

# 1. Java Web 技术栈

## 1.1 B/S架构

Browser/Server, 浏览器/服务器 架构模式, 他的特点是, 客户端只需要浏览器, 应用程序的逻辑和数据都存储在服务器端, 浏览器只需要请求服务器, 获取Web资源, 服务器吧把Web、资源发送给浏览器即可



好处: 易于维护升级, 服务器端升级后, 客户端无需任何部署就可以使用到新的版本

### 1.1.1 静态资源

HTML、CSS、JavaScript、图片等。负责页面展现

### 1.1.2 动态资源

Servlet、JSP 等。负责逻辑处理

### 1.1.3 数据库

负责存储数据

### 1.1.4 HTTP协议

定义通信规则

### 1.1.5 Web服务器

负责解析HTTP协议, 解析请求数据, 并发送响应数据

## 2. HTTP

== HTTP(Hyper Text Transfer Protocol),超文本传输协议, 规定了浏览器和服务器之间数据传输的规则 ==

### HTTP协议特点

- 1、基于TCP协议: 面向连接、安全
- 2、基于请求 响应模型: 一次请求对应一次响应
- 3、HTTP协议是无状态的协议: 对于事务处理没有记忆功能。每次请求 响应都是独立的

缺点: 多次请求不能共享数据。Java中使用会话技术 (Cookie、Session) 来解决这个问题

优点: 速度快

传输控制协议（TCP）：TCP（传输控制协议）定义了两台计算机之间进行可靠的传输而交换的数据和确认信息的格式，以及计算机为了确保数据的正确到达而采取的措施。协议规定了TCP软件怎样识别给定计算机上的多个目的进程如何对分组重复这类差错进行恢复。协议还规定了两台计算机如何初始化一个TCP数据流传输以及如何结束这一传输。TCP最大的特点就是提供的是面向连接、可靠的字节流服务。

用户数据报协议（UDP）：UDP（用户数据报协议）是一个简单的面向数据报的传输层协议。提供的是非面向连接的、不可靠的数据流传输。UDP不提供可靠性，也不提供报文到达确认、排序以及流量控制等功能。它只是把应用程序传给IP层的数据报发送出去，但是并不能保证它们能到达目的地。因此报文可能会丢失、重复以及乱序等。但由于UDP在传输数据报前不用在客户和服务器之间建立一个连接，且没有超时重发等机制，故而传输速度很快。

## 2.1 请求数据格式

1. 请求行：请求数据的第一行。GET表示请求方式，/表示请求资源路径，HTTP/1.1表示协议版本
2. 请求头：第二行开始，格式为Key：value形式
3. 请求体：POST请求的最后一部分，存放请求参数
4. ● 常见的HTTP 请求头：
  - Host: 表示请求的主机名
  - User-Agent: 浏览器版本，例如Chrome浏览器的标识类似Mozilla/5.0 ... Chrome/79，IE浏览器的标识类似Mozilla/5.0 (Windows NT ...) like Gecko；
  - Accept: 表示浏览器能接收的资源类型，如text/\*，image/\*或者/\*/\*表示所有；
  - Accept-Language: 表示浏览器偏好的语言，服务器可以据此返回不同语言的网页；
  - Accept-Encoding: 表示浏览器可以支持的压缩类型，例如gzip, deflate等。

```
GET / HTTP/1.1
Host: www.itcast.cn
Connection: keep-alive
Cache-Control: max-age=0 Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 Chrome/91.0.4472.106
...
```

```
POST / HTTP/1.1
Host: www.itcast.cn
Connection: keep-alive
Cache-Control: max-age=0 Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 Chrome/91.0.4472.106

username=superbaby&password=123456
```

### GET请求和POST请求的区别：

1. GET请求请求参数在请求行中，没有请求体；POST请求请求参数在请求体中
2. GET请求请求参数大小有限制，POST没有

## 2.2 响应数据格式

1. 响应行：响应数据的第一行。其中HTTP/1.1表示协议版本，200表示响应状态码，OK表示状态码描述
2. 响应头：第二行开始，格式为Key：value形式
3. 响应体：最后一部分。存放响应数据

