

软件工具：第 2 部分

(COMS10012 / COMSM0085)

本周（第 6 周）：

- HTTP的
- [HTML全

即将到来：

- CSS（第 7 周）
- JS（第 8 周）

HTTP的

超文本传输协议

迫在眉睫的问题：

- 什么是“超文本”？
- 什么是“传输协议”？

HTTP：协议

由欧洲核子研究中心的Tim Berners-Lee于1989年开发。

作为一组 *RFC* 规范发布。

您本周的练习包括阅读 RFC 7230 的部分内容！

协议

协议是关于系统的协作组件应如何交互的计划。

简单的例子：

客户：给我块 #200

服务器：给你： *0A 2F EE ...*

协议：状态通信

响应不仅仅是请求的数据。例如：

客户：给我块 #45000

服务器：对不起，我找不到那个块。

或

客户：Gimme 块 #200

服务器：对不起，我不知道你在问什么。

或

客户：给我块 #200

服务器：我想你的意思是块 #201，即 *0A 2F EE*

HTTP 状态通信

HTTP传输超文本（当然）。这是协议的*数据* 关注。与前面的玩具示例一样，HTTP 也是一个客户端-服务器 具有请求-响应语义的协议。

但 HTTP 也会传输有关通信方状态的*元数据*。这有两个关键机制：

- 请求和响应标头
- 状态代码

重要的是要理解：元数据是 HTTP 协议的重要组成部分。但是，*元数据与数据*是分开的。超文本文档不必通过HTTP到达。

HTTP 消息结构

从 RFC：

```
HTTP-message  = start-line
                *( header-field CRLF )
```

```
CRLF
[ message-body ]
```

A 可以是 a（用于请求）或 a（用于 响应）。 start-line request-line status-line

标头是可选的，内部是 . field-name : field-value

HTTP 请求

根据 RFC 7230，HTTP 请求行的格式为：

```
method SP request-target SP HTTP-version CRLF
```

- SP = 空格
- CRLF = 回车

很少改变（尽管它可以！ HTTP-version

要了解的关键要素是 和 。 method request-target

HTTP 方法

- GET：检索目标资源的副本
- POST：将有效负载数据提交到目标资源
- HEAD：检索相应 GET 的元数据
- PUT：将目标资源替换为有效负载
- DELETE：删除目标资源

在实践中，许多服务器不会实现或会忽略或请求，以支持使用请求的自定义语义。

```
DELETE PUT POST
```

HTTP 请求目标

简单地说，就是您在请求中定位的资源。

与 URI 的组件相关（统一资源 标识符）。 path query

- path：例如，， 或或 / /files/index.html /user/george/
- query：例如， ?name=welcome&action=view
 - 由一系列参数（，）组成，其值为（， name action welcome view）

同一资源对不同的字符串的响应可能不同。 path query

HTTP 状态行

现在是回应。状态行的格式为：

' status-line = HTTP 版本 SP 状态代码 SP 原因短语 CRLF '

- HTTP-version 我们已经在请求中介绍了这一点
- status code 具有特定含义的 3 位代码
- reason-phrase 说明状态代码

HTTP 状态代码

可以具有非常具体的含义，但按语义的第一个数字分组 意义：

- 1xx 信息（例如， ）。 100 Continue
- 2xx 成功（例如， 200 OK ）
- 3xx 重定向（例如， 301 Moved Permanently ）
- 4xx 客户端错误（例如， 403 Forbidden ）
- 5xx 服务器错误（例如， 500 Internal Server Error ）

HTTP Exchange 示例

请求

```
GET /index.html HTTP/1.1
Host: www.bristol.ac.uk
Connection: close
```

响应

```
HTTP/1.1 200 OK
Content-Type: text/html; charset=UTF-8
Content-Length: 1009
```

```
<!DOCTYPE html>
```

```
<html lang="en">
```

...

内容类型

响应中的一个重要标头是 `Content-Type`

这仍然是元数据。告知客户端响应正文的*数据类型* 将包含。

对于浏览器非常重要，因为客户端可以解释响应正文 人类。

如果我们将响应从 更改为 `Content-Type text/html` `text/plain`

[HTML全

最后，超文本。这个概念是交互式文本。其核心是考虑“带有超链接的文本”。

HTML 是超文本标记语言。一种用于“标记”文本以使其具有交互性（和结构化）的语言。

HTML 文档阅读器（如浏览器）必须解释标记并呈现 结果给用户。

关键要素

- **标记**表示有意义的文档组件。
- 标签和标签内文本的嵌套可以组织文档结构。
- 标记可以具有影响其语义解释的**属性**。

重要的 HTML 标签

一切都嵌套在 `<html>`

`<head>` 与 `<body>`

注意：这是在HTML文档本身中！ 与 HTTP 无关 头。 `<head>`

- `<head>` 包含和标记，用于描述 `<title>` `<meta>` `<body>`
- `<body>` 包含文档的“可见”部分。
 - 大多数文档组件应放置在正文中。
 - 常见内容包括 `„ ... <p>` `<div>` ``

HTML 文档示例

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <title>A web page</title>
  </head>
  <body>
    <h1>An example web-page</h1>
    <p>A paragraph of text here, perhaps with <a href='./another_page/'>a link</a></p>
  </body>
</html>
```

HTML 演示文稿

GUI 浏览器有常见的视觉默认值（例如，蓝色下划线）。

但是文档的语义结构（本来是）与其介绍。

- 考虑浏览器应如何向盲人用户读取 HTML 文档。

在浏览器中，HTML 文档中元素的表示由样式表控制。

下周，我们将讨论如何通过级联样式表（CSS）应用样式。

本周练习

1. 执行简单的 HTTP 客户端-服务器交互
2. 研究 URI 的构造方式
3. 启动“真实”Web 服务器以提供文件
4. 编写简单的 HTML 文档
5. 使用模板引擎让服务器生成 HTML 文档

周五见。