

此页面由社区从英文翻译而来。了解更多并加入 MDN Web Docs 社区。

开始学习 HTML

本文将从 HTML 最基础的部分讲起，对元素（Element）、属性（Attribute）以及可能涉及的一些重要术语进行介绍，并明确它们在 HTML 文档中所处的位置。本文还会讲解 HTML 元素和页面的组织方式，以及其他一些重要的基本语言特性。学习的过程中，我们也会使用 HTML 做一些好玩的事情。

前提：	具备计算机基础知识，安装了 基础软件环境 ，了解基本的 文件组织方法 。
学习目标：	熟悉 HTML 语言的基础知识，掌握几个 HTML 元素的基本用法。

什么是 HTML？

[HTML](#)（HyperText Markup Language，超文本标记语言）是一种用来告知浏览器如何组织页面的标记语言。HTML 可复杂、可简单，一切取决于 web 开发者。HTML 由一系列的[元素](#)组成，这些元素可以用来包围或标记不同部分的内容，使其以某种方式呈现或者工作。两端的[标签](#)可以使内容变成超链接，以连接到另一个页面；使字体表现为斜体等。例如，考虑如下内容：

My cat is very grumpy

如果我们想要将这行文字单独呈现，可以将这行文字封装成一个段落（Paragraph）[<p>](#) 元素：

HTML

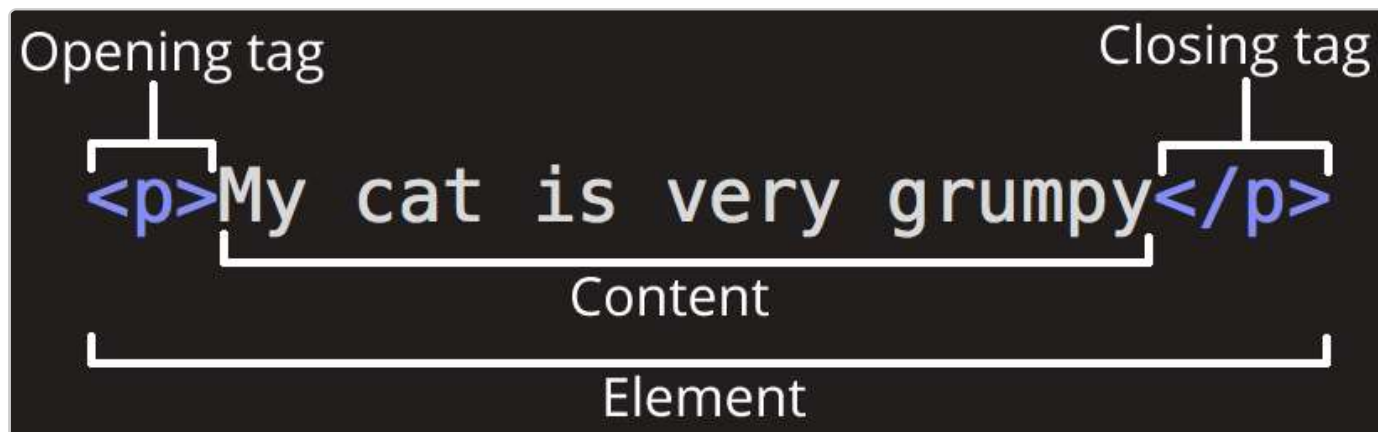
```
<p>My cat is very grumpy</p>
```

备注： HTML 标签不区分大小写。也就是说，输入标签时既可以使用大写字母也可以使用小写字母。例如，标签 [<title>](#) 可以写作 `<title>`、`<TITLE>`、`<Title>`、`<TiTle>`

等，也都可以正常工作。不过，从一致性、可读性来说，最好仅使用小写字母。

剖析一个 HTML 元素

让我们进一步探讨我们的前述的段落元素：



这个元素的主要部分有：

- **开始标签** (Opening tag)：包含元素的名称（本例为 `p`），被左、右角括号所包围。开头标签标志着元素开始或开始生效的地方。在这个示例中，它在段落文本的开始之前。
- **内容** (Content)：元素的内容，本例中就是段落的文本。
- **结束标签** (Closing tag)：与开始标签相似，只是其在元素名之前包含了一个斜杠。这标志着该元素的结束。没有包含关闭标签是一个常见的初学者错误，它可能会产生奇特的结果。

