

# 分布计算环境

北京邮电大学计算机学院

# Chapter 5

## Chapter 5 面向Web的 分布计算环境

- ◆ **Web技术的发展历程**
- ◆ **Web基本支撑技术**
- ◆ **XML技术**
- ◆ **语义WEB**
- ◆ **Web 2.0**
- ◆ **Web Service**
- ◆ **小结**

◆ “Web是一个抽象的（假想的）信息空间” by Tim Berners-Lee

- 作为Internet上的一种应用架构，Web的首要任务就是向人们提供信息和信息服务

◆ 典型的分布式应用架构

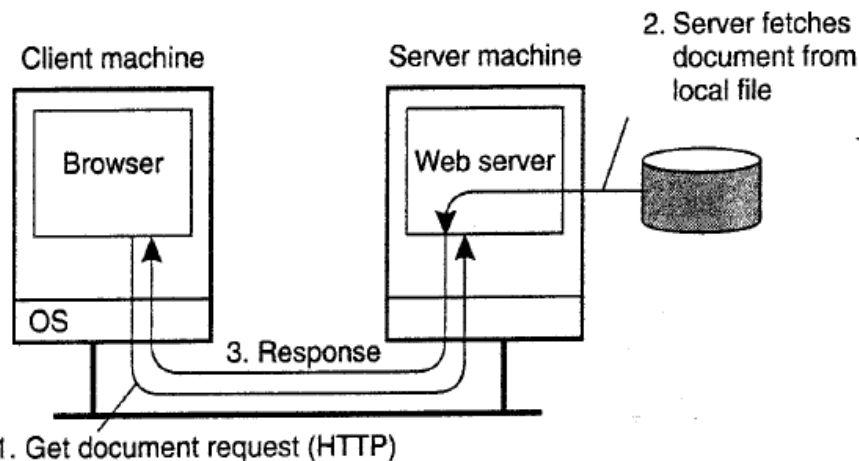
- 在C / S体系基础上扩充而成，Web信息服务应用的最基本模式：B/S结构

◆ 技术特点

- 用超文本技术（HTML）实现信息与信息的连接
- 用统一资源定位技术（URI）实现全球信息的精确定位
- 用新的应用层协议（HTTP）实现分布式的信息共享

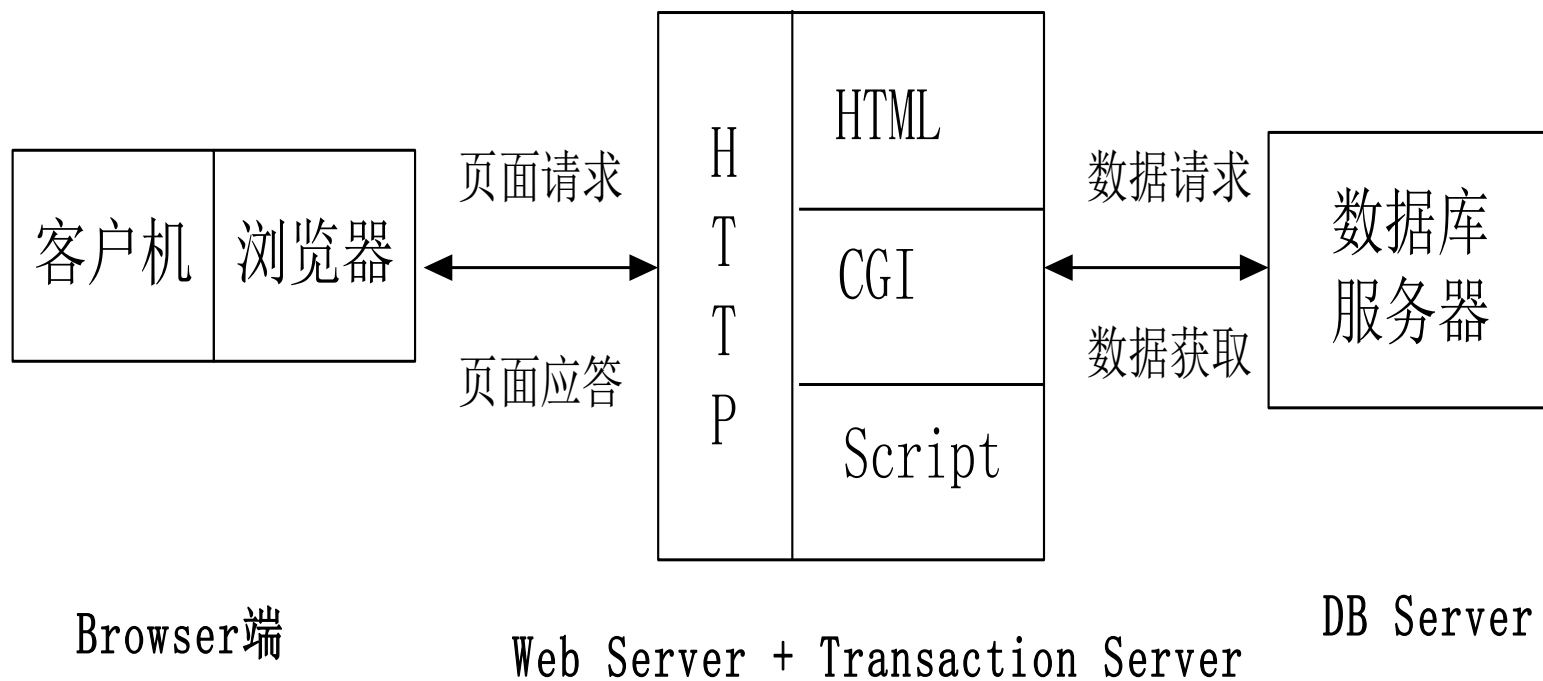
# 第一阶段的Web：静态页面

- ◆ 客户端：通过浏览器访问Internet上各个Web站点，在每一个站点上都有一个主页作为进入一个Web站点的入口
- ◆ 服务器端：每一个Web站点由Web服务器及许多Web页所组成
- ◆ 典型技术：HTML（超文本标记语言），HTTP（超文本传输协议）
- ◆ Web 1.0



## 第二阶段的Web：动态页面

- ◆ 将数据库与Web结合在一起，在Web上发布来自数据库的信息
- ◆ 典型技术：CGI、服务器专用API、ODBC、JDBC等技术
- ◆ Web 1.0 or Web 1.5?



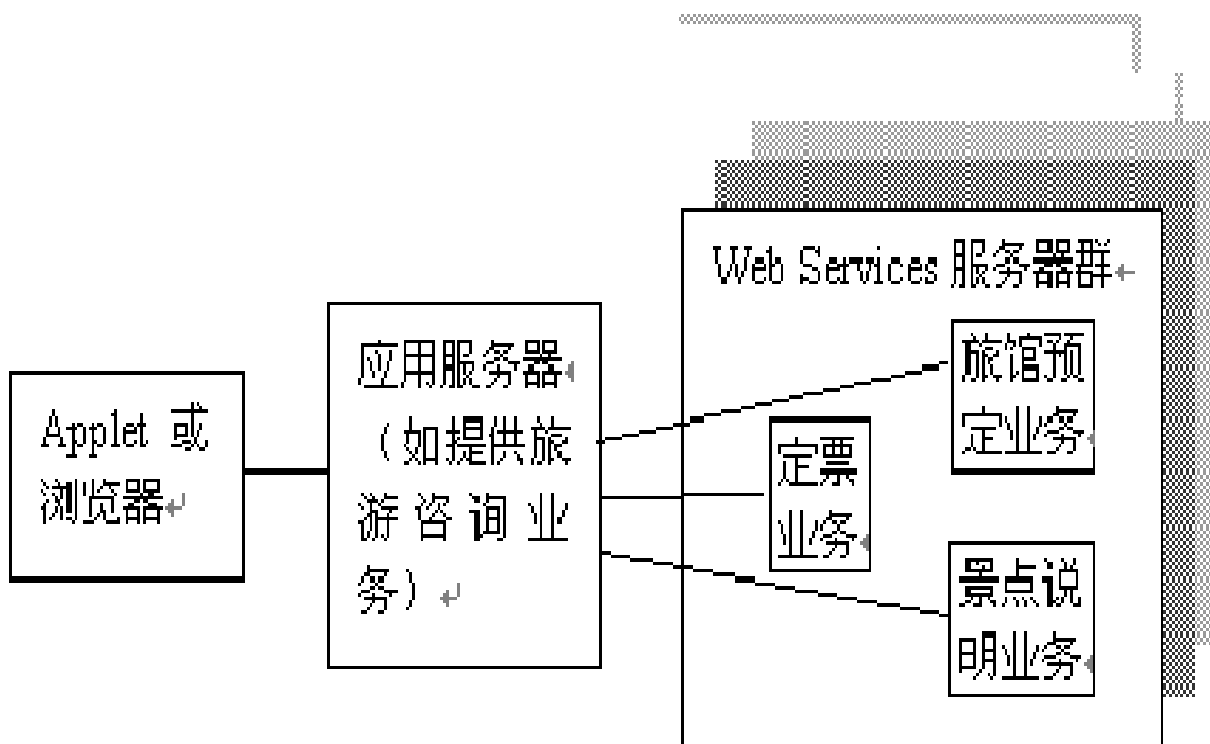
- ◆ 以人为出发点，让所有的人都忙起来，全民织网，然后用软件、机器的力量使这些信息更容易被需要的人找到和浏览
  - 用户：贡献内容，传播内容，提供内容之间的链接关系和浏览路径
- ◆ 主要支撑技术
  - BLOG、RSS、WIKI、SNS等

