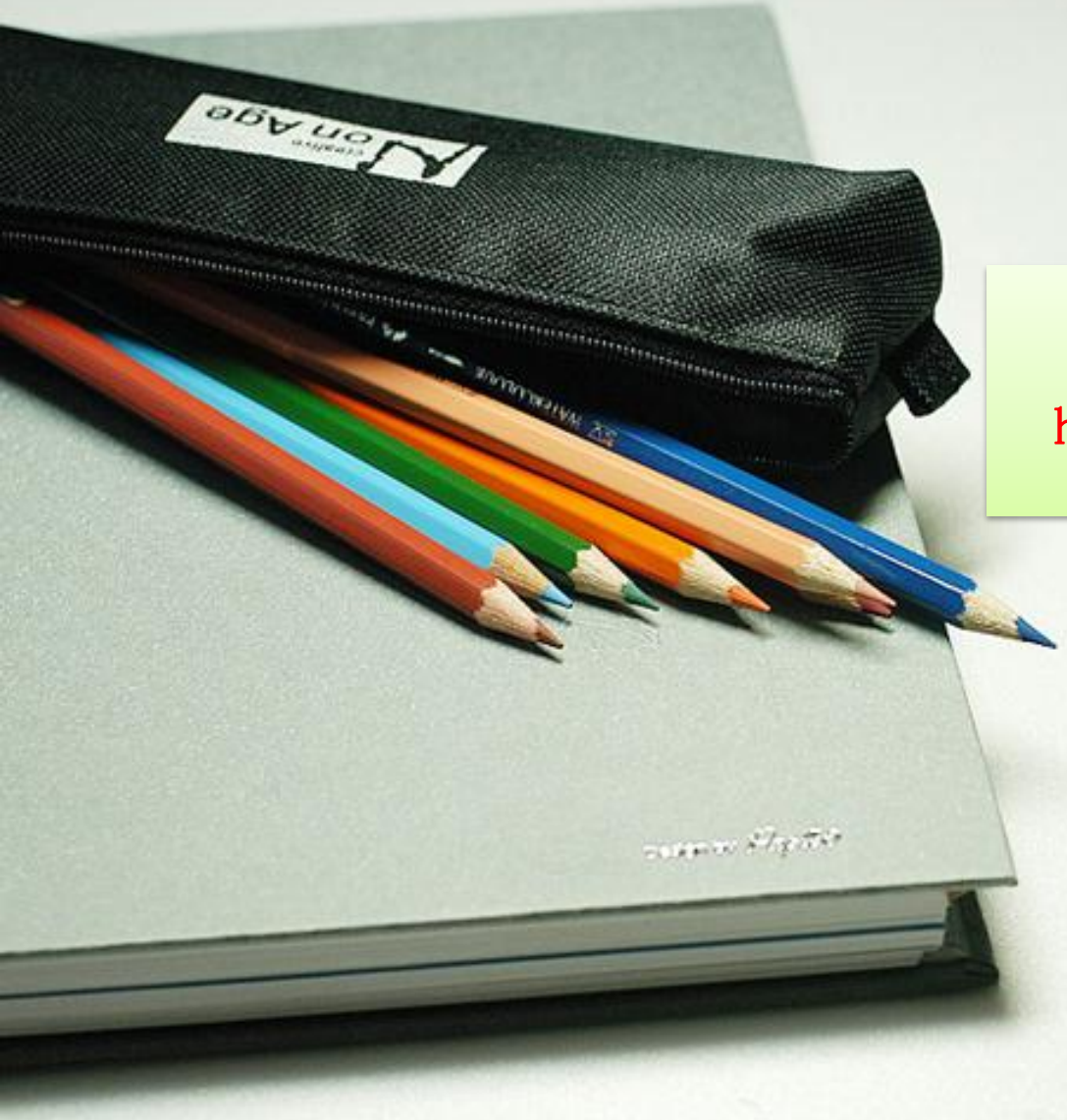


北京圣思园科技有限公司
http://www.shengsiyuan.com

主讲人：张龙

Http协议介绍

- 课程内容
  - 介绍Http协议的相关内容
  - Http URL
  - Http请求和响应
  - Http消息

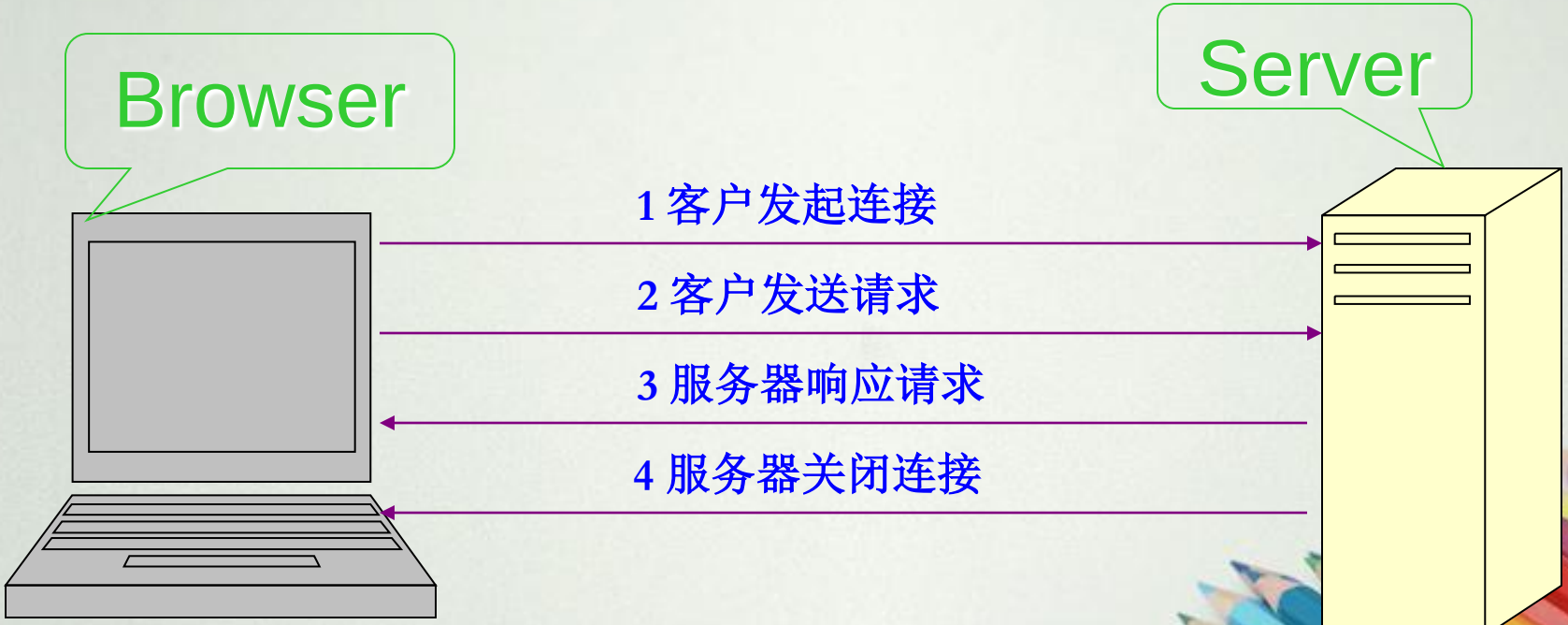
Http协议介绍

- 课程目标
  - 绝大多数的Web开发，都是构建在Http协议之上的Web应用，理解和掌握Http协议，将有助于我们更好地学习和掌握Servlet和JSP技术，以及其他相关的Web开发技术

Http协议介绍

- Http（Hypertext Transfer Protocol）超文本传输协议，从1990年开始就在WWW上广泛应用，是现今在WWW上应用得最多的协议，目前的版本为1.1
- Http是应用层协议，当你上网浏览网页的时候，浏览器和服务器之间就会通过Http在Internet上进行数据的发送和接收
- Http是一个基于请求/响应模式的，无状态的协议(request/response based ,stateless protocol)

浏览器与服务器通信的过程(HTTP1.0)