整个元素即指开始标签、内容、结束标签三部分组成的整体。

主动学习：创建第一个 HTML 元素

在下方可编辑代码区域编辑文本，使用标签 `` 和 `` 来包裹它。在前面放置 `` 以打开元素，在后面放置 `` 以关闭元素。这样编辑使得行内容变成斜体强调！你可以在实时输出区域中查看更新。

如果写错了，可随时按重置按钮重新开始，如果实在想不出来，可按显示答案按钮查看答案。

实时输出

刀枪剑戟 斧钺钩叉

可编辑代码

按 ESC 退出编辑区域，按 Tab 可插入制表符 '\t'

刀枪剑戟 斧钺钩叉

重置

显示答案

嵌套元素

你也可以把元素放到其他元素之中——这被称作**嵌套**。如果我们想要表明我们的小猫脾气**很**暴躁，可以将 `very` 一词嵌套在 `` 元素中，意味着这个单词被着重强调：

HTML

```
<p>My cat is <strong>very</strong> grumpy.</p>
```

你需要确保元素被正确的嵌套：在上面的例子中我们先打开 `p` 元素，然后才打开 `strong` 元素，因此必须先将 `strong` 元素关闭，然后再去关闭 `p` 元素。下面的例子是错误的：

HTML

```
<p>My cat is <strong>very grumpy.</p></strong>
```

所有的元素都需要正确的打开和关闭，这样才能按你所想的方式展现。由于上述示例中的那种重叠，浏览器不得不猜测你的意图。这种猜测可能会导致意想不到的结果。

块级元素和内联元素

在 HTML 中有两种你需要知道的重要元素类别，块级元素和内联元素。

- 块级元素在页面中以块的形式展现。一个块级元素出现在它前面的内容之后的新行上。任何跟在块级元素后面的内容也会出现在新的行上。块级元素通常是页面上的结构元素。例如，一个块级元素可能代表标题、段落、列表、导航菜单或页脚。一个块级元素不会嵌套在一个内联元素里面，但它可能嵌套在另一个块级元素里面。
- 内联元素通常出现在块级元素中并环绕文档内容的一小部分，而不是一整个段落或者一组内容。内联元素不会导致文本换行。它通常与文本一起使用，例如，`<a>` 元素创建一个超链接，`` 和 `` 等元素创建强调。

考虑如下示例：

HTMLPlay

```
<em>第一</em><em>第二</em><em>第三</em>

<p>第四</p>
<p>第五</p>
<p>第六</p>
```

`` 是一个内联元素，所以就像你在下方可以看到的，第一行代码中的三个元素都没有间隙的展示在了同一行。而 `<p>` 是一个块级元素，所以第二行代码中的每个 `p` 元素分别都另起了新的一行展现，并且每个段落间都有一些间隔（这是因为默认的浏览器有着展示 `<p>` 元素的默认 [CSS 样式](#)）。

Play

```
第一第二第三

第四

第五

第六
```

备注： HTML5 重新定义了元素的类别：见[元素内容分类](#) 。尽管这些新的定义更精确，但却比上述的“块级元素”和“内联元素”更难理解，因此在之后的讨论中仍使用旧的定义。

备注： 在这篇文章中提到的“块”和“内联”，不应该与 [CSS 盒子的类型](#) 中的同名术语相混淆。尽管它们默认是相关的，但改变 CSS 显示类型并不会改变元素的分类，也不会影响它可以包含和被包含于哪些元素。防止这种混淆也是 HTML5 摒弃这些术语的原因之一。

备注： 你可以查阅包含了块级元素和内联元素列表的参考页面。参见[块级元素](#)和[内联元素](#)页面。

空元素

不是所有元素都拥有开始标签、内容和结束标签。一些元素只有一个标签，通常用来在此元素所在位置插入/嵌入一些东西。这些元素被称为[空元素](#)。例如：元素 `` 是用来在页面插入一张指定的图片。

