HTML

- `<html>` 与 `</html>` 之间的文本描述网页
- `<body>` 与 `</body>` 之间的文本是可见的页面内容

html标题（heading） > `<h1>` - `<h6>`

html段落（paragraph） > `<p>`

html链接（anchor） > `<a>`

实例：

```
<a herf="http://baidu.com">This is a link to Baidu</a>>
```

html图像（image） > `<img>`

```
<img src="w3school.jpg" width="104" height="142" />
```

html元素

html文档是由html元素定义的。

HTML元素：**开始标签**（开始） - **元素内容**（可以为空内容） - **结束标签**（结束）

大多数html元素具有属性。

在开始标签中添加斜杠，比如 `<br />(换行)`，是关闭空元素的正确方法，

大小写不敏感。

html属性

属性为 HTML 元素提供附加信息。

属性总是以名称/值对的形式出现，比如：`name="value"`。

属性总是在 HTML 元素的*开始标签*中规定。

属性值应该始终被包括在引号内。双引号是最常用的，不过使用单引号也没有问题。

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

```
name='Bill "HelloWorld" Gates'
```

html标题

注释：默认情况下，HTML 会自动地在块级元素前后添加一个额外的空行，比如段落、标题元素前后。

HTML 水平线

`<hr />`标签在 HTML 页面中创建水平线。

hr 元素可用于分隔内容。

HTML 注释

```
<!-- This is a comment -->
```

html段落

注释：浏览器会自动地在段落的前后添加空行。（`<p>` 是块级元素）

提示：使用空的段落标记 `<p></p>` 去插入一个空行是个坏习惯。用 `<br />` 标签代替它！

html样式

style 属性用于改变 HTML 元素的样式。

应该避免使用下面这些标签和属性：

标签	描述
	定义居中的内容。
<font> 和 <basefont>	定义 HTML 字体。
<s> 和 <strike>	定义删除线文本
<u>	定义下划线文本
属性	**描述**
align	定义文本的对齐方式
bgcolor	定义背景颜色
color	定义文本颜色

style 属性淘汰了“旧的” `bgcolor` 属性、旧的 `<font>` 标签、旧的 `"align"` 属性

HTML 样式实例 - 背景颜色

background-color 属性为元素定义了背景颜色：

```
<html>

<body style="background-color:yellow">
<h2 style="background-color:red">This is a heading</h2>
<p style="background-color:green">This is a paragraph.</p>
</body>

</html>
```

HTML 样式实例 - 字体、颜色和尺寸

font-family、color 以及 font-size 属性分别定义元素中文本的字体系列、颜色和字体尺寸：

```
<html>

<body>
<h1 style="font-family:verdana">A heading</h1>
<p style="font-family:arial;color:red;font-size:20px;">A paragraph.</p>
</body>

</html>
```

HTML 样式实例 - 文本对齐

text-align 属性规定了元素中文本的水平对齐方式：

```
<html>

<body>
<h1 style="text-align:center">This is a heading</h1>
<p>The heading above is aligned to the center of this page.</p>
</body>

</html>
```

HTML 文本格式化

文本格式化标签

标签	描述
<b>	定义粗体文本。
<big>	定义大号字。
	定义着重文字。
<i>	定义斜体字。
<small>	定义小号字。
	定义加重语气。
<sub>	定义下标字。
<sup>	定义上标字。
<ins>	定义插入字。
<del>	定义删除字。
<s>	不赞成使用。使用 <del> 代替。
<strike>	不赞成使用。使用 <del> 代替。
<u>	不赞成使用。使用样式（style）代替。

“计算机输出”标签

标签	描述
<code>	定义计算机代码。
<kbd>	定义键盘码。
<samp>	定义计算机代码样本。
<tt>	定义打字机代码。
<var>	定义变量。
<pre>	定义预格式文本。
<listing>	不赞成使用。使用 <pre> 代替。
<plaintext>	不赞成使用。使用 <pre> 代替。
<xmp>	不赞成使用。使用 <pre> 代替。

引用、引用和术语定义

标签	描述
<abbr>	定义缩写。
<acronym>	定义首字母缩写。
<address>	定义地址。
<bdo>	定义文字方向。
<blockquote>	定义长的引用。
<q>	定义短的引用语。
<cite>	定义引用、引证。
<dfn>	定义一个定义项目。

一些代码范例：

```
https://www.w3school.com.cn/tiy/t.asp?f=html_textformatting

https://www.w3school.com.cn/tiy/t.asp?f=html_address

https://www.w3school.com.cn/tiy/t.asp?f=html_abbracronym

https://www.w3school.com.cn/tiy/t.asp?f=html_quotations
```

HTML计算机代码元素

HTML 计算机代码元素

标签	描述
`<code>`	定义计算机代码文本
`<kbd>`	定义键盘文本
`<samp>`	定义计算机代码示例
`<var>`	定义变量
`<pre>`	定义预格式化文本

