



# .NET架构

主讲教师 张智  
计算机学院

## 第3章 HTML5语言

### 3.1 [HTML概述](#)

### 3.2 [基本标签](#)

### 3.3 [HTML表格](#)

### 3.4 [Form表单](#)

【附录1】[文件上传问题](#)

【附录2】[HTML5 的 File API](#)

## 3.1 HTML概述



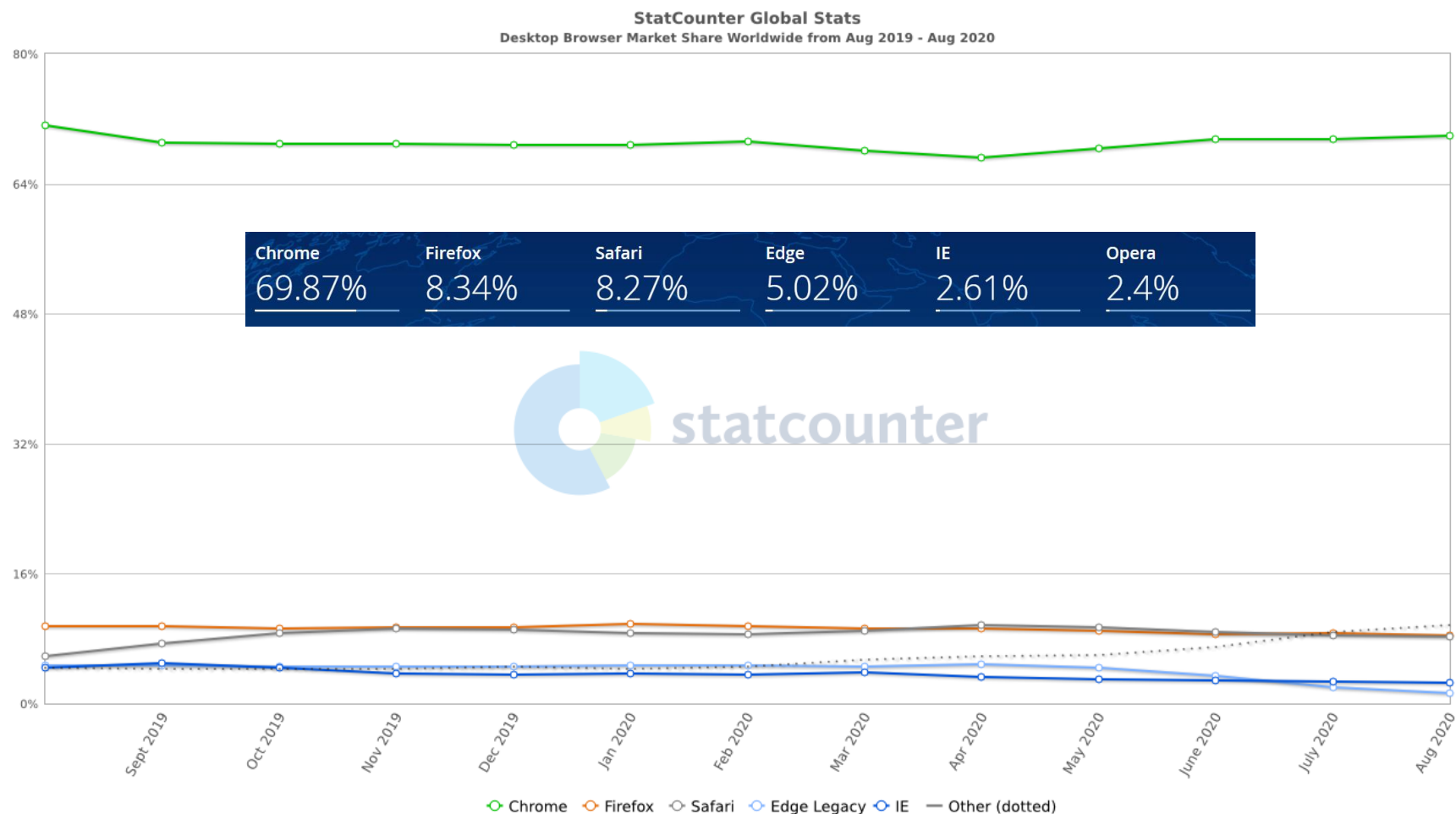
- HTML：是一种用于创建网页的超文本标记语言(Hyper Text Markup Language)

“超文本”就是指页面内可以包含图片、链接，甚至音乐、视频、程序等非文字元素

- HTML语言是Web的基础，可以说Internet是建立在HTML基础之上的。
- HTML5是新一代HTML标准，大部分浏览器已经支持。

## 2020.8月桌面浏览器全球市场份额

数据来源: <http://gs.statcounter.com/>



## HTML5基本结构

HTML文件后缀名：.html 或 .htm

`<!DOCTYPE html>`

html5网页声明

`<html lang="en">`

`<html>` 元素是 HTML 页面根元素

lang属性：声明当前页面的语言类型，"en"英文，"zh"中文

`<head>`

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

`<meta>` 设置网页编码格式为 utf-8

`<title> 文档标题 </title>`

`<title>` 设置文档标题

`</head>`

`<body>`

`<body>` 元素包含了可见的页面内容

...页面内容(各种标签)

`</body>`

`</html>`

## HTML5示例

```
<center>
```

```
<div id="top">
```

```
<h1>千古绝句</h1>
```

```
<hr>
```

```
<h2>《赤壁怀古》</h2>
```

```
<a href="https://www.baidu.com/s?ie=UTF-8&wd=苏轼" target="_blank">苏轼</a>
```

```
<br>
```

```

```

```
<p>大江东去，浪淘尽</p>
```

```
<p>千古东流人物</p>
```

```
<p>故垒西边，人道是</p>
```

```
<p>三国周郎赤壁</p>
```

```
<p>.....</p>
```

```
<hr>
```

```
<p>友情链接: <a href="#">李清照</a> | <a href="#">欧阳修</a> | <a href="#">辛弃疾</a> </p>
```

```
<a href="#top">返回顶部</a>
```

```
</div>
```

```
</center>
```

使用 div 标签对文档结构进行逻辑上的分块

body部分

标题标签

hr 水平线标签

a 超链接标签

img 图片标签

p 段落标签

a 超链接标签

千古绝句

《赤壁怀古》

苏轼



大江东去，浪淘尽

千古东流人物

故垒西边，人道是

三国周郎赤壁

友情链接: [李清照](#) | [欧阳修](#) | [辛弃疾](#)[返回顶部](#)

【返回】

## 3.2 基本标签

- [div和span](#)
- [文本相关](#)
- [图片](#)
- [超链接](#)
- [列表](#)
- [音频和视频](#)
- [综合示例](#)

**【[返回](#)】**

## 1. div和span

**<div> 标签：**块级元素，可用于组合其他 HTML 元素的容器

- div 没有特定的含义，通常用于对文档结构进行逻辑上的分块
- div 的前后会换行

**<span> 标签：**行内元素（内联元素），可用作文本的容器

- span 没有特定的含义
- span 的前后不会换行



## div示例

```
<div id="sports">
```

体育频道

```
<a href="football.html">足球</a>
```

```
<a href="basketball.html">篮球</a>
```

```
</div>
```

