

HTML学习笔记

肖桐 PB18000037

本文摘抄自[菜鸟教程](#)

一、HTML 标签

HTML 标签是由尖括号 `<` 和 `>` 包围的关键词，通常是成对出现的，如 `` 和 ``，分别称为开始标签和结束标签。HTML 标签对大小写不敏感。

[HTML 标签参考手册](#)

1. 常用标签

1. `<!DOCTYPE HTML>`

声明为 HTML5 文档。有助于浏览器中正确显示网页。`doctype` 声明不区分大小写。

2. `<html>`

`<html>` 元素是 HTML 页面的根元素。

Q: 什么是根元素?

A:

3. `<head>`

[HTML 头部元素](#)

元素包含了文档的元数据(metadata)。

如 `<meta charset="utf-8">` 定义网页编码格式为 `utf-8`。因为在大部分浏览器中，直接输出中文会出现中文乱码，这时需要在头部 `<head>` 内将字符声明为 `utf-8` 或者 `gbk`。

Q: 还有哪些元数据，有什么作用?

A:

4. `<title>`

`<title>` 元素描述了浏览器工具栏的标题，即页面的标题。

5. `<base>`

6. `<link>`

`<link>` 标签定义了文档与外部资源之间的关系。同时能够使用外部样式文件来格式化文本

```

1 <link rel="shortcut icon" href="D:\*\...\*.jpg">    <!-- 给浏览器工具栏上加图片 -
->
2
3 <link rel="stylesheet" type="text/css" href="styles.css">    <!-- 这里使用了
"styles.css" 文件来格式化当前的 html 文档 -->

```

7. <style>

在 <style> 元素中可以直接添加样式来渲染 HTML 文档，如：

```

1 <head>
2 <style type="text/css">
3 body {background-color:yellowgreen} <!-- 将整个文档背景色变为黄绿色，或者直接
rgb(..., ..., ...) 决定颜色 -->
4 p {color:blue} <!-- 将段落文字颜色变为蓝色 -->
5     <!-- 类似的也会有：h1 {color:red} 等等 -->
6 </style>
7 </head>

```

8. <body>

<body> 元素包含了可见的页面内容。只有 <body> 区域中的内容才会在浏览器中显示。

Q：如何使用？

A：

9. <h1>

<h1> 元素定义一个大标题，类似于 markdown 中的 #。

同理还有 <h2>, <h3>, ... <h6>

10. <p>

<p> 元素定义一个段落。

如果希望在不产生一个新段落的情况下进行换行，可使用
 标签(空元素)。如：

```

1 <p>
2     这个<br/>
3     段落<br/>
4     演示了分行的效果
5 </p>

```

11. <hr>

<hr> 标签在 HTML 页面中创建水平线。

```

1 <p>这是一个段落</p>
2 <hr/>
3 <p>这是一个段落</p>
4 <hr/>

```

12. <a>

<a> 能够引出一个链接。用法: `链接本文`

如 `这是一个链接`。显而易见, 需要在 href 属性中指定链接的地址。

- target 属性

使用 target 属性, 可以定义被链接的文档在何处显示。如:

```
1 <a href="https://www.ruboon.com" target="_blank">菜鸟教程</a>
```

就会在一个新的窗口打开文档, 若没有 target 属性, 则会在原来的窗口打开文档(即覆盖)。

- id 属性

id 属性可用于创建在一个 HTML 文档书签标记。但这在 HTML 文档中是不显示的, 所以对于读者来说是隐藏的。

Q: 所以怎么用, 有什么用?

A: 可以用来使用页内链接, 如:

```
1 <a href="#c1">查看章节 1</a>
2 <h2><a id="c1">章节 1</a></h2>
```

第一行就可以在页内链接到第二行。

同时 id 属性相当于给对应的元素赋了名称, 可以使用这个"名称"做到很多指定性的设置(如在 CSS 中的设置)。

- style 属性

style 属性用于控制链接的样式, 比如:

```
1 <a href="https://www.ruboon.com/" style="text-decoration:none;">访问
  runoob.com!</a>  <!-- 可以去掉链接的下划线 -->
```

- title 属性

title 属性可以让鼠标悬停在超链接上的时候, 显示该超链接的文字注释。

13.

HTML 图像

 标签用于引入图片。

 有许多重要属性,

- src 属性, 用于确定图片 url
- alt 属性, 用来为图像定义一串可预备的可替换文本。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。
- <height> 属性, 用于设置图像的高度
- <width> 属性, 用于设置图像的宽度

如:

```
1  <!-- -->
2
3 
```

可见 `` 没有结束标签，图片的引用在开始标签内即可完成。

图片链接：将图片载入到 `<a>` 中间即可。如：

```
1 <a href="//www.runoob.com/html/html-tutorial.html">
2 </a>
```

图像映射：创建带有可供点击区域的图像地图，其中的每个区域都是一个超链接。[图像映射](#)

Q：如何使用 `` 导入？

A：使用 `` 的属性 `src` 实现图片导入。

14. `<table>`

[HTML表格](#)