`<code>` 元素不保留多余的空格和折行（`<pre>`可以解决）

HTML样式表

如何使用样式

当浏览器读到一个样式表，它就会按照这个样式表来对文档进行格式化。有以下三种方式来插入样式表：

外部样式表

当样式需要被应用到很多页面的时候，外部样式表将是理想的选择。使用外部样式表，你就可以通过更改一个文件来改变整个站点的外观。

```
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
```

内部样式表

当单个文件需要特别样式时，就可以使用内部样式表。你可以在 head 部分通过 `<style>` 标签定义内部样式表。

```
<head>

<style type="text/css">
body {background-color: red}
p {margin-left: 20px}
</style>
</head>
```

内联样式

当特殊的样式需要应用到个别元素时，就可以使用内联样式。 使用内联样式的方法是在相关的标签中使用样式属性。样式属性可以包含任何 CSS 属性。以下实例显示出如何改变段落的颜色和左外边距。

```
<p style="color: red; margin-left: 20px">
This is a paragraph
</p>
```

HTML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。

HTML 超链接（链接）

超链接可以是一个字，一个词，或者一组词，也可以是一幅图像，您可以点击这些内容来跳转到新的文档或者当前文档中的某个部分。

当您把鼠标指针移动到网页中的某个链接上时，箭头会变为一只小手。

我们通过使用 `<a>` 标签在 HTML 中创建链接。

有两种使用 `<a>` 标签的方式：

1. 通过使用 href 属性 - 创建指向另一个文档的链接
2. 通过使用 name 属性 - 创建文档内的书签

HTML 链接语法

链接的 HTML 代码很简单。它类似这样：

```
<a href="url">Link text</a>
```

href 属性规定链接的目标。

开始标签和结束标签之间的文字被作为超级链接来显示。"链接文本" 不必一定是文本。图片或其他 HTML 元素都可以成为链接。

HTML 链接 - target 属性

使用 Target 属性，你可以定义被链接的文档在何处显示。

HTML 链接 - name 属性

name 属性规定锚（anchor）的名称。

您可以使用 name 属性创建 HTML 页面中的书签。

书签不会以任何特殊方式显示，它对读者是不可见的。

当使用命名锚（named anchors）时，我们可以创建直接跳至该命名锚（比如页面中某个小节）的链接，这样使用者就无需不停地滚动页面来寻找他们需要的信息了。

命名锚的语法：

```
<a name="label">锚（显示在页面上的文本）</a>
```

提示：锚的名称可以是任何你喜欢的名字。

提示：您可以使用 id 属性来替代 name 属性，命名锚同样有效。

HTML 图像

通过使用 HTML，可以在文档中显示图像

图像标签（`<img>`）和源属性（Src）

在 HTML 中，图像由 `<img>` 标签定义。

`<img>` 是空标签，意思是说，它只包含属性，并且没有闭合标签。

要在页面上显示图像，你需要使用源属性（src）。src 指 "source"。源属性的值是图像的 URL 地址。

定义图像的语法是：

```
<img src="url" />
```

URL 指存储图像的位置。如果名为 "boat.gif" 的图像位于 **www.w3school.com.cn** 的 images 目录中，那么其 URL 为 **http://www.w3school.com.cn/images/boat.gif**。

浏览器将图像显示在文档中图像标签出现的地方。如果你将图像标签置于两个段落之间，那么浏览器会首先显示第一个段落，然后显示图片，最后显示第二段。

替换文本属性（Alt）

alt 属性用来为图像定义一串预备的可替换的文本。替换文本属性的值是用户定义的。

```
<img src="boat.gif" alt="Big Boat">
```

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，并且对于那些使用纯文本浏览器的人来说是非常有用的。

基本的注意事项 - 有用的提示：

假如某个 HTML 文件包含十个图像，那么为了正确显示这个页面，需要加载 11 个文件。加载图片是需要时间的，所以我们的建议是：慎用图片。

HTML 表格

你可以使用 HTML 创建表格。

表格标签

表格	描述
<table>	定义表格
<caption>	定义表格标题。
<th>	定义表格的表头。
<tr>	定义表格的行。
<td>	定义表格单元。
<thead>	定义表格的页眉。
<tbody>	定义表格的主体。
<tfoot>	定义表格的页脚。
<col>	定义用于表格列的属性。
<colgroup>	定义表格列的组。

HTML列表

列表标签

标签	描述
	定义有序列表。
	定义无序列表。
	定义列表项。
<dl>	定义定义列表。
<dt>	定义定义项目。
<dd>	定义定义的描述。
<dir>	已废弃。使用 代替它。
<menu>	已废弃。使用 代替它。

