1.1 生成HTTP请求信息
1.1.1 探索之旅从输入网址开始
1.1.2 浏览器先要解析URL
1.1.3 省略文件名的情况
1.1.4 HTTP的基本思路

- **HTTP协议**定义了客户端和服务器之间交互的消息内容和步骤
  - 基本思路：
    - 客户端向服务器发送请求消息
      - 请求消息内容：“对什么” + “进行怎样的操作”
      - 对什么：URI，存放网页数据的文件名或者是一个CGI程序的文件名
      - 怎么做：方法
      - GET： 访问服务器获取网页数据
      - POST：将网页输入框中的信息发送给Web服务器
    - 收到**请求消息**之后，Web服务器会对其中的内容进行解析，通过URI和方法来判断“对什么”“进行怎样的操作”，并根据这些要求来完成自己的工作，然后将结果存放在**响应消息**中。
    - 响应消息的开头有一个**状态码**，它用来表示操作的执行结果是成功还是发生了错误
      - 访问Web服务器时，遇到找不到的文件就会显示出404 Not Found的错误信息
    - 响应消息会被发送回客户端
    - 客户端收到之后，浏览器会从消息中读出所需的数据并显示在屏幕上

1.1.5 客户端生成HTTP请求信息 （http基本思路第一步）

- 浏览器会按照规定的格式来生成请求消息

```
a)请求消息
<方法><空格><URL><空格><HTTP版本>     /// 第一行称为请求行，通过这一行可以大致了解请求的内容
<字段名>:<字段值>    // 这一部分称为消息头，每行包含一个头字段，用于表示请求的附加信息。消息头的行数根据具体情况可变，一直延伸到空行为止
…
…
…
<空行>
<消息体>            //消息体包含客户端向服务器发送的数据，例如用POST方法向Web服务器发送的网页表单数据
```

- 第一行是请求行
  - 最开头的**方法**：告诉Web服务器它应该进行怎样的操作（很多种，需要判断）
    - 判断取决于浏览器的工作状态：
      - eg. 向Web服务器**发送请求消息**： 浏览器顶部的地址栏输入网址，网页中的超级链接，表单中填写信息后点击的“提交”按钮
      - 场景出发浏览器的工作，选用哪种方法由场景决定
      - 场景：在地址栏输入网址并显示网页 ==> 方法：GET
      - 场景：表单 ==> 方法：根据HTML源代码中**表单的属性中指定**使用哪种方法来发送请求，可能是GET也可能是POST
        - Tip：GET方法能够发送的数据只有几百个字节，如果表单中的数据超过这一长度，则必须用POST方法来发送。

```
表单HTML源代码：
<form method=“GET” action=“/cgi/sample.cgi”>  // 对方法的指定（“GET”）和接受表单数据的程序文件名
    <input type=“text” name=“Field1” size=“20”>            //输入的内容 ==> ABCDEFG
    <input type=“submit” value=“SEND” name=“SendButton”>  //提交键
    <input type=“reset” value=“RESET” name=“ResetButton”> //重置键
</form>

请求行【当method=“GET”时的消息】：
GET /cgi/sample.cgi?Field1=ABCDEFG&SendButton=SEND HTTP/1.1  ----  <方法><空格><URL><空格><HTTP版本>
OR 【当method=“POST”时的消息】
POST /cgi/sample.cgi HTTP/1.1
(若干行头字段)
Field1=ABCDEFG&SendButton=SEND // POST方法时，表单填写的信息（ABCDEFG）写在消息体中
```

- 第二行开始为消息头：存放额外的详细信息
  - 消息头的规格中定义了很多项目的细节信息：
    - 日期、客户端支持的数据类型、语言、压缩格式、客户端和服务器的软件名称和版本、数据有效期和最后更新时间等
    - 要想了解这些头字段信息的意思，就需要对**HTTP协议**有非常深入的了解
    - 消息头内容随着浏览器类型、版本号、设置等的不同而不同，大多数情况下消息头的长度为几行到十几行不等
- 当使用POST方法时，需要将表单中填写的信息写在**消息体**中。

```
b)响应消息
<HTTP版本><空格><状态码><空格><响应短语>  //状态行，用来解释状态码的短语
<字段名>:<字段值>  //消息头
…
…
…
<空行>
<消息体>  //消息体包含服务器向客户端发送的数据，例如从文件中读取的数据，或者CGI应用程序输出的数据等。消息体的内容作为二进制数据来处理
```