HTML

Play

```

```

显示如下：

Play



备注：HTML 中，无需在一个空元素的标签末尾添加 `/`，例如 ``。然而，这也是一种有效的语法，当你希望你的 HTML 是有效的 XML 时，这么做也没问题。

属性

元素也可以拥有属性，属性看起来像这样：

Attribute

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

属性包含元素的额外信息，这些信息不会出现在实际的内容中。在上述例子中，这个 `class` 属性是一个识别名称，以后为元素设置样式信息时更加方便。

属性必须包含：

- 一个空格，它在属性和元素名称之间。如果一个元素具有多个属性，则每个属性之间必须由空格分隔。
- 属性名称，后面跟着一个等于号。
- 一个属性值，由一对引号 ("") 引起来。

主动学习：为元素添加属性

另一个例子是关于元素 `<a>` 的——元素 `<a>` 是锚，它使被标签包裹的内容成为一个超链接。锚元素可以添加多种属性，部分如下：

href

这个属性声明超链接的 web 地址。例如 `href="https://www.mozilla.org/"`。

title

`title` 属性为超链接声明额外的信息，比如你将链接至的那个页面。例如 `title="The Mozilla homepage"`。当鼠标悬停在超链接上面时，这部分信息将以工具提示的形式显示。

target

`target` 属性用于指定链接如何呈现出来。例如，`target="_blank"` 将在新标签页中显示链接。如果你希望当前标签页显示链接，忽略这个属性即可。

编辑下面的文本框中的内容，使之变成指向任一个你喜欢的 web 地址的链接。

1. 添加 `<a>` 元素。
2. 添加 `href` 属性和 `title` 属性。
3. 指定 `target` 属性，使得点击链接时在新标签页打开。

你可以在实时输出区域看到你修改的内容。你应该可以看到一个链接，当鼠标移上此链接时会显示 `title` 属性值，当点击此链接时会跳转到 `href` 属性指定的 web 地址。记住：在元素名和属性名之间以及两个属性之间要有一个空格。

如果写错了，可随时按重置按钮重新开始，如果实在想不出来，可按显示答案按钮查看答案。

实时输出

A link to my favorite website.

可编辑代码

按 ESC 退出编辑区域，按 Tab 可插入制表符 '\t'

```
<p>A link to my favorite website.</p>
```

重置

显示答案

布尔属性

有时你会看到没有值的属性，这也是完全可以接受的。这些属性被称为布尔属性。布尔属性只能有一个值，这个值一般与属性名称相同。例如，考虑 [disabled](#) 属性，你可以将其分配给表单输入元素。用它来禁用表单输入元素，这样用户就不能输入了。被禁用的元素通常有一个灰色的外观。示例如下：

HTML

Play

```
<input type="text" disabled="disabled" />
```

方便起见，我们完全可以将其写成以下形式：

HTML

Play

```
<!-- 使用 disabled 属性来防止终端用户输入文本到输入框中 -->
```

```
<input type="text" disabled />
```

```
<!-- 下面这个输入框不包含 disabled 属性，所以用户可以向其中输入 -->
```

```
<input type="text" />
```


作为参考，上面的例子还包括一个非禁用的表单输入元素。上面两段 HTML 代码产生的效果如下：

HTMLPlay

省略包围属性值的引号

如果你看了很多其他网站的代码，你可能会遇到一些奇怪的标记风格，包括没有引号的属性值。在某些情况下它是被允许的，但是其他情况下会破坏你的标记。例如，针对之前的链接示例，我们可以像这样写一个只拥有一个 href 属性的版本：

HTMLPlay

```
<a href=https://www.mozilla.org/>favorite website</a>
```

然而，当我们再添加一个 title 属性时，就会出现问题：

HTMLPlay

```
<a href=https://www.mozilla.org/ title=The Mozilla homepage>favorite website</a>
```

