

了解互联网（也叫因特网）

● 互联网的组成：

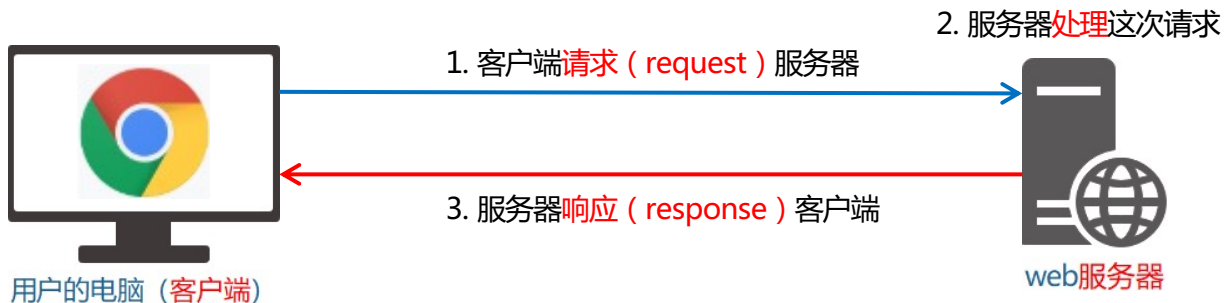
- 服务器：负责**存放和对外提供资源或服务**的电脑（性能比个人电脑高很多）
- 客户端：个人电脑，用来**消费资源或服务**
- 网线：用来**传输**资源和数据