## 2.2.1状态码分类

状态码分类	类别	原因短语
1XX	Informational（信息性状态码）	接受的请求正在处理
2XX	Success（成功状态码）	请求正常处理完毕
3XX	Redirection（重定向状态码）	需要进行附加操作以完成请求
4XX	Client Error（客户端错误状态码）	服务器无法处理请求
5XX	Server Error（服务器错误状态码）	服务器处理请求出错

## 2.2.2常用状态码

状态码	英文描述	解释
200	OK	客户端请求成功，即 <b>处理成功</b> ，这是我们最想看到的状态码
302	Found	指示所请求的资源已移动到由Location响应头给定的URL，浏览器会自动重新访问到这个页面
304	Not Modified	告诉客户端，你请求的资源至上次取得后，服务端并未更改，你直接用你本地缓存吧。隐式重定向
400	Bad Request	客户端请求有 <b>语法错误</b> ，不能被服务器所理解
403	Forbidden	服务器收到请求，但是 <b>拒绝提供服务</b> ，比如：没有权限访问相关资源
404	Not Found	<b>请求资源不存在</b> ，一般是URL输入有误，或者网站资源被删除了
428	Precondition Required	<b>服务器要求有条件的请求</b> ，告诉客户端要想访问该资源，必须携带特定的请求头
429	Too Many Requests	<b>太多请求</b> ，可以限制客户端请求某个资源的数量，配合Retry-After(多长时间后可以请求)响应头一起使用
431	Request Header Fields Too Large	<b>请求头太大</b> ，服务器不愿意处理请求，因为它的头部字段太大。请求可以在减少请求头域的大小后重新提交。
405	Method Not Allowed	请求方式有误，比如应该用GET请求方式的资源，用了POST
500	Internal Server Error	<b>服务器发生不可预期的错误</b> 。服务器出异常了，赶紧看日志去吧
503	Service Unavailable	<b>服务器尚未准备好处理请求</b> ，服务器刚刚启动，还未初始化好
511	Network Authentication Required	<b>客户端需要进行身份验证才能获得网络访问权限</b>

状态 | [Status - HTTP 中文开发手册 - 开发者手册 - 腾讯云开发者社区-腾讯云\(tencent.com\)](#)