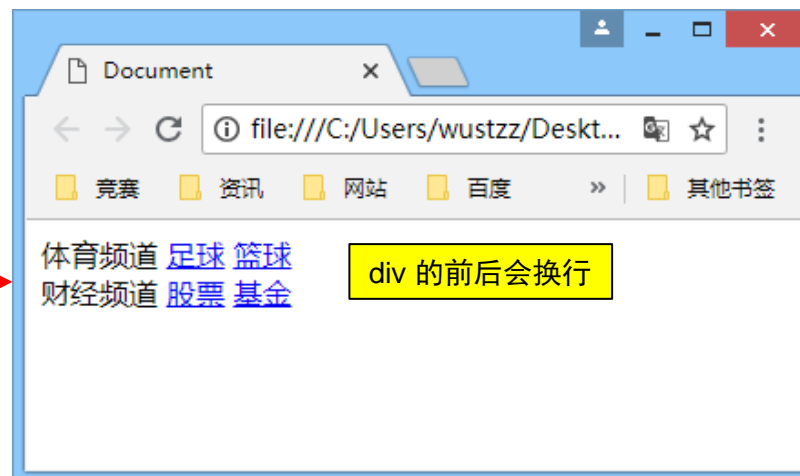
```
<div class="finance">
```

财经频道

```
<a href="stock.html">股票</a>
```

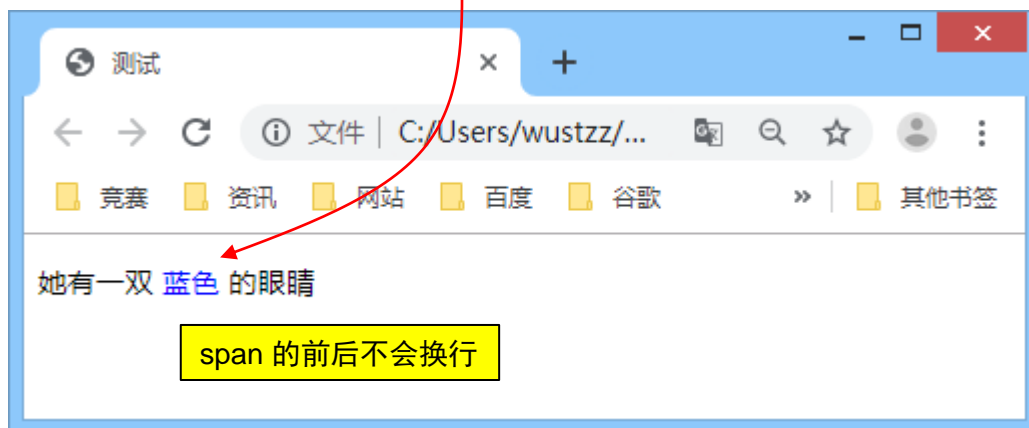
```
<a href="fund.html">基金</a>
```

```
</div>
```



## span示例

<p>她有一双 <span style="color:blue">蓝色</span> 的眼睛</p>



## 2. 文本相关

align属性不推荐使用, 用 style="text-align:center;" 样式代替

标题标签: `<h1>` - `<h6>`

`<h1 align="left/center/right">...</h1>`

`<h1 align="center">欢迎光临我的主页</h1>`

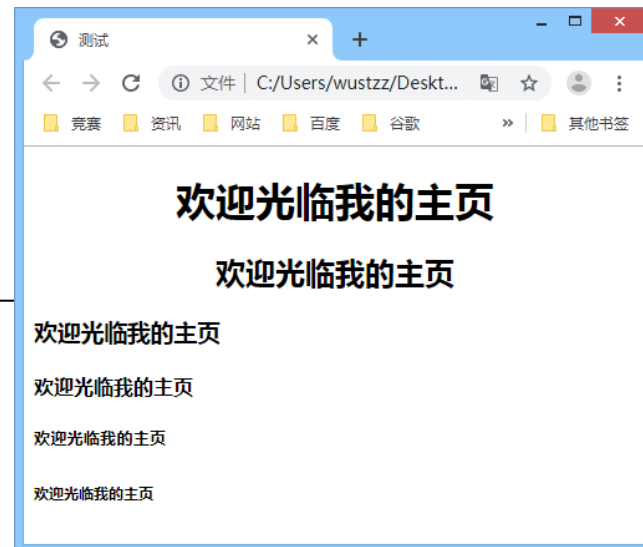
`<h2 style="text-align:center;">欢迎光临我的主页</h2>`

`<h3>欢迎光临我的主页</h3>`

`<h4>欢迎光临我的主页</h4>`

`<h5>欢迎光临我的主页</h5>`

`<h6>欢迎光临我的主页</h6>`



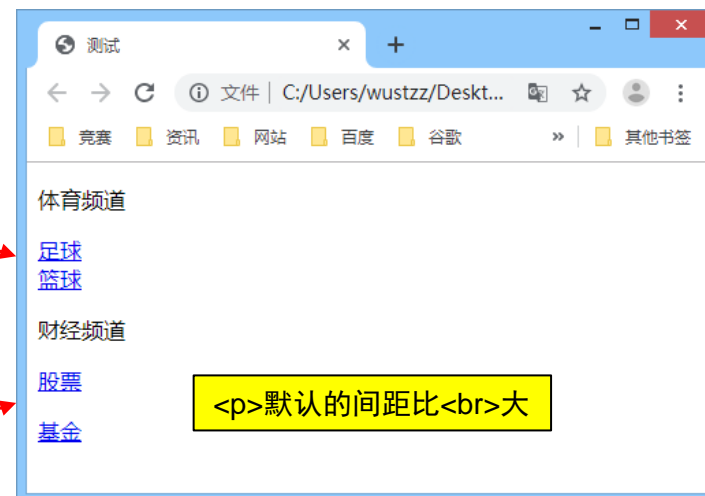
特别提醒：不要在<p>里嵌套块级元素，如嵌套p或div



段落标签：<p align="center/left/right"> 一段文字 </p>

换行标签：<br> 或者 <br/>

```
<div id="sports">
    <p>体育频道</p>
    <a href="football.html">足球</a>
    <br>
    <a href="basketball.html">篮球</a>
</div>
<div class="finance">
    <p>财经频道</p>
    <p><a href="stock.html">股票</a></p>
    <p><a href="fund.html">基金</a></p>
</div>
```



