

# HTML

---

## HTML

[简介](#)

[基础](#)

[声明](#)

[HTML标签](#)

[标题](#)

[段落](#)

[注释](#)

[水平线](#)

[文本格式化标签](#)

[超链接](#)

[图像](#)

[<img>标签](#)

[<map>标签](#)

[表格](#)

[框架](#)

[脚本标签](#)

[HTML元素](#)

[元素](#)

[属性](#)

[头部元素](#)

[列表元素](#)

[区块元素](#)

[表单元素](#)

[颜色](#)

[字符实体](#)



# HTML

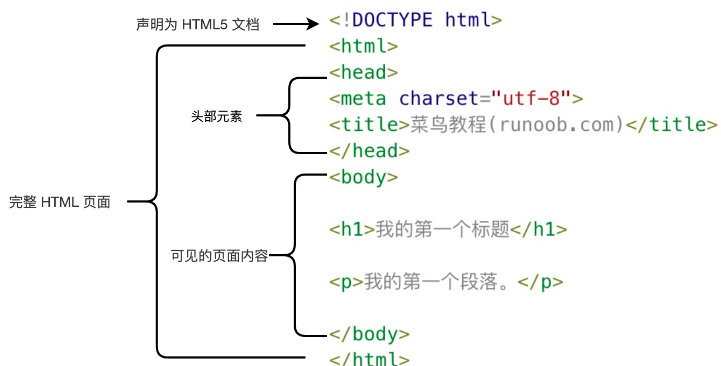
## 简介

**超文本标记语言**（英语：HyperText Markup Language，简称：HTML）是一种用于创建网页的标准标记语言。

HTML产生于1990年，1997年HTML4成为互联网标准，并广泛应用于互联网应用的开发。

HTML5是Web中核心语言HTML的规范，用户使用任何手段进行网页浏览时看到的内容原本都是HTML格式的，在浏览器中通过一些技术处理将其转换成为了可识别的信息。

## HTML预览



- `<!DOCTYPE html>` 声明为 HTML5 文档
- `<html>` 元素是 HTML 页面的根元素
- `<head>` 元素包含了文档的元 (meta) 数据，如 `<meta charset="utf-8">` 定义网页编码格式为 utf-8。
- `<title>` 元素描述了文档的标题
- `<body>` 元素包含了可见的页面内容
- `<h1>` 元素定义一个大标题
- `<p>` 元素定义一个段落

## 浅析html,.css,js区别

- html

用于勾勒出网页的结构和内容

- CSS

用于美化网页

- JavaScript

让网页呈现动态的数据和效果

jQuery

web项目基础

浏览器

HTTP协议

服务器

网页存到服务器上，运在浏览器上

## 基础

### 声明

为了让浏览器识别html版本，正确显示网页。减少中文乱码等问题。

声明不区分大小写。



Plain Text

```
1  HTML5
2  <!DOCTYPE html>
3  html 4.01
4  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
5  "http://www.w3.org/TR/html4/loose.dtd">
6  html 1.0
7  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
8  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

对于中文网页需要使用 `<meta charset="utf-8">` 声明编码，否则会出现乱码。有些浏览器(如 360 浏览器)会设置 GBK 为默认编码，但是，只要在HTML文件里写了`<!doctype hmtl>`和 `<meta charset="utf-8" />`，浏览器将会按照utf-8编码解析网页。

不过，一定要确保文档编码和meta设置的编码，一定要相同。比如，在记事本保存时，选择编码为utf-8，保存。（代码无需变动）。另一种，改变方式可以变动meta中所设置的编码和文档编码保持一致。

utf-8和utf8的区别

"UTF-8" 是标准写法, php 在 Windows 下边英文不区分大小写, 所以也可以写成 "utf-8"。"UTF-8" 也可以把中间的"- "省略, 写成 "UTF8"。

在数据库中只能使用"utf8"(MySQL) 在MySQL的命令模式中只能使用"utf8", 不能使用"utf-8", 也就是说在PHP程序中只能使用 "set names utf8(不加小横杠)", 如果你加了"- "此行命令将不会生效, 但是在 PHP 中 header 时却要加上"- "

