

# Web工作原理

---

北京理工大学计算机学院  
金旭亮

# 不要泯灭你的好奇心！

① 我们常说的Web是什么？

② Web上如此丰富的信息，是怎样放上去的？

③ Web网站是怎么回事？网站和网页是什么关系？

④ 为什么我在浏览器中输入一个神秘的“http://...”，互联网居然知道我想访问的那个网站在哪里！

⑤

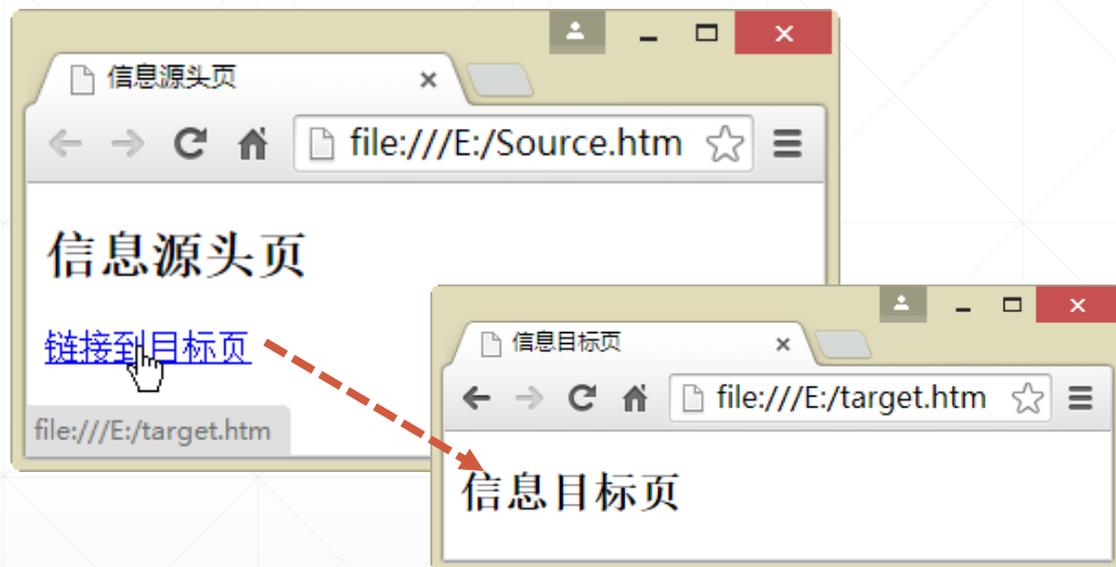
.....

# Web上的资源

---

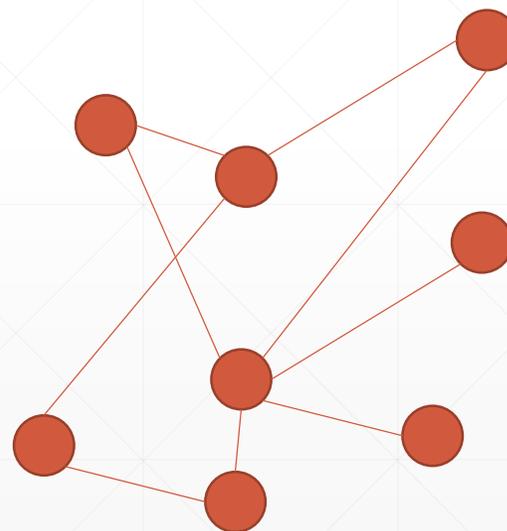
你从哪里来？我的朋友！

互联网上的信息，被看成是一种“**Web Resource (Web资源)**”。



相互链接的Web Resource构成了互联网信息的**网状结构**。

Web资源最常见的形式是**网页**。



# 设计Web网页的三大要素

Web网页是最主要的一种Web资源，通常，我们使用以下三种“东西”设计一个网页：

HTML

定义网页要显示的内容和总体结构

CSS

定义样式规则，确定网页在浏览器中呈现的样子

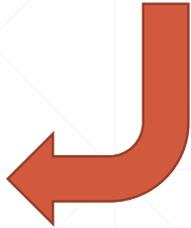
JavaScript

编写程序，让网页能响应用户的操作，完成各种各样的功能

Web资源太多了，我们怎么样才找到需要的资源？

为了能正确地定位特定的资源，可以给每个资源一个唯一的标识，“**URL (Uniform Resource Locator: 统一资源定位器)**”起到的就是这个作用，俗称“**网址**”。

