

一、前序知识

1. 什么是web

Web (World Wide Web) 即全球广域网，也称为万维网，它是一种基于超文本和HTTP的、全球性的、动态交互的、跨平台的分布式图形信息系统。是建立在Internet上的一种网络服务，为浏览者在Internet上查找和浏览信息提供了图形化的、易于访问的直观界面，其中的文档及超级链接将Internet上的信息节点组织成一个互为关联的网状结构。

总结：Web在计算机网页开发设计中就是网页的意思

2. web可以做什么

PC端

- 京东商城、淘宝网、学院网站

移动端

- Android APP、微信小程序

3. 软件架构

应用软件分为两大类：

C/S架构，特点：需要安装、偶尔更新、不跨平台、开发更具针对性。

B/S架构，特点：无需安装、无需更新、可跨平台、开发更具通用性。

名词解释：C => client (客户端)、B => browser (浏览器)、S => server (服务器)。

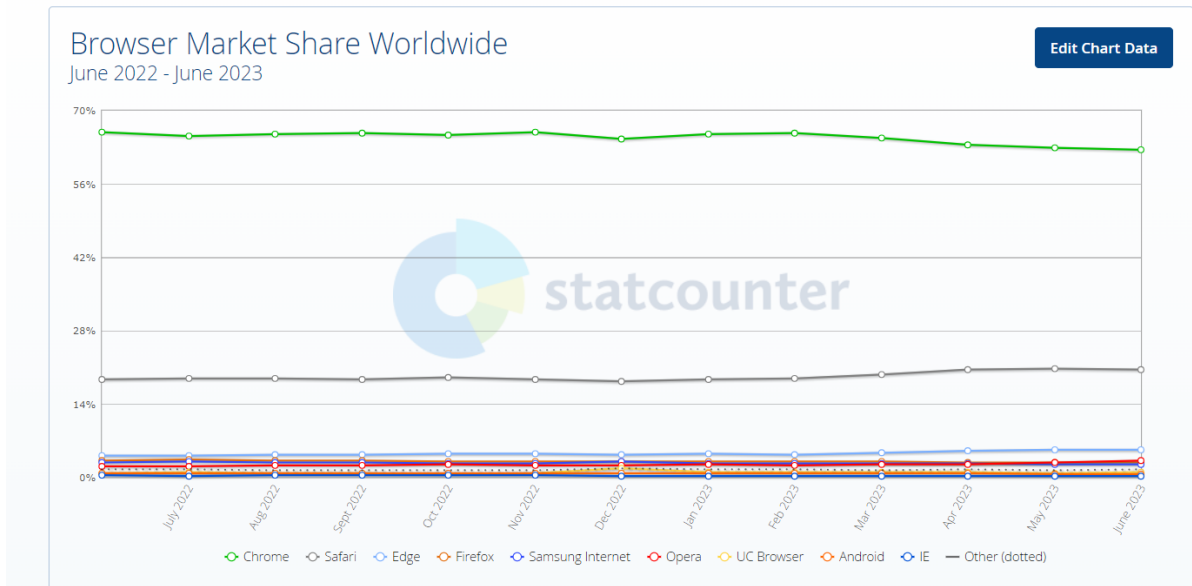
服务器：为软件提供数据的设备（在背后默默的付出）。

前端工程师，主要负责编写 B/S架构中的网页（呈现界面、实现交互）。

4. 浏览器相关知识

浏览器是网页运行的平台，常见的浏览器有：谷歌(Chrome)、Safari、IE、火狐(Firefox)、欧朋(Opera)等，以上这些是常用的五大浏览器。

1.各大浏览器市场份额：



2. 常见浏览器的内核:



早期使用 **webkit** 内核，现在使用 **Blink** 内核。

Chrome



使用 **webkit** 内核。

Safari



使用 **Trident** 内核。

IE



使用 **Gecko** 内核。

Firefox



早期使用 **Presto** 内核（已放弃维护），现在使用 **Blink** 内核。

Opera

五大浏览器
四大内核



其它

在上述内核的基础上，加上了一些精美的UI界面（皮肤）、实用的小功能。

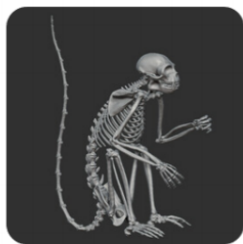
5. 网页相关概念

1. 网址：我们在浏览器中输入的地址。
2. 网页：浏览器所呈现的每一个页面。
3. 网站：多个网页构成了一个网站。
4. web标准：

web标准不是某一个标准,而是由W3C和其他标准化组织制定的一系列标准的集合。

主要包括：结构（Structure）、表现（Presentation）和行为（Behavior）三个方面。

结构



HTML

表现



CSS

行为



JavaScript

结构标准：结构用于对网页元素进行整理和分类，主要学的是HTML。

表现标准：表现用于设置网页元素的板式、颜色、大小等外观样式，主要指的是CSS

行为标准：行为是指网页模型的定义及交互的编写，主要学的是JavaScript、jQuery

二、HTML简介

1. 什么是HTML?