```
<?php header('Content-Type: text/html; charset=UTF-8'); ?>
```

即

除了在命令 "mysql\_query(set names utf8)" 外一律用大写"UTF-8"。

## HTML标签

HTML标签对于大小写不敏感: <P>等同于<p>。

- HTML 标签是由尖括号包围的关键词, 比如
- 标签分为开放标签和闭合标签。
- HTML 标签通常是成对出现的, 但也有例外,<br>和</br>这两个都是单独存在的, 本身也都是开放标签和闭合标签,不包含内容, 所以只是个标签。

▼	Plain Text
1	<标签>内容<标签>

单标签 等于无内容的双标签

▼	Plain Text
1	<student/>

双标签

▼	Plain Text
1	<student "id=""uid"></student>

浏览器并不会直接显示HTML标签, 但是通过标签来决定如何展示HTML页面的内容.

## 标题

标题 (Heading) 是通过 <h1> – <h6> 标签进行定义的。

▼	Plain Text
1	<h1>hello</h1>
2	<h2>hello</h2>

尽量不要使用标题这个标签来生成粗体或者大号字体，这种可能会打乱原有的文档结构。

## 段落

### <p>标签（块级元素）

HTML可以将文档分成若干段落。

折行，不产生新段落的情况下进行换行，可以借助<br>标签

▼ Plain Text |

```
1 <p>hello</p>
2 <p>你好<br>bob</p>
```

## 注释

可以将注释插入 HTML 代码中，这样可以提高其可读性，使代码更易被人理解。浏览器会忽略注释，也不会显示它们。

▼ Plain Text |

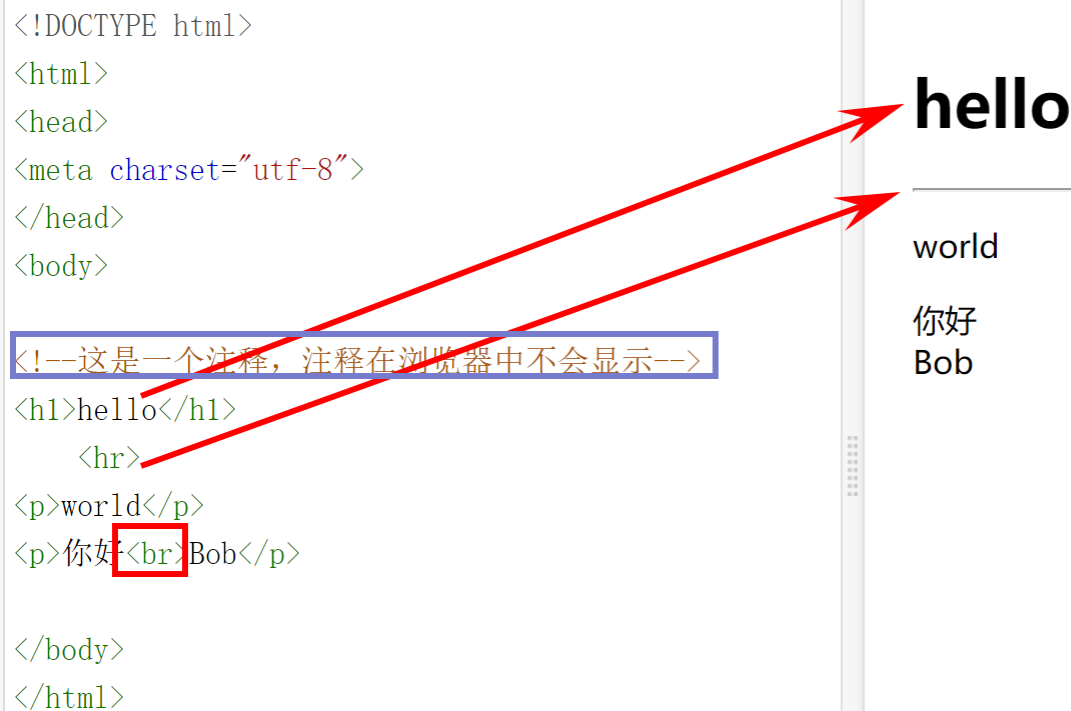
```
1 <!--这是注释-->
```

## 水平线

<hr>标签在HTML页面中创建水平线，可以用于分隔内容。

▼ Plain Text |