<http://study.163.com/curricula/cs/grade-4.htm>

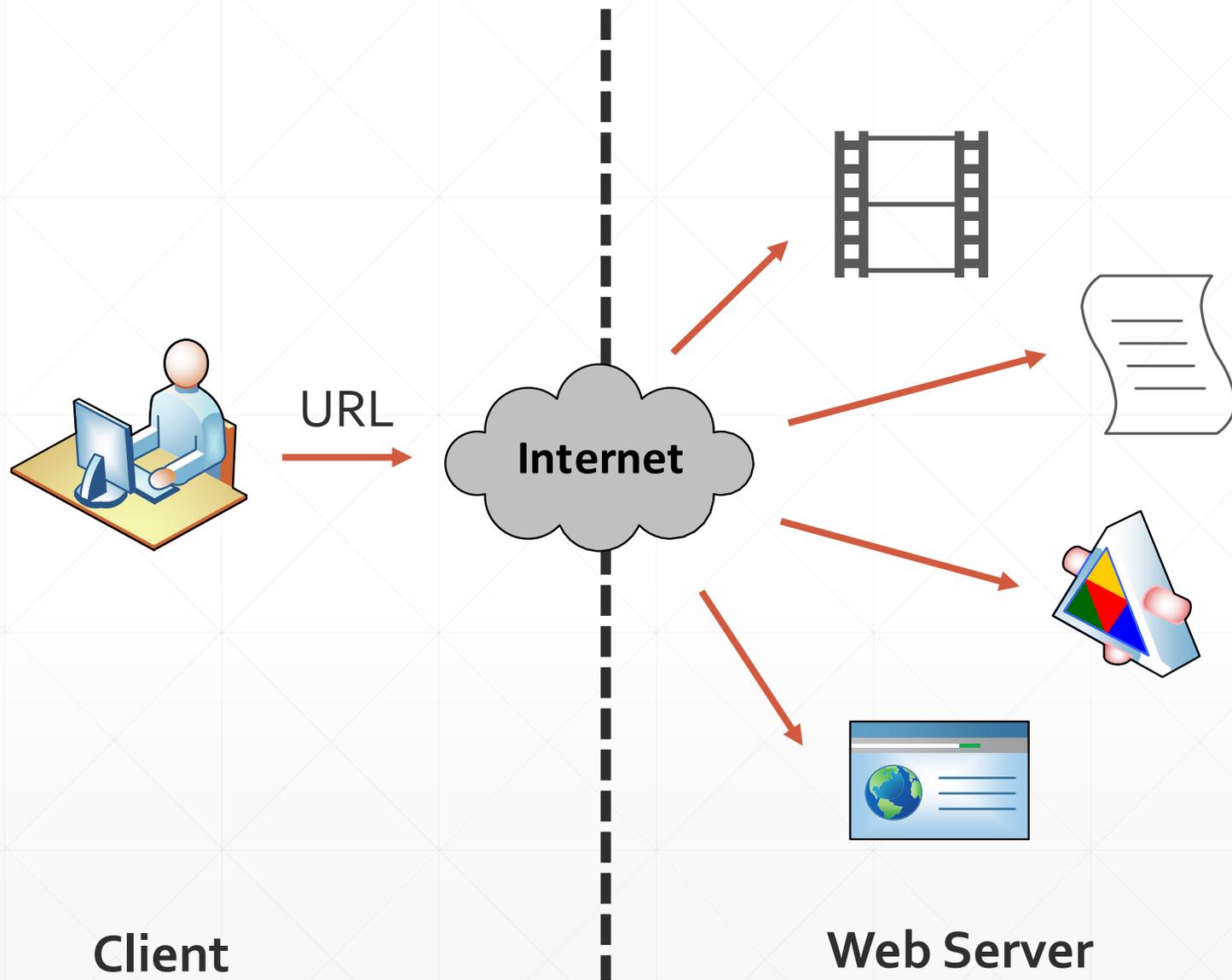


整个网页是一个Web Resource



网页中的每个图片，其实也是一个Web Resource



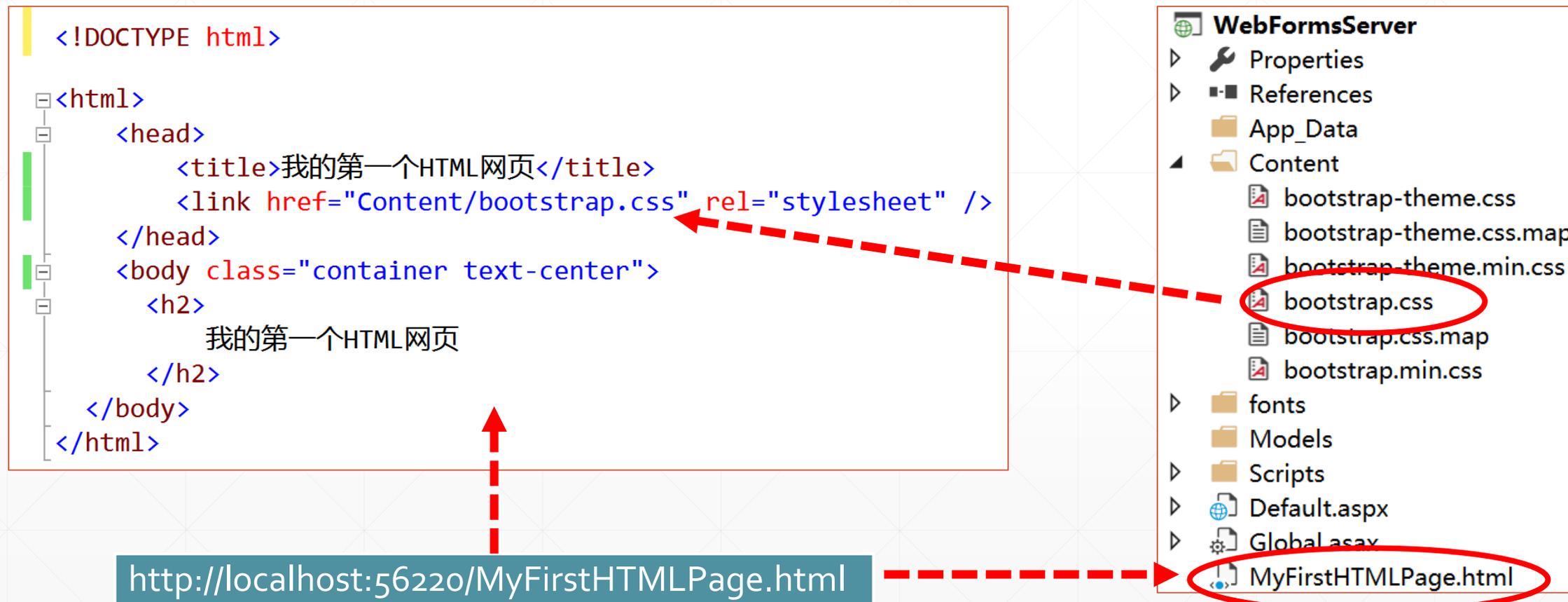


URL实际上把Web Server与Web Client两种不同类型的应用程序给**隔离**开了，从而让两者可以相互独立地采用不同的技术、平台和软件架构开发。

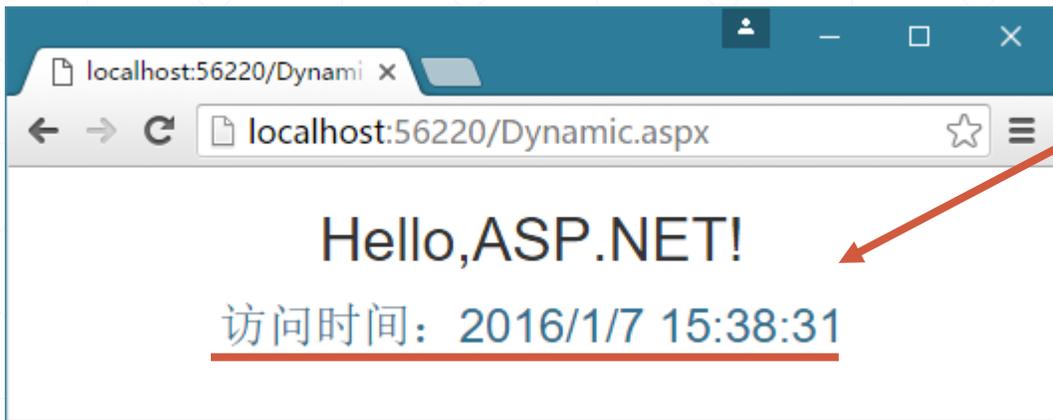
# 有两种主要的Web Resource提供方式:



早期的互联网站，基本上是网页的“仓库”，网页一旦创建好，不管什么人什么时候访问，都是一样的，所以是“静态”的。



用户看到的这个网页，实际上是由Web应用程序**动态**生成并装配出来的。



现在的互联网站，都具备很强的交互性，绝大多数内容都是“动态生成”的，早已进入“**动态**”时代。

```
无标题 - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
<!DOCTYPE html>
<html>
  <head>
    <link href="Content/bootstrap.css" rel="stylesheet" />
  </head>
  <body class="container text-center">
    <div>
      <h2>Hello, ASP.NET!</h2>
      <span id="lblInfo" class="text-info" style="font-size:X-Large;">访问时间: 2016/1/7 15:38:31</span>
    </div>
  </body>
</html>
```

动态生成的部分

# 两种风格的“动态资源”URL-1

当前互联网上有两种主要风格的URL，一种直接在URL中指明文件（比如XXX.php，XXX.aspx等），可称为“**具体的**”URL，其响应通常基于一个网页模板而生成：

```
http://www.example.com/URL.aspx?first=100&second=abc
```

JSP，PHP，ASP.NET Web Forms 等开发技术均支持这种开发方式。



1. 这种网页的生成方式，你可以把网页模板想象成是一张试卷，上面有很多空和问题要回答。
2. 生成网页的过程就是你考试时在试卷上填空和答题的过程，考完交卷，你得到一个考试分数作为考试结果。
3. 类似地，用户所得到的网页生成的结果就是你最后在浏览器中所看到那个最终的网页。

## 两种风格的“动态资源”URL-2

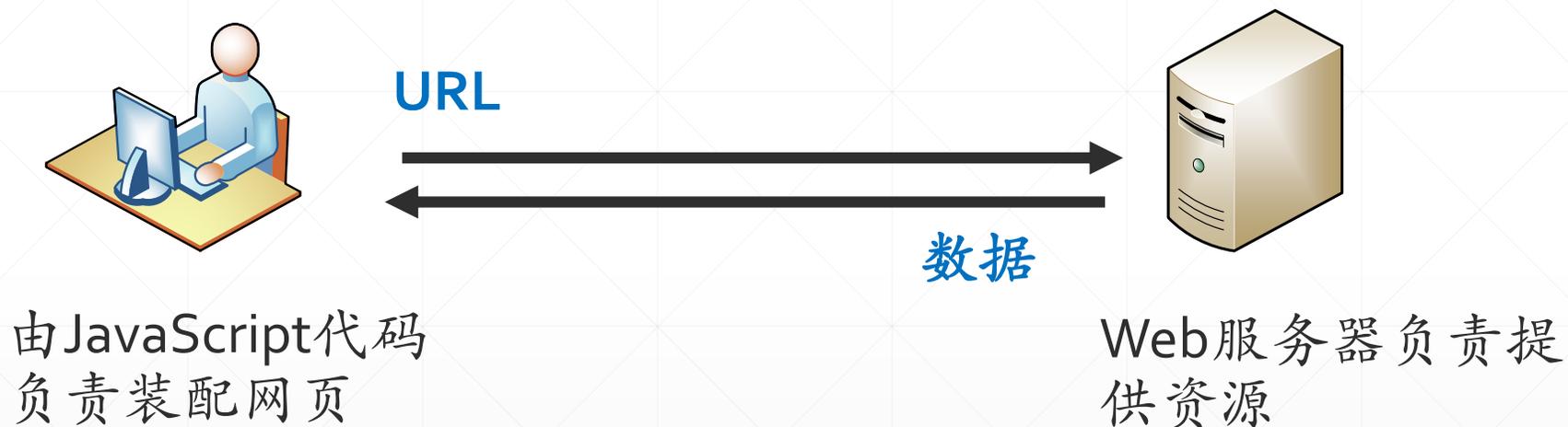
另一种是更强调URL的直观性，可称为“**抽象的**”URL，这种类型的URL，通常需要经过一个“**路由系统**”的翻译过程，其响应的生成过程五花八门：

`http://en.wikipedia.org/wiki/Uniform_resource_locator`



作为入门级别的课程，这块内容不介绍。

动态网页的生成工作，过去严重依赖于Web Server（比如ASP.NET Web Forms和MVC开发的许多网站），近年来流行一种“**单页面应用（SPA: Single Page Application）**”，主要依靠JavaScript程序在客户端（包括PC和手机）的浏览器中完成这个任务。



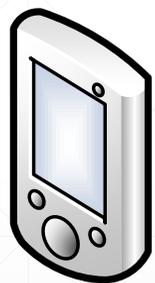
单页面应用通常分为两块：服务端技术框架 + 前端技术框架

ASP.NET Web API等

AngularJS, jQuery等

# REST风格的URL

随着手机等智能移动设备大量地连入互联网，一种被称为**RESTful API**的Web资源访问方式开始流行起来.....



POST /products/5

GET /products



**REST = URL + HTTP方法 (GET/POST/PUT/DELETE)**

这些东西是什么，目前你暂时不用管它，以后学到了就会懂了，它们都是与Web通讯协议——HTTP相关的知识.....

# 互联网公司面试宝典：URL相关的概念辨析

## URI

- Uniform Resource Identifier
- 用于唯一地标识一个Web Resource

## URL

- Uniform Resource Locator
- 是一种最常用的URI，即“网址”

## URN

- Uniform Resource Name
- 是另一种URI。它给一个Web资源设定了一个整个网络唯一的“名字”。

# URL的深入剖析

---

让我仔细瞧瞧你的模样.....

在Web网页中，通常使用特定的标记配合URL引用特定的资源，比如<a><img><link><script>等，一个典型的URL如下所示：

http://www.jinxuliang.com/forum/Home/Read?SubjectId=5027#top



# URL中的查询字符串与编码



在百度中搜索“慕课”，最后会得到下面的这个URL.....