持续连接（Persistent Connections）

- 在Http1.0中，当连接建立后，浏览器发送一个请求，服务器回应一个消息，之后，连接就被关闭。当浏览器下次请求的时候，需要重新建立连接，很显然这种需要不断建立连接的通信方式开销比较大。早期的Web页面通常只包含HTML文本，因此即使建立连接的开销比较大，也不会有太大的影响。而现在的Web页面往往包含多种资源（图片，动画，声音等），每获取一种资源，就建立一次连接，这样就增加了HTTP服务器的开销，造成了Internet上的信息堵塞

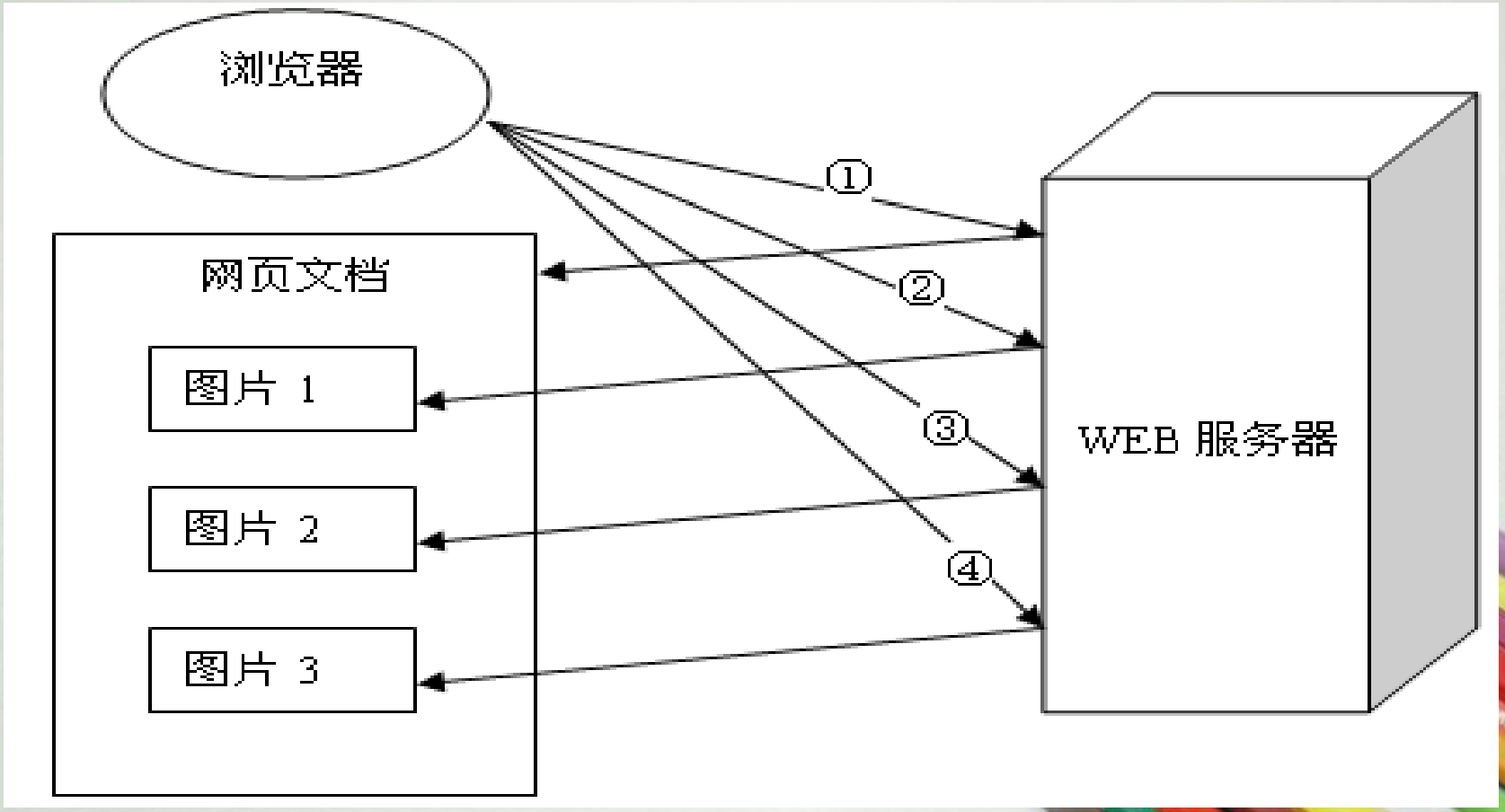
持续连接（Persistent Connections）

- 因此在Http1.1版本中，给出了一个持续连接（Persistent Connections）的机制，并将其作为Http1.1中建立连接的缺省行为。通过这种连接，浏览器可以在建立一个连接之后，发送请求并得到回应，然后继续发送请求并再次得到回应。而且，客户端还可以发送流水线请求，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

持续连接（Persistent Connections）

- 一个WEB站点每天可能要接收到上百万的用户请求，为了提高系统的效率，HTTP 1.0规定浏览器与服务器只保持短暂的连接，浏览器的每次请求都需要与服务器建立一个TCP连接，服务器完成请求处理后立即断开TCP连接，服务器不跟踪每个客户也不记录过去的请求。但是，这也造成了一些性能上的缺陷，例如，一个包含有许多图像的网页文件中并没有包含真正的图像数据内容，而只是指明了这些图像的URL地址，当WEB浏览器访问这个网页文件时，浏览器首先要发出针对该网页文件的请求，当浏览器解析WEB服务器返回的该网页文档中的HTML内容时，发现其中的<img>图像标签后，浏览器将根据<img>标签中的src属性所指定的URL地址再次向服务器发出下载图像数据的请求