```
1 <h1>hello</h1>
2 <hr>
3 <h1>world</h1>
```



## 文本格式化标签

一些 HTML 标签除了具有一定的语义（含义）外，还有默认的风格，例如**<b>**（加粗），通过这些标签我们无需借助 CSS 就可以为网页中的内容定义样式。在这些具有语义和默认样式的标签中，有许多是针对文本的，通过这些标签我们可以格式化文本（为文本添加样式），例如使文本加粗、倾斜或者添加下划线等。

### 文本格式化标签

标签	描述
<b>&lt;b&gt;...&lt;/b&gt;</b>	加粗标签中的字体
<b>&lt;em&gt;...&lt;/em&gt;</b>	强调标签中的内容，并使标签中的字体倾斜
<b>&lt;i&gt;...&lt;/i&gt;</b>	定义标签中的字体为斜体
<b>&lt;small&gt;...&lt;/small&gt;</b>	定义标签中的字体为小号字体
<b>&lt;strong&gt;...&lt;/strong&gt;</b>	强调标签中的内容，并将字体加粗
<b>&lt;sub&gt;...&lt;/sub&gt;</b>	定义下标文本
<b>&lt;sup&gt;...&lt;/sup&gt;</b>	定义上标文本
<b>&lt;ins&gt;...&lt;/ins&gt;</b>	定义文档的其余部分之外的插入文本

<code>&lt;del&gt;...&lt;/del&gt;</code>	在文本内容上添加删除线
<code>&lt;code&gt;...&lt;/code&gt;</code>	定义一段代码
<code>&lt;kbd&gt;...&lt;/kbd&gt;</code>	用来表示文本是通过键盘输入的
<code>&lt;samp&gt;...&lt;/samp&gt;</code>	定义程序的样本
<code>&lt;var&gt;...&lt;/var&gt;</code>	定义变量
<code>&lt;pre&gt;...&lt;/pre&gt;</code>	定义预格式化的文本，被该标签包裹的文本会保留所有的空格和换行符，字体也会呈现为等宽字体
<code>&lt;abbr&gt;...&lt;/abbr&gt;</code>	用来表示标签中的内容为缩写形式
<code>&lt;address&gt;...&lt;/address&gt;</code>	用来定义文档作者的联系信息，被该标签包裹的文本通常会以斜体呈现，并在文本前面换行
<code>&lt;bdo&gt;...&lt;/bdo&gt;</code>	定义标签中的文字方向
<code>&lt;blockquote&gt;...&lt;/blockquote&gt;</code>	定义一段引用的文本，例如名人名言，文本会换行输出，并在左右两边进行缩进
<code>&lt;q&gt;...&lt;/q&gt;</code>	定义一段短的引用，浏览器会将引用的内容使用双引号包裹起来
<code>&lt;cite&gt;...&lt;/cite&gt;</code>	表示对某个文献的引用，例如书籍或杂志的名称，文本会以斜体显示
<code>&lt;dfn&gt;...&lt;/dfn&gt;</code>	用来定义一个术语，标签中的文本会以斜体呈现

### 关于 `<b>` 和 `<strong>`

在显示上，这两个标签都可是加粗文本，那么为什么会有两个功能"相同"的标签呢？而且好像 `strong` 并非个例。

- 首先，这两个并非完全相同。比如，如果使用网页阅读器阅读网页（盲人使用），`strong` 会重读，`b` 则不会。
- 其次，从起源上来说，`strong` 是为了在未来建设语义网而诞生的。应该知道的是，`html` 是负责显示的标记，不能表示语义。也就是说，浏览器知道这个标签如何显示，而不知道标签所标记的内容应该是什么含义。而 `strong` 在语义上走出了第一步。

示例：

```

1  <b>这个文本是加粗的</b>
2  <br />
3  <strong>这个文本是加粗的</strong>
4
5  <br />
6  这个文本包含
7  <sub>下标</sub>
8  <br />
9  这个文本包含
10 <sup>上标</sup>
11
12 <br>
13 <del>这是删除线</del>
14 <br>
15 <p>前面是宋体, <dfn>后面是斜体</dfn></p>

```