- ◆ 让计算机能够“理解”分布在网上的信息和知识，并在“理解”的前提下更好地处理、利用这些信息和知识，使Web成为支持全球化知识共享的智能信息服务平台，为人们提供优质的服务
  
- ◆ 属于哪个阶段？
  - 目前还没有成功到为其定位一个阶段



## ◆ 广域异构网络环境下，应用的互操作

### ■ Web Service



## ◆ Web的基本支撑技术

- ◆ HTML、HTTP
- ◆ CSS、JavaScript
- ◆ ASP、JSP/Servlet、PHP

## ◆ 数据的表示、组织和共享

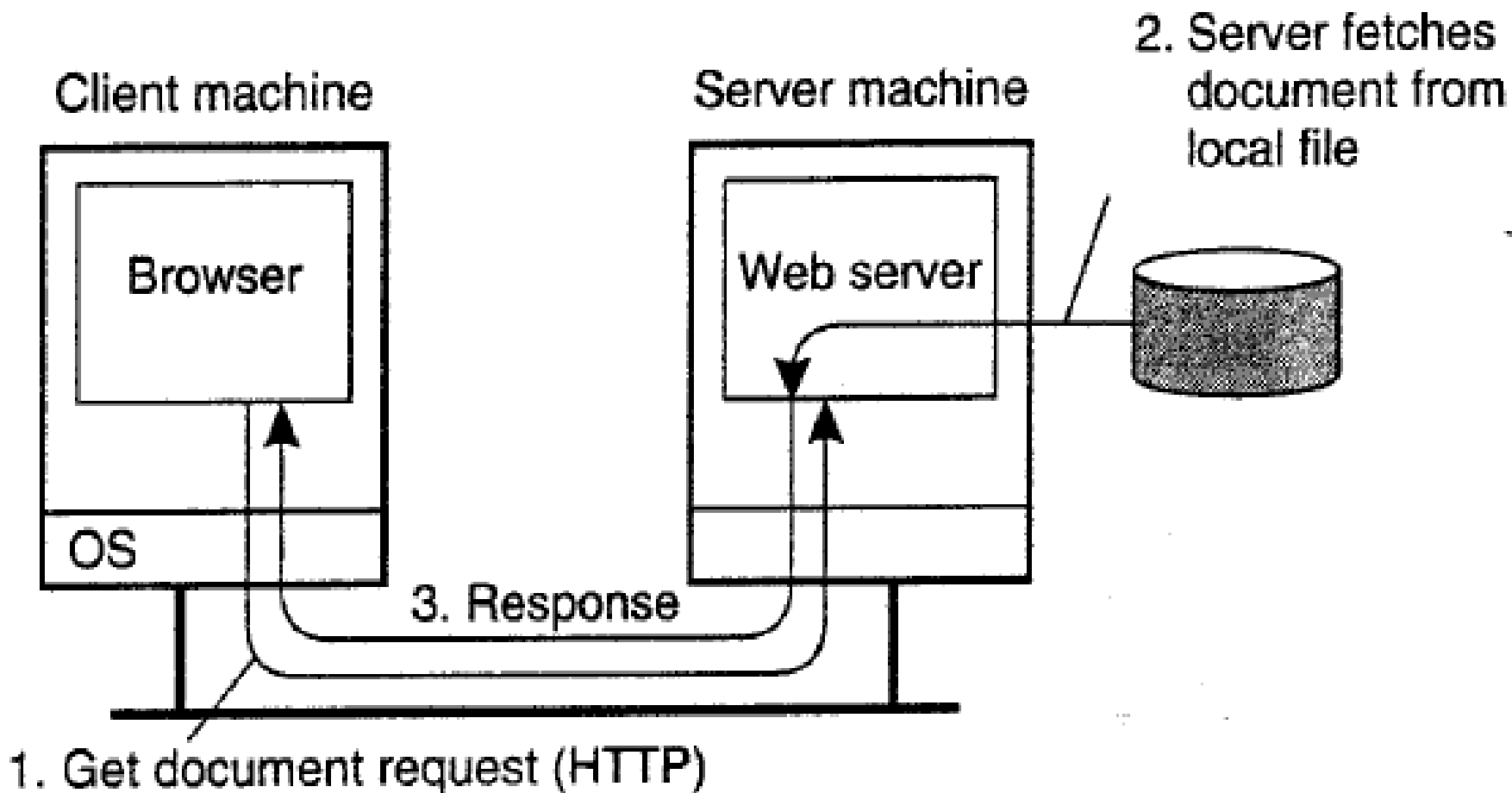
- ◆ XML
- ◆ 语义Web
- ◆ Web 2.0

## ◆ 异构应用程序的互操作

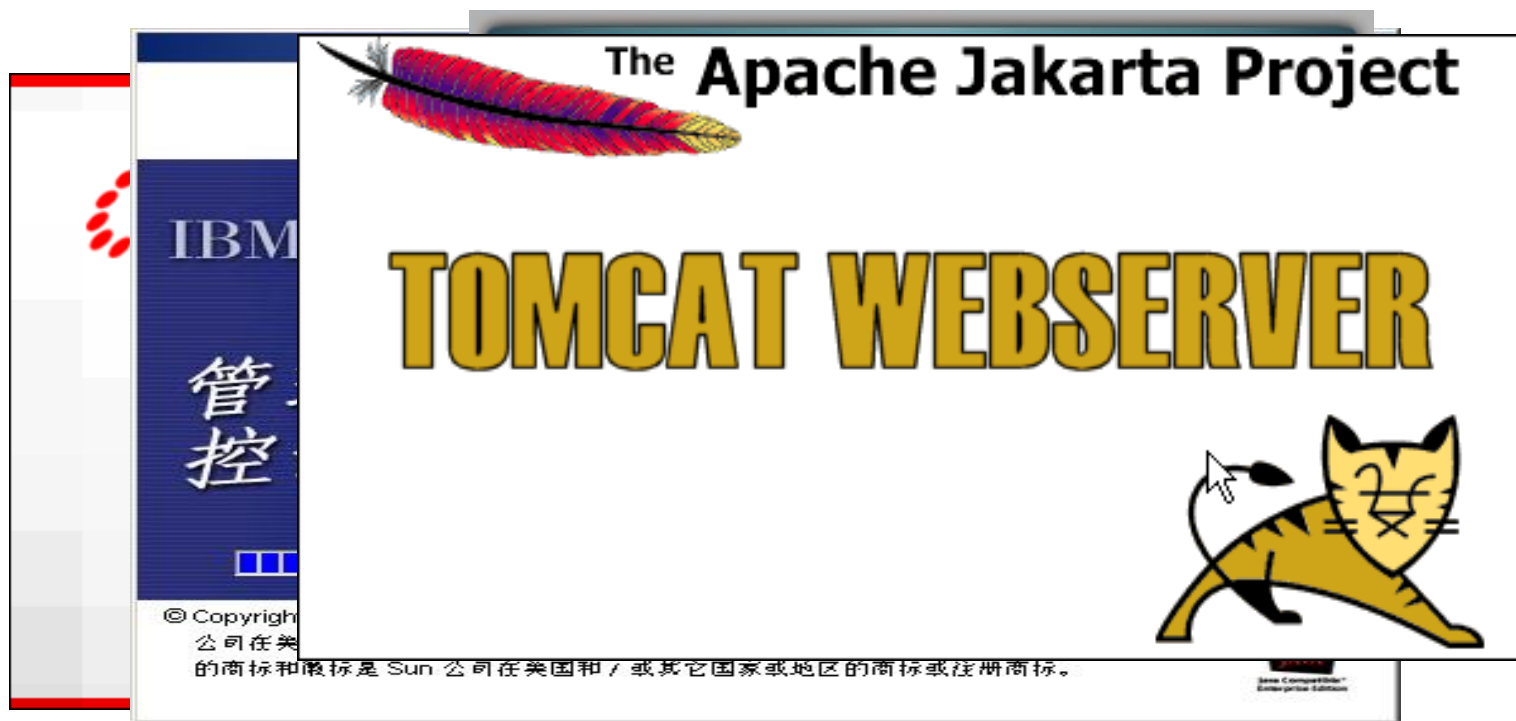
- ◆ Web Service

- ◆ Web技术的发展历程
- ◆ Web基本支撑技术
- ◆ XML技术
- ◆ 语义WEB
- ◆ Web 2.0
- ◆ Web Service
- ◆ 小结

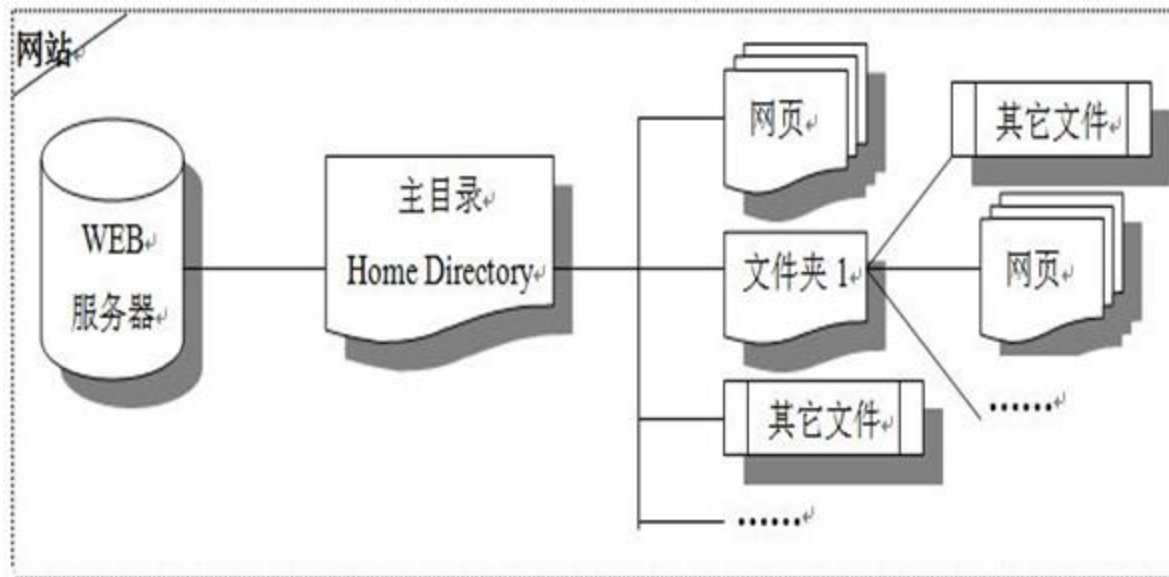
# 传统Web网站的常见组织 (Web1.0)



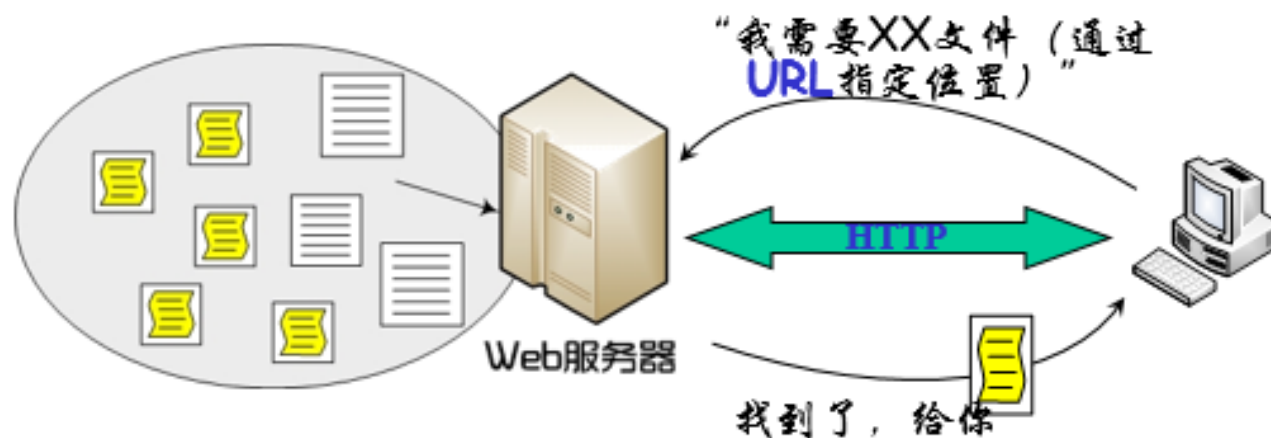