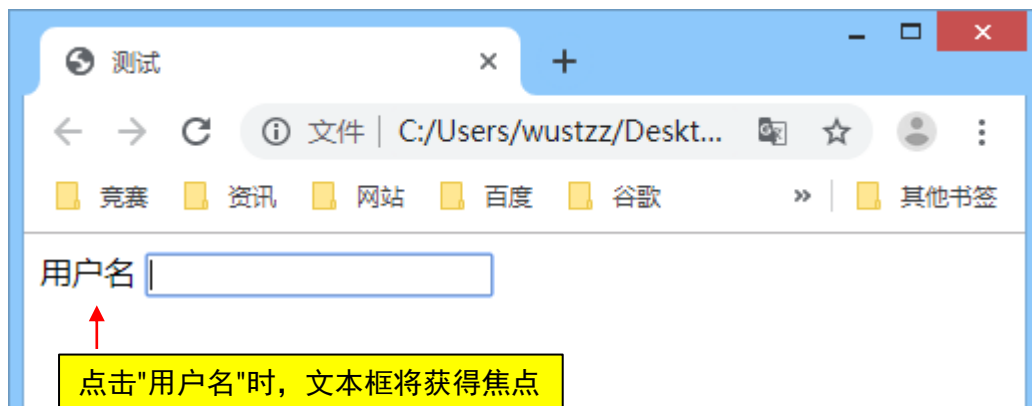
for属性：绑定一个控件id值，当用户点击label标签内文本，就会将焦点切换到绑定的控件上。

<label> 标签：

<label for="..."> 标签文字 </label>

<label for="username">用户名</label>

<input type="text" id="username" />

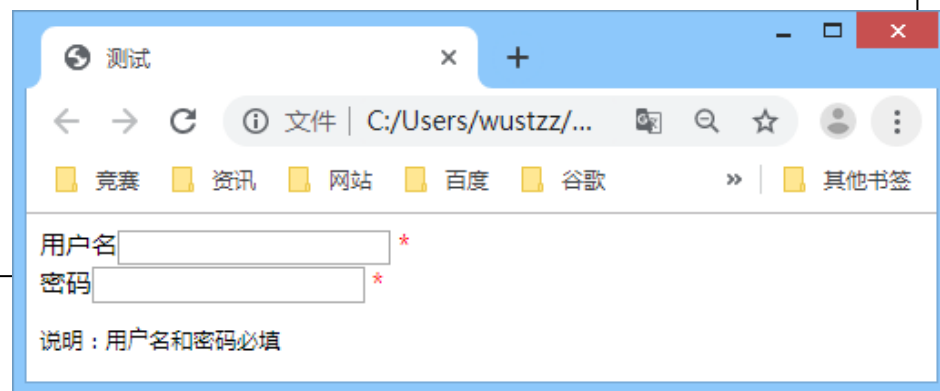


嵌套写法：嵌套时可省略 for 属性

<label> 用户名<input type="text" /> </label>

示例：

```
<div id="login">
    <label>用户名<input type="text" name="username"/></label>
    <span style="color:red"> * </span>
    <br/>
    <label>密码<input type="password" name="psd"/></label>
    <span style="color:red">*</span>
</div>
<div id="tip" style="font-size:12px;">
    <p>说明：用户名和密码必填</p>
</div>
```



## 格式标签：部分

粗体：<b>      斜体：<i>      下划线：<u>      删除线：<strike>

下标：<sub>    上标：<sup>

小号字：<small>      大号字：<big>

强调(加粗)：<strong>    强调(斜体)：<em>

突出显示(黄色背景)：<mark>

<p><b><i> <u>欢迎光临</u> </i></b></p>

<p>X<sup>2</sup></p>

<p><big>big文字</big></p>

<p><small>small文字</small></p>

<strong>强调的内容(粗体,语气更强)</strong> <br>

<em>强调的内容(斜体)</em> <br>

<mark>突出显示</mark>

欢迎光临

x<sup>2</sup>

big文字

small文字

**强调的内容(粗体,语气更强)**

*强调的内容(斜体)*

**突出显示**

## 特殊符号(实体标签): 部分

显示结果	特殊符号
空格	&nbsp;
<	&lt;
>	&gt;
&	&amp;
£	&pound;
¥	&yen;
©	&copy;
®	&reg;
™	&trade;
×	&times;

**HTML注释:** <!-- 注释文字 -->



**【[返回](#)】**



### 3. 图片

图片标签:

```

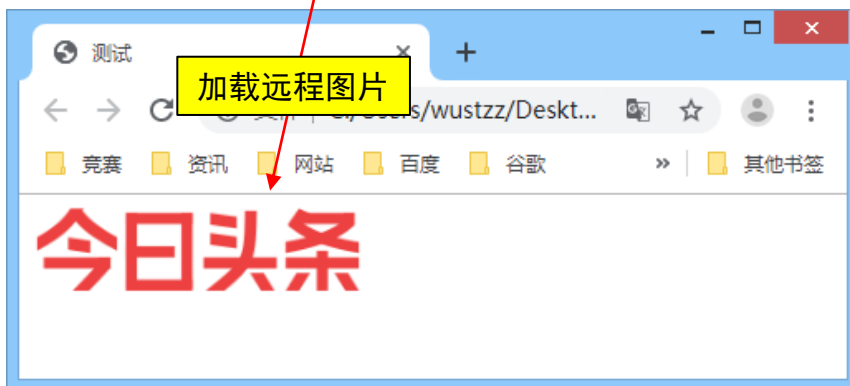
```

单位是像素点

无法载入图像时显示alt信息

```

```



[【返回】](#)

## 4. 超链接

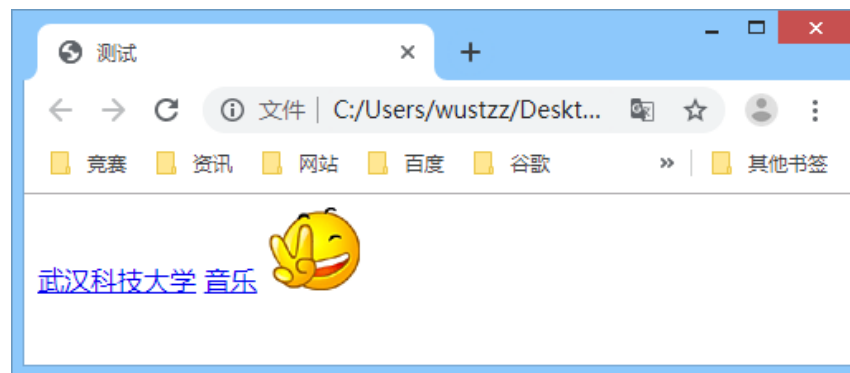
超链接标签: `<a href="url"> 链接文字 </a>`

新窗口打开页面

`<a href="http://www.wust.edu.cn" target="_blank">武汉科技大学</a>`

`<a href="music/she.mp3">音乐</a>`

`<a href="http://www.wust.edu.cn">  </a>` ← 图片超链接



## 超链接锚点

跳转到当前页的某一个位置(锚点, 一般用id值):

`<a href="#id值">链接文字</a>`

`<a href="#">页面的最上部</a>`

例如:

`<a href="#test">点击跳到test</a>`

注意观察一下地址栏

...

...

`<div id="test">`

...

`</div>`

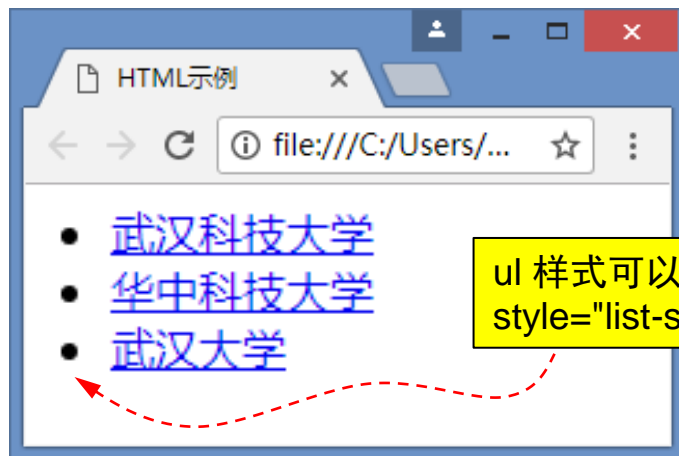
...