经过编码的URL  
查询字符串  
(加粗显示)

`http://www.baidu.com/s?wd=%E6%85%95%E8%AF%BE&ie=utf-8&f=8&rsv_bp=1&rsv_idx=1&tn=baidu&rsv_pq=fb788b390000eod6&rsv_t=1f21c%2Ba23VfGaf8%2BbXOCIUcQ6sQ4ERo3Qq6gcGxoK7hvSWnpIldg5qBoYQs`

之所以要对URL进行编码，主要是为了处理各种各样的特殊字符，以方便网络传输和计算机进行处理。

一些被视为“危险”的字符（比如汉字，空格等）会被转换，几个例子：

下面这些字符则被认为是“安全的”

危险的ASCII字符	进行URL编码
空格	%20
!	%21
"	%22
#	%23
\$	%24
%	%25
&	%26

英文字母：a~z,A~Z

数字：123...90

常用符号：\$-\_.+#'(),



为什么有些字符被认为是“危险的”？



因为这些字符要不在URL中有特殊的含义，要不然会给服务端程序解析带来麻烦，举两个例子：

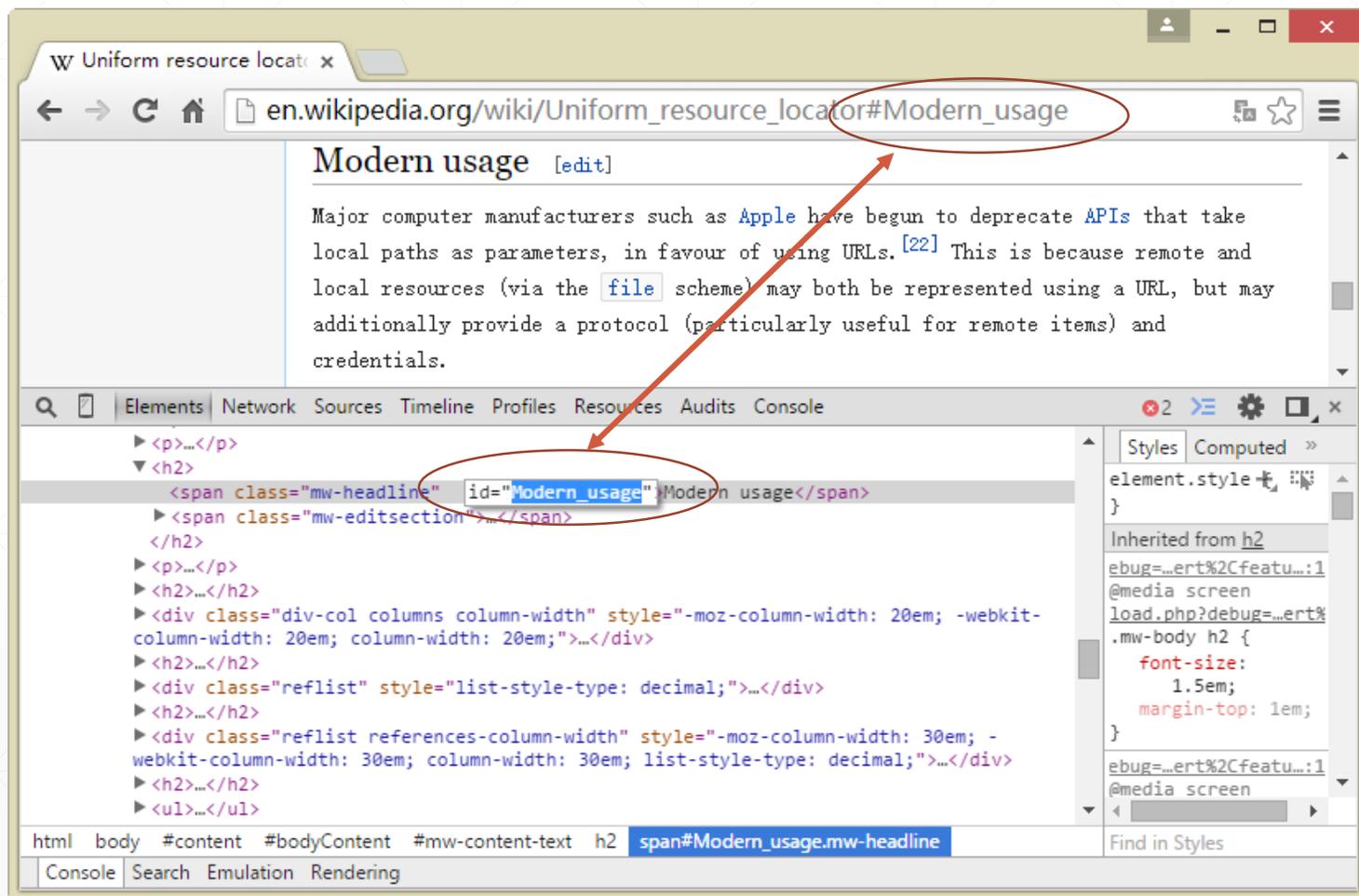
空格：

请问：“abc”这个字符串后头包容几个空格？你能看得出来吗？

&：

在URL中，&被用来分隔多个查询参数，如果要查询的值中有一个“&”，比如“abc&d”不加转换的话，Web Server就会错误地把“&”的前后看成是两个要查询的参数“abc”和“d”，不会把它当成一个整体“abc&d”进行处理。

fragment通常用于代表网页中的某个特定位置。



Wikipedia中使用fragment显示页面中的不同部分，但要注意，所有操作均在本地完成，没有任何网络操作。

在设计网页时，我们可以使用fragment跳转到特定页面（或同一页面）中的特定位置（比如长网页中经常出现的“**回到开头**”功能）

# Web资源的类型

---

你究竟有几个好妹妹？

互联网上的资源有多种类型：文字、图片、视频、Flash动画、JavaScript代码.....