- ◆ Web服务器又称**WWW服务器**、网站服务器、站点服务器，将本地信息用**超文本**（Hypertext）组织，为用户在Internet上**搜索和浏览信息**提供服务



- ◆ Web在提供信息服务之前，所有信息都必须以文件方式事先存放在Web服务器所管辖磁盘中某个文件夹下，其中包含了由超文本标记语言HTML (HyperText Markup Language) 组成的文本文件，我们称这些文本文件为超链接文件，又称网页文件，或称Web页面文件（Web Page）



- ◆ 当用户通过浏览器在地址栏输入访问网站的网址时，实际上就是向某个Web服务器发出调用某个页面的请求。Web服务器收到页面调用请求后，从磁盘中调出该网页进行相关处理后，传回给浏览器显示
- ◆ 网站的网址由**统一资源定位符URL**指定
  - URL字串分成三个部分：协议名称、主机名和文件名（包含路径）：  
`http://www.yahoo.com.cn/index.htm`
- ◆ 请求和响应通过HTTP协议传送

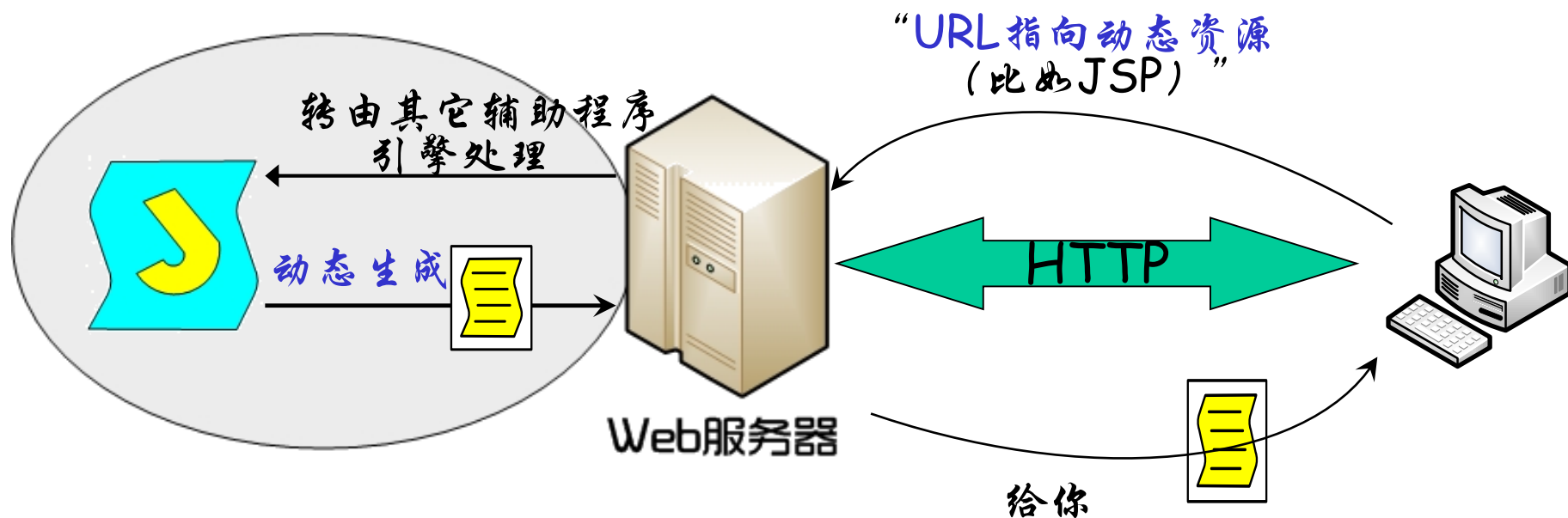


Web服务器存放着各类  
**Web资源** (文件、图片等)