`<a href="#">回到顶部TOP</a>`

[【返回】](#)

## 5. 列表 -- 无序列表ul

```
<ul>
  <li><a href="http://www.wust.edu.cn">武汉科技大学</a></li>
  <li><a href="http://www.hust.edu.cn">华中科技大学</a></li>
  <li><a href="http://www.whu.edu.cn">武汉大学</a></li>
</ul>
```



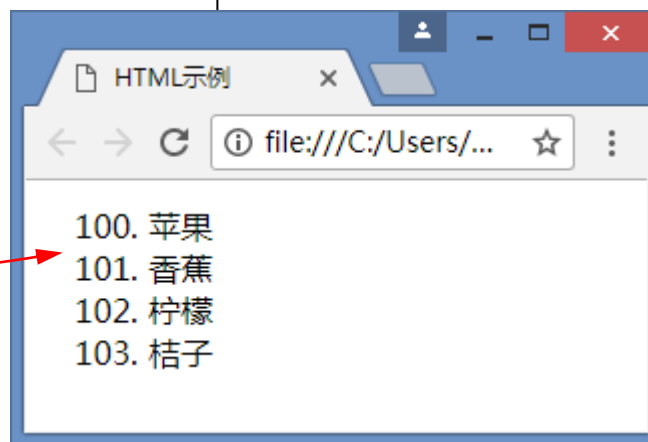
### list-style-type部分取值:

none: 无标记  
disc: 实心圆(默认)  
circle: 空心圆  
square: 实心方块  
lower-roman: 小写罗马数字  
...

## 列表 -- 有序列表ol

默认序号从1开始

```
<ol start="100">  
  <li>苹果</li>  
  <li>香蕉</li>  
  <li>柠檬</li>  
  <li>桔子</li>  
</ol>
```



[【返回】](#)

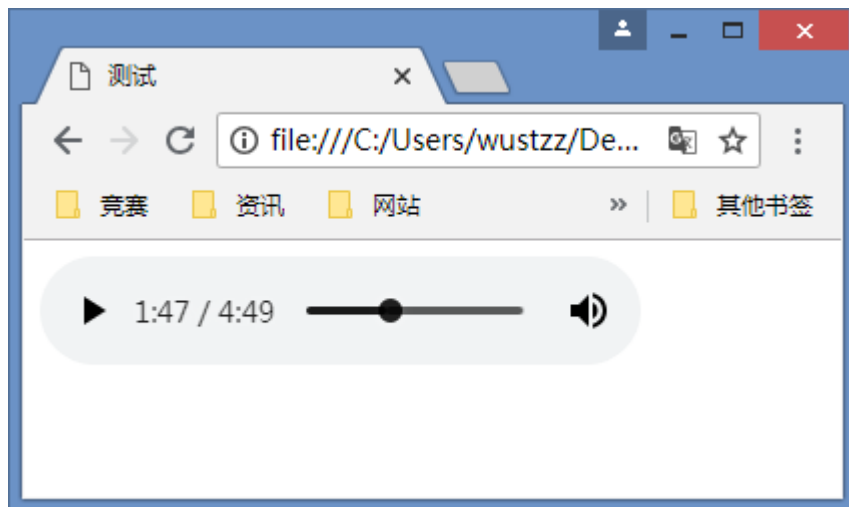
## 6. 音频：mp3、wav、ogg格式

```
<audio src="music/不再犹豫.mp3" controls="controls">
```

你的浏览器不支持audio元素

```
</audio>
```

自动播放属性：autoplay="autoplay"  
循环播放属性：loop="loop"

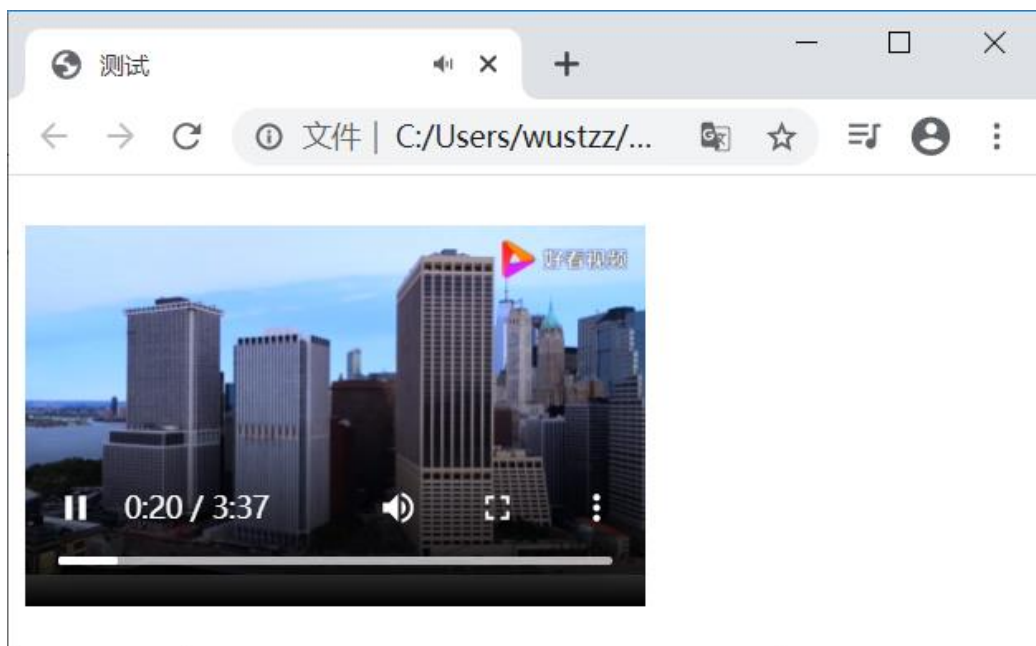


视频：mp4、webm、ogg格式

```
<video src="video/纽约航拍.mp4" width="300" height="200" controls="controls">
```