此时浏览器会误解你的标记，它会把 title 属性理解为三个属性——title 的属性值为 The，另外还有两个布尔属性 Mozilla 和 homepage，很明显不是我们所期望的！并且在这个编码里面它会报错或者出现异常行为。试一试把鼠标移动到链接上，看会显示什么 title 文字！

Play

[favorite website favorite website](#)

我们建议始终添加引号——这样可以避免很多问题，并且使代码更易读。

使用单引号还是双引号？

在到目前为止，本章内容所有的属性都是由双引号来包裹。然而，你也许在一些 HTML 中也见过单引号。这只是风格的问题，你可以从中选择一个你喜欢的。以下两种情况都可以：

HTML

```
<a href='https://www.example.com'>示例站点链接</a>
```

```
<a href="https://www.example.com">示例站点链接</a>
```

但应该注意单引号和双引号不能在一个属性值里面混用。下面的语法是错误的：

HTML

```
<a href="https://www.example.com">示例站点链接</a>
```

在一个 HTML 中已使用一种引号，你可以在此引号中嵌套另外一种引号：

HTML

```
<a href="https://www.example.com" title="你觉得'好玩吗'? ">示例站点链接</a>
```

如果你想将英文引号（单引号或双引号）当作文本显示在 html 中，你就必须使用 [HTML 实体引用](#)。像这样的代码就会破坏显示：

HTML

```
<a href='https://www.example.com' title='Isn't this fun?'>示例站点链接</a>
```

要这样做：

HTML

```
<a href="https://www.example.com" title="Isn't this fun?">示例站点链接</a>
```

剖析 HTML 文档

单独的 HTML 元素本身并不十分有用。接下来，我们来看看单个元素是如何组合成整个 HTML 页面的：

HTML

```
<!doctype html>
<html lang="zh-CN">
  <head>
    <meta charset="utf-8" />
    <title>我的测试站点</title>
  </head>
  <body>
    <p>这是我的页面</p>
  </body>
</html>
```

这里我们有：

1. `<!DOCTYPE html>`：声明文档类型。早期的 HTML（大约 1991-1992 年）文档类型声明类似于链接，规定了 HTML 页面必须遵从的良好规则，能自动检测错误和其他有用的东西。文档类型使用类似于这样：

HTML

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

文档类型是一个历史遗留问题，需要包含它才能使其他东西正常工作。现在，只需要知道

`<!DOCTYPE html>` 是最短的有效文档声明！

2. `<html></html>`： [<html>](#) 元素。这个元素包裹了页面中所有的内容，有时被称为根元素。
3. `<head></head>`： [<head>](#) 元素。这个元素是一个容器，它包含了所有你想包含在 HTML 页面中但**不在 HTML 页面中显示**的内容。这些内容包括你想在搜索结果中出现的关键字和页面描述、CSS 样式、字符集声明等等。以后的章节中会学到更多相关的内容。
4. `<meta charset="utf-8">`： [<meta>](#) 元素。这个元素代表了不能由其他 HTML 元相关元素表示的元数据，比如 [<base>](#)、[<link>](#)、[<script>](#)、[<style>](#) 或 [<title>](#)。 [charset](#) 属性将你的文档的字符集设置为 UTF-8，其中包括绝大多数人类书面语言的大多数字符。有了这个设置，页面现在可以处理它可能包含的任何文本内容。没有理由不对它进行设置，它可以帮助避免以后的一些问题。
5. `<title></title>`： [<title>](#) 元素。这设置了页面的标题，也就是出现在该页面加载的浏览器标签中的内容。当页面被加入书签时，页面标题也被用来描述该页面。

6. `<body></body>` : [<body>](#) 元素。包含了你访问页面时所有显示在页面上的内容，包含文本、图片、视频、游戏、可播放音频轨道等等。