● 上网的本质：通过互联网来**相互传输数据**（举例：看电影、聊天）



如何传送数据

- 浏览器请求一个资源的流程是：**请求 - 处理 - 响应**（演示：百度图片）



和写信进行类比

- 写信通讯的流程是：**写好信并交给邮差** - **收到信并读信** - **回信**

李白



杜甫

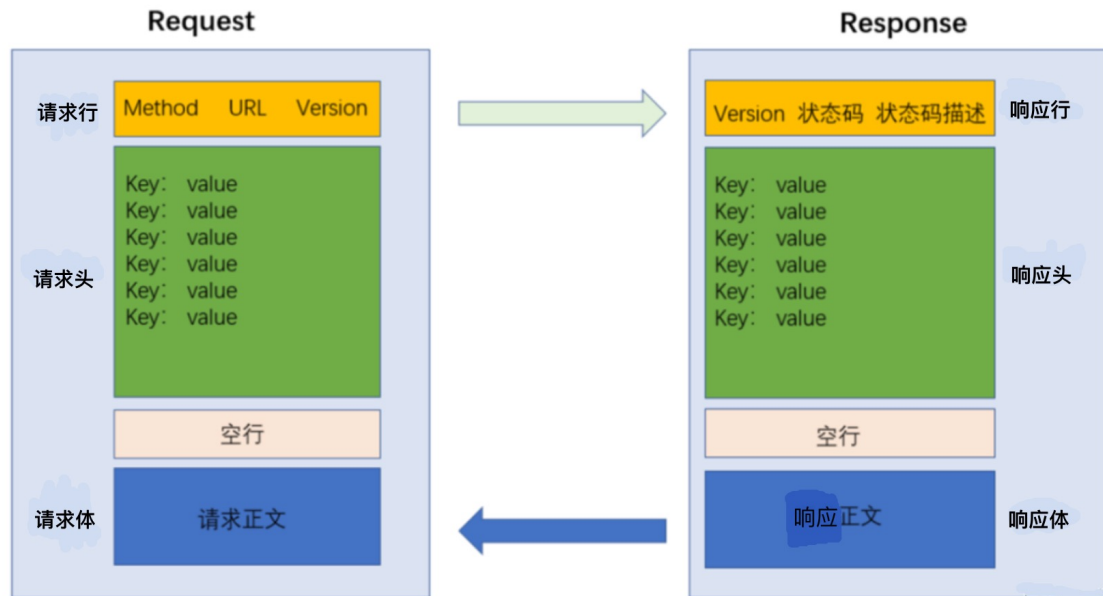


1	5	0	0	5	6	← 这里写收信人邮编	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;"></td> <td style="width: 50%; text-align: center;">贴 邮 票 处</td> </tr> </table>		贴 邮 票 处
	贴 邮 票 处								
<div style="display: flex; justify-content: space-between;"> 黑龙江省哈尔滨市道外区 ← 收信人地址, 地址要详细, 所以可能要占两行或者三行 </div>									
<div style="display: flex; justify-content: space-between;"> 王某 ← 收信人姓名, 字体稍大一些 </div>									
<div style="display: flex; justify-content: space-between;"> 黑龙江省哈尔滨市道外区 ← 寄信人地址、姓名。 </div>									
						邮政编码 100053 ←			

- 在互联网中传输信息时也必须遵守固定的规则（一般称为：协议）
- 作用：保证在因特网中能**正确无误地传输数据**
- 但互联网通讯协议不止一种，常见的协议有：
 1. DNS域名系统；
 2. FTP文件传输协议；
 3. Telnet远程终端协议；
 - 4. HTTP超文本传送协议（浏览器使用的就是http协议）；**
 5. SMTP电子邮件协议；
 6. SSH 加密传输协议；
 7. etc.....

了解HTTP协议

(带学生看真正的格式)



提问

1. 什么是协议？
 - ✓ 规则或约定
2. 什么是http协议？它的作用是什么？
 - ✓ 是互联网的一种通讯协议，作用：保证在因特网中能**正确无误地传输数据**
3. http协议的格式为？
4. http协议的格式能随意更改吗？

■ method (请求方式)

- 常见：

- get
- post
- delete
- etc.....

- 各个method之间的区别（两个层面）：

- 技术：get请求没有请求体、post请求可以有请求体，也可以没有

- 含义：

- get：获取资源（eg：查看购物车里的商品）
- post：提交数据（eg：注册、登录）
- delete：删除服务器上的某个资源（eg：删除购物车里的某个商品）

- 作用：以开门见山地方式告诉服务器这个请求的目的是什么

- 工作中到底用哪一个？通常不由前端决定，一般由后端java决定

序号	方法	描述
1	GET	(查询)发送请求来获得服务器上的资源，请求体中不会包含请求数据，请求数据放在协议头中。
2	POST	(新增)向服务器提交资源（例如提交表单或上传文件）。数据被包含在请求体中提交给服务器。
3	PUT	(修改)向服务器提交资源，并使用提交的新资源，替换掉服务器对应的旧资源。
4	DELETE	(删除)请求服务器删除指定的资源。
5	HEAD	HEAD 方法请求一个与 GET 请求的响应相同的响应，但没有响应体。
6	OPTIONS	获取http服务器支持的http请求方法，允许客户端查看服务器的性能，比如ajax跨域时的预检等。
7	CONNECT	建立一个到由目标资源标识的服务器的隧道。
8	TRACE	沿着到目标资源的路径执行一个消息环回测试，主要用于测试或诊断。
9	PATCH	是对 PUT 方法的补充，用来对已知资源进行局部更新。