全称：HyperText Markup Language（超文本标记语言）。

超文本：暂且简单理解为“超级的文本”，和普通文本比，内容更丰富。

标 记：文本要变成超文本，就需要用到各种标记符号。

语 言：每一个标记的写法、读音、使用规则，组成了一个标记语言。

2. 相关国际组织（了解）

1. IETF

全称：Internet Engineering Task Force（国际互联网工程任务组），成立于1985年底，是一个权威

的互联网技术标准化组织，主要负责互联网相关技术规范的研发和制定，当前绝大多数国际互联网技术标准均出自IETF。官网：<https://www.ietf.org>

2. W3C

全称：World Wide Web Consortium（万维网联盟），创建于1994年，是目前Web技术领域，最具影

响力的技术标准机构。共计发布了200多项技术标准和实施指南，对互联网技术的发展和应用到基础性、根本性的支撑作用，官网：<https://www.w3.org>

3. WHATWG

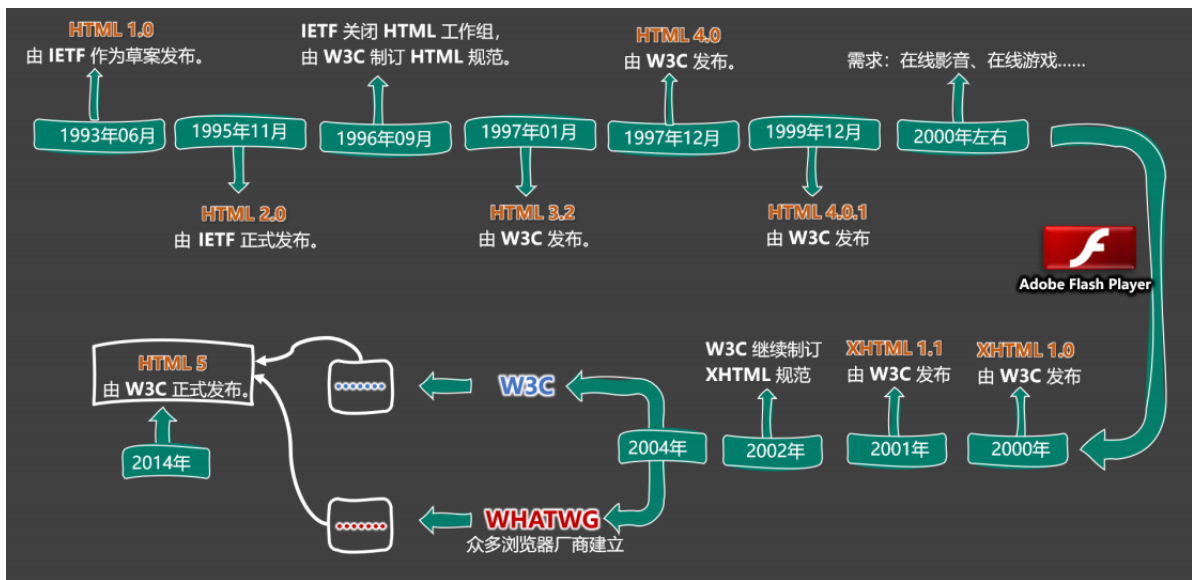
全称：Web Hypertext Application Technology Working Group（网页超文本应用技术工作小组）成立

于2004年，是一个以推动网络HTML 5 标准为目的而成立的组织。由Opera、Mozilla基金会、苹果，

等这些浏览器厂商组成。官网：<https://whatwg.org/>

3. HTML发展史（了解）

从 HTML 1.0 开始发展，期间经历了很多版本，目前HTML的最新标准是：HTML 5，具体发展史如图（了解即可）。



4. HTML结构

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title></title>
  </head>
  <body>

</body>
</html>
```

!DOCTYPE html: 文档类型

html :所有HTML标签中的一个根节点。最大的标签 跟标签

head :头部标签, 文档的头部描述了文档的各种属性和信息, 包括文档的标题

meta charset="UTF-8" :设置字符编码为UTF-8

title :文档的标题, 让页面拥有一个属于自己的标题

body:文档的主体,body元素包含文档的所有内容(例如: 文本、超链接、图像、表格、列表等)

5. HTML初体验

第一步: 鼠标右键 => 新建 => 文本文档 => 输入以下内容, 并保存。

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>第一个网页</title>
  </head>
  <body>
    率真存厚、立志有恒, 奋斗为本, 成就客户, 创造分享, 共同成长
  </body>
</html>
```