# 3.web服务器——tomcat

Tomcat是Apache软件基金会一个核心项目，是一个开源免费的轻量级Web服务器，支持Servlet/JSP少量JavaEE规范

# 4.网页

## 4.1 网页的结构

1. 结构——HTML
- 相当于人体的骨架

## 2. 表现——CSS

外在表现（所有能看到的，如：颜色、大小、样式）

## 3. 行为——JavaScript

人与网页的一个交互，JS响应用户操作

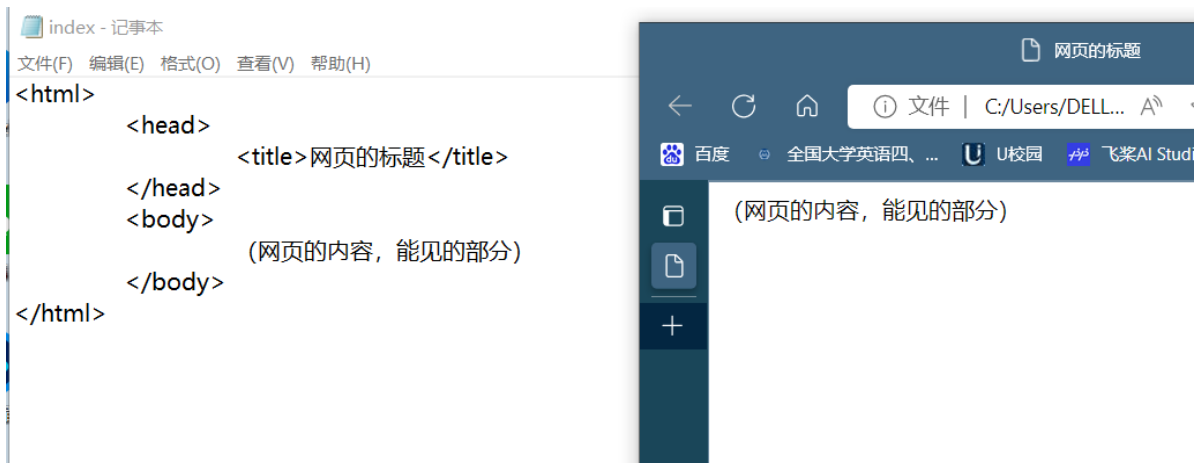
# 5.HTML

HTML（Hypertext Markup Language）超文本标记语言

它负责网页的三个要素中的结构

HTML使用标签的形式来标识网页中的不同组成部分

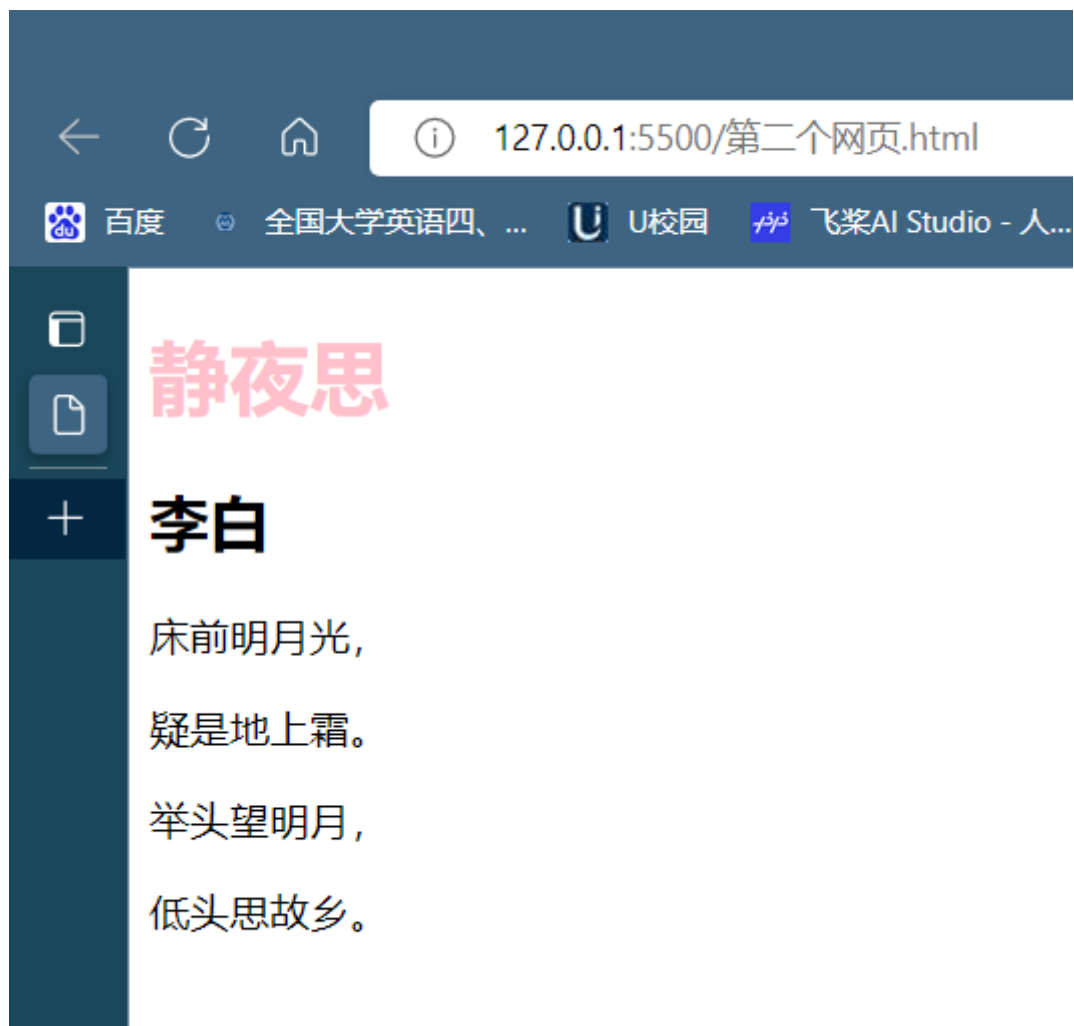
所谓超文本指的是超链接，使用超链接可以让我们从一个页面跳到另一个页面



```

<> 第二个网页.html > ...
1  <!--
2  |   快捷方式（自动生成）：1、html->html5； 2、!->Tab键
3  -->
4  <!doctype html>
5  <!--
6  |   文档声明(doctype)
7  |   - 文档声明用来告诉浏览器当前网页的版本
8  |   <!doctype html>
9  |   <!Doctype HTML>
10 -->
11 <!--
12 |   html的根标签（根元素），网页中的所有内容都要写在根元素的里边
13 -->
14 <html>
15   <head>
16   <!--head是网页的头部，head中的内容不会在网页中直接出现，主要用来帮助浏览器或搜索引擎来解析网页-->
17   <meta charset="utf-8">
18   <!-- 可以通过meta标签来设置网页的字符集（charset），避免乱码问题-->
19   <title>标签的属性</title>
20   <!--title中的内容会显示在浏览器的标签栏-->
21   </head>
22   <body>
23   <!--
24   |   body是html的子元素，表示网页的主体
25   -->
26   <h1><font color="pink">静夜思</font></h1>
27   <!--
28   |   属性--><font color="pink">在标签中(开始或者结束标签)还可以设置属性
29   |   属性是一个名值对(x = y)
30   |   属性用来设置标签中的内容如何显示
31   |   属性和标签名或者其他属性应该使用空格隔开
32   -->
33   <h2>李白</h2>
34   <p>床前明月光，</p>
35   <p>疑是地上霜。</p>
36   <p>举头望明月，</p>
37   <p>低头思故乡。</p>
38   </body>
39 </html>
40