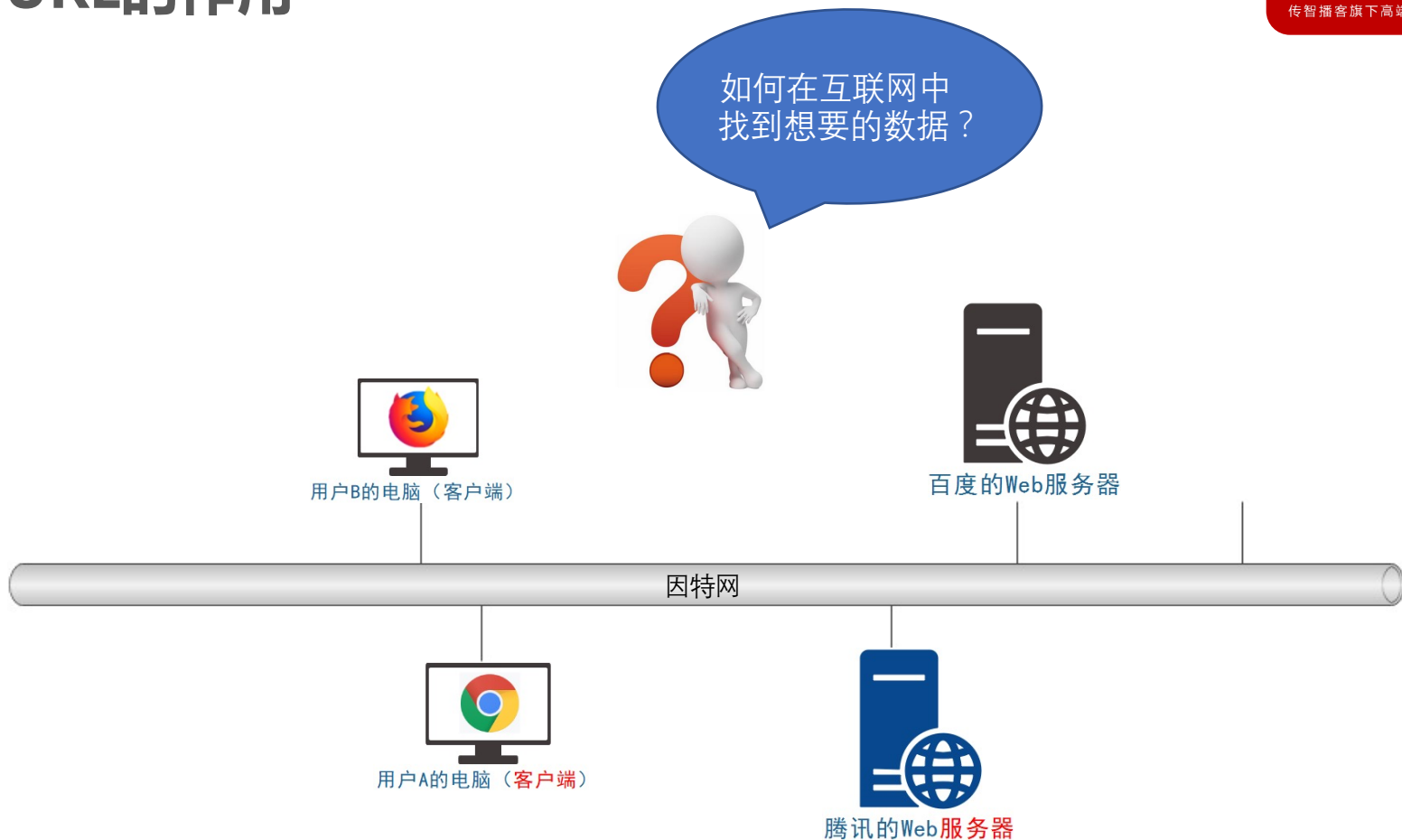
- URL (UniformResourceLocator 译：**统一资源定位符**)

- URL的作用：用于**标识**互联网上**每个**资源的**唯一**存放位置

- 例如：

1. https://gimg2.baidu.com/image_search/src=http%3A%2F%2Fimg.ws.126.net%2F%3Furl%3Dhttp%253A%252F%252Fdingyue.ws.126.net%252F2021%252F1025%252F1aa7ccc2j00r1it0x002fc000hs00lzc.jpg%26thumbnail%3D650x2147483647%26quality%3D80%26type%3Djpg&refer=http%3A%2F%2Fimg.ws.126.net&app=2002&size=f9999,10000&q=a80&n=0&g=0n&fmt=auto?sec=1658368929&t=c590f82f7594fac5b5684cfabc7e1572
2. https://gimg2.baidu.com/image_search/src=http%3A%2F%2Fimg-pre.ivsky.com%2Fimg%2Fbizhi%2Fpre%2F201910%2F29%2Fzhangmingen.jpg&refer=http%3A%2F%2Fimg-pre.ivsky.com&app=2002&size=f9999,10000&q=a80&n=0&g=0n&fmt=auto?sec=1660390479&t=68cc5d34f28b72177082238ace4ee6a5