第二步: 修改后缀为 .html , 然后双击打开即可。

这里的后缀名，使用 .htm 也可以，但推荐使用更标准的 .html。

程序员写的叫 **源代码**，要交给浏览器进行渲染。

6. HTML标签

在HTML页面中，带有“<>”符号的元素被称为HTML标签，如上面提到的、< head >、



并列关系



7. 字符集



```
<meta charset="UTF-8">
```

utf-8是目前最常用的字符集编码方式，常用的字符集编码方式还有gbk和gb2312。

gb2312 简单中文 包括6763个汉字

BIG5 繁体中文 港澳台等用

GBK包含全部中文字符 是GB2312的扩展，加入对繁体字的支持，兼容GB2312

UTF-8则包含全世界所有国家需要用到的字符



记住一点，以后我们统统使用**UTF-8** 字符集，这样就避免出现字符集不统一而引起乱码的情况了。

8. HTML常用标签

首先 HTML和CSS是两种完全不同的语言，我们学的是结构，就只写HTML标签，认识标签就可以了。不会再给结构标签指定样式了。

HTML标签有很多，这里我们学习最为常用的，后面有些较少用的，我们可以查下手册就可以了。

8.1. 排版标签

排版标签主要和css搭配使用，显示网页结构的标签，是网页布局最常用的标签。

8.1.1. 标题标签（熟记）

单词缩写： head 头部. 标题

为了使网页更具有语义化，我们经常会在页面中用到标题标签，HTML提供了6个等级的标题，即：

、

1. 、

1.1. 、

1.1.1. 、

1.1.1.1. 、

1.1.1.1.1.

标题标签语义：作为标题使用，并且依据重要性递减

基本语法格式： 标题文本

1.1.2. 段落标签（熟记）

单词缩写： paragraph 段落

在网页中要把文字有条理地显示出来，离不开段落标签，就如同我们平常写的文章一样，整个网页可以分为若干个段落，段落标签即：



```
<p>文本内容</p>
```

是HTML文档中最常见的标签，默认情况下，文本在一个段落中会根据浏览器窗口的大小自动换行。

1.1.3. 水平线标签（认识）

单词： horizontal 横线

在网页中常常看到一些水平线将段落与段落之间隔开，使得文档结构清晰，层次分明。这些水平线可以通过插入图片实现，也可以简单地通过标签来完成



```
<hr/>是单标签
```

在网页中显示默认样式的水平线

课堂练习：新闻页面

40.6摄氏度：上海创出140年气象史上高温新纪录

2016年7月27日 10:58:26 来源： 新华网

新华网上海7月26日电（记者李荣）26日13时36分至47分这一时间段，上海徐家汇气象观测站测得当日最高温达40.6摄氏度。这是上海有气象记录以来140年的高温新纪录，打破了此前1934年创下的40.2摄氏度的历史极值。

上海已经连续2天发出了最高等级的红色高温警报。上海中心气象台首席服务官满莉萍说，今年副热带高压强度特别强，对上海及江南地区的控制“实在太稳定了”，整个7月份基本上都处在它的势力范围之内。25日上海已出现了气象史上7月份“第四高”的高温值，这使得26日的“基础”气温就很高，超过了30摄氏度，然后不断地升温。此外，26日白天风小，又吹的是西南风，特别是在中午之后这一个最易出现高温的时段，光照又比较强，所以气温“直线飙升”，一举冲破历史极值，出现了“创纪录”的极端酷暑天。

。在上海历史上，出现40摄氏度以上极端高温的几率并不大。根据相关资料，中心城区徐家汇观测站140年来仅出现了5次记录，除了这一次的新纪录，还有就是1934年7月12日的40.2摄氏度；1934年8月25日、2009年7月20日、2010年8月13日的40摄氏度。

由于气温实在太高，上海26日下午不少地区出现了热对流天气。气象台说，首先是浦东地区，下起了热雷雨。到15时05分，全市大部分地区出现了分散性的雷电活动和热雷雨，中心城区徐家汇等地都响起了隆隆的雷声。

1.1.4. 换行标签 (熟记)

单词缩写: break 打断,换行

在HTML中, 一个段落中的文字会从左到右依次排列, 直到浏览器窗口的右端, 然后自动换行。如果希望某段文本强制换行显示, 就需要使用换行标签

```
<br />
```