```



## 5.1 HTML实体

HTML 中的预留字符必须被替换为字符实体。

## HTML 中有用的字符实体

注释：实体名称对大小写敏感！

显示结果	描述	实体名称	实体编号
	空格	&nbsp;	&#160;
<	小于号	&lt;	&#60;
>	大于号	&gt;	&#62;
&	和号	&amp;	&#38;
"	引号	&quot;	&#34;
'	撇号	&apos; (IE不支持)	&#39;
¢	分 (cent)	&cent;	&#162;
£	镑 (pound)	&pound;	&#163;
¥	元 (yen)	&yen;	&#165;
€	欧元 (euro)	&euro;	&#8364;
§	小节	&sect;	&#167;
©	版权 (copyright)	&copy;	&#169;
®	注册商标	&reg;	&#174;
™	商标	&trade;	&#8482;
×	乘号	&times;	&#215;
÷	除号	&divide;	&#247;

## 5.2 meta标签

meta主要用于设置网页中的一些元数据

name -- 制定网页的名称

charset -- 制定网页的字符集

content -- 制定的数据的内容

keywords -- 表示网站的关键字

description -- 网站的描述会显示在搜索引擎的结果中

title -- 标签的内容会作为搜索结果的超链接显示

```
<> 静夜思.html    <> 第二个网页.html    <> 实体.html    <> meta标签.html ●
<> meta标签.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name = "keywords" content = "HTML, 前端, CSS3">
6      <meta name = "description" content = "这是一个不错的网站">
7      <meta http-equiv = "refresh" content="3;url=https://www.baidu.com">
8      <title>meta标签</title>
9  </head>
10 <body>
11
12 </body>
13 </html>
```