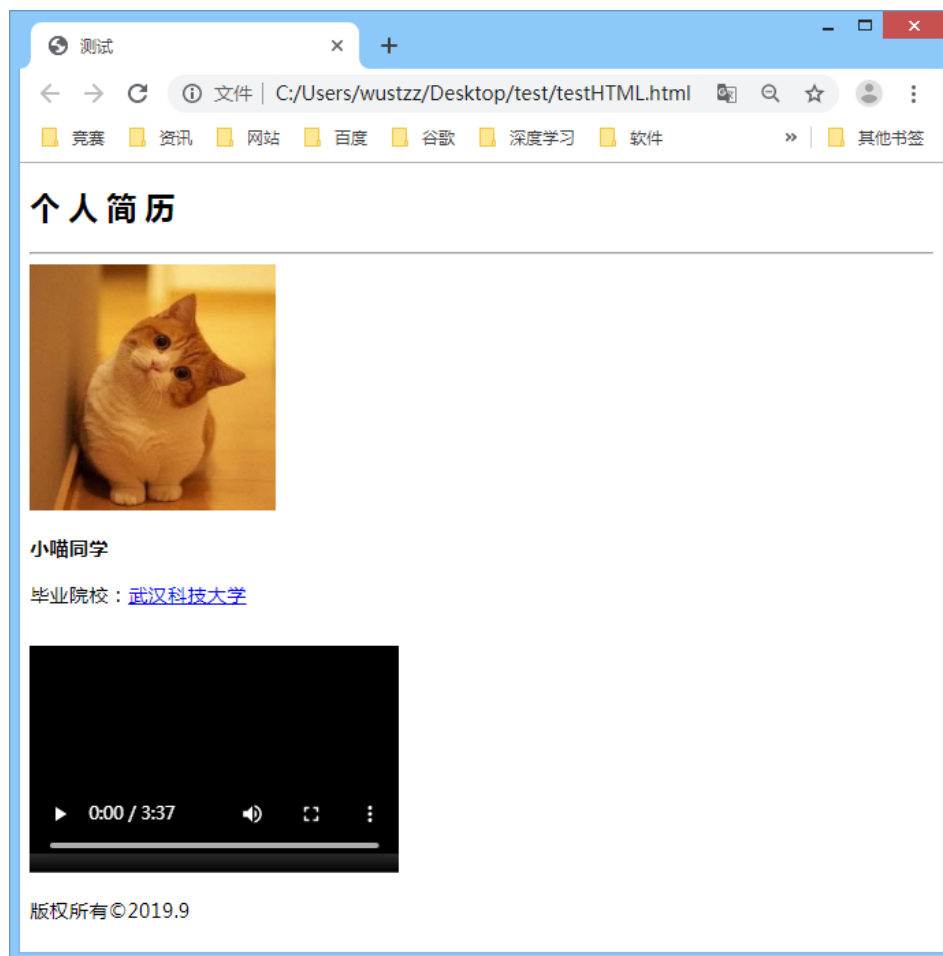
你的浏览器不支持video元素

```
</video>
```



【[返回](#)】

## HTML综合示例





body部分

**<div class="header">**`<h2>个人简历</h2>``<hr>````<p><strong>小喵同学</strong></p>` ← strong标签用于强调`<p>毕业院校: <a href="http://www.wust.edu.cn">武汉大学</a></p>`**</div>****<div class="content">**`<video src="video/纽约航拍.mp4" width="300" height="200" controls="controls">`

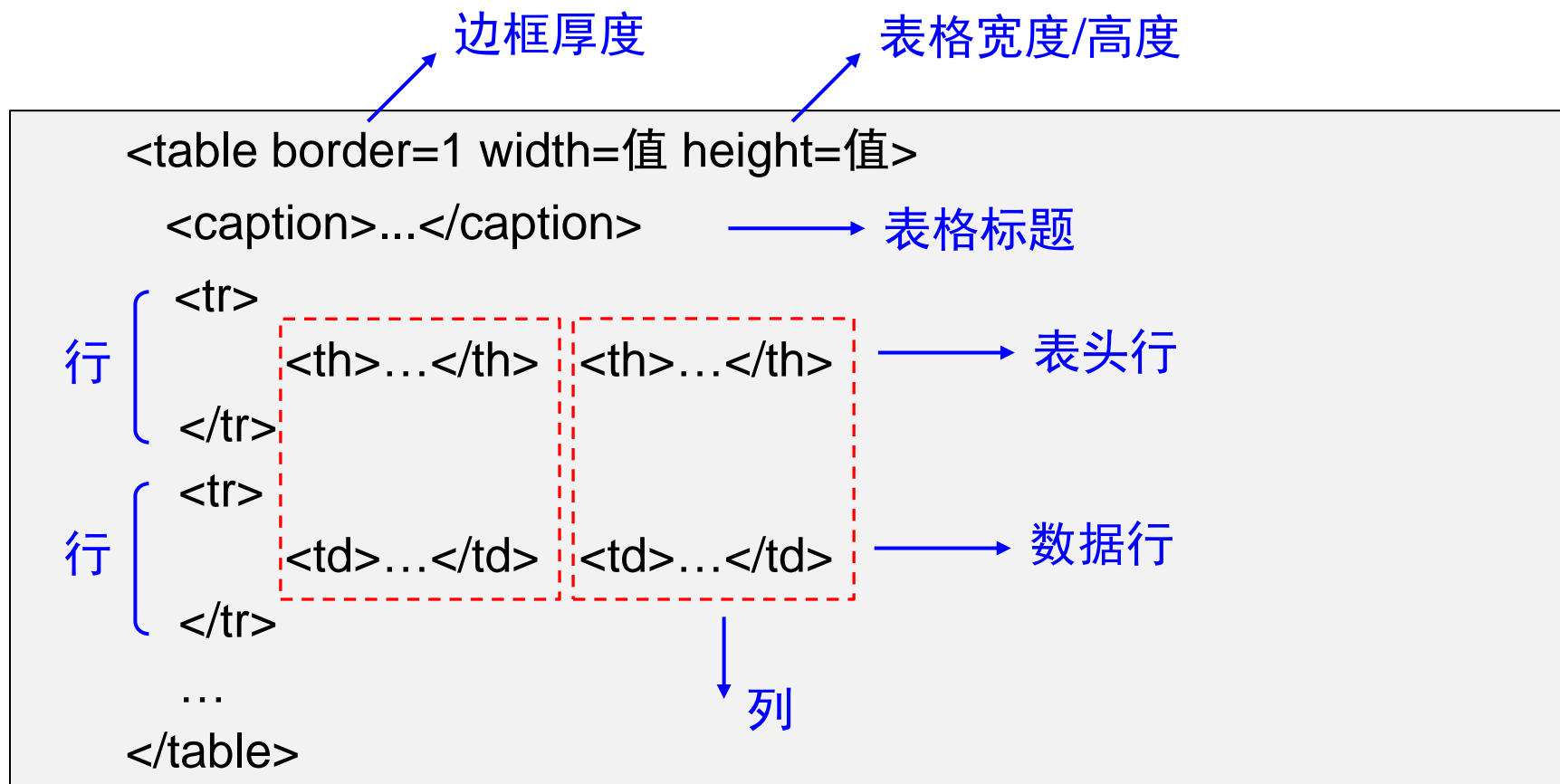
你的浏览器不支持video元素

`</video>`**</div>****<div class="footer">**`<p>版权所有&copy;2020.9</p>`**</div>**

特殊字符

**【返回】**

### 3.3 HTML表格



## 表格示例

测试

文件 | C:/Users/wustzz/Desktop/test/testHTML....