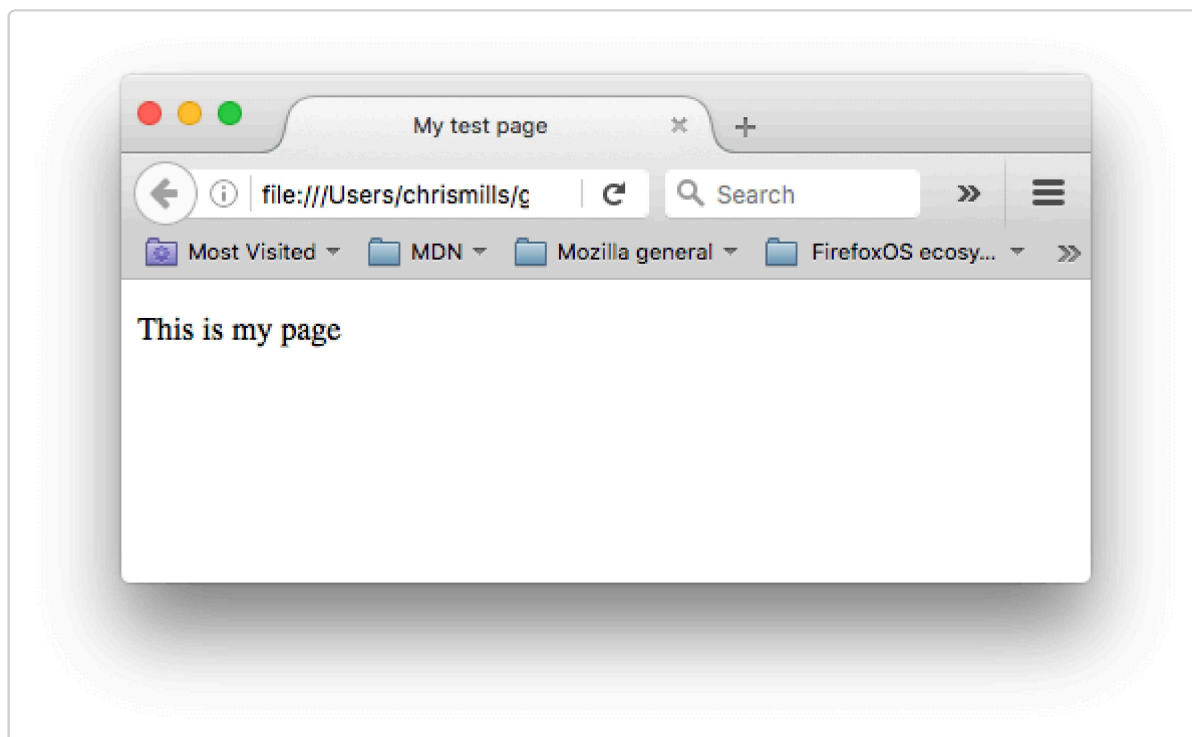
主动学习：为 HTML 文档添加一些特征

如果你想在你的本地练习写一些 HTML 页面，你可以这样做：

1. 复制上面的 HTML 页面例子。
2. 在文字编辑器创建一个新文件。
3. 粘贴代码到这个文件。
4. 保存为 `index.html`。

备注： 可在 [MDN 学习区代码仓库](#) 上查看该示例。

你可以打开浏览器看看这段代码的效果是什么样的，然后改变代码刷新浏览器看看新的结果。最开始的代码是这样的效果：



在这个练习中，你可以在电脑上本地编辑代码，如前所述，也可以下面的样本窗口中编辑（在这种情况下，可编辑的样本窗口只代表 [<body>](#) 元素的内容）。通过执行以下任务来磨练你的技能：

- 就在 `<body>` 元素开始标签下方，为这个文档添加一个主标题。这个主标题应该被包含在 `<h1>` 开始标签和 `</h1>` 结束标签之间。
- 编辑这个段落以包含一些你感兴趣的文本。
- 把重要的字词包含在开始标记 `` 和结束标记 `` 之间可以使他们以粗体显示。
- 在你的文档中添加一个超文本链接，[如前所述](#)。
- 在段落下方向你的文档添加一张图片，[如前所述](#)。如果你尝试对不同的图片（在你的本地电脑或是在 Web 的其他位置上）添加链接，那你就更棒了。

