

前端学习web-learn

[html-learn](#)

[简介](#)

[元素](#)

[属性](#)

[标题和段落](#)

[文本格式化](#)

[链接](#)

[头部](#)

[HTML <head> 元素](#)

[HTML <title> 元素](#)

[HTML <meta> 元素](#)

[如何使用CSS](#)

[表单](#)

[HTML 表单和输入](#)

[颜色名](#)

[HTML 颜色名](#)

目前所有浏览器都支持以下颜色名。

[CSS](#)

[语法](#)

[id和class选择器](#)

[插入样式表的方式](#)

[如何插入样式表](#)

html、css、js分别对应布局、样式和动作，其实换做现实中的实物，也可以这样来定义，例如手机，它的布局就是长方形的不太厚的，样式的不同造就了不同得到手机外观，例如看到苹果手机的设计就知道这个品牌，而动作也就是功能，手机的功能有哪些呢。

关联电脑文件夹： `D:\08projects\frontend\01-web-learn`

学习地址：

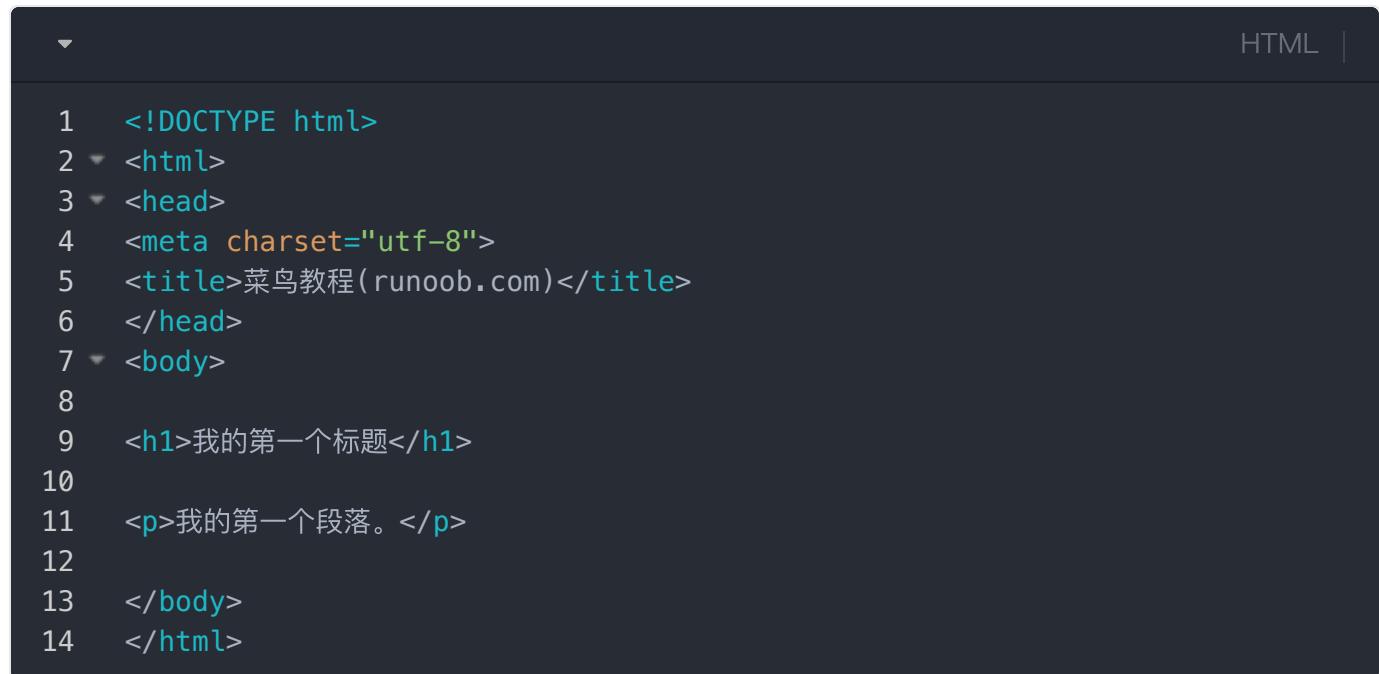
html-learn

<!doctype> 声明必须位于 HTML5 文档中的第一行,使用非常简单:

<!DOCTYPE html>

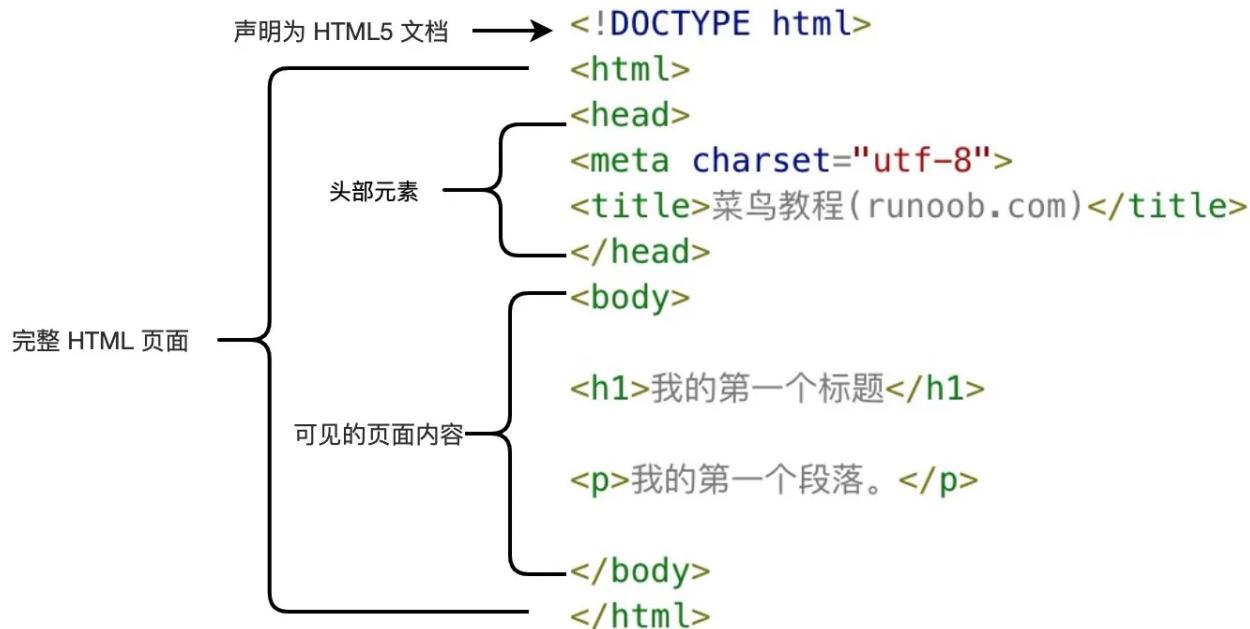
对于中文网页需要使用 <meta charset="utf-8"> 声明编码, 否则会出现乱码。

简介



A screenshot of a code editor window titled "HTML". The code editor displays an HTML document with line numbers on the left. The code includes a doctype declaration, an HTML element, a head element containing a meta tag for UTF-8 encoding, a title element with the text "菜鸟教程(runoob.com)", a body element containing an H1 element with the text "我的第一个标题" and a P element with the text "我的第一个段落。".