URL的作用



完整的url样子



完整的网址(url)中文版本: 协议://主机名(域名):端口号/路径/文件名?查询字符串#锚点

英文版本: protocol://hostname:portname/pathname/?search#hash

URL编码 - 导入

- 问题一：在html代码中，为什么不能在页面中直接显示<>等？
- 问题二：在js代码中，为什么字符串里不能有和两边相同的引号？
- 答：浏览器分不清
- 如何解决：对特殊字符进行转义（即用别的字符去表示）

1. 在URL中 /、?、=、&、# 等都是有特殊含义的，如果想把这些字符变成普通字符，就需要进行转义
 2. 除此之外，为了安全起见，URL中不允许使用非ASCII 字符（如：中文、日文、韩文等），还有空格
- ◆ 所以工作中在URL里尽量不要使用特殊字符和非ASCII 字符，但由于特殊情况非要使用，可以通过下面这些js内置函数进行转义（原理：使用一些ASCII 字符去表示特殊字符和非ASCII 字符）

js对URL编码与解码的方法

- JS提供了2套函数对URL进行编码和解码：

➤ encodeURI/decodeURI （ 主要对非ASCII 字符进行编码，不对 /、?、=、&、# 等这些字符进行编码）

```
encodeURI('http://www.liulongbin.top:3006/黑马/getbooks?id=1&bookname=西游记')  
// 输出字符串http://www.liulongbin.top:3006/%E9%BB%91%E9%A9%AC/getbooks?id=1&bookname=%E8%A5%BF%E6%B8%B8%E8%AE%B0  
decodeURI('%E9%BB%91%E9%A9%AC')  
// 输出字符串  黑马
```

➤ encodeURIComponent/decodeURIComponent（ 对非ASCII 字符和 /、?、=、&、# 等都进行编码）

```
encodeURIComponent("6&10");  
// 输出字符串  6%2610  
encodeURIComponent("6%2610 ");  
// 输出字符串  6&10
```

好消息：浏览器会自动对URL进行编码（在内部使用encodeURI）



提问

1. url的作用？
2. url的格式是什么？各自的作用是什么？（背）
3. URL里不允许有哪些字符？
4. 工作中什么情况下我们需要手动对url进行编码？
5. encodeURI和encodeURIComponent的区别？

- HTTP的所有版本：

- HTTP/0.9 1989年
- HTTP/1.0 1996 年
- **HTTP/1.1 1997 年 (主流)**
- HTTP/2 2015 年
- HTTP/3 未正式发布

- 作用：对这次请求的一个说明（类似于：商品说明）



- 作用：对这次请求的一个说明（类似于：商品说明）

```
if-modified-since: Wed, 07 Jan 1970 00:00:00 GMT
if-none-match: 50f90032c5edb07eb0d31dde98daef65
referer: https://image.baidu.com/
sec-ch-ua: ".Not/A)Brand";v="99", "Google Chrome";v="103", "Chromium";v="103"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "macOS"
sec-fetch-dest: document
sec-fetch-mode: navigate
sec-fetch-site: same-site
sec-fetch-user: ?1
upgrade-insecure-requests: 1
user-agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/103.0.0.0 Safari/537.36
```

请求体

- 作用：真正的数据

■ Status Code (状态码)

- 全部：

分类	分类描述
1**	信息，服务器收到请求，需要请求者继续执行操作（实际开发中很少遇到 1** 类型的状态码）
2**	成功，操作被成功接收并处理
3**	重定向，需要进一步的操作以完成请求
4**	客户端错误，请求包含语法错误或无法完成请求
5**	服务器错误，服务器在处理请求的过程中发生了错误

- 常见的状态码有：

- 200 请求成功
- 404 服务器没找到该资源

■ Status Message (状态信息)

