## **什么是HTTP？**

HTTP是 Hyper Text Transfer Protocol的简称，中文名超文本传输协议，是一个基于TCP/IP 协议来传输超文本的应用层协议。

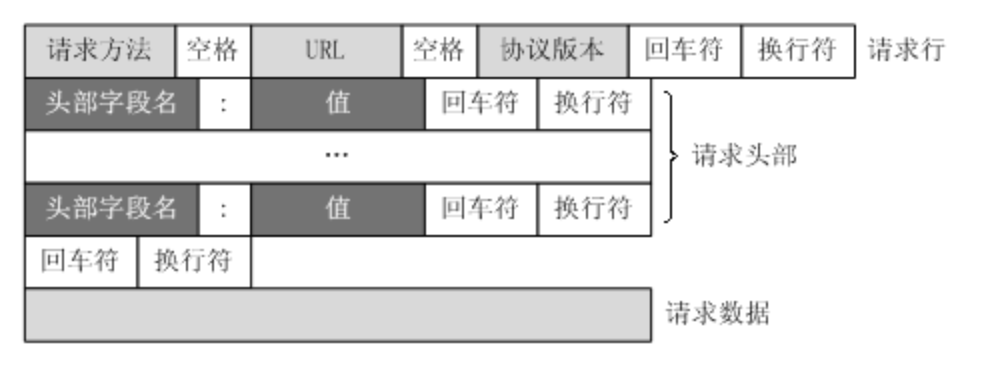
HTTP协议工作与客户端-服务器架构上，浏览器作为HTTP的客户端通过URL向HTTP服务器端发送请求。服务端收到请求后向客户端发送相应信息。HTTP的默认端口为80.

* HTTP是无连接的。即每次只处理一个请求，服务器处理完客户端的请求，并收到客户端的应答后就断开连接。
* HTTP是无状态的。对于请求的处理没有记忆能力，如果后续处理需要前面的信息，则必须重传。

## **HTTP 消息结构**

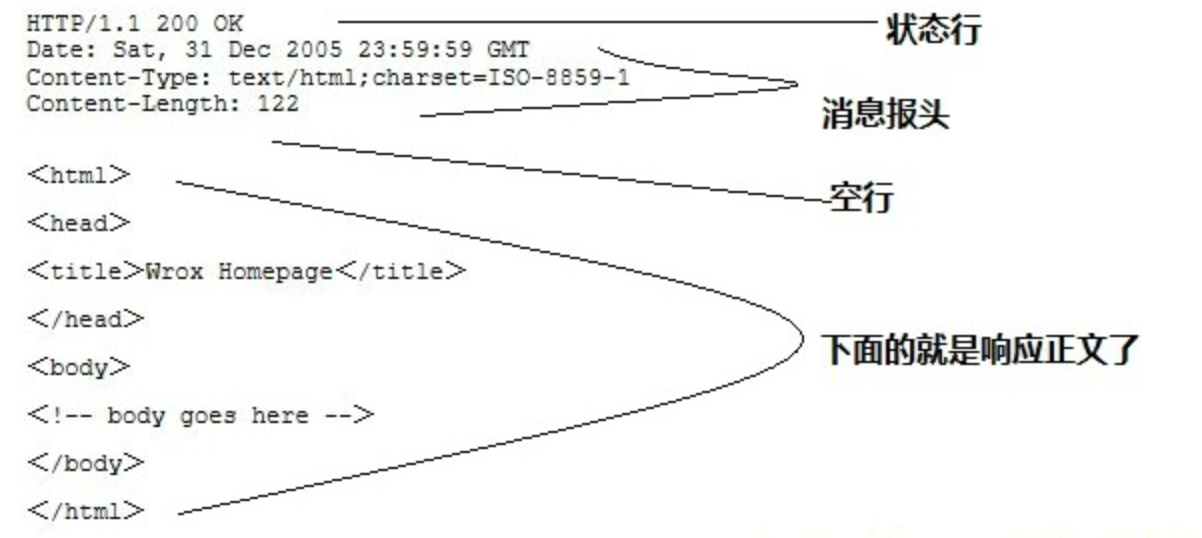
### **客户端请求**

客户端发送一个HTTP请求到服务器的请求消息包括以下格式：请求行（request line）、请求头部（header）、空行和请求数据四个部分组成，下图给出了请求报文的一般格式。