1.1.5. div span标签(重点)

div span 是没有语义的 是我们网页布局主要的2个盒子

div 就是 division 的缩写 分割, 分区的意思 其实有很多div 来组合网页。

span, 跨度, 跨距; 范围

语法格式:

```
<div> 这是头部 </div>    <span>今日价格</span>
```

生活中的例子: div 是大包装袋, span 是小包装袋。

1.1.6. 块级元素和行内元素

1. **块级元素**: 独占一行 (排版标签都是块级元素)。
2. **行内元素**: 不独占一行 (目前只学了: input, 稍后会学习更多)。
3. **使用原则**:

1. **块级元素** 中能写 **行内元素** 和 **块级元素** (简单记: 块级元素中几乎什么都能写)。

2. **行内元素** 中能写 **行内元素**, 但不能写 **块级元素**。

3. 一些特殊的规则:

- h1~h6 不能互相嵌套。
- p 中不要写块级元素。

备注: marquee 元素设计的初衷是: 让文字有动画效果, 但如今我们可以通过 CSS 来实现了, 而且还可以实现的更加炫酷, 所以 marquee 标签已经: **过时了** (废弃了), 不推荐使用。

1.1.7. 文本格式化标签(熟记)

在网页中有时需要为文字设置粗体、斜体及下划线效果。这时就需要用到HTML中的文本格式化标签, 使文字以特殊的方式显示

标签	显示效果
和	文字以粗体方式显示
和	文字以斜体方式显示
和	文字以加删除线的方式显示
和	文字以加下划线方式显示

1.1.8. 标签属性

使用HTML制作网页时，如果想让HTML标签提供更多的信息，可以使用HTML标签的属性加以设置。其基本语法格式如下：

```
<标签名 属性1="属性值1" 属性2="属性值2" ...> 内容 </标签名>
```

在上面的语法中，

1. 标签可以拥有多个属性，必须写在开始标签中，位于标签名后面。
2. 属性之间不分先后顺序，标签名与属性、属性与属性之间均以空格分开。
3. 任何标签的属性都有默认值，省略该属性则取默认值。

采取 键值对 的格式 key="value" 的格式

示例：

```
<hr width=400/  
属性是宽度  
值是 400
```

1.1.9. 图像标签img (重点)

单词：image 图像

HTML网页中任何元素的实现都要依靠HTML标签，要想在网页中显示图像就需要使用图像标签

其基本语法格式如下：

```

```

标记属性

属性	属性值	描述
src	URL	图像的路径
alt	文本	图像不能显示时提示的文字
title	文本	鼠标悬停时显示的内容
width	像素	设置图像的宽度
height	像素	设置图像的高度
border	数字	设置图像边框的宽度

常见图片格式

a. jpg格式

概述：扩展名为 .jpg 或 .jpeg，是一种有损的压缩格式（把肉眼不容易观察出来的细节丢弃了）。

主要特点：**支持的颜色丰富、占用空间较小**、不支持透明背景、不支持动态图。

使用场景：对图片细节**没有极高要求**的场景，例如：网站的产品宣传图等。——该格式网页中很常见。

b. png格式

概述：扩展名为 .png，是一种无损的压缩格式，能够更高质量的保存图片。

主要特点：**支持的颜色丰富**、占用空间略大、**支持透明背景**、不支持动态图。

使用场景：①想让图片有透明背景；②想更高质量的呈现图片；例如：公司logo图、重要配图等。

c. bmp 格式

概述：扩展名为 .bmp，不进行压缩的一种格式，在最大程度上保留图片更多的细节。

主要特点：**支持的颜色丰富、保留的细节更多**、占用空间极大、不支持透明背景、不支持动态图。

使用场景：对图片细节**要求极高**的场景，例如：一些大型游戏中的图片。（网页中很少使用）

d. gif格式

概述：扩展名为 .gif，仅支持256种颜色，色彩呈现不是很完整。

主要特点：支持的颜色较少、**支持简单透明背景、支持动态图**。

使用场景：网页中的动态图片。

e. webp格式：

概述：扩展名为 .webp，谷歌推出的一种格式，专门用来在网页中呈现图片。

主要特点：具备上述几种格式的优点，但兼容性不太好，一旦使用务必要解决兼容性问题。

使用场景：网页中的各种图片。

f. base64 格式

1. 本质：一串特殊的文本，要通过浏览器打开，传统看图应用通常无法打开。
2. 原理：把图片进行 base64 编码，形成一串文本。
3. 如何生成：靠一些工具或网站。
4. 如何使用：直接作为 img 标签的 src 属性的值即可，并且不受文件位置的影响。
5. 使用场景：一些较小的图片，或者需要和网页一起加载的图片。

1.1.10. 链接标签(重点)