HTML其他

块 - div

类 - class

布局

响应式Web设计

框架 - 同一个窗口中显示多个页面（很难打印整张页面，处理更多的HTML文档）- frame

背景 -bgcolor（十六进制数、RGB、颜色名、图片（url））

JavaScript 使 HTML 页面具有更强的动态和交互性。

HTML script 元素

`<script>` 标签用于定义客户端脚本，比如 JavaScript。

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。

必需的 type 属性规定脚本的 MIME 类型。

JavaScript 最常用于图片操作、表单验证以及内容动态更新。

html `<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

HTML 文件路径

路径	描述
<img src="picture.jpg">	picture.jpg 位于与当前网页相同的文件夹
<img src="images/picture.jpg">	picture.jpg 位于当前文件夹的 images 文件夹中
<img src="/images/picture.jpg">	picture.jpg 当前站点根目录的 images 文件夹中
<img src="../picture.jpg">	picture.jpg 位于当前文件夹的上一级文件夹中

HTML 字符实体

HTML 中的预留字符必须被替换为字符实体。

HTML 实体(大小写敏感)

在 HTML 中，某些字符是预留的。

在 HTML 中不能使用小于号（<）和大于号（>），这是因为浏览器会误认为它们是标签。

如果希望正确地显示预留字符，我们必须在 HTML 源代码中使用字符实体（character entities）。

字符实体类似这样：

```
&entity_name;

或者

&#entity_number;
```

如需显示小于号，我们必须这样写：< 或 <

提示：使用实体名而不是数字的好处是，名称易于记忆。不过坏处是，浏览器也许并不支持所有实体名称（对实体数字的支持却很好）。

不间断空格（non-breaking space）

HTML 中的常用字符实体是不间断空格( )。

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   字符实体。

URL - Uniform Resource Locator

当您点击 HTML 页面中的某个链接时，对应的 **标签指向万维网上的一个地址。**

统一资源定位器（URL）用于定位万维网上的文档（或其他数据）。

网址，比如 http://www.w3school.com.cn/html/index.asp，遵守以下的语法规则：

```
scheme://host.domain:port/path/filename
```

解释：

- scheme - 定义因特网服务的类型。最常见的类型是 http
- host - 定义域主机（http 的默认主机是 www）
- domain - 定义因特网域名，比如 w3school.com.cn
- :port - 定义主机上的端口号（http 的默认端口号是 80）
- path - 定义服务器上的路径（如果省略，则文档必须位于网站的根目录中）。
- filename - 定义文档/资源的名称

编者注：URL 的英文全称是 Uniform Resource Locator，中文也译为“统一资源定位符”。

URL Schemes

以下是其中一些最流行的 scheme：

Scheme	访问	用于...
http	超文本传输协议	以 http:// 开头的普通网页。不加密。
https	安全超文本传输协议	安全网页。加密所有信息交换。
ftp	文件传输协议	用于将文件下载或上传至网站。
file		您计算机上的文件。

URL 编码

URL 只能使用 **ASCII 字符集**来通过因特网进行发送。

由于 URL 常常会包含 ASCII 集合之外的字符，URL 必须转换为有效的 ASCII 格式。

URL 编码使用 "%" 其后跟随两位的十六进制数来替换非 ASCII 字符。

URL 不能包含空格。URL 编码通常使用 + 来替换空格。

HTML 表单

HTML 表单用于搜集不同类型的用户输入。

<form> 元素

HTML 表单用于收集用户输入。

<form> 元素定义 HTML 表单：

实例

```
<form>
.
form elements
.
</form>
```

HTML 表单包含表单元素。

表单元素指的是不同类型的 input 元素、复选框、单选按钮、提交按钮等等。

Action 属性

*action 属性*定义在提交表单时执行的动作。

向服务器提交表单的通常做法是使用提交按钮。

通常，表单会被提交到 web 服务器上的网页。

在上面的例子中，指定了某个服务器脚本来处理被提交表单：

```
<form action="action_page.php">
```

如果省略 action 属性，则 action 会被设置为当前页面。

Method 属性

*method 属性*规定在提交表单时所用的 HTTP 方法（*GET* 或 *POST*）：

```
<form action="action_page.php" method="GET">
```

或：

```
<form action="action_page.php" method="POST">
```

何时使用 GET?

您能够使用 GET（默认方法）：

如果表单提交是被动的（比如搜索引擎查询），并且没有敏感信息。

当您使用 GET 时，表单数据在页面地址栏中是可见的：

```
action_page.php?firstname=Mickey&lastname=Mouse
```

注释：GET 最适合少量数据的提交。浏览器会设定容量限制。

何时使用 POST?