如果写错了，可随时按[重置](#)按钮重新开始，如果实在想不出来，可按[显示答案](#)按钮查看答案。

Play

实时输出

相思无用，惟别而已。别期若有定，千般煎熬又何如？莫道黯然销魂，何处柳暗花明？

可编辑代码

按 ESC 退出编辑区域，按 Tab 可插入制表符'\t'

<p>相思无用，惟别而已。别期若有定，千般煎熬又何如？莫道黯然销魂，何处柳暗花明？</p>

重置

显示答案

HTML 中的空白

在上面的例子中，你可能已经注意到了在代码中包含了很多的空格——这是没有必要的；下面的两个代码片段是等价的：

HTML

<p>狗 狗 很 呆 萌。</p>

<p>狗 狗 很
 呆 萌。</p>

无论你在 HTML 元素的内容中使用多少空格（包括一个或多个空白字符或换行），当渲染这些代码的时候，HTML 解释器会将连续出现的空白字符减少为一个单独的空格符。

如果你的代码有很好的格式化，就会更容易理解你的代码中发生了什么。在我们的 HTML 中，每个嵌套元素都比它所在的元素多缩进了两个空格。你可以选择格式化的风格（例如，每一级缩进多少个空格），但你至少应该考虑格式化。

实体引用：在 HTML 中包含特殊字符

在 HTML 中，字符 <、>、"、' 和 & 是特殊字符。它们是 HTML 语法自身的一部分，那么你将如何将它们包含进你的文本中呢，比如说如果你真的想要在文本中使用符号 & 或者小于号，而不想让它们被浏览器视为代码并被解释？

我们必须使用字符引用——表示字符的特殊编码，它们可以在那些情况下使用。每个字符引用以符号 & 开始，以分号 (;) 结束。

原义字符	等价字符引用
<	<
>	>
"	"

原义字符	等价字符引用
'	'
&	&

等价字符引用可以很容易记住，因为它使用的文本可以被看作是小于“<”，引号是“"”，其他的也是如此。要找到更多关于实体引用的信息，请参见 [XML 和 HTML 字符实体引用列表](#)（维基百科）。

在下面的示例中，有两个自然段：

HTMLPlay

```
<p>HTML 中用 <p> 来定义段落元素。</p>
```

```
<p>HTML 中用 &lt;p> 来定义段落元素</p>
```

在下面的实时输出中，你会看到第一段是错误的，因为浏览器会认为第二个 `<p>` 是开始一个新的段落！第二段是正确的，因为我们用字符引用来代替了角括号（`<` 和 `>` 符号）。

Play

```
HTML 中用
来定义段落元素。
HTML 中用 <p> 来定义段落元素
```


HTML 拥有在代码中写注释的机制。浏览器会忽略注释，有效地使注释对用户来说不可见。注释的目的是让你在代码中加入注释，以解释你的逻辑或编码。如果你在离开很久后回到一个代码库，以至于你不能完全记住它，这就非常有用。同样，当不同的人在进行修改和更新时，注释也是非常宝贵的。

为了将一段 HTML 中的内容置为注释，你需要将其用特殊的记号 `<!--` 和 `-->` 包裹起来，比如：

HTMLPlay

```
<p>我在注释外！</p>

<!-- <p>我在注释内！</p> -->
```

正如你下面所见的那样，第一段出现在了实时输出中，但是第二段却没有。

Play

我在注释外！

总结

你已经来到了这篇文章的结尾——希望你享受基础的 HTML 学习的旅程。

在这里你应该可以理解 HTML 语言的全貌和基本的工作原理。你应该还学会了一些元素和属性的使用。在这个模块的后续文章中，我们会深入一些你已经见过的东西的细节，并且展示语言的一些其他概念。

备注： 当你开始学习更多的 HTML 知识时，可能也想了解一些层叠样式列表 ([CSS](#)) 的基础知识。CSS 是一种用来设计网页样式的语言（比如，用它改变字体、颜色或页面布局等）。你很快就会发现，HTML 和 CSS 能很好地协调配合。

参见

- [使用 CSS 为 HTML 元素应用颜色](#)

Help improve MDN

Was this page helpful to you?

[Learn how to contribute.](#)



This page was last modified on 2023年9月19日 by [MDN contributors](#).