- 作用：对状态码进行说明
- 常见的状态信息有：
 - ok



提问

1. content-type的作用？没有content-type会发生什么事情？
2. get请求中一般有content-type吗？为什么？

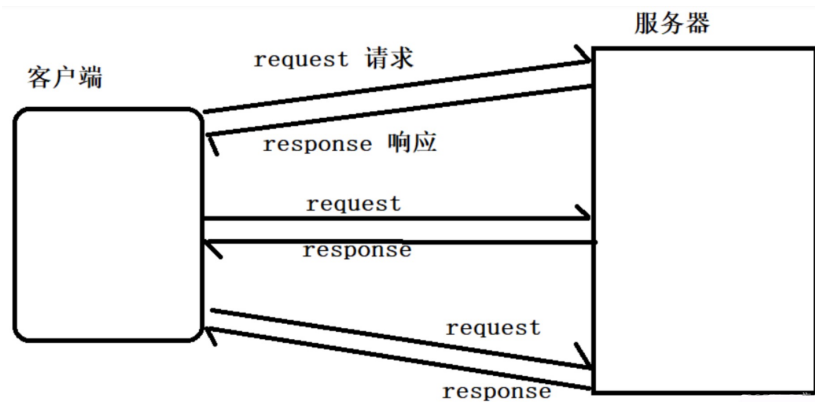
■ 浏览器发送请求的方式

- 浏览器向服务器发送请求的方式有很多：

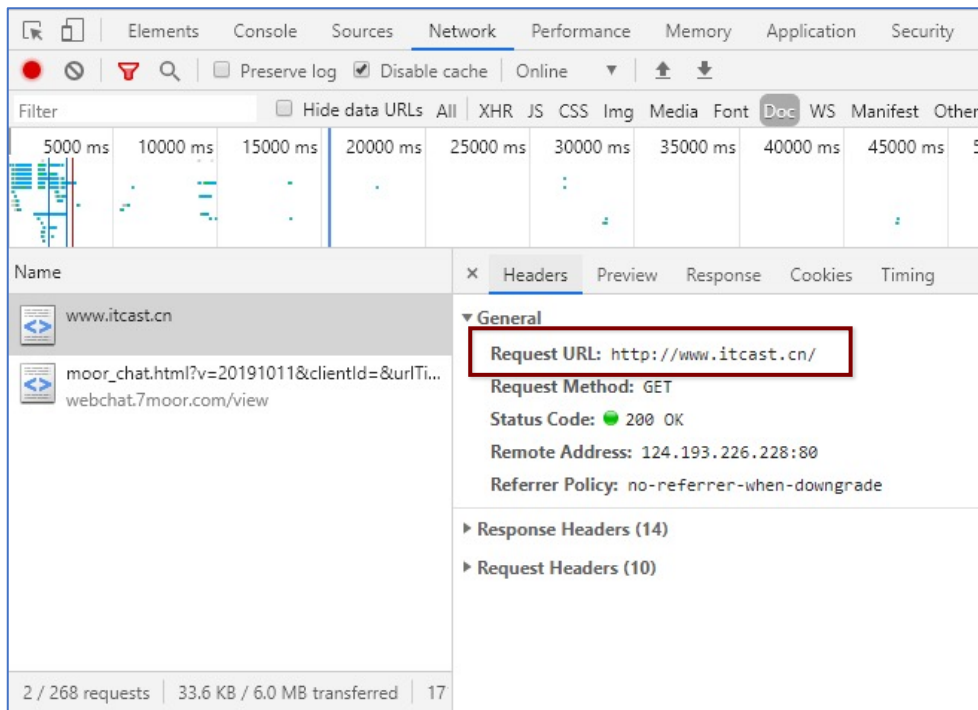
- 在浏览器地址栏输入url按回车
- 点击 链接
- js中的location.href="新的url地址"
- <link>、<script>、
- css里的background-image : url()
- form表单
- js中的XMLHttpRequest （即：ajax）
- etc.....