- ◆ 静态Web资源(比如HTML页面): 指Web页面中供人们浏览的信息数据保持不变
  - Web浏览器请求静态资源, Web服务器通过HTTP协议把该HTML文档内容 (原封不动地) 直接发送给浏览器
- ◆ 动态Web资源: 指Web页面中供人们浏览的信息数据是由程序动态产生的, 不同时间访问到的内容可能各不相同
  - Web浏览器请求动态资源 (如JSP&Servlet), Web服务器转交给相应程序引擎处理, 由程序动态生成结果返回给Web服务器, 再经由Web服务器返回给浏览器, 返回的 (通常) 仍然是HTML格式的内容
    - ➔ 有人称为Web 1.5



# 动态页面访问的基本方式



Web服务器存放着各类  
Web资源 (文件、图片等)

- ◆ Web技术几乎汇集了当前信息处理的所有技术手段。由于Web正处在日新月异的高速发展之中，它所覆盖的技术领域和层次深度也在不断改变
- ◆ Web1.x阶段的基本技术是Web领域的基本支撑技术，主要包括：
  - HTML、CSS
  - HTTP
  - JavaScript、VBScript、Java Applet
  - CGI、PHP、JSP / Servlet、ASP/ASP.NET
  - .....

- ◆ 超文本**标记语言**Hyperlink Text Markup Language, Web世界的第一语言
  - “超文本”就是指页面内可以包含图片、链接，甚至音乐、程序等非文字元素
  - 标记语言：写给浏览器的语言，由浏览器负责解释标记进行内容的呈现
  - 负责在浏览器和Web服务器间传递信息
- ◆ 用HTML语言编写的网页文件，也称**HTML页面文件**，或称**HTML文档**，是由 HTML标记组成的描述性文本。HTML标记可以说明文字、图形、动画、声音、表格、链接等
- ◆ HTML文档是普通的文本文档（ASCII），不含任何与平台和程序相关的信息，它们可以被任何文本编辑器读取

- ◆ `<html>...</html>` 创建一个超文本标记语言文档
- ◆ `<body>...</body>` 文档体，文档的可见部分。
- ◆ `<u>...</u>` 下划线（英文全称：Underline）
- ◆ `<font color=00ff00>...</font>`；字体 颜色
- ◆ `<font size=1>...</font>`；字体 大小等于1（最小）。
- ◆ `<p>...</p>` 创建一个段落（英文全称：**Paragraphs**）
- ◆ `<p align="">` 将段落按左、中、右对齐
- ◆ `<Table>...</Table>` 定义整个表格
- ◆ `<tr>...</tr>` 标记表示一行
- ◆ `<td>...</td>` 标记表示一列



# HTML文档例

[illegible]

- ◆ HTML文档通过在文本文件中添加**标记**符，告诉浏览器如何显示其中的内容（如：文字如何处理，画面如何安排，图片如何显示等）
- ◆ 浏览器按顺序阅读网页文件，然后根据标记符解释和显示其标记的内容
- ◆ 对书写出错的标记将不指出其错误，且不停止其解释执行过程，编制者只能通过显示效果来分析出错原因和出错部位
- ◆ 需要注意的是，对于不同的浏览器，对同一标记符可能会有不完全相同的解释，因而可能会有不同的显示效果

- ◆ HTML5的目标是取代已有的HTML 4.01和XHTML 1.0 标准。它希望能够减少对Flash、Silverlight、JavaFX等外部插件的依赖，并且提供更多能有效增强网络应用的API，并能够在移动设备上支持多媒体
  - 2004年提出，2014年正式公开发布

- ◆ 八大新特性



## 新的表单功能

**<Input>**标签，包含多种类型，用于收集特定数据。  
可用于所有浏览器。

```
<!DOCTYPE html>
<meta charset="UTF-8">
<html>
<head>html5中新的表单</head>
<body>
  <input type=color> color<br/>
  <input type=datetime-local>
    datetime-local<br/>
  <input type=email>email<br/>
  <input type=number>numbrt<br/>
  <input type=range>range<br/>
  <input type=time>time<br/>
  <input type=tel>tel<br/>
  <input type=url>url<br/>
  <input type=week>week</br>

</body>
</html>
```



新表单.html

file:///C:/Users/samsung/I

html5中新的表单

color

2015/03/24 --:-- datetime-local

561282230@qq.com email

8 numbrt

range

10:00 time

15009290000 tel

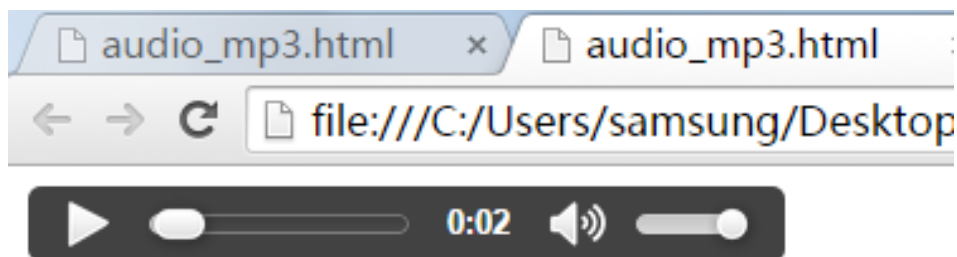
www.baidu.com url

2015 年第 13 周 week



## audio音频标签

```
<!DOCTYPE html>
<html>
<body>
  <audio controls="controls">
    <source src="example.mp3" type="audio/mp3"/>
    <embed height="500" width="500" src="example.mp3" />
  </audio>
</body>
</html>
```



## Video视频标签

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
    <video width="640" height="480"  
    controls="controls" autoplay="autoplay">
```

```
        <source src="example.mp4"  
        type="video/mp4" />
```

```
    </video>
```

```
</body>
```

```
</html>
```



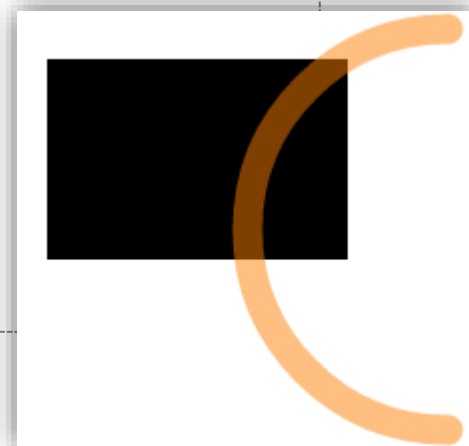
## Canvas 画布元素

传统的网页，总是使用GIF或者JPEG来显示图像，这种图形是需要事先画好的“静态”的图像。而Canvas，则是用Javascript的一种绘图手段  
可以用它来画图、合成图象、或做简单的(和不那么简单的)动画