```

2  <html>
3  <head>
4  <meta charset="utf-8">
5  </head>
6
7  <body>
8  <b>这个文本是加粗的</b>
9  <br />
10 <strong>这个文本是加粗的</strong>
11
12 <br />
13 这个文本包含
14 <sub>下标</sub>
15 <br />
16 这个文本包含
17 <sup>上标</sup>
18
19 <br>
20 <del>这是删除线</del>
21 <br>
22 <p>前面是宋体, <dfn>后面是斜体</dfn></p>
23 </body>

```

这个文本是加粗的  
 这个文本是加粗的  
 这个文本包含 下标  
 这个文本包含 上标  
 这是删除线  
 前面是宋体 后面是斜体

## 超链接

HTML 使用超级链接与网络上的另一个文档相连。几乎可以在所有的网页中找到链接。点击链接可以从一张页面跳转到另一张页面。



**<a>标签来设置超文本链接**，在标签<a> 中使用了href属性来描述链接的地址。

▼ Plain Text |

```
1 <a href="链接地址">链接文本</a>
2 <a href="#">回到顶部</a>
```

链接文本不一定要文本，可以是图片，HTML其他元素都行。

- **target属性**

使用 target 属性，你可以定义被链接的文档在何处显示。

1. target="\_blank":在新窗口中浏览新的页面。
2. target="\_self":在同一个窗口打开新的页面。
3. target="\_parent":在父窗口中打开新的页面。（页面中使用框架才有用）
4. target="\_top" :以整个浏览器作为窗口显示新页面。（突破了页面框架的限制）

▼ Plain Text |

```
1 <a href="https://www.baidu.com/" target="_blank"
2   rel="noopener noreferrer">访问百度!</a>
3
4 <p>如果你将 target 属性设置为"_blank";, 链接将在新窗口打开, 同时
5   加上rel="noopener noreferrer"意思是不会打开其他的网站,
6   因为恶意病毒可能会修改你的浏览器空白页地址。。</p>
7
```

- **id属性**

id 属性可用于创建一个 HTML 文档书签.(注意, id不能重复)

书签不会以任何特殊方式显示, 即在 HTML 页面中是不显示的。

在HTML文档中插入ID:

▼ Plain Text |

```
1 <a id="tips">有用的提示部分</a>
```

在HTML文档中创建一个链接到"有用的提示部分(id="tips") ":

▼ Plain Text |

```
1 <a href="#tips">访问有用的提示部分</a>
```

或者，从另一个页面创建一个链接到"有用的提示部分(id="tips") "：

```
1 <a href="https://www.baidu.com/html/html-links.html#tips">访问有用的提示部分</a>
```

注意：

请始终将正斜杠添加到子文件夹。一般情况，依赖服务器配置。

1. 主要问了区分访问的是文件还是文件夹，比如访问路径为<http://www.a.com/b>浏览器不知道访问的是文件还是文件夹，会先访问一遍确认是那种类型，如果是文件夹则会加上/,重新访问地址为<http://www.a.com/b/>的路径。
2. 如果带后缀名<http://www.a.com/b.html>可以明确知道访问的是文件，不需要再次请求。

## 图像

### <img>标签

图像由<img> 标签定义。

<img> 是空标签，只包含属性，并且没有闭合标签。

### 源属性 (src)

源属性的值就是图像的URL地址

### Alt属性

图像定义一串预备的可替换的文本。

在浏览器无法载入图像时，替换文本属性告诉读者她们失去的信息。此时，浏览器将显示这个替代性的文本而不是图像。为页面上的图像都加上替换文本属性是个好习惯，这样有助于更好的显示信息，。

### 高度和宽度属性

height（高度）与width（宽度）属性用于设置图像的高度与宽度。

```
1 
```

## 绝对路径与相对路径

## 1绝对路径

从盘符开始写出图片的绝对路径

C:/images/1.jpg或者/home/soft01/images/1.jpg

缺点

1.路径过长

2.不方便迁移，路径易报错。

## 2相对路径

只需要写出图片和网页的相对位置。方便迁移，简单。

1.平级（同目录下）

2.下级 

3.上级 

（照片最好全放到一个文件下里，使用上级最方便）

（然后统一放到src下）



## <map>标签

<map> 标签用于客户端图像映射。

**图像映射指带有可点击区域的一幅图像。**

<img>中的 usemap 属性可引用 <map> 中的 id 或 name 属性（取决于浏览器），所以我们应同时向 <map> 添加 id 和 name 属性。

**area 元素永远嵌套在 map 元素内部。**

area 元素可定义图像映射中的区域。

1、矩形：(左上角顶点坐标为(x1,y1)，右下角顶点坐标为(x2,y2))

```
<area shape="rect" coords="x1,y1,x2,y2" href=url>
```

2、圆形：(圆心坐标为(X1,y1)，半径为r)

```
<area shape="circle" coords="x1,y1,r" href=url>
```

3、多边形：(各顶点坐标依次为(x1,y1)、(x2,y2)、(x3,y3) .....)