您应该使用 POST：

如果表单正在更新数据，或者包含敏感信息（例如密码）。

POST 的安全性更加，因为在页面地址栏中被提交的数据是不可见的。

`<input>` 元素

`<select>` 元素（下拉列表）

```
<select name="cars">
<option value="volvo">Volvo</option>
<option value="saab">Saab</option>
<option value="fiat">Fiat</option>
<option value="audi">Audi</option>
</select>
```

`<option>` 元素定义待选择的选项。

列表通常会把首个选项显示为被选选项。

您能够通过添加 selected 属性来定义预定义选项。

实例

```
<option value="fiat" selected>Fiat</option>
```

`<textarea>` 元素

<textarea> 元素定义多行输入字段（*文本域*）：

实例

```
<textarea name="message" rows="10" cols="30">
The cat was playing in the garden.
</textarea>
```

`<button>` 元素

<button> 元素定义可点击的*按钮*:

实例

```
<button type="button" onclick="alert('Hello World!')">Click Me!</button>
```

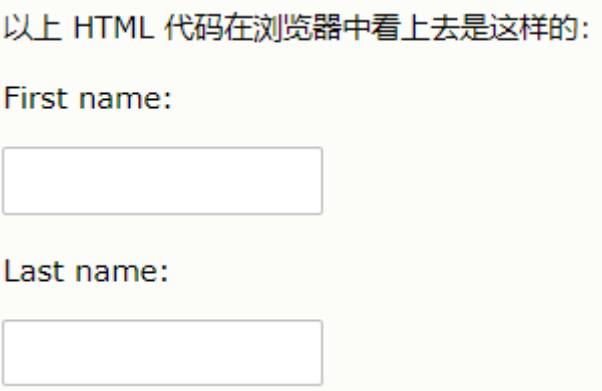
HTML 输入类型

输入类型：text

<input type="text"> 定义供*文本输入*的单行输入字段：

```
<form>
 First name:<br>
<input type="text" name="firstname">
<br>
 Last name:<br>
<input type="text" name="lastname">
</form>
```

以上 HTML 代码在浏览器中看上去是这样的：



输入类型：password

`<input type="password">` 定义*密码字段*:

```
<form>
 User name:<br>
<input type="text" name="username">
<br>
 User password:<br>
<input type="password" name="psw">
</form>
```

以上 HTML 代码在浏览器中看上去是这样的:

User name:

User password:

注释：password 字段中的字符会被做掩码处理（显示为星号或实心圆）。

输入类型：submit

`<input type="submit">` 定义*提交*表单数据至*表单处理程序*的按钮。

表单处理程序（form-handler）通常是包含处理输入数据的脚本的服务器页面。

在表单的 action 属性中规定表单处理程序（form-handler）：

```
<form action="action_page.php">
First name:<br>
<input type="text" name="firstname" value="Mickey">
<br>
Last name:<br>
<input type="text" name="lastname" value="Mouse">
<br><br>
<input type="submit" value="Submit">
</form>
```

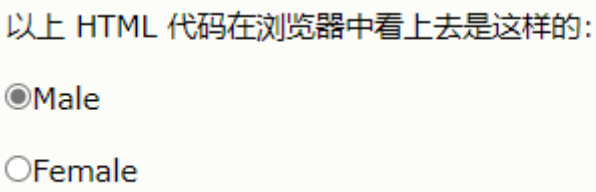

```
<form action="action_page.php">
First name:<br>
<input type="text" name="firstname" value="Mickey">
<br>
Last name:<br>
<input type="text" name="lastname" value="Mouse">
<br><br>
<input type="submit">
</form>
```

Input Type: radio

`<input type="radio">` 定义单选按钮。

Radio buttons let a user select ONLY ONE of a limited number of choices:

```
<form>
<input type="radio" name="sex" value="male" checked>Male
<br>
<input type="radio" name="sex" value="female">Female
</form>
```



Input Type: checkbox

`<input type="checkbox">` 定义复选框。

复选框允许用户在有限数量的选项中选择零个或多个选项。

```
<form>
<input type="checkbox" name="vehicle" value="Bike">I have a bike
<br>
<input type="checkbox" name="vehicle" value="Car">I have a car
</form>
```

以上 HTML 代码在浏览器中看上去是这样的：

☐I have a bike

☐I have a car

Input Type: button

`<input type="button>` 定义*按钮*。

```
<input type="button" onclick="alert('Hello World!')" value="Click Me!">
```

HTML5 输入类型

HTML5 增加了多个新的输入类型：

- color
- date
- datetime
- datetime-local
- email
- month
- number
- range
- search
- tel
- time
- url
- week

注释：老式 web 浏览器不支持的输入类型，会被视为输入类型 text。

输入类型：number

`<input type="number">` 用于应该包含数字值的输入字段。

您能够对数字做出限制。

根据浏览器支持，限制可应用到输入字段。

```
<form>
  Quantity (between 1 and 5):
  <input type="number" name="quantity" min="1" max="5">
</form>
```