① 我们采用哪种方式来区分资源的类型？

② 浏览器怎么知道它所接收的是哪种类型的数据？

# 一个神秘的术语.....

“**MIME**” 用于标识Web资源类型。

**MIME** : Multipurpose Internet Mail Extensions

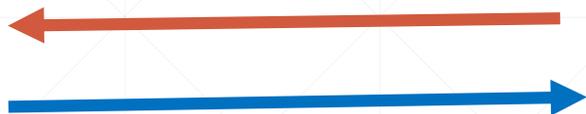
Web上，MIME为每种类型的资源提供了一个统一的标准字符串，比较完美地解决了资源类型的标识问题。

# 使用MIME标识Web资源的类型

MIME中的类型字符串 **==** 类型/子类型

MIME类型字符串	描述
application/json	JSON字符串资源
image/png	PNG格式的图片资源
text/html	HTML文档资源
video/mp4	MP4视频资源
.....	.....

# MIME的用法



```
HTTP/1.1 200 OK  
Content-Length: 95267  
Content-Type: image/png
```

```
Accept: text/html,image/png
```

浏览器：我能处理哪种类型的数据.....

Web Server：“来，给你，这些数据的类型是png格式的图片，共有95267字节”。



“**字节 (byte)**”是信息的度量单位之一，另外一些常见的单位有“**K**”、“**兆 (M)**”、“**G**”、“**T**”等，相信你在购买手机或电脑时，一定见过它们！

# 浏览器与Web Server间的交互（打招呼）实例：

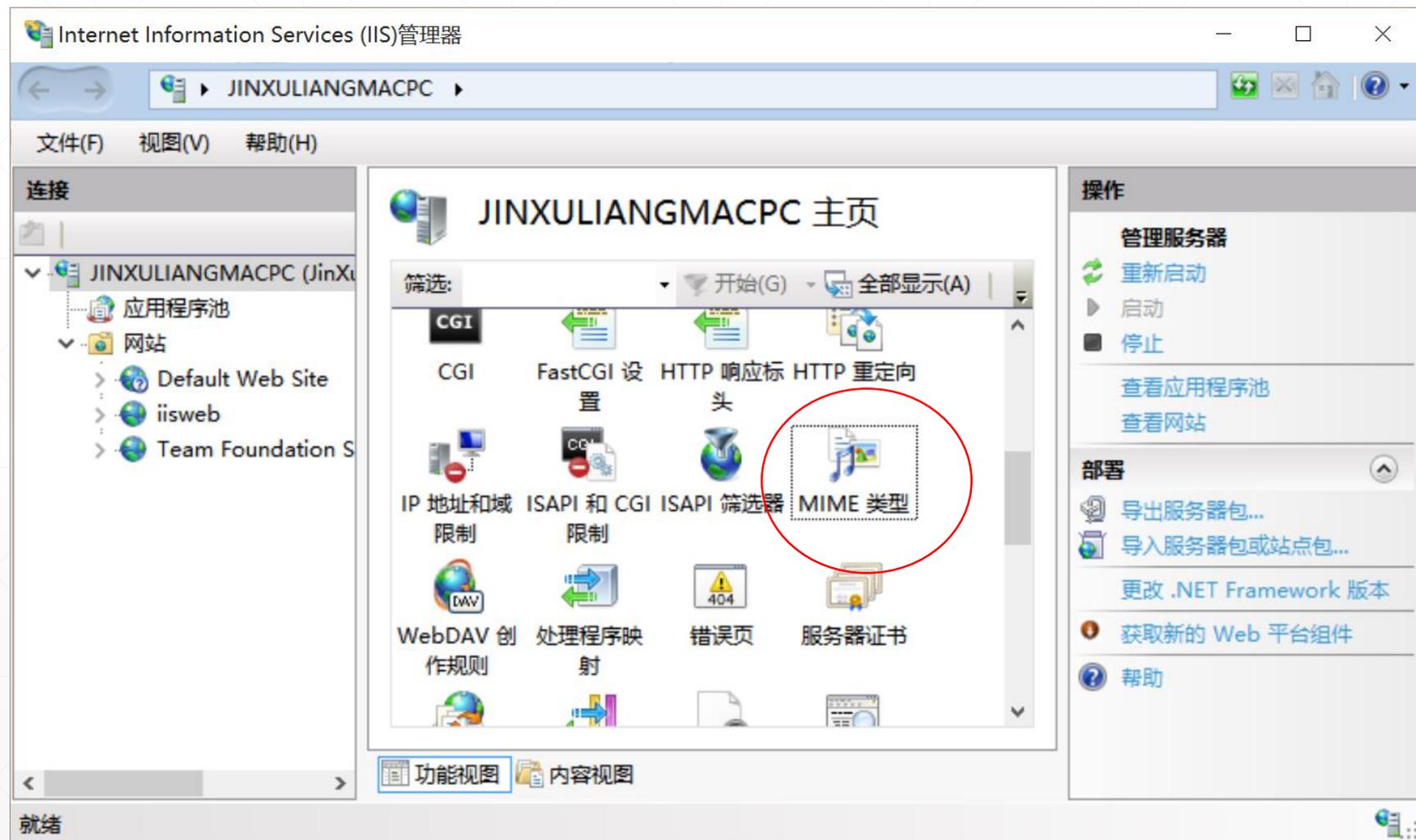
The screenshot shows the Chrome DevTools Network tab. The left sidebar lists network requests, with 'subjectListService.js' selected. The main panel displays the request details for this resource:

- General:**
  - Remote Address: 182.92.1.231:80
  - Request URL: http://jinxuliang.com/forum/MyScripts/subject\_list/subjectListService.js
  - Request Method: GET
  - Status Code: 200 OK (from cache)
- Response Headers:**
  - Accept-Ranges: bytes
  - Content-Length: 740
  - Content-Type: application/x-javascript
  - Date: Thu, 20 Aug 2015 10:31:19 GMT
  - ETag: "fbe43121462ed01:0"
  - Last-Modified: Mon, 12 Jan 2015 08:59:53 GMT
  - Server: Microsoft-IIS/7.5
  - X-Powered-By: ASP.NET
- Request Headers (3):** (Expanded)