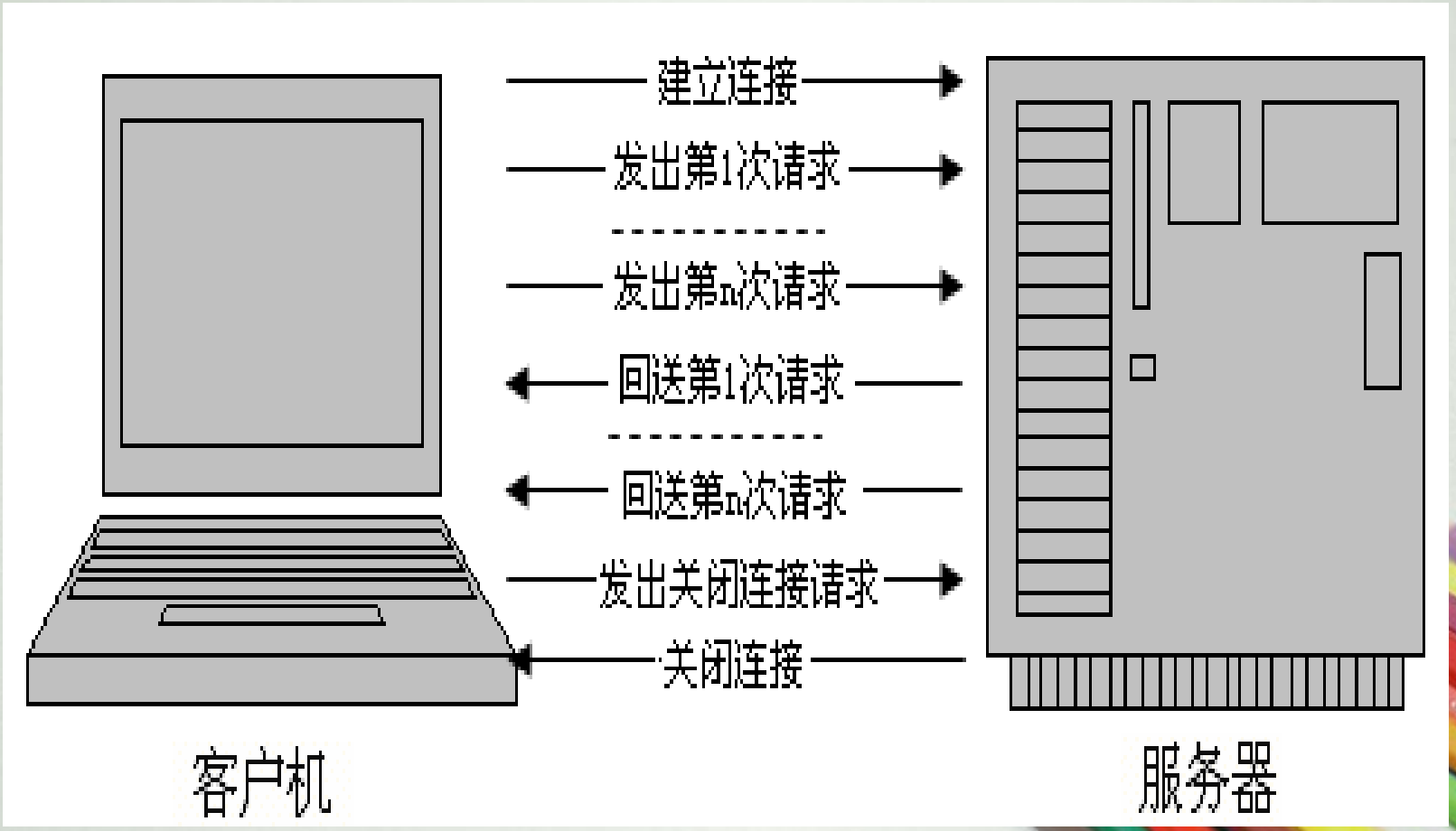
持续连接（Persistent Connections）



持续连接（Persistent Connections）

- 显然，访问一个包含有许多图像的网页文件的整个过程包含了多次请求和响应，每次请求和响应都需要建立一个单独的连接，每次连接只是传输一个文档和图像，上一次和下一次请求完全分离。即使图像文件都很小，但是客户端和服务器端每次建立和关闭连接却是一个相对比较费时的过程，并且会严重影响客户机和服务器的性能。当一个网页文件中包含Applet，JavaScript文件，CSS文件等内容时，也会出现类似上述的情况。

浏览器与服务器通信的过程(HTTP1.1)



HTTP URL

- 格式：
  - http://host[: port] [abs_path]
  - 其中http表示要通过HTTP协议来定位网络资源。
  - Host表示合法的Internet主机域名或IP地址（以点分十进制格式表示）
  - Port用于指定一个端口号，拥有被请求资源的服务器主机监听该端口的TCP连接。如果port是空，则使用缺省的端口80。
  - abs_path指定请求资源的URI（Uniform Resource Identifier,统一资源标识符），如果URL中没有给出abs_path,那么当它作为请求URI时，必须以“/”的形式给出。通常这个工作浏览器就帮我们完成了

URL 与 URI

- URI 纯粹是一个符号结构，用于指定构成Web资源的字符串的各个不同部分