### **服务端响应**

HTTP响应也由四个部分组成，分别是：状态行、消息报头、空行和响应正文。



## **HTTP的请求方法**

HTTP1.0定义了三种方法，分别为GET、POST和HEAD方法。HTTP1.1新增了六种方法PUT、DELETE、OPTIONS、PATCH、TRACE 和 CONNECT 方法。这里只介绍常用的五种方法：

### **1.GET**

用来获取资源，不进行服务器数据操作，因此请求数据中没有Body。

### **2.POST**

向指定资源提交数据进行处理请求（例如提交表单或者上传文件）。数据被包含在请求体中。POST 请求可能会导致新的资源的建立和/或已有资源的修改。

### **3.PUT**

向指定资源提交数据进行处理请求（例如提交表单或者上传文件）。数据被包含在请求体中。POST 请求可能会导致新的资源的建立和/或已有资源的修改。用于修改资源，有Body

### **4.DELETE**

请求服务器删除指定的页面。没有Body

### **5.HEAD**

类似于 GET 请求，只不过返回的响应中没Body内容，用于获取报头

## **HTTP的响应头**

HTTP 的请求头示例如下：

GET /home.html HTTP/1.1

Host: developer.mozilla.org

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.9; rv:50.0) Gecko/20100101 Firefox/50.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8

Accept-Language: en-US,en;q=0.5

Accept-Encoding: gzip, deflate, br

Content-Type:text/html;charset=UTF-8

Referer: https://developer.mozilla.org/testpage.html

Connection: keep-alive

Upgrade-Insecure-Requests: 1

If-Modified-Since: Mon, 18 Jul 2016 02:36:04 GMT

If-None-Match: "c561c68d0ba92bbeb8b0fff2a9199f722e3a621a"

Cache-Control: max-age=0





### **Content-Type**

Content-Type 用于定义网络文件的类型和网页的编码，决定浏览器将以什么样的形式，什么样的编码读取这个文件。语法格式：

Content-Type: text/html; charset=utf-8

Content-Type: multipart/form-data; boundary=something

常见的媒体格式类型如下：

* text/html ： HTML格式
* text/plain ：纯文本格式
* text/xml ： XML格式
* image/gif ：gif图片格式
* image/jpeg ：jpg图片格式
* image/png：png图片格式

以application开头的媒体格式类型：

* application/xhtml+xml ：XHTML格式
* application/xml： XML数据格式
* application/atom+xml ：Atom XML聚合格式
* application/json： JSON数据格式
* application/pdf：pdf格式
* application/msword ： Word文档格式
* application/octet-stream ： 二进制流数据（如常见的文件下载）
* application/x-www-form-urlencoded ： 中默认的encType，form表单数据被编码为key/value格式发送到服务器（表单默认的提交数据的格式）

另外一种常见的媒体格式是上传文件之时使用的：

* multipart/form-data ： 需要在表单中进行文件上传时，就需要使用该格式

## **一次完整的HTTP请求**

首先进行DNS域名解析（本地浏览器缓存、操作系统缓存或者DNS服务器），首先会搜索浏览器自身的DNS缓存（缓存时间比较短，大概只有1分钟，且只能容纳1000条缓存）

如果浏览器自身的缓存里面没有找到，那么浏览器会搜索系统自身的DNS缓存

如果还没有找到，那么尝试从 hosts文件里面去找

在前面三个过程都没获取到的情况下，就去域名服务器去查找，

三次握手建立 TCP 连接

在HTTP工作开始之前，客户端首先要通过网络与服务器建立连接，HTTP连接是通过 TCP 来完成的。HTTP 是比 TCP 更高层次的应用层协议，根据规则，只有低层协议建立之后，才能进行高层协议的连接，因此，首先要建立 TCP 连接，一般 TCP 连接的端口号是80；

客户端发起HTTP请求

服务器响应HTTP请求

客户端解析html代码，并请求html代码中的资源

浏览器拿到html文件后，就开始解析其中的html代码，遇到js/css/image等静态资源时，就向服务器端去请求下载

客户端渲染展示内容

关闭 TCP 连接

一般情况下，一旦服务器向客户端返回了请求数据，它就要关闭 TCP 连接，然后如果客户端或者服务器在其头信息加入了这行代码 Connection:keep-alive ，TCP 连接在发送后将仍然保持打开状态，于是，客户端可以继续通过相同的连接发送请求，也就是说前面的3到6，可以反复进行。保持连接节省了为每个请求建立新连接所需的时间，还节约了网络带宽。