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <meta charset="utf-8">
5  <title>菜鸟教程(runoob.com)</title>
6  </head>
7  <body>
8
9  <h1>我的第一个标题</h1>
10
11 <p>我的第一个段落。</p>
12
13 </body>
14 </html>
```



只有 <body> 区域（白色部分）才会在浏览器中显示。

元素

推荐使用小写的html标签

属性

下列是html中的大多数属性

属性名	适用元素	说明
<code>id</code>	所有元素	为元素指定唯一的标识符。
<code>class</code>	所有元素	为元素指定一个或多个类名，用于 CSS 或 JavaScript 选择。
<code>style</code>	所有元素	直接在元素上应用 CSS 样式。
<code>title</code>	所有元素	为元素提供额外的提示信息，通常在鼠标悬停时显示。

<code>data-*</code>	所有元素	用于存储自定义数据，通常通过 JavaScript 访问。
<code>href</code>	<code><a></code> , <code><link></code>	指定链接的目标 URL。
<code>src</code>	<code></code> , <code><script></code> , <code><iframe></code>	指定外部资源（如图片、脚本、框架）的 URL。
<code>alt</code>	<code></code>	为图像提供替代文本，当图像无法显示时显示。
<code>type</code>	<code><input></code> , <code><button></code>	指定输入控件的类型（如 <code>text</code> , <code>password</code> , <code>checkbox</code> 等）。
<code>value</code>	<code><input></code> , <code><button></code> , <code><option></code>	指定元素的初始值。
<code>disabled</code>	表单元素	禁用元素，使其不可交互。
<code>checked</code>	<code><input type="checkbox"></code> , <code><input type="radio"></code>	指定复选框或单选按钮是否被选中。
<code>placeholder</code>	<code><input></code> , <code><textarea></code>	在输入框中显示提示文本。
<code>target</code>	<code><a></code> , <code><form></code>	指定链接或表单提交的目标窗口或框架（如 <code>_blank</code> 表示新标签页）。
<code>readonly</code>	表单元素	使输入框只读。
<code>required</code>	表单元素	指定输入字段为必填项。

<code>autoplay</code>	<code><audio></code> , <code><video></code>	自动播放媒体。
<code>onclick</code>	所有元素	当用户点击元素时触发 JavaScript 事件。
<code>onmouseover</code>	所有元素	当用户将鼠标悬停在元素上时触发 JavaScript 事件。
<code>onchange</code>	表单元素	当元素的值发生变化时触发 JavaScript 事件。

标题和段落

标题 (Heading) 是通过 `<h1>` – `<h6>` 标签进行定义的。

`<h1>` 定义最大的标题。 `<h6>` 定义最小的标题。

段落是通过 `<p>` 标签定义的。

文本格式化

<code></code>	定义粗体文本
<code></code>	定义着重文字
<code><i></code>	定义斜体字
<code><small></code>	定义小号字
<code></code>	定义加重语气
<code><sub></code>	定义下标字
<code><sup></code>	定义上标字
<code><ins></code>	定义插入字
<code></code>	定义删除字

链接

常用的空链接

方法	作用	是否会跳转	场景适用性
<code>href="#"</code>	导航到页面顶部	是	占位符, 捕获点击事件
<code>href="javascript:void(0)"</code>	阻止默认行为, 不刷新页面	否	阻止跳转, 配合 JS 使用
<code>href=""</code>	刷新当前页面	是	需要页面刷新时
<code>href="about:blank"</code>	打开空白页面	是	新窗口打开空白页面
<code>role="button"</code>	链接表现为按钮, 无默认行为	否	配合 JS 实现按钮功能, 无跳转

头部

HTML `<head>` 元素

`<head>` 元素包含了所有的头部标签元素。在 `<head>` 元素中你可以插入脚本 (scripts) , 样式文件 (CSS) , 及各种meta信息。

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

HTML `<title>` 元素

`<title>` 标签定义了不同文档的标题。

`<title>` 在 HTML/XHTML 文档中是必需的。

`<title>` 元素:

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

HTML <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。

元数据可以使用于浏览器（如何显示内容或重新加载页面），搜索引擎（关键词），或其他Web服务。

如何使用CSS

CSS 是在 HTML 4 开始使用的,是为了更好的渲染HTML元素而引入的.

CSS 可以通过以下方式添加到HTML中:

- 内联样式 – 在HTML元素中使用"style" 属性
- 内部样式表 –在HTML文档头部 <head> 区域使用<style> 元素 来包含CSS
- 外部引用 – 使用外部 CSS 文件

最好的方式是通过外部引用CSS文件.

在本站的HTML教程中我们使用了内联CSS样式来介绍实例，这是为了简化的例子，也使得你能更容易在线编辑代码并在线运行实例。

你可以通过本站的 [CSS 教程](#) 学习更多的 CSS 知识。

表单

HTML 表单和输入

HTML 表单用于收集用户的输入信息。

HTML 表单表示文档中的一个区域，此区域包含交互控件，将用户收集到的信息发送到 Web 服务器。

HTML 表单通常包含各种输入字段、复选框、单选按钮、下拉列表等元素。

以下是一个简单的HTML表单的例子：

- `<form>` 元素用于创建表单，`action` 属性定义了表单数据提交的目标 URL，`method` 属性定义了提交数据的 HTTP 方法（这里使用的是 "post"）。
- `<label>` 元素用于为表单元素添加标签，提高可访问性。
- `<input>` 元素是最常用的表单元素之一，它可以创建文本输入框、密码框、单选按钮、复选框等。`type` 属性定义了输入框的类型，`id` 属性用于关联 `<label>` 元素，`name` 属性用于标识表单字段。
- `<select>` 元素用于创建下拉列表，而 `<option>` 元素用于定义下拉列表中的选项。

颜色名

HTML 颜色名

目前所有浏览器都支持以下颜色名。

141个颜色名称是在HTML和CSS颜色规范定义的（17标准颜色，再加124）。下表列出了所有颜色的值，包括十六进制值。

 提示：17标准颜色：黑色，蓝色，水，紫红色，灰色，绿色，石灰，栗色，海军，橄榄，橙，紫，红，白，银，蓝绿色，黄色。点击其中一个颜色名称（或一个十六进制值）就可以查看与不同文字颜色搭配的背景颜色。

CSS

语法

CSS 规则由两个主要的部分构成：选择器，以及一条或多条声明：



选择器通常是您需要改变样式的 HTML 元素。

每条声明由一个属性和一个值组成。

属性 (property) 是您希望设置的样式属性 (style attribute) 。每个属性有一个值。属性和值被冒号分开。

CSS 声明总是以分号 ; 结束，声明总以大括号 {} 括起来：