- URL 是一种特殊类型的URI，它包含了用于查找某个资源的足够的信息。其他的URI，例如：mailto:zhanglong217@yahoo.com.cn,则不属于URL，因为它里面不存在根据该标识符来查找的任何数据。这种URI成为URN（通用资源名）

HTTP 请求

- 客户端通过发送HTTP请求向服务器请求对资源的访问
- HTTP请求由三部分组成，分别是：请求行，消息报头，请求正文

请求行

- 请求行以一个方法符号开头，后面跟着请求URI和协议的版本，以CRLF作为结尾。请求行以空格分隔。除了作为结尾的CRLF外，不允许出现单独的CR或LF字符，格式如下：
  - Method Request-URI HTTP-Version CRLF
- Method表示请求的方法，Request-URI是一个统一资源标识符，标识了要请求的资源，HTTP-Version表示请求的HTTP协议版本， CRLF表示回车换行。例如：
  - GET /test.html HTTP/1.1 (CRLF)

HTTP请求 — 方法

方法	作用
GET	请求获取由Request-URI所标识的资源
POST	在Request-URI所标识的资源后附加新的数据
HEAD	请求获取由Request-URI所标识的资源的响应消息报头
DELETE	请求服务器删除由Request-URI所标识的资源
TRACE	请求服务器回送收到的请求信息，主要用于测试或诊断
CONNECT	保留将来使用
OPTIONS	请求查询服务器的性能，或者查询与资源相关的选项和需求
PUT	请求服务器存储一个资源，并用Request-URI作为其标识