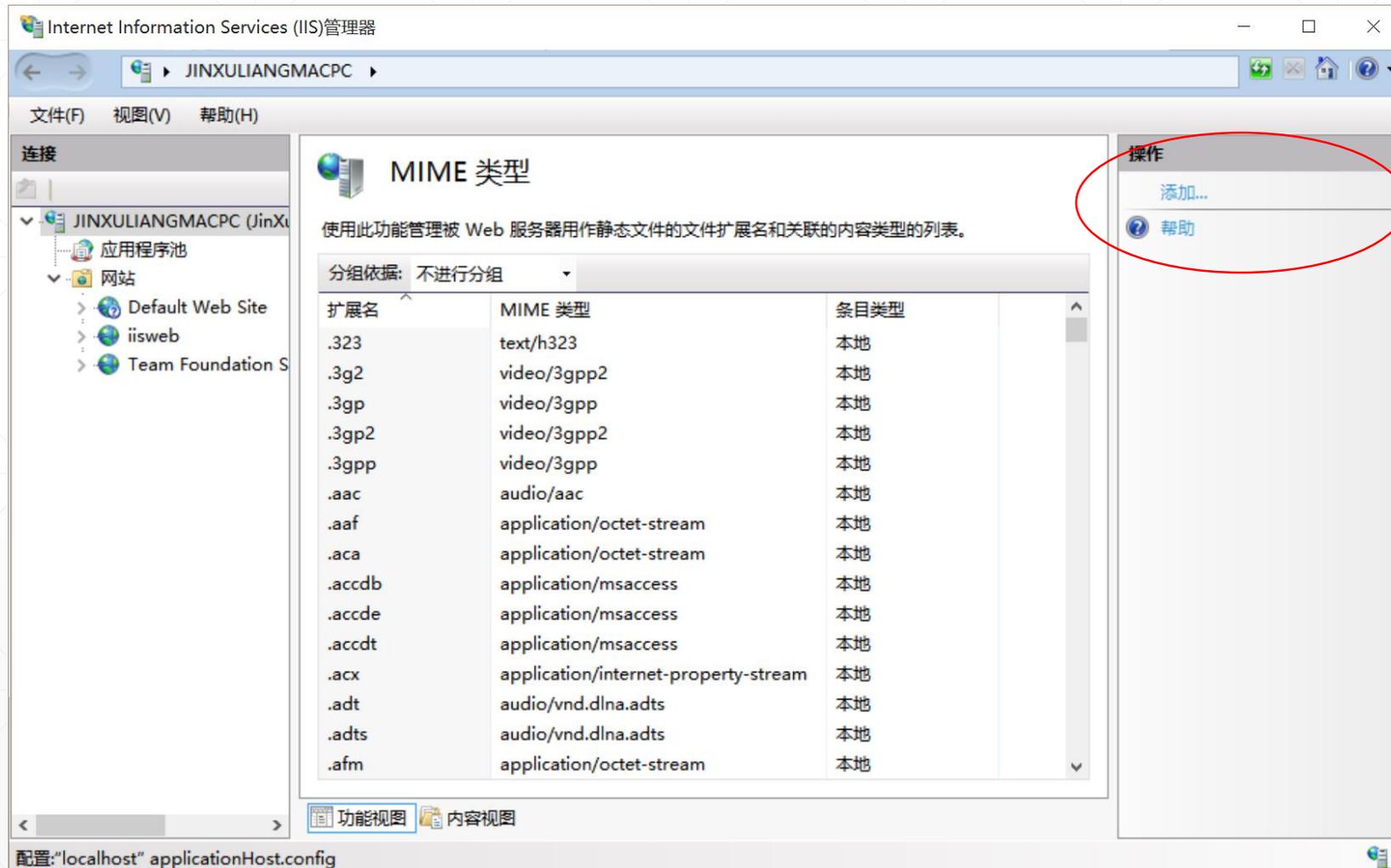
浏览器向jinxuliang.com请求  
subjectListService.js资源

jinxuliang.com通知浏览器，subjectListService.js是  
一个包容JavaScript代码的资源

# 在IIS中指定MIME类型



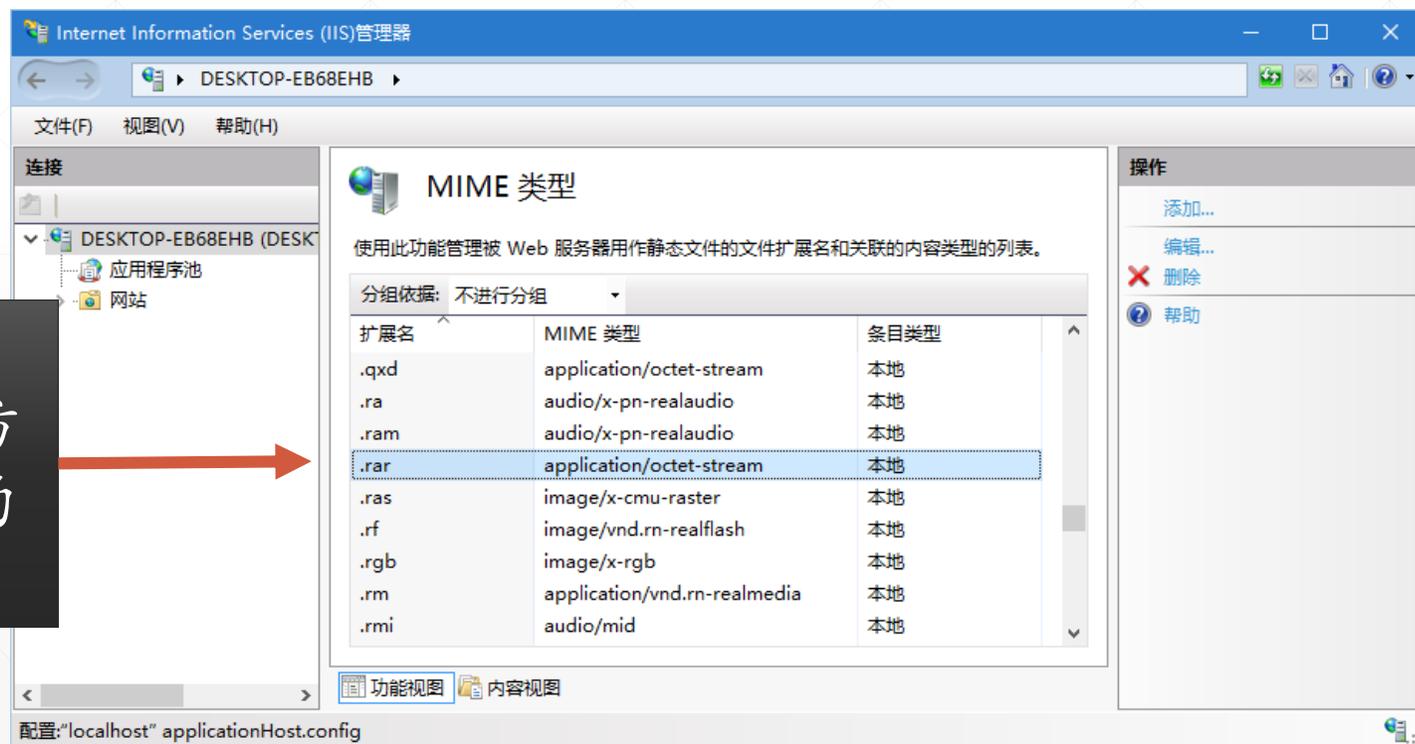
当给IIS添加针对特定的MIME时，IIS就具备了向客户端提供这种类型资源的能力：