```
<area shape="poly" coords="x1,y1,x2,y2 ....." href=url>
```

```

1 
3
4 <map name="planetmap">
5   <area shape="rect" coords="0,0,82,126" href="sun.htm" alt="Sun">
6   <area shape="circle" coords="90,58,3" href="mercur.htm" alt="Mercury">
7   <area shape="circle" coords="124,58,8" href="venus.htm" alt="Venus">
8 </map>
9 //shape指的是点击区域的形状,
10 coords指的应该是链接区域在图片中的坐标 (像素为单位)

```

## 表格

表格由 `<table>` 标签来定义。

- table 表格声明范围
- tr 行
- td 行里的单元格
- tr 声明表头

表格中的标签

标签	描述
<code>&lt;table&gt;</code>	定义表格
<code>&lt;th&gt;</code>	定义表格的表头
<code>&lt;tr&gt;</code>	定义表格的行
<code>&lt;td&gt;</code>	定义表格单元
<code>&lt;caption&gt;</code>	定义表格标题
<code>&lt;colgroup&gt;</code>	定义表格列的组
<code>&lt;col&gt;</code>	定义用于表格列的属性
<code>&lt;thead&gt;</code>	定义表格的页眉
<code>&lt;tbody&gt;</code>	定义表格的主体

<tfoot>

定义表格的页脚

实例：

```

1  <table border="1"cellspacing="0" width="30%" >
2  <caption>单元格间距</caption>
3      <tr>
4          <td>alice</td>
5          <td>bob</td>
6      </tr>
7      <tr>
8          <td>cc</td>
9          <td>dd</td>
10     </tr>
11 </table>
12
13 <table border="1" cellpadding="0" width="30%" >
14 <caption>单元格边距</caption>
15     <tr>
16         <td>Alice</td>
17         <td>Bob</td>
18     </tr>
19     <tr>
20         <td>cc</td>
21         <td>dd</td>
22     </tr>
23 </table>
24
25
26 <table border="1">
27     <tr>
28         <td rowspan="2" >alice</td>
29         <td>bob</td>
30     </tr>
31     <tr>
32         <td>cc</td>
33     </tr>
34 </table>
35
36 <table border="1">
37     <tr>
38         <td colspan="2">bob</td>
39     </tr>
40     <tr>
41         <td >alice</td>
42         <td>cc</td>
43     </tr>
44 </table>
45

```

```

46 <table border="1">
47   <caption>带有标题的表格</caption>
48   <tr>
49     <th>A</th>
50     <th>B</th>
51   </tr>
52 </table>
53
54 <table border="1">
55 <caption>多元化的表格</caption>
56 <tr>
57   <td>
58     <p>这是一个段落</p>
59     <p>这是另一个段落</p>
60   </td>
61   <td>这个单元格包含一个表格:
62     <table border="1">
63 <caption>表格中的表格</caption>
64   <tr>
65     <td>A</td>
66     <td>B</td>
67   </tr>
68   <tr>
69     <td>C</td>
70     <td>D</td>
71   </tr>
72   </table>
73   </td>
74 </tr>
75 <tr>
76   <td>这个单元格包含一个列表
77     <ul>
78       <li>apples</li>
79       <li>bananas</li>
80       <li>pineapples</li>
81     </ul>
82   </td>
83   <td>HELLO</td>
84 </tr>
85 </table>

```





```

<!DOCTYPE html>
<html>
<body>

<iframe src="demo_iframe.htm" name="iframe_a"></iframe>
<p><a href="https://www.runoob.com" target="iframe_a">
  RUNOOB.COM</a></p>

<p>因为 a 标签的 target 属性是名为
  iframe_a 的 iframe 框架，
  所以在点击链接时页面会显示在 iframe 框架中。</p>