■ 客户端与服务器的通信过程

- 一个网页中有许许多多的资源，**每一个资源**都是通过 **请求 - 处理 - 响应** 的方式从服务器获取回来的。

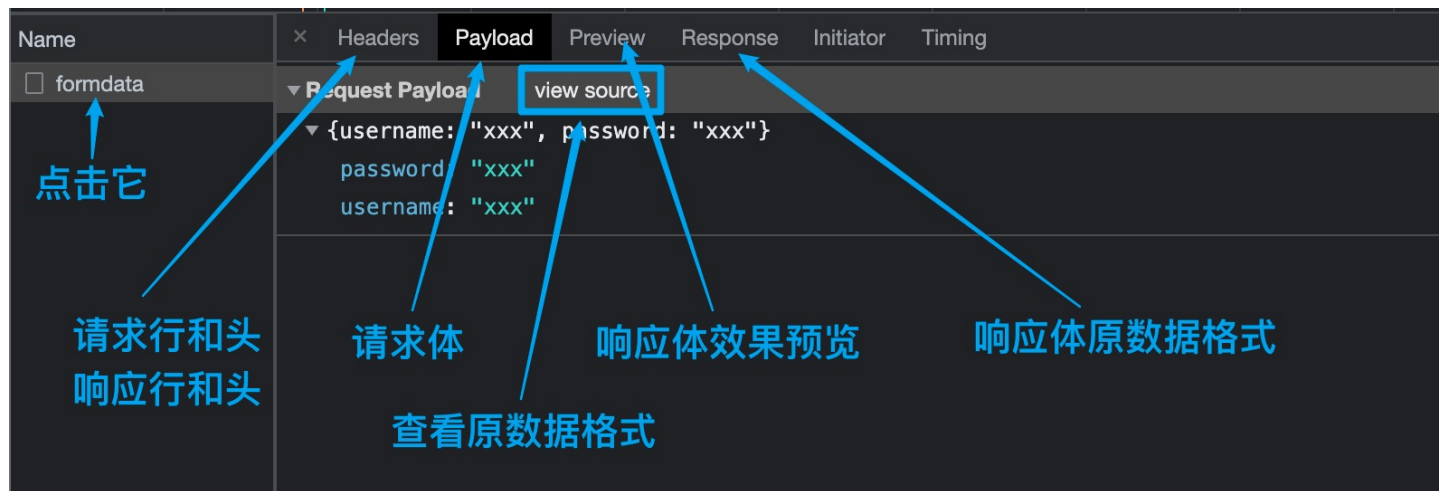


■ 使用调试工具抓包



1. Chrome 的开发者工具
2. 切换到 Network 面板
3. 选中某个标签，然后刷新页面
4. 点击某个http请求查看它的内容

■ 使用调试工具抓包



- 当用户在浏览器的地址栏里输入了www.jd.com，发生了什么？（面试题）
 1. 先向服务器发出一个http请求，拿到html代码后开始解析html代码
 2. 当碰到<link>、<script>、标签时，再发出http请求（有多少个标签就发出多少个http请求）
 3. 等css、js代码请求回来后就开始解析css和js代码，如果在css、js代码里还有background-image：url()等需要资源的代码，就再发出http请求（有多少个就发出多少个http请求）
 4. 哪个请求回来了就解析并渲染请求回来的资源
- 注意：第一个请求一定是去请求html代码，因为一个url对应一个资源，www.jd.com这个url只能请求到一个资源。把html代码请求回来后，通过<link>、<script>、这些标签再去请求别的资源。

Content-Type (数据类型)

- 请求头里有一个非常重要的字段：Content-Type
- 作用：告诉对方body里的数据是什么格式（原因：发送过去的都是0和1）（带学生去network里看）
- 常见的Content-Type值有：
 - html格式字符串 （Content-Type：text/html）
 - css格式字符串 （Content-Type：text/css）
 - js格式字符串 （Content-Type：application/javascript）
 - png图片 （Content-Type：image/png）
 - jpg图片 （Content-Type：image/jpeg）
 - pdf文件 （Content-Type：application/pdf）
 - word文件 （Content-Type：application/msword）
 - 普通的字符串 （Content-Type：text/plain）
 - form表单独有的格式 （Content-Type：application/x-www-form-urlencoded）
 - json格式的字符串 （Content-Type：application/json）
 - form表单上传文件 （Content-Type：multipart/form-data）
 - etc.....



提问

1. 当用户在浏览器的地址栏里输入了www.jd.com，发生了什么？