在HTML中创建超链接标签非常简单，只需要用标签环绕需要链接的对象即可，其基本语法格式如下：



```
<a href="跳转的目标" target="目标窗口的弹出方式">文本或图像</a>
```

href:用于指定链接目标的URL地址，当为标签应用href属性时，它就具有了超链接的功能。Hypertext Reference的缩写。意思是超文本引用。

Target: 用于指定链接页面的打开方式，其取值有self和blank两种，其中self为默认值，blank为在新窗口中打开方式

1.1.11. base标签

base可以设置整体链接的打开方式，其基本语法格式如下：

```
<base target="_blank"/>
```

1.1.12. 特殊字符标签（理解）

HTML为特殊字符准备了专门的替代代码

特殊字符	描述	字符代码
	空格符	
>	大于号	<
<	小于号	>
&	和号	&
¥	人民币	¥
©	版权	©
℃	摄氏度	℃
±	正负号	±
×	乘号	×
÷	除号	÷
2	平方2（上标2）	²
3	立方3（上标3）	³

1.1.13. 注释标签

如果需要在HTML文档中添加一些便于阅读和理解但又不需要显示在页面中的注释文字，就需要使用注释标签。

基本语法：

```
<!-- 注释语句 -->
```

注释的内容不会显示在浏览器，但是作为HTML文档内容的一部分，查看源代码时可以看到。

1.1.14. 路径

实际工作中，通常新建一个文件夹专门用于存放图像文件，这时再插入图像，就需要采用“路径”的方式来指定图像文件的位置

路径分为：

- 相对路径
 - 图像文件和HTML文件位于同一文件夹：只需输入图像文件的名称即可，如。

- 图像文件位于HTML文件的下一级文件夹：输入文件夹名和文件名，之间用“/”隔开，如- 图像文件位于HTML文件的上一级文件夹：在文件名之前加入“../”，如果是上两级，则需要使用“../ ../”，以此类推，如- 绝对路径
 - “D:\web\img\logo.gif”，或完整的网络地址

1.1.15. 列表标签

在HTML页面中让文字或图表展现出整齐有序的格式

列表分类：

- 无序列表 (ul)

无序列表ul(重点)基本语法如下：

```
<ul>
  <li>列表项一</li>
  <li>列表项二</li>
  <li>列表项三</li>
</ul>
```

``中只能嵌套``标签，直接在``标签中输入其他标签或文字的做法是不被允许的。
``与``之间相当于一个容器，可以容纳所有元素。

- 有序列表 (ol)

有序列表 (ol) 有序列表即为有排列顺序的列表，其各列表项按照一定的顺序排列定义，其基本语法如下：

```
<ol>
  <li>列表项一</li>
  <li>列表项二</li>
  <li>列表项三</li>
</ol>
```

- 自定义列表 (dl)

自定义列表 (dl) 自定义列表常用于对术语或名词解释，自定义列表的列表项前没有任何项目符号。其基本语法格式如下：

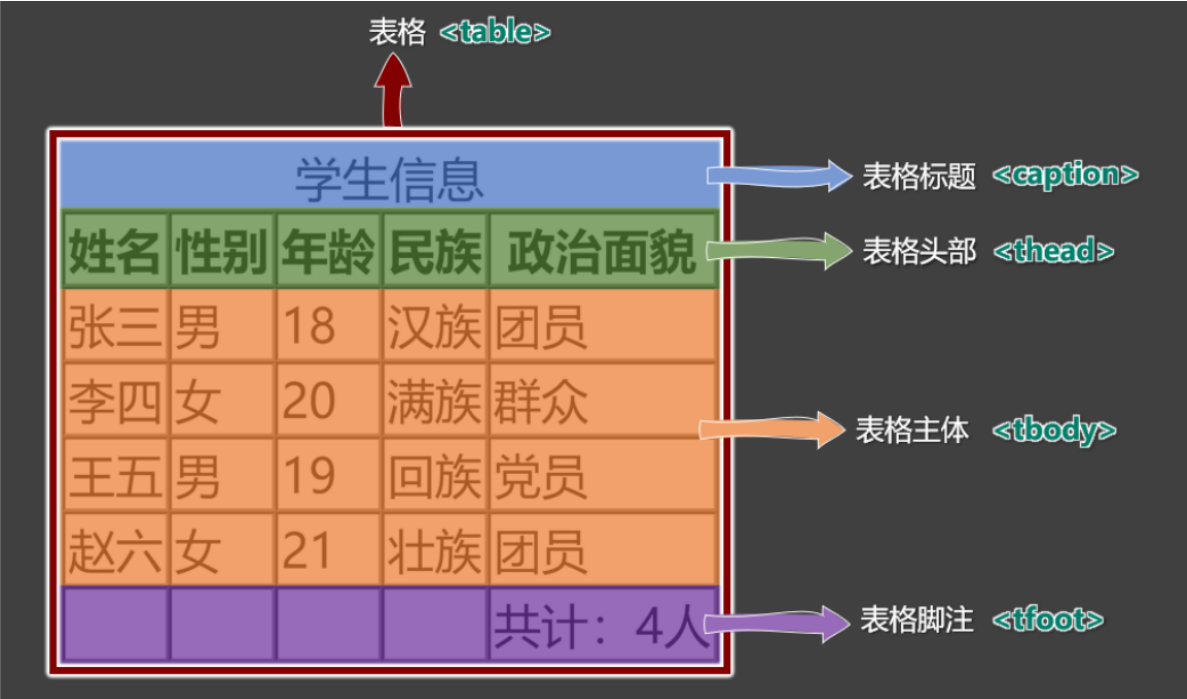
```
<dl>
  <dt>名词1</dt>
  <dd>名词1解释1</dd>
  <dd>名词1解释2</dd>
  <dd>名词1解释3</dd>
</dl>
```