</body>
</html>

```



**注意：** 因为 a 标签的 target 属性是名为 iframe\_a 的 iframe 框架，所以在点击链接时页面会显示在 iframe 框架中。

网页对于嵌套的设置：

X-Frame-Options

这个有三个可配置值

- DENY：表示该网站页面不允许被嵌套，即便是在自己的域名的页面中也不能进行嵌套。
- SAMEORIGIN：表示该页面可以在相同域名页面中被嵌套展示。
- ALLOW-FROM uri：表示该页面可以在指定来源页面中进行嵌套展示。

## 脚本标签

<script>标签

- <script> 标签用于定义客户端脚本。
- <script> 元素既可包含脚本语句，也可通过 src 属性指向外部脚本文件。

<noscript>标签

- <noscript> 标签提供无法使用脚本时的替代内容，比方在浏览器禁用脚本时，或浏览器不支持客户端脚本时。
- <noscript>元素可包含普通 HTML 页面的 body 元素中能够找到的所有元素。
- 只有在浏览器不支持脚本或者禁用脚本时，才会显示 <noscript> 元素中的内容：

```

1  <script>
2  document.write("Hello World!")
3  </script>
4  <noscript>抱歉，你的浏览器不支持 JavaScript!</noscript>

```

## HTML元素

### 元素

元素语法：


- 元素以开始标签起始，以结束标签终止。
- 元素内容是开始标签与结束标签之间的内容。
- 空元素在开始标签中进行关闭（以开始标签的结束而结束）
- 不少元素具有空内容。
- 大多数元素可以嵌套其他元素。

```

1  <p>这是一个段落。</p>，这一整个就是元素。

```

标签加属性加内容为元素



```

<p class="editor-note">My cat is very grumpy</p>

```

```

test
1  <html>
2
3  <body>
4  <p>hello</p>
5  </body>
6
7  </html>

```

### 属性

- 元素中可以设置属性
- 属性一般添加到开始标签中

- 属性总是以名称或者值对的形式出现，例如：name="value"或者name='value'
- 属性和属性值对大小写都行。
- 属性值始终被包括在引号内，双引号或单引号皆可。

在某些个别的情况下，比如属性值本身就含有双引号，那么就必须使用单引号，例如：

```
name='hello "bob" '
```



Plain Text

```
1 <div style="color: antiquewhite">hello</div>
```

## 头部元素

<head> 元素包含了所有的头部标签元素。在 <head>元素中声明元信息，指示浏览器在何处可以找到样式表，包含的脚本等等。

可以添加在头部区域的元素标签为：<title>，<style>，<meta>，<link>，<script>，<noscript> 和 <base>。

### <title>元素

<title>标签定义了不同文档的标题。

在HTML/XHTML文档是必需的。

<title> 元素：

1. 定义了浏览器工具栏的标题
2. 当网页添加到收藏夹时，显示在收藏夹中的标题
3. 显示在搜索引擎结果页面的标题

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <meta charset="utf-8">
5  <title>文档标题</title>
6  </head>
7
8  <body>
9  文档内容.....
10 </body>
11
12 </html>
```

### <base>元素

<base> 标签描述了基本的链接地址/链接目标，该标签作为HTML文档中所有的链接标签的默认链接。

```
1  <head>
2  <base href="http://www.baidu.com" target="_blank">
3  </head>
```

### <link>元素

<link> 标签定义了文档与外部资源之间的关系。

<link> 标签通常用于链接到样式表。

```
1  <head>
2  <link rel="stylesheet" type="text/css" href="test.css">
3  </head>
```

### <style> 元素

<style> 标签定义了HTML文档的样式文件引用地址。

在<style> 元素中你也可以直接添加样式来渲染 HTML 文档。

```
1 <head>
2 <style type="text/css">
3 body{
4     background-color:red;
5 }
6 </style>
7 </head>
```

## <meta>元素

meta标签描述了一些基本的元数据。

<meta> 标签提供了元数据.元数据也不显示在页面上，但会被浏览器解析。

META 元素通常用于指定网页的描述，关键词，文件的最后修改时间，作者，和其他元数据。

常用属性：content、http-equiv、charset

```
1 //为搜索引擎定义关键词：
2 <meta name="keywords" content="HTML, CSS, XML, XHTML, JavaScript">
3 //为网页定义描述内容：
4 <meta name="description" content="免费 Web & 编程 教程">
5 定义网页作者：
6 <meta name="author" content="ktrol">
7 每30秒钟刷新当前页面
8 <meta http-equiv="refresh" content="30">
```