1.1.6 发送请求后会收到响应

- 将请求消息发送出去后，Web服务器会返回响应消息。关于响应消息我们将在第6章详细介绍
- 第一行的内容为状态码和响应短语（表示的内容一致，用途不同），用来表示请求的执行结果是成功还是出错
  - 状态码是一个数字，主要用来向程序告知执行的结果
  - 响应短语是一段文字，用来向人们告知执行的结果
- 返回响应消息之后，浏览器会将数据提取出来并显示在屏幕上，我们就能够看到网页的样子了
  - 如果网页只有文字，浏览器请求&服务器响应到此为止
  - 网页中还包括图片等资源，会在相应位置嵌入形如 <img src=“image1.jpg”>的标签，浏览器会在显示文字时搜索相应的标签，当遇到图片相关的标签时，会在屏幕上留出用来显示图片的空间，再次访问Web服务器请求获取相应的图片并显示在预留的空间中。
    - 比如一个网页中包含三张图片，那么获取网页加上获取图片，一共需要向Web服务器发送4条请求
- 浏览器需要判断所需的文件，然后获取这些文件并显示在屏幕上，这一系列工作的整体指挥也是浏览器的任务之一，而Web服务器却毫不知情
- 服务器的任务是对每一条单独的请求返回1条响应

1.1.7 实例

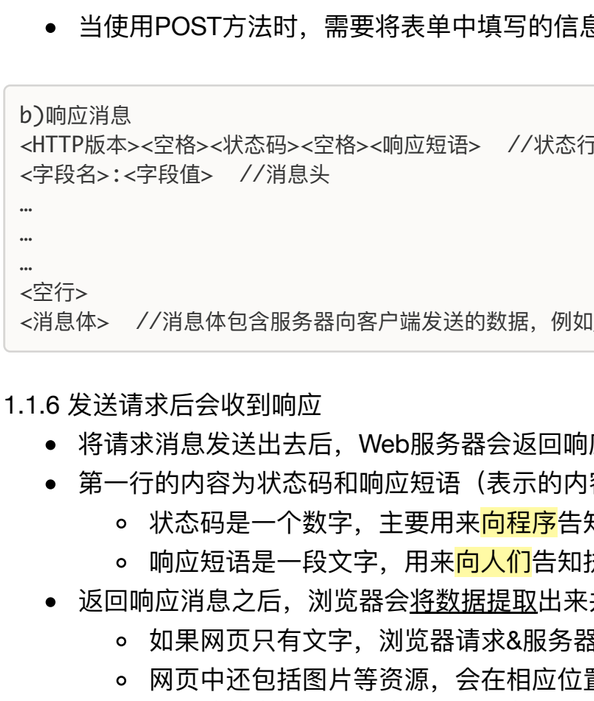

图 1.7 HTTP 消息示例

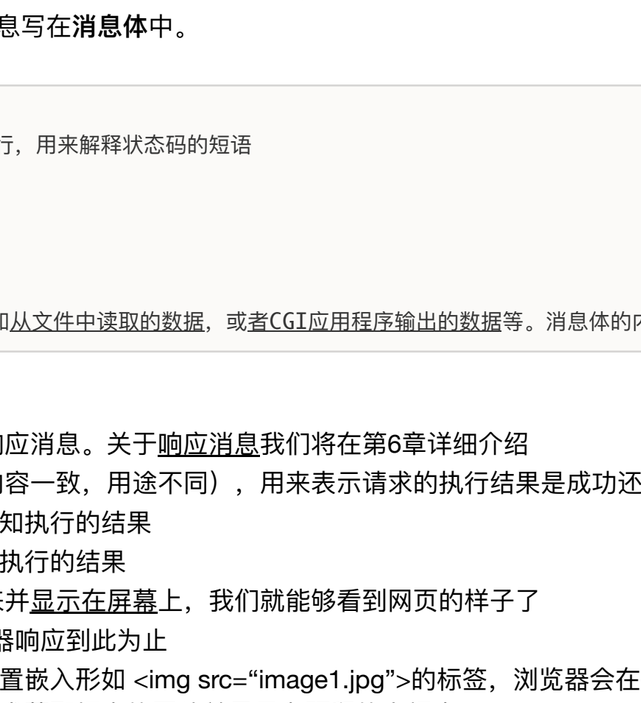

图 1.7 （续）

图 1.7 （续）