```
<canvas id="canvas" width="300" height="300"></canvas>

<script>
  var ctx = document.getElementById("canvas").getContext("2d");
  ctx.fillRect(20, 25, 150, 100);

  ctx.beginPath();
  ctx.arc(220, 110, 100, Math.PI * 1/2, Math.PI * 3/2);
  ctx.lineWidth = 15;
  ctx.lineCap = 'round';
  ctx.strokeStyle = 'rgba(255, 127, 0, 0.5)';
  ctx.stroke();
</script>
```



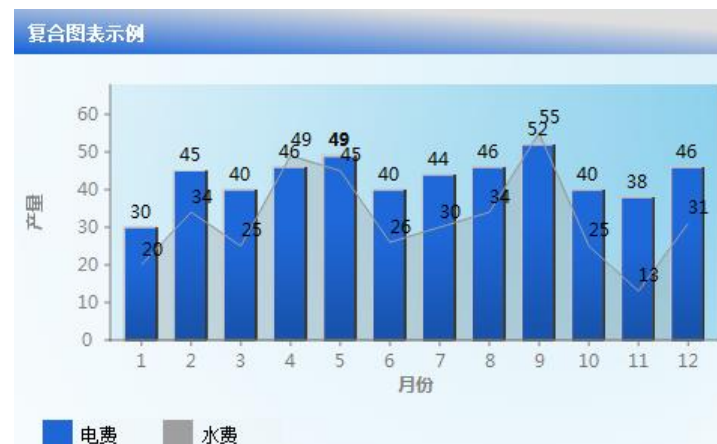
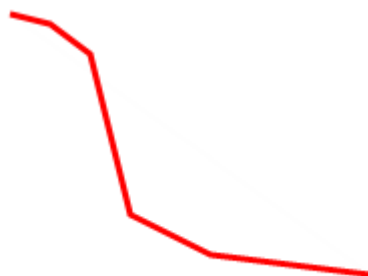
## SVG 矢量图形

**SVG (Scalable Vector Graphics)** : 可缩放矢量图形, 使用 XML 来描述二维图形和绘图程序的语言。

可以在浏览器中构造 矩形、圆形、椭圆、线条、多边形、折线、路径、滤镜效果、渐变效果, 和动画等。

## 复合图表

### 简单的图形元素



- ◆ CSS（Cascading Style Sheets，层叠样式单）是W3C协会为弥补HTML在显示属性设定上的不足而制定的一套扩展样式标准，是网页的美容师
  - 最初的HTML只包含很少的显示属性。为了满足页面设计者的要求，HTML添加了很多显示功能，HTML也就变得越来越杂乱、臃肿
- ◆ CSS为HTML标记语言提供了一种样式描述，定义了其中元素的显示方式
  - 提供了丰富的文档样式外观，以及设置文本和背景属性的能力
  - 允许为任何元素创建边框，以及元素边框与其他元素间的距离，以及元素边框与元素内容间的距离
  - 允许随意改变文本的大小写方式、修饰方式以及其他页面效果

◆ 一个CSS样式表由许多样式规则组成，用来控制网页元素的显示方式。规则的形式为：

■ 选择符{属性1： 值1； 属性2： 值2； ……}

◆ 规则例：

■ `td, input, select, body {font-family:Verdana;font-size:12px;}`

■ `select, body, textarea {background:#fff;font-size:12px;}`

■ `img {border:none}`

■ `h1 {color: #ff0000}`

- ◆ 丰富的样式定义
- ◆ 易于使用和修改：可将样式定义在HTML文档中，也可声明在一个专门的CSS文件中供HTML页面引用
- ◆ 多页面应用： CSS样式表可单独存放在一个CSS文件中，多个页面可使用同一个CSS样式表，实现多个页面风格的统一
- ◆ 页面压缩：将样式的声明单独放到CSS样式表中，可以大大减少页面大小。样式表还会被浏览器保存在缓存里，加快了复用同一样式表的网页的下载速度
- ◆ 层叠：对一个元素可多次设置同一样式，将使用最后一次设置的属性值进行显示

- ◆ 一种解释型脚本语言，广泛用于Web应用开发
  - 解释型脚本语言，代码不进行预编译
  - 主要用来向HTML页面添加交互行为，添加各式各样的动态功能，为用户提供更流畅美观的浏览效果
  - 可以直接嵌入HTML页面，但写成单独的js文件有利于结构和行为的分离
  - 跨平台特性，在绝大多数浏览器的支持下，可以在多种平台下运行（如Windows、Linux、Mac、Android、iOS等）
- ◆ 它的解释器被称为JavaScript引擎，为浏览器的一部分



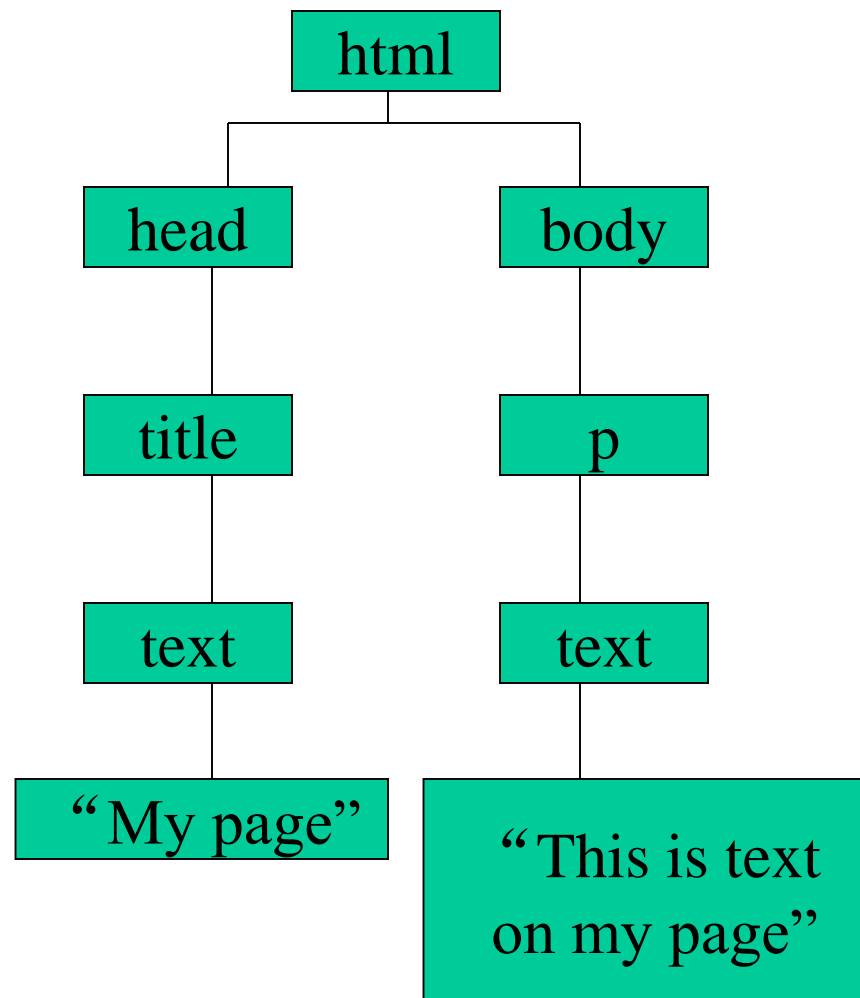
```
01      <html>                                <!--HTML文档开始-->
02      <body>                                <!--文档体开始-->
03      <script language="JavaScript">        <!--脚本程序-->
04          document.write("Hello World!"); // 输出经典的Hello world
05      </script>                             <!--脚本结束-->
06      </body>                             <!--文档体结束-->
07      </html>                             <!--HTML文档结束-->
```