1.1.16. 表格标签

1.1.16.1. 基本结构

1.一个完整的表格由：**表格标题**、**表格头部**、**表格主体**、**表格脚注**，四部分组成。



2.表格中涉及的标签

标签名	说明
table	表格
caption	表格标题
thead	表格头部
tbody	表格主体
tfoot	表格注脚
tr	每一行
th、td	每一个单元格)

(备注：表格头部中用 th，表格主体、表格脚注中用：td

示例代码：

```
<table border="1">
  <!-- 表格标题 -->
  <caption>学生信息</caption>
  <!-- 表格头部 -->
  <thead>
    <tr>
      <th>姓名</th>
      <th>性别</th>
      <th>年龄</th>
      <th>民族</th>
      <th>政治面貌</th>
    </tr>
  </thead>

  <!-- 表格主体 -->
  <tbody>
    <tr>
      <td>张三</td>
      <td>男</td>
      <td>18</td>
      <td>汉族</td>
      <td>团员</td>
    </tr>
    <tr>
      <td>李四</td>
      <td>女</td>
      <td>20</td>
      <td>满族</td>
      <td>群众</td>
    </tr>
    <tr>
      <td>王五</td>
      <td>男</td>
      <td>20</td>
      <td>回族</td>
      <td>党员</td>
    </tr>
    <tr>
      <td>赵六</td>
      <td>女</td>
      <td>21</td>
      <td>壮族</td>
      <td>团员</td>
    </tr>
  </tbody>

  <!-- 表格脚注 -->
  <tfoot>
    <tr>
      <td></td>
      <td></td>
      <td></td>
      <td></td>
      <td>共计：4人</td>
    </tr>
```

```
</tfoot>
</table>
```

1.1.16.2. 常用属性

属性名	含义	常用属性
border	设置表格的边框（默认border="0"无边框）	像素值
cellspacing	设置单元格与单元格边框之间的空白间距	像素值（默认为2像素）
cellpadding	设置单元格内容与单元格边框之间的空白间距	像素值（默认为2像素）
width	设置表格的宽度	像素值
height	设置表格的高度	像素值
align	设置表格在网页中的水平对齐方式	left、center、right

1.1.16.3. 合并单元格

跨行合并：rowspan 跨列合并：colspan

课堂练习1：课程表

18 华为大数据 8 班 课 程 表					
节次/星期	周一	周二	周三	周四	周五
第1-2节		平面设计 吴建 201A_504			
第3-4节	JavaWeb应用设计与实战 熊慧志 301B_303	JavaWeb应用设计与实战 熊慧志 301B_303	Hadoop集群设计与开发 冯奎 (301B_405)	JavaWeb应用设计与实战 熊慧志 301B_303	Python爬虫开发 严庄 301B404
第5-6节	Hadoop集群设计与开发 冯奎 (301B_405)	Python爬虫开发 严庄 301B404		Python爬虫开发 严庄 301B404	
第7-8节					
第9-10节					

个人简历

基本资料				
姓 名		性 别		
年 龄		电 话		
现居住地				
毕业院校		专 业		
求职意向				
主修课程				
专业技能				
技能证书				
自我评价				

1.1.17. 表单标签

在html中，form标签被用于定义表单域，即创建一个表单，以实现用户信息的收集和传递，form中的所有内容都会被提交给服务器。创建表单的语法如下：

```
<form action="" method="" name="" id="">
    各种表单控件
</form>
```

- 1、Action：在表单收集到信息后，需要将信息传递给服务器进行处理，action属性 用于指定接收并处理表单数据的服务器程序的url地址
- 2、method：用于设置表单数据的提交方式，其取值为get或post
- 3、name：用于指定表单的名称，以区分同一个页面中的多个表单
- 4、id：用于指定表单的编号

