机1（可以使客户端，也可以是服务器端），设置Sequence Number和Acknowledgment Number，向主机2发送一个FIN报文段；此时，主机1进入FIN\_WAIT\_1状态；这表示主机1没有数据要发送给主机2了；
2. 第二次分手：主机2收到了主机1发送的FIN报文段，向主机1回一个ACK报文段，Acknowledgment Number为Sequence Number加1；主机1进入FIN\_WAIT\_2状态；主机2告诉主机1，我已经知道你没有数据要发送了；
3. 第三次分手：主机2向主机1发送FIN报文段，请求关闭连接，同时主机2进入CLOSE\_WAIT状态；
4. 第四次分手：主机1收到主机2发送的FIN报文段，向主机2发送ACK报文段，然后主机1进入TIME\_WAIT状态；主机2收到主机1的ACK报文段以后，就关闭连接；此时，主机1等待2MSL后依然没有收到回复，则证明Server端已正常关闭，那好，主机1也可以关闭连接了。

TCP协议是一种面向连接的、可靠的、基于字节流的运输层通信协议。TCP是全双工模式，这就意味着，当主机1发出FIN报文段时，只是表示主机1已经没有数据要发送了，主机1告诉主机2，它的数据已经全部发送完毕了；但是，这个时候主机1还是可以接受来自主机2的数据；当主机2返回ACK报文段时，表示它已经知道主机1没有数据发送了，但是主机2还是可以发送数据到主机1的；当主机2也发送了FIN报文段时，这个时候就表示主机2也没有数据要发送了，就会告诉主机1，我也没有数据要发送了，之后彼此就会愉快的中断这次TCP连接。