```
01      <script language="javascript">        // 脚本程序开始
02      <!--
03      var poets = new Array( "王维", "杜甫", "李白", "白居易" );    // 创建数组
04      document.write("古代几个大诗人: <br>");    // 输出标题
05      for( n in poets )                    // 逐个输出数组元素
06      {
07          document.write( "<li>" + poets[n] );    // 输出诗人的名字
08      }
09      -->
10      </script>                            <!--脚本程序结束-->
```

- ◆ 嵌入动态文本于HTML页面
- ◆ 对浏览器事件做出响应
- ◆ 读写HTML元素
- ◆ 在数据被提交到服务器之前验证数据
- ◆ 检测访客的浏览器信息
- ◆ 控制cookies，包括创建和修改等
- ◆ 基于Node.js技术可以进行服务器端编程

- ◆ 如果JavaScript只是执行一些计算、循环等操作，而不能操作html也就失去了它存在的意义。JavaScript通过DOM操作html页面，让html变成Dhtml
- ◆ DOM（文档对象模型Document Object Model）就是html页面的模型，将每个标签都做为一个对象
- ◆ JavaScript通过调用DOM中的属性、方法就可以对网页中的 头、段落、列表、文本框、风格、ID等等元素进行编程控制
  - 比如通过操作文本框的DOM对象，就可以读取文本框中的值、设置文本框中的值

- ◆ `<html>`
- ◆ `<head>`
- ◆ `<title>My page</title>`
- ◆ `</head>`
- ◆ `<body>`
- ◆ `<p>This is text on my page</p>`
- ◆ `</body>`
- ◆ `</html>`



- ◆ getElementById(): 根据元素的Id获得对象
- ◆ getElementsByName(): 根据元素的name获得对象
- ◆ getElementsByTagName(): 获得指定标签名称的元素数组
- ◆ createElement('element'): 创建一个节点
- ◆ appendChild(node): 追加一个节点
- ◆ removeChild(node): 移除一个节点
- ◆ replaceChild(new,old): 替换一个节点
- ◆ insertBefore(new,参照): 把节点加到前面（插到某个节点前面）

- ◆ JavaScript Object Notation , JavaScript 对象表示法
- ◆ 是存储和交换文本信息的语法，是轻量级的文本数据交换格式
  - ◆ 比 XML 更小、更快，更易解析
- ◆ JSON 使用 Javascript语法来描述数据对象，但是 JSON 仍然独立于语言 and 平台
  - ◆ JSON 解析器和 JSON 库支持许多不同的编程语言。
  - ◆ 非常多的动态（PHP，JSP，.NET）编程语言都支持JSON

```
var myObj, x;  
myObj = { "name": "Mary", "id": 10000, "description": null };  
  
x = myObj.name;
```

- ◆ Dynamic HTML，就是动态的html，是相对传统的静态的html而言的一种制作网页的概念
- ◆ 它并不是一门新的语言，只是HTML、CSS和客户端脚本的一种集成，即一个页面中包括html+css+javascript（或其它客户端脚本）
- ◆ DHTML不是一种技术、标准或规范，只是一种将目前已有的网页技术、语言标准整合运用，制作出能在下载后仍然能实时变换页面元素效果的网页设计概念

- ◆ HTTP协议（HyperText Transfer Protocol，超文本传输协议）是用于从Web服务器传输超文本到本地浏览器的传输协议
- ◆ 工作于客户端-服务端架构上。浏览器作为HTTP客户端，通过URL向HTTP服务端即WEB服务器发送所有请求
- ◆ 基于TCP/IP通信协议来传递数据（HTML 文件, 图片文件, 查询结果等）
  - 默认端口号为80，但是也可以改为8080或者其他端口

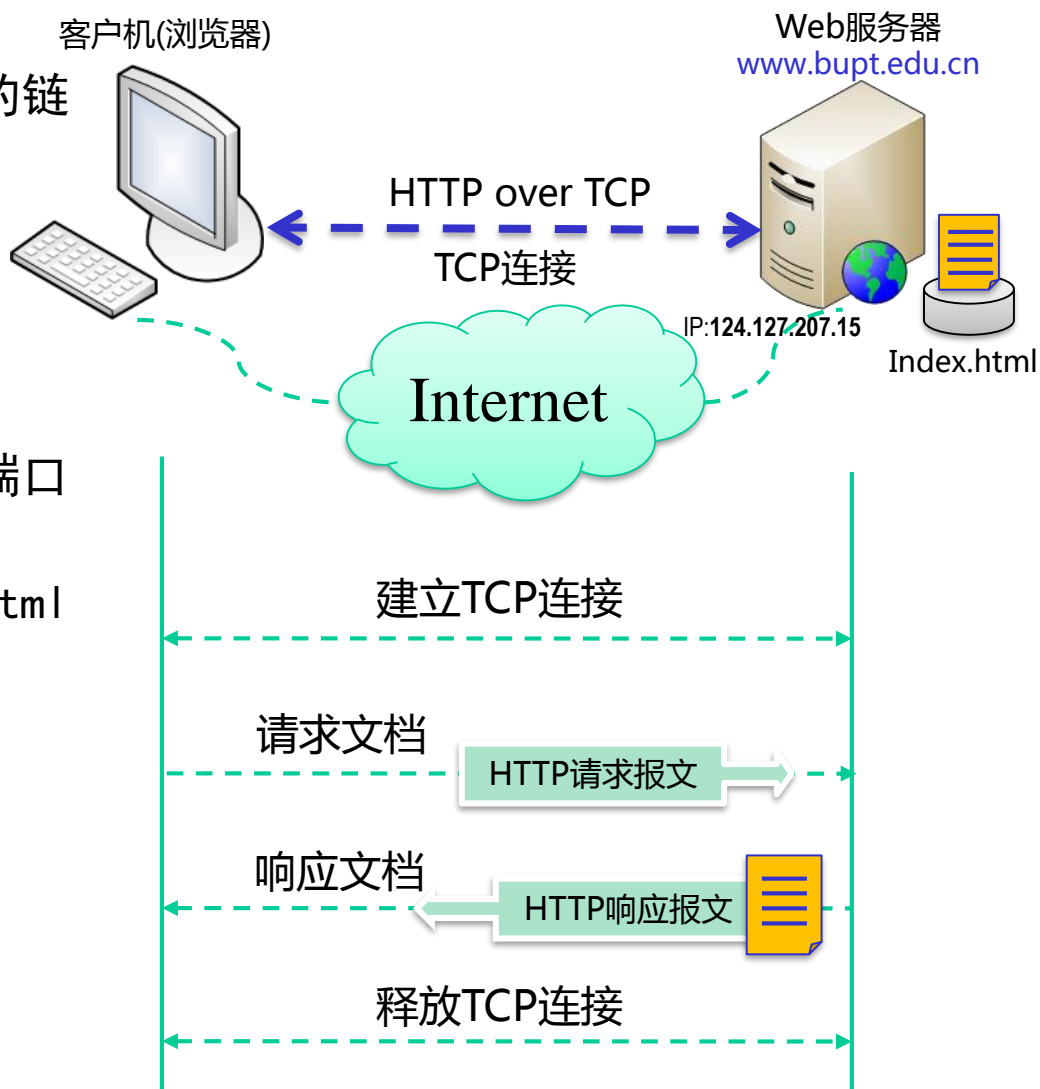


# HTTP的请求响应交互

- ◆ 在用户点击URL为 `http://www.bupt.edu.cn/index.html` 的链接后，浏览器执行以下动作：

- 浏览器分析超链接中的URL
- 浏览器向DNS请求解析 `www.bupt.edu.cn` 的IP地址
- DNS将解析出的IP地址 `124.127.207.15` 返回浏览器

- ◆ 浏览器与服务器建立TCP连接(80端口)
- ◆ 浏览器请求文档：GET `/index.html`
- ◆ 服务器给出响应，将文档 `index.html` 发送给浏览器
- ◆ 释放TCP连接
- ◆ 浏览器显示 `index.html` 中的内容



- ◆ HTTP消息分为请求消息和响应消息两类
- ◆ 请求消息：从客户端(浏览器)向Web服务器发送的请求报文
- ◆ 请求消息例：