## 列表元素

无序列表使用 <ul> 标签，有序列表使用<ol> 标签。

```

1  <ul>
2  <li>A</li>
3  <li>B</li>
4  </ul>
5
6  <ol>
7  <li>C</li>
8  <li>D</li>
9  </ol>

```

<h4>无序列表:</h4>

```

<ul>
<li>A</li>
<li>B</li>
</ul>

```

**无序列表:**

- A
- B

<h4>有序列表:</h4>

```

<ol>
<li>C</li>
<li>D</li>
</ol>

```

**有序列表:**

1. C
2. D

自定义列表以 **<dl>** 标签开始。每个自定义列表项以 **<dt>** 开始。每个自定义列表项的定义以 **<dd>** 开始。

<h4>一个自定义列表: </h4>

```

<dl>
  <dt>Coffee</dt>
  <dd>- black hot drink</dd>
  <dt>Milk</dt>
  <dd>- white cold drink</dd>
</dl>

```

**一个自定义列表:**

Coffee  
- black hot drink  
Milk  
- white cold drink

## 区块元素

块级元素显示时通常会以新行来开始（和结束），<h1>, <p>, <ul>, <table>。

内联元素显示时通常不会以新行开始，<b>, <td>, <a>, <img>。

独立成行的元素为块元素，行内元素不能独立成行。

HTML <div> 元素是块级元素，它可用于组合其他 HTML 元素的容器。

<div> 元素没有特定的含义。除此之外，由于它属于块级元素，浏览器会在其前后显示折行。

如果与 CSS 一同使用，<div> 元素可用于对大的内容块设置样式属性。

<div> 元素的另一个常见的用途是文档布局。它取代了使用表格定义布局的老式方法。使用 <table> 元素进行文档布局不是表格的正确用法。<table> 元素的作用是显示表格化的数据。

- <span> 元素是内联元素，可用作文本的容器。
- <span> 元素也没有特定的含义，当与 CSS 一同使用时，<span> 元素可用于为部分文本设置样式属性。

Plain Text

```
1 <p>区块元素</p>
2 <div style="color:red">
3   <h3>这是一个在 div 元素中的标题。</h3>
4   <p>这是一个在 div 元素中的段落。</p>
5 </div>
6 <br>
7 <p>内联元素</p>
8 <p>北京<span style="color: blue">北京大学</span><p>
9 <p>北京<b>北京大学</b></p>
10 <p>北京<u>北京大学</u></p>
```

```
<p>区块元素</p>
<div style="color:red">
  <h3>这是一个在 div 元素中的标题。</h3>
  <p>这是一个在 div 元素中的段落。</p>
</div>
<br>
<p>内联元素</p>
<p>北京<span style="color: blue">北京大学</span><p>
<p>北京<b>北京大学</b></p>
<p>北京<u>北京大学</u></p>

</body>
</html>
```

区块元素

**这是一个在 div 元素中的标题。**

这是一个在 div 元素中的段落。

内联元素

北京北京大学

北京**北京大学**

北京北京大学

## 表单元素

HTML 表单用于收集用户的输入信息。

## 表单标签

标签	描述
<code>&lt;form&gt;</code>	定义供用户输入的表单
<code>&lt;input&gt;</code>	定义输入域
<code>&lt;textarea&gt;</code>	定义文本域（一个多行的输入控件）
<code>&lt;label&gt;</code>	定义了 <code>&lt;input&gt;</code> 元素的标签，一般为输入标题
<code>&lt;fieldset&gt;</code>	定义了一组相关的表单元素，并使用外框包含起来
<code>&lt;legend&gt;</code>	定义了 <code>&lt;fieldset&gt;</code> 元素的标题
<code>&lt;select&gt;</code>	定义了下拉选项列表
<code>&lt;optgroup&gt;</code>	定义选项组
<code>&lt;option&gt;</code>	定义下拉列表中的选项
<code>&lt;button&gt;</code>	定义一个点击按钮
<code>&lt;datalist&gt;</code>	指定一个预先定义的输入控件选项列表
<code>&lt;keygen&gt;</code>	定义了表单的密钥对生成器字段
<code>&lt;output&gt;</code>	定义一个计算结果