```
p {color:red;text-align:center;}
```

注释是用来解释你的代码，并且可以随意编辑它，浏览器会忽略它。

CSS 注释以 /* 开始，以 */ 结束，实例如下：

id 和 class 选择器

 ID 属性不要以数字开头，数字开头的 ID 在 Mozilla/Firefox 浏览器中不起作用。

class 属性

class 选择器用于描述一组元素的样式，class 选择器有别于 id 选择器，class 可以在多个元素中使用。

class 选择器在 HTML 中以 class 属性表示，在 CSS 中，类选择器以一个点 . 显示：

在以下的例子中，所有拥有 center 类的 HTML 元素均为居中。多个 class 选择器可以使用空格分开

插入样式表的方式

如何插入样式表

插入样式表的方法有三种：

- 外部样式表(External style sheet)

- 内部样式表(Internal style sheet)
- 内联样式(Inline style)

当样式需要应用于很多页面时，外部样式表将是理想的选择。在使用外部样式表的情况下，你可以通过改变一个文件来改变整个站点的外观。每个页面使用 `<link>` 标签链接到样式表。`<link>` 标签在（文档的）头部：

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

浏览器会从文件 `mystyle.css` 中读到样式声明，并根据它来格式文档。

外部样式表可以在任何文本编辑器中进行编辑。文件不能包含任何的 `html` 标签。样式表应该以 `.css` 扩展名进行保存。下面是一个样式表文件的例子：

内部样式表

当单个文档需要特殊的样式时，就应该使用内部样式表。你可以使用 `<style>` 标签在文档头部定义内部样式表，就像这样：

内联样式

由于要将表现和内容混杂在一起，内联样式会损失掉样式表的许多优势。请慎用这种方法，例如当样式仅需要在一个元素上应用一次时。

要使用内联样式，你需要在相关的标签内使用样式（style）属性。Style 属性可以包含任何 CSS 属性。本例展示如何改变段落的颜色和左外边距：

```
<p style="color:sienna; margin-left:20px">这是一个段落。</p>
```

多重样式优先级

样式表允许以多种方式规定样式信息。样式可以规定在单个的 HTML 元素中，在 HTML 页的头元素中，或在一个外部的 CSS 文件中。甚至可以在同一个 HTML 文档内部引用多个外部样式表。

一般情况下，优先级如下：

（内联样式） **Inline style** > （内部样式） **Internal style sheet** > （外部样式） **External style sheet** > 浏览器默认样式