```
GET /somedir/page.html HTTP/1.1
Host:www.bupt.edu.cn_
Connection:close
User-agent:Mozilla/4.0
Accept-language:zh-cn
```

◆ 方法(Method)是对所请求对象所进行的操作,也就是一些命令。请求报文中的操作有

方法(操作)	含义	方法(操作)	含义
<b>GET</b>	请求读取一个Web页面	<b>HEAD</b>	请求读取一个Web页面的首部
<b>POST</b>	附加一个命名资源(如Web页面)	<b>PUT</b>	请求存储一个Web页面
<b>DELETE</b>	删除Web页面	<b>TRACE</b>	用于测试,要求服务器送回收到的请求
<b>CONNECT</b>	用于代理服务器	<b>OPTION</b>	查询特定选项

## ◆ 从Web服务器到客户机(浏览器)的应答

## ◆ 响应消息例:

HTTP/1.1 200 OK

Connection:close

Date: Thu, 13 Oct 2005 03:17:33 GMT

Server: Apache/2.0.54 (Unix)

Last—Modified:Mon,22 Jun 1998 09;23;24 GMT

Content—Length:6821

Content—Type:text/html

(数据 数据 数据 数据 数据.....)

- ◆ 状态码 (Status-Code) 是响应报文状态行中包含的一个 3 位数字，指明特定的请求是否被满足，如果没有满足，原因是什么。状态码分为以下五类

状态码	含义	例子
1xx	通知信息	100=服务器正在处理客户请求
2xx	成功	200=请求成功(OK)
3xx	重定向	301=页面改变了位置
4xx	客户错误	403=禁止的页面；404=页面未找到
5xx	服务器错误	500=服务器内部错误；503=以后再试

# HTTP消息结构:首部字段或消息头

头(header)	类型	说明
User- Agent	请求	关于浏览器和它平台的信息，如Mozilla5.0
Accept	请求	客户能处理的页面的类型，如text/html
Accept-Charset	请求	客户可以接受的字符集，如Unicode-1-1
Accept-Encoding	请求	客户能处理的页面编码方法，如gzip
Accept-Language	请求	客户能处理的自然语言，如en(英语)，zh-cn(简体中文)
Host	请求	服务器的DNS名称。从URL中提取出来，必需。
Authorization	请求	客户的信息凭据列表
Cookie	请求	将以前设置的Cookie送回服务器，可用来作为会话信息
Date	双向	消息被发送时的日期和时间
Server	响应	关于服务器的信息，如Microsoft-IIS/6.0
Content-Encoding	响应	内容是如何被编码的（如gzip）
Content-Language	响应	页面所使用的自然语言
Content-Length	响应	以字节计算的页面长度
Content-Type	响应	页面的MIME类型
Last-Modified	响应	页面最后被修改的时间和日期，在页面缓存机制中意义重大
Location	响应	指示客户将请求发送给别处，即重定向到另一个URL
Set-Cookie	响应	服务器希望客户保存一个Cookie

## ◆ 非持久性连接

- 即浏览器每请求一个Web文档，就创建一个新的连接，当文档传输完毕后，连接就立刻被释放
- HTTP1.0、HTTP0.9采用此连接方式

## ◆ 持久性连接

- 即在一个连接中，可以进行多次文档的请求和响应。服务器在发送完响应后，并不立即释放连接，浏览器可以使用该连接继续请求其他文档
- HTTP/1.1默认使用持久连接

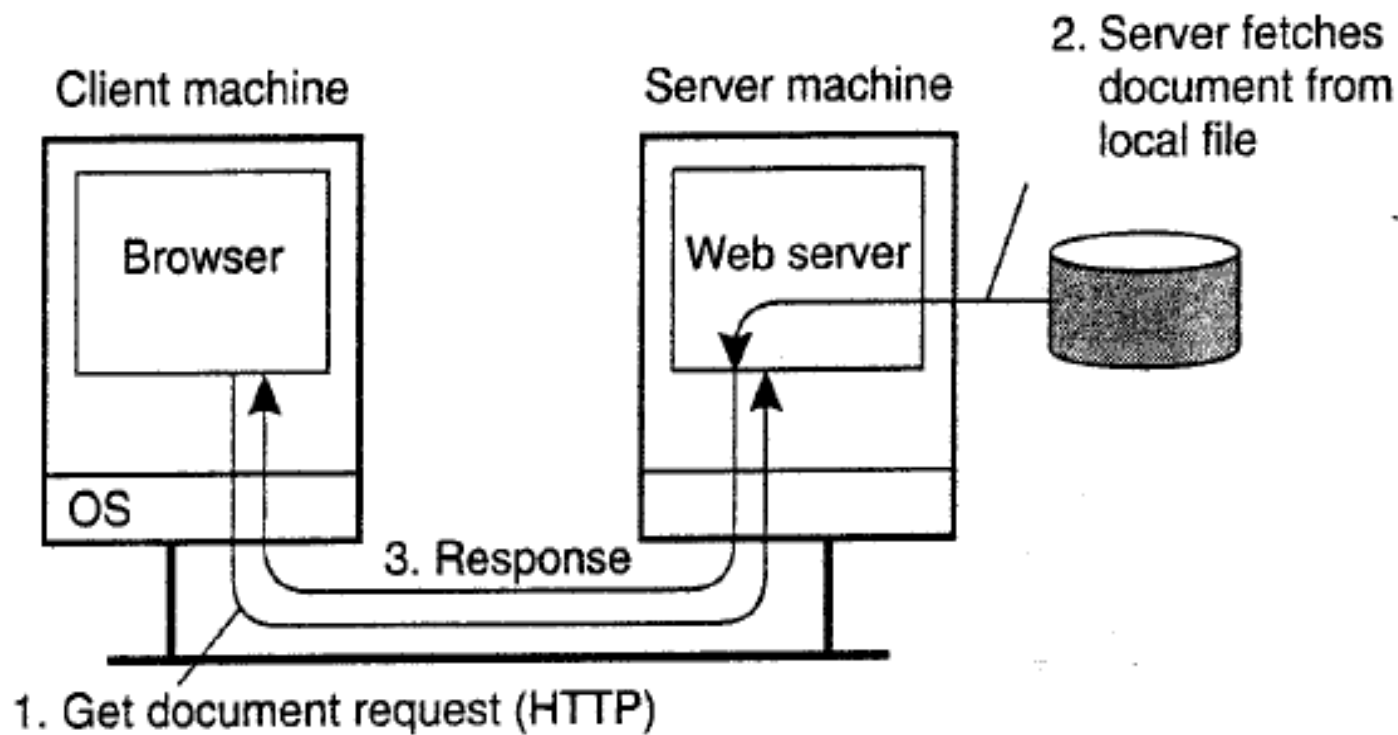
# HTTP的无状态性（1）

- ◆ 无状态协议，是指协议对于交互性场景没有记忆能力，协议本身并不保留之前一切的请求或响应报文的信息
- ◆ 对于一般的Web资源而言，一个URL对应着唯一的超文本，而HTTP服务器也绝对公平公正，不管请求者是谁，之前还请求过什么，它都会根据接收到的URL请求返回相同的超文本。**每次请求都是独立的**
  - 这样，记录用户的行为状态变得毫无意义，所以，HTTP协议被设计为无状态协议符合它本身的需求
- ◆ HTTP的无状态性简化了服务器的设计，使其更容易支持大量并发的HTTP请求
  - 集群、cache都方便

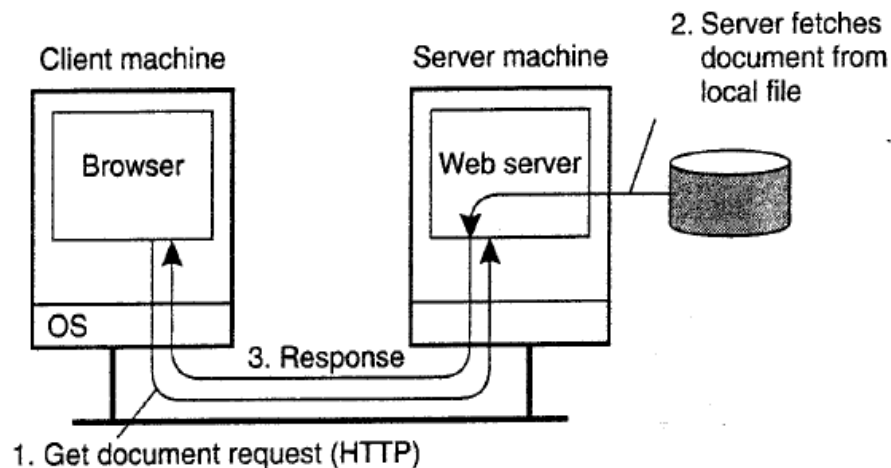


- ◆ 为支持有状态的服务，HTTP引入了Cookie技术，例如
  - 当客户端访问服务器时，服务器根据需求设置 Session，将会话信息保存在服务器上，同时将标识Session 的 SessionId 传递给客户端浏览器，浏览器将这个 SessionId 进行保存
  - 在这个会话期间，浏览器每次请求都会额外加上这个参数值，服务器根据这个 SessionId，就能取得该客户的会话信息进行有针对性的响应
- ◆ Cookie技术为传递用于标识客户身份或者会话ID的信息提供了支持，但这些信息并不是协议自身保存的
  - HTTP仍然是一个无状态协议

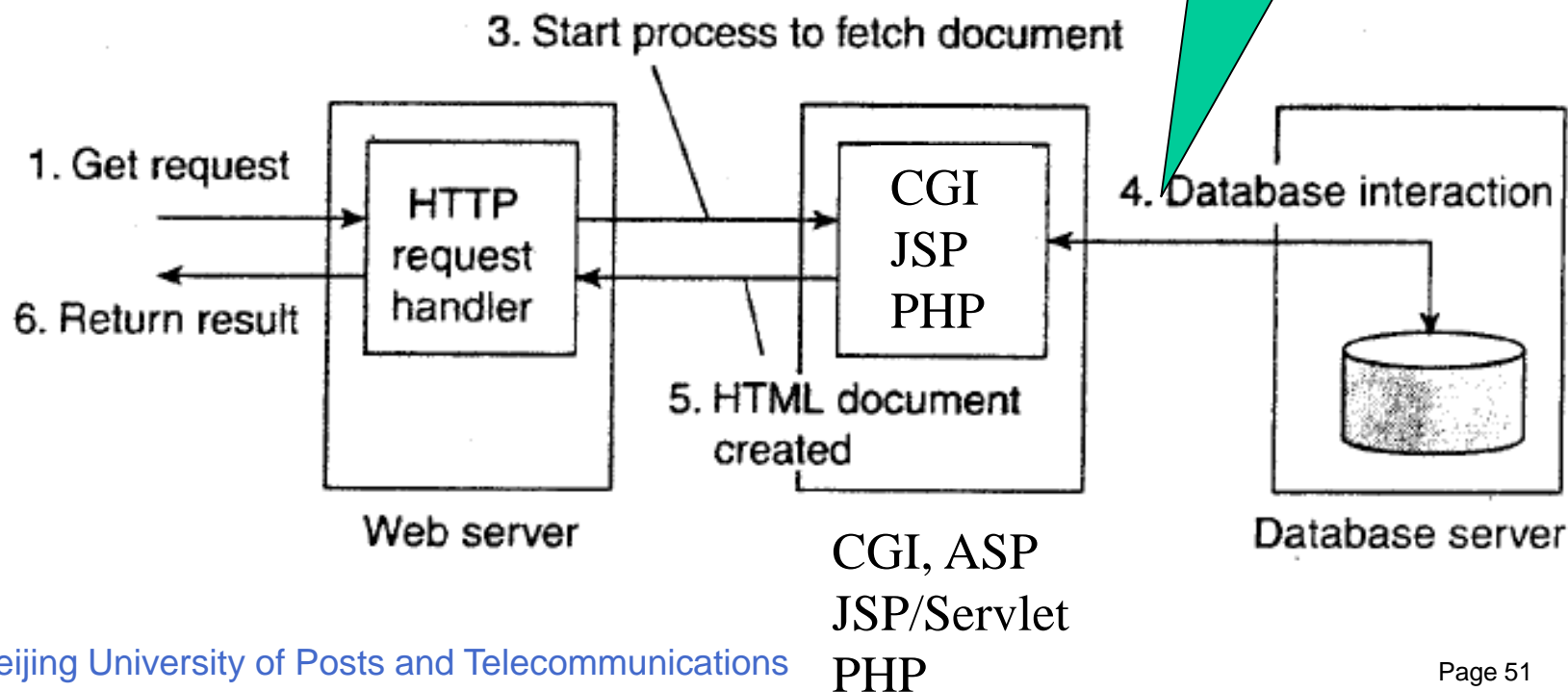
# 传统Web网站的常见组织 (Web1.0)



# 多层结构 (Web 1.5?)



如果业务逻辑复杂，可增加业务逻辑层



- Java类
- 运行在服务端
- 不需要图形界面
- 接收HTTP请求，动态生成HTTP响应