竞赛 资讯 网站 标题( `<caption>` ) 软件 其他书签

边框(border)

3行 `<tr>`

学生名单

表头行( 3个 `<th>` )

学号	姓名	院系
2017001	张无忌	计算机学院
2017002	令狐冲	计算机学院

数据行 ( 每行3个 `<td>` )

## 参考代码

试一下百分比，如：50%

```
<table border=1 width=600>
  <caption>学生名单</caption>
  <tr>
    <th>学号</th>    <th>姓名</th>    <th>院系</th>
  </tr>
  <tr>
    <td>2017001</td> <td>张无忌</td> <td>计算机学院</td>
  </tr>
  <tr>
    <td>2017002</td> <td>令狐冲</td> <td>计算机学院</td>
  </tr>
</table>
```

## tr、th、td 的对齐属性

**align:** 水平对齐, 取值 left | center | right

**valign:** 垂直对齐, 取值 top | middle | bottom

或使用样式 `style="text-align:center;"` 或 `"vertical-align:middle"`

```
<table border=1>
  <caption style="caption-side:bottom">学生名单</caption>
  ...
  <tr align="center">
    <td>2004001</td>  <td>张无忌</td>  <td>计算机学院</td>
  </tr>
  ...
</table>
```

将caption放到表格下方

此行 td 文字全部居中

colspan和rowspan属性:

设置当前单元格(th/td)横跨的列数或行数

```
<table border=1 width=600>  
  <caption>学生名单</caption>
```

```
<tr>
```

```
  <th>学号</th> <th>姓名</th> <th>院系</th>
```

```
  <th colspan="2">操作</th>
```

```
</tr>
```

```
<tr>
```

```
  <td>2017001</td> <td>张无忌</td> <td>计算机学院</td>
```

```
  <td>编辑</td> <td>删除</td>
```

```
</tr>
```

```
  ...
```

```
</table>
```

设置当前单元格横跨2列

学生名单

学号	姓名	院系	操作	
2017001	张无忌	计算机学院	编辑	删除
2017002	令狐冲	计算机学院	编辑	删除

## 修改边框样式

学生名单

学号	姓名	院系	操作	
2017001	张无忌	计算机学院	编辑	删除
2017002	令狐冲	计算机学院	编辑	删除

```
<style>
```

```
  table {
```

```
    border-collapse: collapse; → 边框合并为单一的边框
```

```
  }
```

```
  th, td {
```

```
    border: 1px solid blue; → 设置边框宽度、边框式样和颜色
```

```
  }
```

```
</style>
```

边框式样的一些取值：

none：无边框    solid：实线

dashed：虚线    dotted：点状

## 课堂练习

专业设置表及在校人数表

录名	专业人数			
计算机学院	计算机科学技术专业			
	16级	17级	18级	19级
	150人	150人	200人	220人
	软件工程专业			
	16级	17级	18级	19级
	80人	80人	90人	90人
外语学院	英语专业			
	16级	17级	18级	19级
	60人	80人	90人	90人



# 课后作业 (不添加样式)

个人简历

file:///C:/Users/wustzz/Desktop/test/个人简历.html

竞赛 资讯 网站 百度 谷歌 深度学习 知识库 软件 WebRTC 课程相关 其他书签

## 个人简历



小猫同学

求职意向：软件开发

基本信息	姓名	小橘子	出生年月	1999.9
	年龄	19	政治面貌	团员
	学历	本科	籍贯	湖北武汉
	电话	18908643860	邮箱	<a href="mailto:zz@qq.com">zz@qq.com</a>
	毕业院校	武汉科技大学计算机学院		

专业技能	<ul style="list-style-type: none"> <li>语言类：CET4，6</li> <li>软件类：               <ol style="list-style-type: none"> <li>前端开发(html5/css3/js等)</li> <li>移动开发(Android方向)</li> <li>Web开发(asp.net/j2ee等)</li> </ol> </li> <li>荣誉类：               <ol style="list-style-type: none"> <li>特等奖学金</li> <li>校级优秀学生</li> <li>拔尖人才</li> </ol> </li> </ul>

个人风采		

版权所有 2018.9.9

【返回】

## 3.4 Form表单

- 表单概念
- Form常用元素

## 1. 表单概念

- Form：为用户输入创建**表单**。
- 其中可包含文本框、密码框、单/复选框、列表框、按钮等。
- 表单通常用于**向指定的 URL 提交用户数据**。

Form示例 - Windows Internet Explorer

姓名: wustzz

密码: .....

性别: ☒男 ☐女

爱好: ☒足球 ☒篮球 ☐网球

专业: 软件工程

特长: .NET编程, J2EE编程, C语言编程

附件: 浏览...

简介: 哈哈

提交 重填

提交数据 处理新页面

Form示例 - Windows Internet Explorer

姓名: wustzz

密码: 123456

性别: 男

爱好: 足球 篮球

专业: 软件工程

特长: .NET编程 J2EE编程

附件:

简介: 哈哈

## Form基本结构

```
<html>
<body>
  <form action="URL" method="get或post">
    表单元素
    ...
    <input type="submit" value="提交">
    <input type="reset" value="重填">
  </form>
</body>
</html>
```

提交后跳转的页面

传送数据的方法

备注：form 元素是块级元素，其前后会产生换行

## Form属性

属性名称	取值	说明
id	表单id串	定义表单的唯一id
name	表单名称	定义表单的唯一名称
action	URL	当提交按钮被按下时，接受和处理表单数据的网页URL
method	get/post	用于向 action URL 传送数据的 HTTP 方法 <ul style="list-style-type: none"><li>• get方法(默认) — 数据量少，会在地址栏显示数据</li><li>• post方法 — 数据量多</li></ul>
target	_blank _self _parent _top	在一个新的未命名的窗口载入文档 在相同的框架或窗口中载入目标文档 把文档载入父窗口或包含了超链接引用的框架的框架集 把文档载入包含该超链接的窗口,取代任何当前正在窗口中显示的框架

## Form示例

跳转到数据处理页面

```
<form action="url页面" method="get">
```

用户名: <input type="text" name="username"> <br/>

密 码: <input type="password" name="psd" maxlength=6>

<br/>

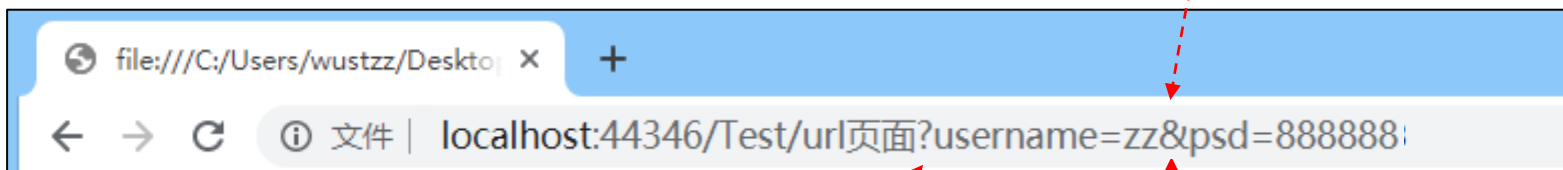
注意: 传值要用name属性  
没有name属性的<input>的数据不会被提交

<input type="submit" value="提交">

<input type="reset" value="重填">

get方法提交数据会在地址栏中显示

```
</form>
```



get方法: 跳转页面后用问号"?"连接请求串

多个值(name)之间用单个"&"连接

[【返回】](#)