# 让网站支持下载特定类型的文件。

默认情况下，Web Server会阻止外界对特定类型文件的访问请求，比如，我们不能向IIS提出要求：我要下载“XXX.rar”文件！

如果我们确定Web Server可以允许外界对特定类型文件的访问请求，只需给其添加相应的MIME类型即可。



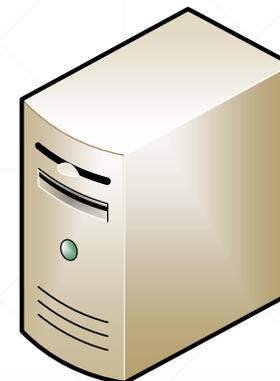
同一资源可能有多种表现形式，比如中文/英文，  
XML/Json等等

# Web上的“内容协商”机制



accept: application/json

{info:"hello",value:100}



客户端可以向服务端表明：  
“我期望获取哪种类型的资源”

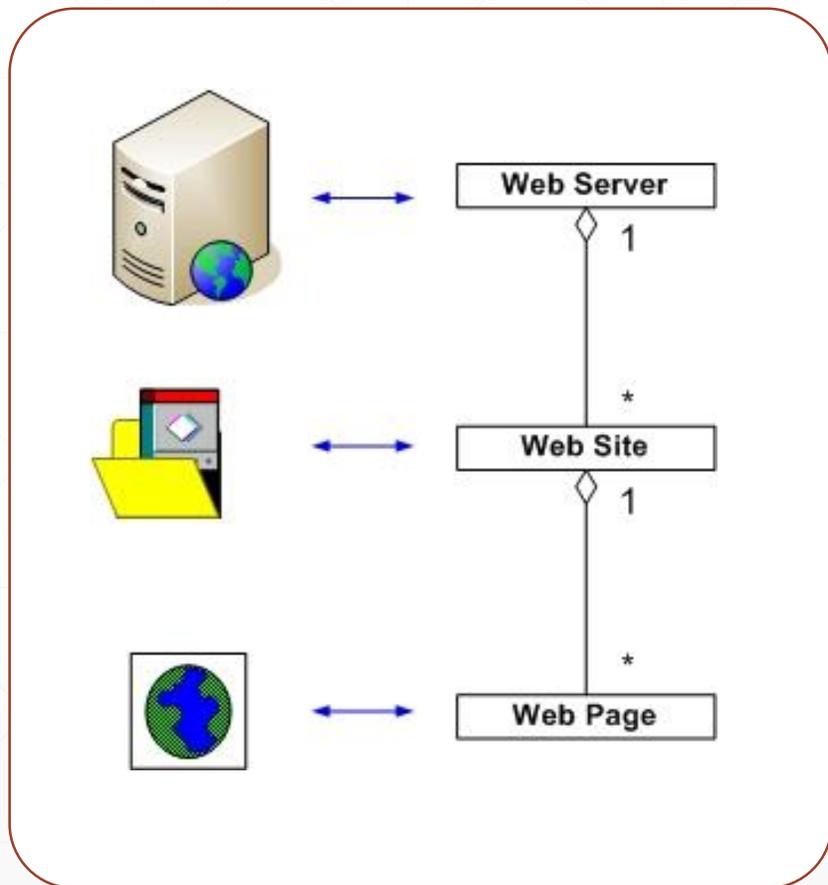
服务器依据客户端请求，  
向其提供所需的资源

# 使用IIS开发Web网站

---

光说不练，到老一场空！

# 网站 ( Web Site ) 的概念



一个**Web Server**上可以承载多个**Web Site** (网站) , 一个Web Site可以包容多个**Web网页** (或**Web应用程序**和**虚拟目录**) 。