```
public class GetDemo extends HttpServlet{
    public void doGet(HttpServletRequest req,
        HttpServletResponse resp)
        throws ServletException, IOException{
        resp.setContentType("text/html");
        PrintWriter out = resp.getWriter();
        String username = req.getParameter("uname");
        String password = req.getParameter("userpw");
        out.println("<HTML>");
        .....
        out.println("</HTML>");
        out.close();
    }
}
```

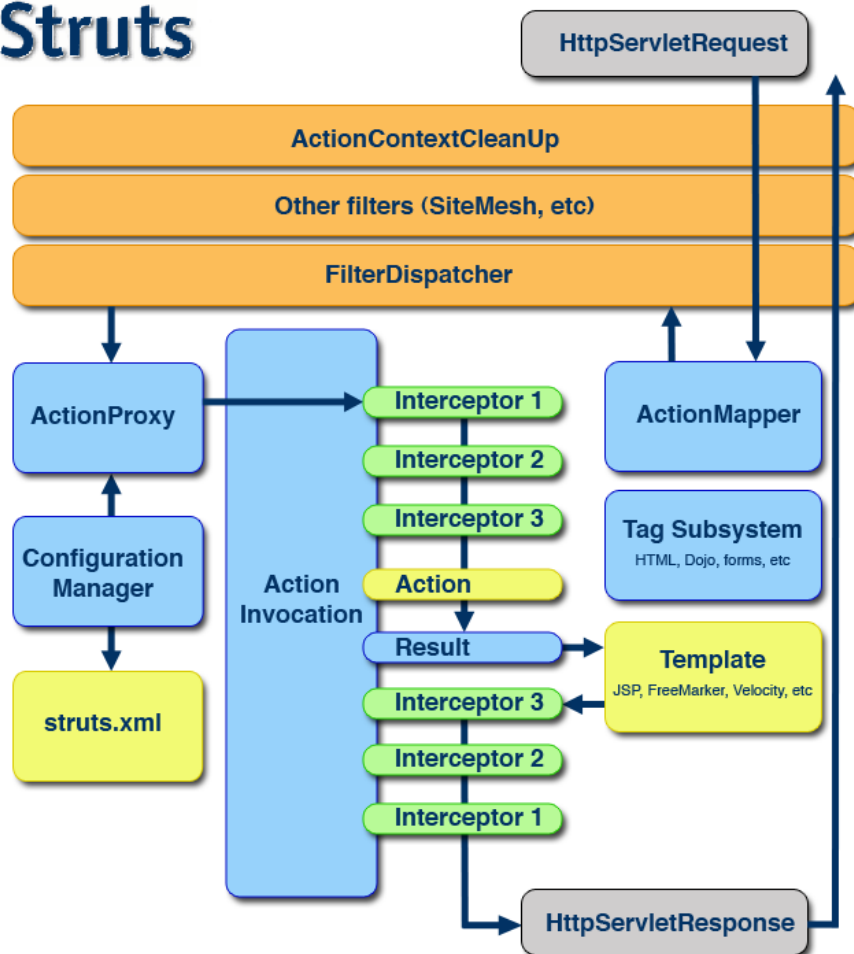
- ◆ HTML document: 嵌入了JSP特定的tag; 在页面中加入Java代码, 动态生成页面的内容
- ◆ 在服务器端, JSP页面被编译成Java Servlet执行

```
<html>
<head><title>Ch2 Simple jsp</title></head>
<body>
<%@ page import="java.util.*" %>
<h2>Date and Time: <%= new Date().toString() %></h2>
<hr>
<% for(int i=0; i<10; i++) %>
hello<br>
<% %>
how are u?<br>
</body>
</html>
```

- ◆ JQuery: 简洁的JavaScript库, 使用户能更方便地处理HTML documents、events、实现动画效果, 并且提供AJAX交互。实际上是类库
- ◆ Backbone.js: MVC JavaScript 框架, 可用于创建复杂的JavaScript应用程序和单页界面
- ◆ Bootstrap: 基于 HTML、CSS、JAVASCRIPT 的Web 框架。是来自Twitter的CSS / HTML框架
- ◆ Angular.js: MV\* (Model-View-Whatever, 不管是MVC或者MVVM, 统归MDV(model Drive View)) JavaScript框架。是Google推出的一个应用框架
- ◆ React.js: 用于构建可重复使用的UI组件, 是一个视图库, 对应于MVC中的V。起源于 Facebook

- ◆ Struts: 是一个采用JavaServlet/JSP 实现MVC Web应用的框架。

## Struts



Key:

■ Servlet Filters 
 ■ Struts Core 
 ■ Interceptors 
 ■ User created



## ◆ Web 服务器、Web网页、浏览器

- 静态和动态网页

## ◆ 主要技术例

- HTML、CSS
- JavaScript、DOM、JSON
- HTTP
- JSP/Servlets