## 2. Form常用元素

- 文本框、密码框    隐藏域
- 单选框    复选框
- 列表框
- 文件域
- 文本区域
- 提交和重置按钮    一般按钮
- Button按钮
- 日期域
- 区间域

【[返回](#)】

## 1. 文本框和密码框

用户名:	<input type="text" value="wustzz"/>
密码:	<input type="password" value="长度8个字符"/>

<form>

用户名:

设置初始值

悬浮提示

获得焦点  
autofocus="autofocus"简写

<input type="text" name="username" value="wustzz" title="提示信息" autofocus>

<br/>

密码:

输入的最大长度

占位提示(输入时消失)

<input type="password" name="psd" maxlength="8" placeholder="长度8个字符">

</form>

其他属性:

1. 只读属性: readonly="readonly"
2. 禁用属性: disabled="disabled"
3. 自动填充属性: autocomplete="on" (配合提交按钮)

【[返回](#)】



## 2. 隐藏域

隐藏域在页面中对于用户是不可见的。目的：收集信息以利于表单处理程序使用。提交数据时，有name值的隐藏域其value值会一并提交。

```
<form action="1.html" >  
  <input type="text" name="username" >  
  <input type="hidden" name="test" value="someValue">  
  <input type="submit">  
</form>
```

备注：可以通过js设置/读取value值

## 隐藏域示例：设置隐藏域值并提交

```
<form action="1.html" onsubmit="return foo()">
```

```
  <input type="text" id="username">
```

```
  <input type="hidden" id="h" name="h">
```

```
  <input type="submit">
```

```
</form>
```

```
<script>
```

JavaScript代码

```
  function foo() {
```

```
    var u = document.getElementById("username");
```

```
    var h = document.getElementById("h");
```

```
    h.value = u.value + "666"; //简单处理一下
```

```
    return true;
```

```
  }
```

```
</script>
```

注意：本例只有隐藏域设置了name属性

【[返回](#)】

### 3. 单选框

性别: ☒ 男 ☐ 女

```
<form>
```

性别:

```
<label>
```

```
  <input type="radio" name="sex" value="male" checked="checked"> 男
```

```
</label>
```

```
<label>
```

```
  <input type="radio" name="sex" value="female"> 女
```

```
</label>
```

```
</form>
```

套一个label的好处：点击选项文字也可以选中

设置为选中状态

name值相同才能互斥

提交的实际值

[【返回】](#)

## 4. 复选框

爱好: ☒ 足球 ☐ 篮球

```
<form>
```

爱好:

```
<label>
```

```
  <input type="checkbox" name="like" value="football">足球
```

```
</label>
```

```
<label>
```

```
  <input type="checkbox" name="like" value="basketball">篮球
```

```
</label>
```

```
</form>
```

复选框使用相同的name值主要是为取值方便

[【返回】](#)

## 5. 列表框

专业:

专业特长:

- .NET编程
- J2EE编程
- Java编程
- Android编程

size=1 为下拉列表框(默认)  
size>1 为列表框

**<select** id或name="名称" size=可见行数>

**<option value="值1" >内容1</option>**

...

**<option value="值n" >内容n</option>**

**</select>**

显示内容和实际值可以相同

可增加 selected="selected" 指定默认选中项

## 单选下拉列表示例：

专业：  ▼

```
<form>
  专业：
  <select name="major">
    <option value="0">计算机科学与技术</option>
    <option value="1" selected="selected" >软件工程</option>
    <option value="2">网络工程</option>
  </select>
</form>
```

## 多选列表框示例：

```
<form>
```

专业特长： <br/>

```
<select name="master" size=4 multiple="multiple">
```

```
<option value= "0" >.NET编程</option>
```

```
<option value= "1" >J2EE编程</option>
```

```
<option value= "2" >Java编程</option>
```

```
<option value= "3" >Android编程</option>
```

```
<option value= "4" >C语言编程</option>
```

```
</select>
```

```
</form>
```

多选（配合ctrl或shift键）



[【返回】](#)

## 6. 文件域

附件：  test.html

浏览选择本机的文件，上传需要另外添加代码

<form >

附件： <input type="file" name="myfile" accept="文件类型">

</form>



文件类型值	描述
缺省	所有文件
image/*	图像文件
audio/*	声音文件
video/*	视频文件
MIME_type	某个有效MIME类型，不带参数

[【返回】](#)



## 7. 文本区域

简历:

请填写个人简历

```
<form>  
简历: <br/>  
<textarea name="intro" cols="50" rows="10">请填写个人简历</textarea>  
</form>
```

文本域大小(行列数)

[【返回】](#)

## 8. 提交和重置按钮

submit按钮与form的action属性自动绑定

```
<input type="submit" value="提交">  
<input type="reset" value="重填">
```

用户名:

密码:

性别: ☒ 男 ☐ 女

爱好: ☐ 足球 ☐ 篮球

专业:

简历:

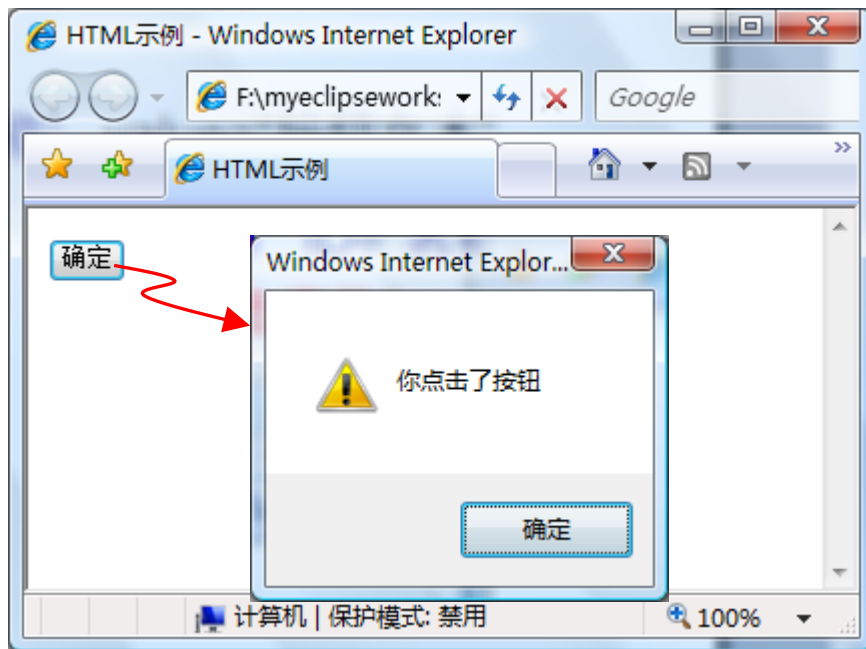
[【返回】](#)

## 9. 一般按钮

事件属性

```
<input type="button" value="确定" onclick="alert('你点击了按钮') " >
```

JavaScript代码



[【返回】](#)