在HTML中，一个完整的表单通常由表单控件（也称为表单元素）、提示信息和表单域3个部分构成。

表单控件：

包含了具体的表单功能项，如单行文本输入框、密码输入框、复选框、提交按钮、重置按钮等。

提示信息：

一个表单中通常还需要包含一些说明性的文字，提示用户进行填写和操作。

表单域：

他相当于一个容器，用来容纳所有的表单控件和提示信息，可以通过他定义处理表单数据所用程序的url地址，以及数据提交到服务器的方法。如果不定义表单域，表单中的数据就无法传送到后台服务器。

1.1.17.1. input控件

<input />标签为单标签，type属性为其最基本的属性，其取值有多种，用于指定不同的控件类型。除了type属性之外，<input />标签还可以定义很多其他的属性，其常用属性如下表所示。

属性	属性值	描述
type	text	单行文本输入框
	password	密码输入框
	radio	单选按钮
	checkbox	复选框
	button	普通按钮
	submit	提交按钮
	reset	重置按钮
	image	图像形式的提交按钮
	file	文件域
name	由用户自定义	控件的名称
value	由用户自定义	input控件中的默认文本值
size	正整数	input控件在页面中的显示宽度
checked	checked	定义选择控件默认被选中的项
maxlength	正整数	控件允许输入的最多字符数

1.1.17.2. label标签（理解）

label 标签为 input 元素定义标注（标签）。

作用： 用于绑定一个表单元素, 当点击label标签的时候, 被绑定的表单元素就会获得输入焦点

如何绑定元素呢？

for 属性规定 label 与哪个表单元素绑定。

示例代码：

```
<label for="male">Male</label>
<input type="radio" name="sex" id="male" value="male">
```

1.1.17.3. textarea控件(文本域)

如果需要输入大量的信息，就需要用到<textarea></textarea>标签。通过textarea控件可以轻松地创建多行文本输入框，其基本语法格式如下：

```
<textarea cols="每行中的字符数" rows="显示的行数">
  文本内容
</textarea>
```

文本域 textarea

登录

1.1.17.4. 下拉菜单

使用select控件定义下拉菜单的基本语法格式如下

```
<select>
  <option>选项1</option>
  <option>选项2</option>
  <option>选项3</option>
  ...
</select>
```

1.1.17.5. 按钮标签

button 按钮

disabled 属性：设置按钮不可用。

type 属性：设置按钮的类型，值：submit（默认）、reset、button

语法格式：

```
<button>按钮标签</button>
```

1.1.17.6. fieldset 与 legend的使用（了解）

fieldset 可以为表单控件分组、legend 标签是分组的标题。

示例：

```
<fieldset>
  <legend>主要信息</legend>
  <label for="zhanghu">账户： </label>
  <input id="zhanghu" type="text" name="account" maxlength="10"><br>
  <label>
    密码：
    <input id="mima" type="password" name="pwd" maxlength="6">
  </label>
  <br>
  性别：
  <input type="radio" name="gender" value="male" id="nan">
  <label for="nan">男</label>
  <label>
    <input type="radio" name="gender" value="female" id="nv">女
  </label>
</fieldset>
```

三、HTML5新增特性

1. 新增语义化标签

1.1. 新增布局标签

标签名	语义	单/双标签
header	整个页面，或部分区域的头部	双
footer	整个页面，或部分区域的底部	双
nav	导航	双
article	文章、帖子、杂志、新闻、博客、评论等。	双
section	页面中的某段文字，或文章中的某段文字（里面文字通常里面会包含标题）。	双
aside	侧边栏	双
main	文档的主要内容 (WHATWG 没有语义， IE 不支持)，几乎不用。	双

1.2. progress 标签

语义：显示某个任务完成的进度的指示器，一般用于表示进度条，双标签，例如：工作完成进度等。

常用属性如下：

属性	值	描述
max	数值	规定目标值。
value	述职	规定当前值。

1.3. 新增列表标签

标签名	语义	单/双标签
datalist	用于搜索框的关键字提示	双
details	用于展示问题和答案，或对专有名词进行解释	双
summary	写在 details 的里面，用于指定问题或专有名词	双

示例代码：

```
<input type="text" list="mydata">
  <datalist id="mydata">
    <option value="周冬雨">周冬雨</option>
    <option value="周杰伦">周杰伦</option>
    <option value="温兆伦">温兆伦</option>
    <option value="马冬梅">马冬梅</option>
  </datalist>
```