网站中的网页放置在文件夹中，IIS将这种文件夹视为一个“**虚拟目录 (Virtual Directory)**”。

http://localhost/MyFirstWeb/MyFirstHTMLPage.html

URL = 主机 + 虚拟目录 + 网页



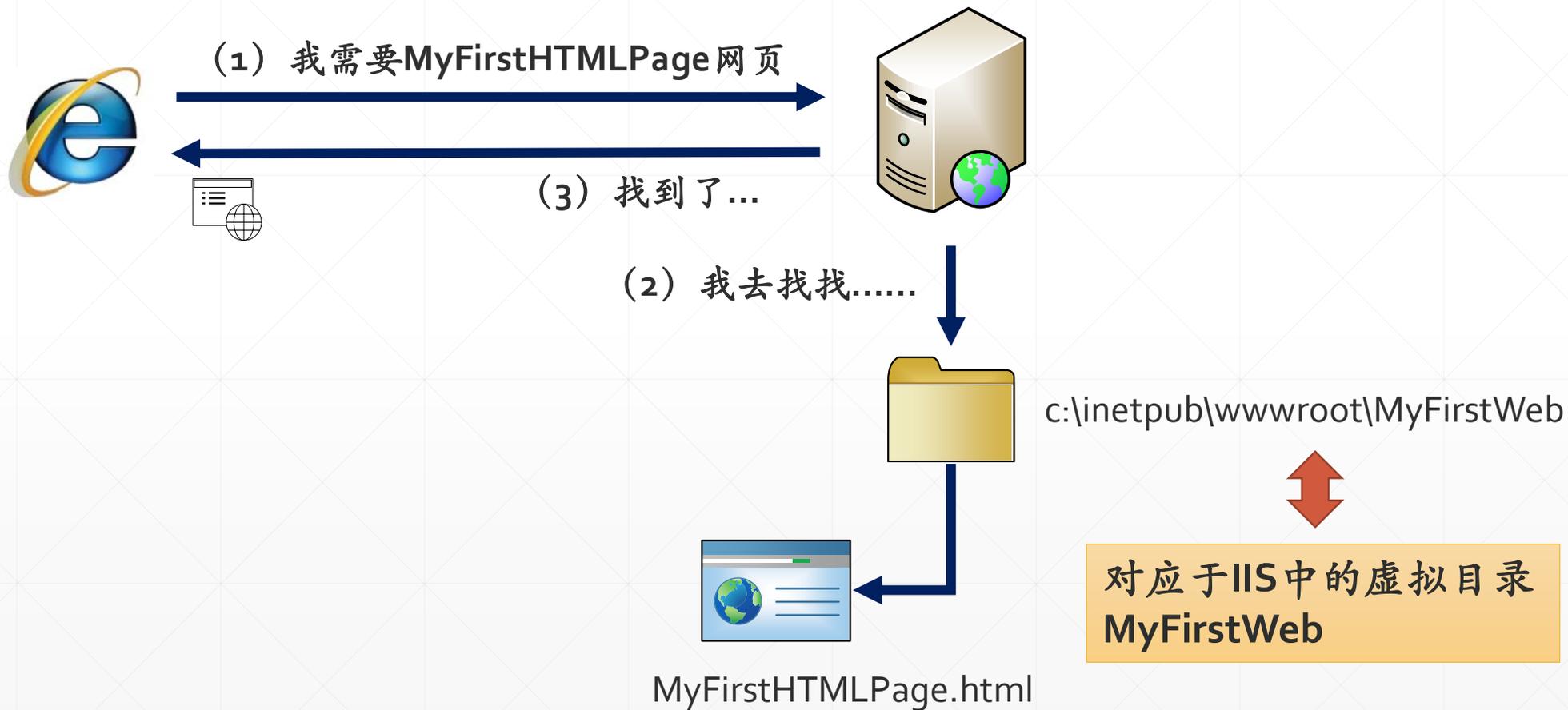
IIS中的虚拟目录，通常对应着一个物理文件夹。虚拟目录的名字与物理文件夹的名字，**可以不一样**。

默认情况下，IIS Web服务器默认网站的“**根虚拟目录 (/)**”对应的物理路径为：c:\inetpub\wwwroot

承载于IIS Web服务器中的一个Web Site，可以包容多个虚拟目录。

# Web网页的定位过程

http://localhost/MyFirstWeb/MyFirstHTMLPage.html

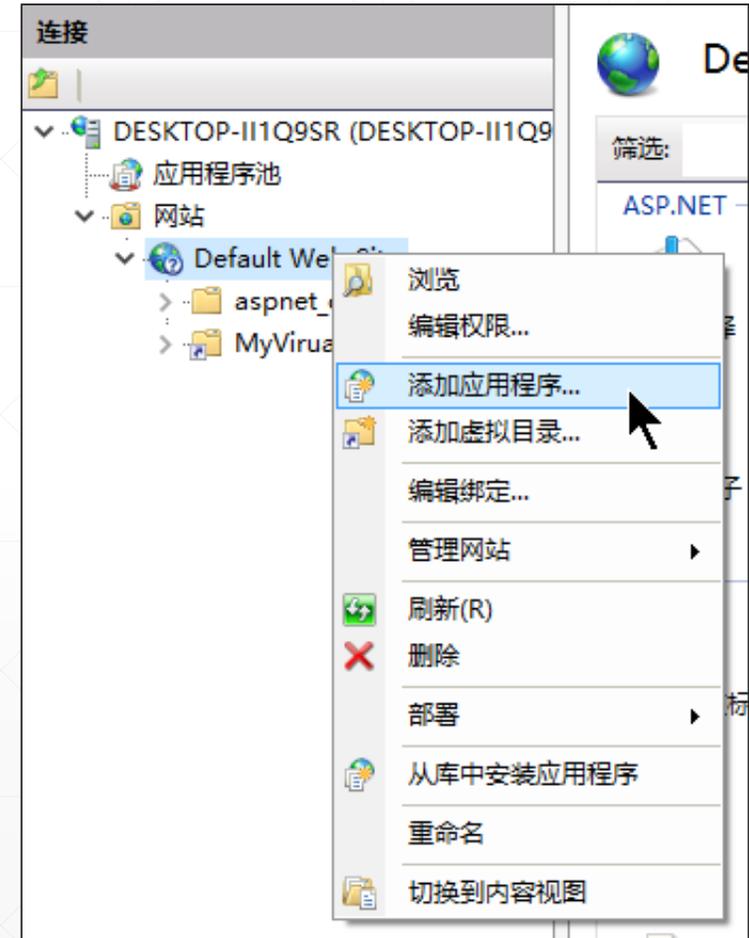


# 子“应用程序”和“虚拟目录”

在IIS中，一个“Web Site/应用程序”下，可以再创建子“**应用程序**”和“**虚拟目录**”，构成一个多级的结构。

每一个“**应用程序**”和“**虚拟目录**”实际上都对应着一个文件夹，可以往其中放置网页文件。

通常情况下，只需创建一个“**应用程序**”，之后依据需要再向其中添加一个或多个“**虚拟目录**”，再向虚拟目录中添加网页文件。



添加应用程序

网站名称: Default Web Site  
路径: /

别名(A): MyWebSite      应用程序池(L): DefaultAppPool      选择(E)...

示例: 销售

物理路径(P): I:\MyWebSite      ...

传递身份验证

连接为(C)...      测试设置(G)...

启用预加载(N)

确定      取消

使用这个名字访问网站：  
<http://localhost/MyWebSite>

网页放到这个文件夹中

如果MyWebSite文件夹中有一个index网页，则以下网址可以访问它：

<http://localhost/MyWebSite/index.html>

localhost表示“本机”

# 跟着老师一起做.....

- ① 在硬盘上创建一个文件夹用于保存网页
- ② 使用喜欢的文本编辑器创建网页，并保存到这个文件夹中
- ③ 在IIS中创建Web应用程序指向这个文件夹
- ④ 通过浏览器访问网页
- ⑤ 创建子文件夹，其中放置更多的网页
- ⑥ 通过在网址中添加虚拟目录访问子文件夹中的网页