## 10. Button按钮

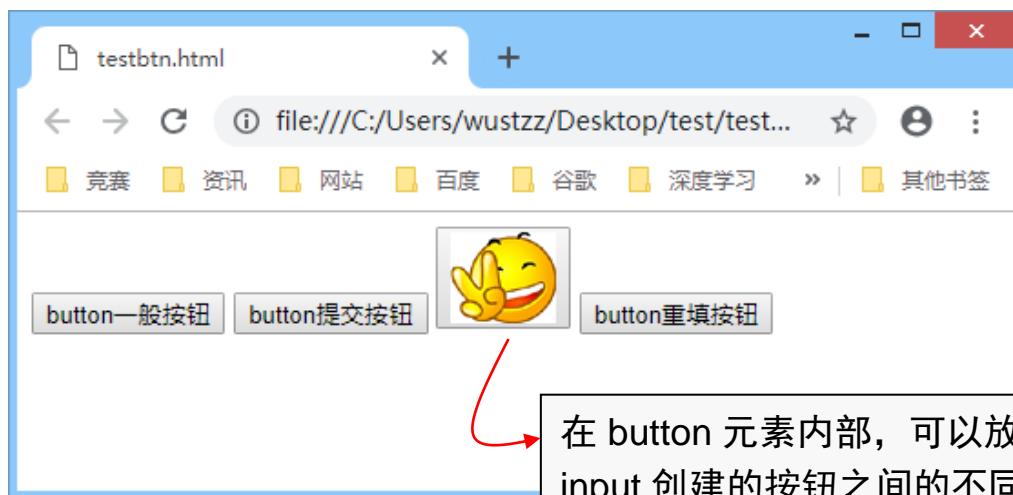
1. type不要省略
2. button文字不要放到value属性中

`<button type="button"> button一般按钮 </button>`

`<button type="submit"> button提交按钮 </button>`

`<button type="submit">  </button>`

`<button type="reset"> button重填按钮 </button>`



在 button 元素内部，可以放置文本或图像，这是与 input 创建的按钮之间的不同之处。

[【返回】](#)

## 11. 日期域

日期 2018/09/01 x ▴ ▾ ▼

2018年09月 ▾ ◀ ● ▶

周一	周二	周三	周四	周五	周六	周日
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

<form>

<input type="date" name="mydate">

</form>

[【返回】](#)

## 12. 区间域



```
<form>
```

```
成绩<input type="range" min="0.0" max="100.0" step="0.5" value="60.0"
id="score" name="score" >
```

```
    <span id="msg"></span>
```

```
    <input type="submit" value="提交">
```

```
</form>
```

```
<script>
```

```
    document.getElementById("score").onchange=function(){
        var msg = document.getElementById("msg");
        msg.innerHTML=this.value;
```

```
    }
```

```
</script>
```

onchange事件

课后作业  
(不添加样式)

欢迎注册会员

手机号码： 必填

创建密码： 必填

注册邮箱： 必填

验证码： UJFg 

性别：☒ 男 ☐ 女

生日：

年龄：

籍贯：

个人学历：

月薪：

个人爱好：☐ 唱歌 ☐ 跑步 ☐ 游泳

个人照片： 未选择任何文件

个人简介：

[【返回】](#)

## 附录1：文件上传问题

当一个表单包含时，表单的enctype必须指定为multipart/form-data，method必须指定为post，浏览器才能正确编码并以multipart/form-data格式发送表单的数据。（当然完成上传还需要配合后台代码）

```
<form action="Upload" method="post" enctype="multipart/form-data">
```

```
附件： <input type="file" name="myfile" >
```

```
</form>
```



enctype值	描述
application/x-www-form-urlencoded	默认，在发送前对所有字符进行编码(将空格转换为+字符，特殊字符转换为ASCII HEX值)
multipart/form-data	不对字符编码，当使用文件上传控件的表单时，该值是必须的
text/plain	将空格转换为+字符，但不编码特殊字符



## 后台C#代码：后台控制器添加文件上传功能

```
public ActionResult Upload()
{
    HttpPostedFileBase file = Request.Files["myfile"];
    if (file == null) return Content("上传失败");
    var fileName = Path.Combine( Request.MapPath("~/Upload"),
                    Path.GetFileName(file.FileName) );

    try
    {
        file.SaveAs(fileName);
        return Content("上传成功");
    }
    catch
    {
        return Content("上传失败");
    }
}
```

指定上传位置

先在当前网站根目录下新建一个名为Upload文件夹

## 上传添加时间戳代码：

```
public ActionResult Upload()
{
    HttpPostedFileBase file = Request.Files["photo"];
    if (file == null) return Content("上传失败");
    filename = file.FileName;
    filename=filename.Substring(0,filename.IndexOf(".")) +
        GetTimeStamp() + "." +
        filename.Substring(filename.IndexOf(".") + 1);
    var fileName = Path.Combine(Request.MapPath("~/Upload"),
                                Path.GetFileName(filename));

    try
    {   file.SaveAs(fileName);       return Content("上传成功");   }
    catch
    {   return Content("上传失败");   }
}
```

此处 filename 定义为类成员变量

文件名加时间戳

## 时间戳：后台代码

```
public static string GetTimeStamp()
{
    TimeSpan ts = DateTime.UtcNow - new DateTime(1970, 1, 1, 0, 0, 0, 0);
    return Convert.ToInt64(ts.TotalSeconds).ToString();
}
```

[【返回】](#)

## 附录2： HTML5 的 File API

HTML5 的 File API 提供了 File 和 FileReader 两个主要对象，可以使用JS 代码获得文件信息并读取文件

图片预览：



选择文件 cat.jpg

文件: cat.jpg

大小: 16536

修改: Wed Aug 15 2018 16:41:52 GMT+0800 (中国标准时间)

## Form表单代码

```
<form method="post" action="#" enctype="multipart/form-data">
  <p>图片预览: </p>
  <p></p>
  <div id="test-image-preview" style="border: 1px solid silver; width: 100%; height: 200px;
background-repeat: no-repeat; background-position: center;">
  </div>
  <p></p>
  <p>
    <input type="file" id="test-image-file" name="test">
  </p>
  <p id="test-file-info">没有选择文件</p>
</form>
```

```
<script>
    var fileInput = document.getElementById('test-image-file');
    var info = document.getElementById('test-file-info');
    var preview = document.getElementById('test-image-preview');

    fileInput.addEventListener('change', function () { // 监听change事件
        preview.style.backgroundImage = ""; // 清除背景图片
        if (!fileInput.value) {
            info.innerHTML = '没有选择文件';
            return;
        }

        var file = fileInput.files[0]; // 获取File引用
        // 获取File信息:
        info.innerHTML = '文件: ' + file.name + '<br>' + '大小: ' + file.size + '<br>' + '修改: ' + file.lastModifiedDate;
        if (file.type !== 'image/jpeg' && file.type !== 'image/png' && file.type !== 'image/gif') {
            alert('不是有效的图片文件!');
            return;
        }

        var reader = new FileReader(); // 读取文件
        reader.onload = function (e) { // 发起一个异步操作来读取文件内容
            var data = e.target.result; // data串形如: 'data:image/jpeg;base64,/9j/4AAQSk...(base64编码)...'
            preview.style.backgroundImage = 'url(' + data + ')';
        };

        reader.readAsDataURL(file); // 以DataURL的形式读取文件
    });
</script>
```

服务器对Base64解码就可以得到原始文件的二进制内容



【完】