表单元素（声明表单范围，声明数据信息提交到服务器的范围）

表单控件（用来给用户输入数据）

### 1. input元素，输入类型由type属性决定。

- 文本框`<input type="text">`  
value：默认输入的值，maxlength：最大长度，readonly：设置只读。
- 密码框`<input type="password">`
- 文件框`<input type="file"/>`
- 隐藏框`<input type="hidden" value="kk"/>`
- 单选框`<input type="radio">`  
name：组名，同组radio互斥，checked:设置默认选中
- 复选框`<input type="checkbox">`  
checked 设置默认选中
- 提交按钮`<input type="submit">`



method属性, 用来定义表单数据的提交方式

- post: 指的是 HTTP POST 方法, 表单数据会包含在表单体内然后发送给服务器, 用于提交敏感数据, 如用户名与密码等。
- get: 默认值, 指的是 HTTP GET 方法, 表单数据会附加在 action 属性的 URL 中, 并以 ? 作为分隔符, 一般用于不敏感信息, 如分页等。例如: `https://www.baidu.com/?page=1`, 这里的 `page=1` 就是 get 方法提交的数据。
- 重置按钮 `<input type="reset">`
- 普通按钮 `<input type="button">`

## 2. 其他元素 (3个)

**label:** 用来管理表单中的文本, 可以将文本和控制件联系在一起。

**select:** 下拉选, selected 默认选中

**textarea** 文本域, cols 指定最大列数, rows 指定指定最大行数, readonly

```
1 <form action="https://www.baidu.com">
2   <p>
3     账号:<input type="text"
4     value="baidu"
5     maxlength="12">
6   </p>
7   <p>
8     密码:<input type="password"3. maxlength="16">
9   </p>
10  <p>
11    头像: <input type="file"/>
12  </p>
13  <p>
14    <input type="hidden" value="kk"/>
15  </p>
16  <p>
17    性别:
18    <input type="radio" name="sex" checked/>男
19    <input type="radio" name="sex"/>女
20  </p>
21  <p>
22    兴趣:
23    <input type="checkbox" checked/>music
24    <input type="checkbox"/>reading
25    <input type="checkbox"/>football
26  </p>
27  <p>
28    <input type="submit" value="注册"/>
29    <input type="reset" value="重置"/>
30    <input type="button" value="试试"/>
31  </p>
32
33  <p>
34    <input type="checkbox" id="a"/>
35    <label for="a">I agree</label>
36  </p>
37  <p>
38    城市:
39    <select>
40      <option>请选择</option>
41      <option> beijing</option>
42      <option>shanghai</option>
43      <option>guangzhou</option>
44      <option>changsha</option>
45    </select>
```

```

46         </p>
47         <p>
48             简介:
49             <textarea cols="5" rows="5">想啥呢</textarea>
50         </p>
51     </form>

```

账号:

密码:

头像:  No file chosen

性别: ☒ 男 ☐ 女



隐藏框可以去浏览器开发者工具中查看

兴趣: ☒ music ☐ reading ☐ football

☐ I agree

城市:

想啥呢

简介:

## 颜色

颜色由一个十六进制符号来定义，这个符号由红色、绿色和蓝色的值组成（RGB）。

RGBA 的意思是（Red-Green-Blue-Alpha）它是在 RGB 上扩展包括了“alpha”通道，运行对颜色值设置透明度，该透明度的设置范围为0-1。

可以去该网站进行颜色对比。



HTML颜色代码表 🌈

RT

## 字符实体

HTML 中的预留字符必须被替换为字符实体。  
一些在键盘上找不到的字符也可以使用字符实体来替换。

不间断空格

浏览器总是会截短 HTML 页面中的空格。如果您在文本中写 10 个空格，在显示该页面之前，浏览器会删除它们中的 9 个。如需在页面中增加空格的数量，您需要使用 &nbsp; 字符实体。

实体名称对大小写敏感！

显示结果	描述	实体名称
	空格	&nbsp;
<	小于号	&lt;
>	大于号	&gt;
&	和号	&amp;
"	引号	&quot;
'	撇号	&apos; (IE不支持)
¢	分	&cent;
£	镑	&pound;
¥	人民币/日元	&yen;
€	欧元	&euro;
§	小节	&sect;
©	版权	&copy;
®	注册商标	&reg;
™	商标	&trade;
×	乘号	&times;
÷	除号	&divide;