1.4. 新增文本标签

1.4.0.7. 文本注音

标签名	语义	单**/双标签**
ruby	包裹需要注音的文字	双
rt	写注音， rt 标签写在 ruby 的里面	双

示例代码：

```
<ruby>
  <span>魑魅魍魉</span>
  <rt>chī mèi wǎng liǎng </rt>
</ruby>
```

1.4.0.8. 文本标记

标签名	语义	单**/双标签**
mark	标记	双

注意： W3C 建议 mark 用于标记搜索结果中的关键字。

1.5. 新增表单功能

1.5.1. 表单控件新增属性

属性名	功能
placeholder	提示文字（注意：不是默认值， value 是默认值），适用于 文字输入类 的表单控件。
required	表示该输入项必填，适用于 除按钮外 其他表单控件。
autofocus	自动获取焦点，适用于所有表单控件。
autocomplete	自动完成，可以设置为 on 或 off，适用于 文字输入类 的表单控件。
pattern	填写正则表达式，适用于文本输入类表单控件。

示例代码：

用户名应该只包含小写字母;没有大写字母、 数字或其他特殊字符，并且长度不超过15个字符

```
<form action="#">
  <input type="text" name="username" placeholder="请输入用户名" pattern="[a-z]{1,15}">
</form>
```

1.5.2. input新增属性值

属性名	功能
email	邮箱 类型的输入框，表单提交时会验证格式，输入为空则不验证格式。
url	url 类型的输入框，表单提交时会验证格式，输入为空则不验证格式。
number	数字 类型的输入框，表单提交时会验证格式，输入为空则不验证格式。
search	搜索 类型的输入框，表单提交时不会验证格式。
tel	电话 类型的输入框，表单提交时不会验证格式，在移动端使用时，会唤起数字键盘
range	范围 选择框，默认值为 50，表单提交时不会验证格式。
color	颜色 选择框，默认值为黑色，表单提交时不会验证格式。
date	日期 选择框，默认值为空，表单提交时不会验证格式。
month	月份 选择框，默认值为空，表单提交时不会验证格式。
week	周 选择框，默认值为空，表单提交时不会验证格式。
time	时间 选择框，默认值为空，表单提交时不会验证格式。
datetime-local	日期+时间 选择框，默认值为空，表单提交时不会验证格式。

1.5.3. form 标签新增属性

属性名	功能
novalidate	如果给 form 标签设置了该属性，表单提交的时候不再进行验证。

1.6. 新增多媒体标签

1.6.1. 视频标签

```
<video> 标签用来定义视频，它是双标签。
语法格式：
  <video>
    <source src="视频.mp4"/>
  </video>
```

属性	值	描述
src	URL地址	视频地址
width	像素值	设置视频播放器的宽度
height	像素值	设置视频播放器的高度
controls		向用户显示视频控件（比如播放/暂停按钮）
muted		视频静音
autoplay		视频自动播放
loop		循环播放
poster	URL地址	视频封面
preload	auto / metadata / none	none：不预加载视频、metadata：仅预先获取视频的元数据、 auto：可以下载整个视频文件，即使用户不希

示例代码：

```
<video controls width="400px" height="400px">  
  <source src="视频.mp4"/>  
</video>
```

1.6.2. 音频标签

```
<audio> 标签用来定义音频，它是双标签。  
<!--语法格式-->
```

属性	值	描述
src	URL地址	视频地址
controls		向用户显示视频控件（比如播放/暂停按钮）
muted		音频静音
autoplay		音频自动播放
loop		循环播放
preload	auto / metadata / none	none：不预加载视频、metadata：仅预先获取视频的元数据、 auto：可以下载整个视频文件，即使用户不希

示例代码：



```
<audio controls>
  <source src="音频.mp3"/>
</audio>
```

2. HTML5兼容性处理

- 添加元信息，让浏览器处于最优渲染模式。



```
<!--设置IE总是使用最新的文档模式进行渲染-->
<meta http-equiv="X-UA-Compatible" content="IE=Edge">
<!--优先使用 webkit ( Chromium ) 内核进行渲染，针对360等壳浏览器-->
<meta name="renderer" content="webkit">
```

- 使用 html5shiv 让低版本浏览器认识 H5 的语义化标签。



```
<!--[if lt ie 9]>
<script src="../sources/js/html5shiv.js"></script>
<![endif]-->
```

- 拓展

lt 小于

lte 小于等于

gt 大于

gte 大于等于

! 逻辑非

- 示例



```
<!--[if IE 8]>仅IE8可见<![endif]-->
<!--[if gt IE 8]>仅IE8以上可见<![endif]-->
<!--[if lt IE 8]>仅IE8以下可见<![endif]-->
<!--[if gte IE 8]>IE8及以上可见<![endif]-->
<!--[if lte IE 8]>IE8及以下可见<![endif]-->
<!--[if !IE 8]>非IE8的IE可见<![endif]-->
```