HTTP请求 — 方法

- GET方法用于获取由Request-URI所标识的资源的信息，常见形式是：
  - GET Request-URI HTTP/1.1
- 当我们通过在浏览器的地址栏中直接输入网址的方式去访问网页的时候，浏览器采用的就是GET方法向服务器获取资源

HTTP请求 — 方法

- POST方法用于向服务器发送请求，要求服务器接受附在请求后面的数据。POST方法在表单提交的时候用的最多
- 采用POST方法提交表单的例子

  POST /login.jsp HTTP/1.1 (CRLF)
  Accept:image/gif (CRLF) (….)
  Host:www.sample.com (CRLF)(….)
  ….
  Cache-Control:no=cache (CRLF)
  (CRLF)
  username=hello&password=123456

HTTP请求 — 方法

- HEAD方法与GET方法几乎是一样的，他们的区别在于HEAD方法只是请求消息报头，而不是完整的内容。对于HEAD请求的回应部分来说，它的HTTP头部中包含的信息与通过GET请求所得到的信息是相同的。利用这个方法，不必传输整个资源的内容，就可以得到Request-URI所标识的资源的信息。这个方法通常用于测试超链接的有效性，是否可以访问，以及最近是否更新等

HTTP请求 — 方法

- 当我们在HTML中提交表单时，浏览器会根据你的提交方法是get还是post，采用相应的在HTTP协议中的GET或POST方法，向服务器发出请求。
- 注意：在HTML文档中，书写get和post，不区分大小写，但HTTP协议中的GET和POST只能是大写形式

HTTP响应

- 在接收和解释请求消息后，服务器会返回一个HTTP响应消息
- 与HTTP请求类似，HTTP响应也是由三个部分组成，分别是：状态行，消息报头，响应正文

HTTP响应—状态行

- 状态行由协议版本，数字形式的状态代码，相应的状态描述组成，各元素之间以空格分隔，除了结尾的CRLF（回车换行）序列外，不允许出现CR或LF字符。格式如下：
  - HTTP-Version Status-Code Reason-Phrase CRLF
- HTTP-Version表示服务器HTTP协议的版本，Status-Code表示服务器发回的响应代码，Reason-Phrase表示状态代码的文本描述，CRLF表示回车换行，例如：
  - HTTP/1.1 200 OK (CRLF)

HTTP响应—状态代码与状态描述

- 状态代码由三位数字组成，表示请求是否被理解或被满足，状态描述给出了关于状态代码的简短文本描述
- 状态代码的第一个数字定义了响应的类别，后面两个数字没有具体的分类。第一个数字有五种可能的取值
  - 1xx：指示信息—表示请求已接收，继续处理
  - 2xx：成功—表示请求已经被成功接收，理解，接受
  - 3xx：重定向—要完成请求必须进行更进一步的操作
  - 4xx：客户端错误—请求有语法错误或请求无法实现
  - 5xx：服务器端错误—服务器未能实现合法的请求

HTTP响应—状态代码与状态描述

状态代码	状态描述	说明
200	OK	客户端请求成功
400	Bad Request	由于客户端请求有语法错误，不能被服务器所理解
401	Unauthorized	请求未经授权，这个状态代码必须和WWW-Authenticate报头域一起使用
403	Forbidden	服务器收到请求，但是拒绝提供服务，服务器通常会在响应正文中给出不提供服务的原因
404	Not Found	请求的资源不存在，例如：输入了错误的URL
500	Internal Server Error	服务器发生不可预期的错误，导致无法完成客户的请求
503	Service Unavailable	服务器当前不能够处理客户端的请求，在一段时间之后，服务器可能会恢复正常

HTTP消息

- HTTP消息由客户端到服务器的请求和服务器到客户端的响应组成。请求消息和响应消息都是由开始行，消息报头（可选），空行（只有CRLF的行），消息正文（可选）组成。
- 对于请求消息，开始行就是请求行，对于响应消息，开始行就是状态行

实验

- 实验工具 ： Telnet
- HTTP协议与TELNET协议都是基于TCP协议