`<table>` 用于创建 HTML 表格。每个表格有若干行，由标签 `<tr>`、`</tr>` 定义；每行被分割为若干个单元格，由标签 `<td>`、`</td>` 定义。

表格的表头使用 `<th>` 标签定义，大多数浏览器会把表头显示为粗体居中的文本。

`<caption>` 标签为表格起个标题名。

如：

```
1 <table border="1"> <!-- border属性设置边框，border="0" 则无边框 -->
2   <caption>.....</caption> <!-- 作为表格的标题 -->
3   <tr>
4     <th>Header 1</th> <!-- 表头 -->
5     <th>Header 2</th>
6   </tr>
7   <tr>
8     <td>row 1, cell 1</td>
9     <td>row 1, cell 2</td>
10  </tr>
11  <tr>
12    <td>row 2, cell 1</td>
13    <td>row 2, cell 2</td>
14  </tr>
15 </table>
```

还有跨行和跨列的表格单元格的制作，与 Latex 有些相似，利用 `<th>`、`<tr>`、`<td>` 属性 `colspan`（列合并）和 `rowspan`（行合并）实现。如：

```
1 <td colspan="2">学生信息表</td> <!-- 列合并，合并2列 -->
```

更多骚操作详见上面的菜鸟教程链接。

`<table>` 还可以用来创建 HTML 布局，骚操作详见[HTML布局](#)

15.

 标签用于定义 HTML 无序列表，此列表项目使用粗体圆点进行标记。在 中使用 标签创建新的列表项。比如：

```
1 <ul>
2     <li>Coffee</li>
3     <li>Milk</li>
4 </ul>
```

16.

 标签用于定义 HTML 有序列表，此列表项目使用数字进行标记。在 中使用 标签创建新的列表项。比如：

```
1 <ol>
2     <li>Coffee</li>
3     <li>Milk</li>
4 </ol>
```

17. <div>

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。

使用范例如下：

```
1 <body>
2
3 <div id="container" style="width:500px">
4
5 <div id="header" style="background-color:#FFA500;">
6 <h1 style="margin-bottom:0;">主要的网页标题</h1></div>
7
8 <div id="menu" style="background-
9 color:#FFD700;height:200px;width:100px;float:left;">
10 <b>菜单</b><br>
11 HTML<br>
12 CSS<br>
13 JavaScript</div>
14
15 <div id="content" style="background-
16 color:#EEEEEE;height:200px;width:400px;float:left;">
17 内容在这里</div>
18
19 <div id="footer" style="background-color:#FFA500;clear:both;text-
20 align:center;">
21 版权 © runoob.com</div>
22 </div>
23 </body>
```

18. ``

HTML `` 元素是内联元素，可用作文本的容器。

`` 元素也没有特定的含义。

当与 CSS 一同使用时，`` 元素可用于为部分文本设置样式属性。

19. `<!-- -->`

HTML 注释，如：`<!-- 这是一个注释 -->`

2. 格式化标签

[HTML 文本格式化](#)

1. ``

`` 是将文本加粗的格式化标签，如：`加粗文本`

2. `<i>`

`<i>` 是将文本变为斜体的标签，如：`<i>斜体文本</i>`

3. `<sub>`

`<sub>` 定义下标字

4. `<sup>`

`<sup>` 定义上标字

5. ``

`` 定义删除字

二、HTML 元素

1. HTML 空元素

没有内容的 HTML 元素被称为**空元素**。空元素是在开始标签中关闭的。

`
` 就是没有关闭标签的空元素（`
` 标签定义换行）。

在 XHTML、XML 以及未来版本的 HTML 中，所有元素都必须被关闭。在开始标签中添加斜杠，比如 `
`，是关闭空元素的正确方法

三、HTML 属性

- HTML 元素可以设置属性
- 属性可以在元素中添加附加信息
- 属性一般描述于**开始标签**
- 属性总是以**键值对**的形式出现，如：`name="value"`，也能使用单引号

HTML 属性对大小写也不敏感，但是推荐使用小写属性。

四、页面布局

页面元素与布局是 HTML 最重要的基础和难点，它考验了各种技术的综合运用能力。

页面元素与布局的**核心技巧**：

1. HTML 非常强调代码的**语义化**。比如过去常常用表格来实现页面布局，这在现在看来是很糟糕的语义实现。而现在，在做 HTML 页面布局时，应首先考虑代码的语义化，尽量使代码中不包含冗余的 DOM 结构，在此基础上尽可能使用 CSS 在完成页面布局。
2. 绝大部分的页面布局都是浮动 float、定位 position 和内外边框 margin & padding 三者的有机结合。
3. 要对布局有更加深入的理解，还需要掌握各种元素的呈现方式 (display) 及其与整个页面空间的关系，这也就是所谓的"文档流"。

1. 文档流

摘自[CSDN](#)

所谓的文档流 (normal flow, 也被称为“普通流”)，指的是就是元素排版布局过程中，元素会自动从左往右，从上往下地遵守这种流式排列方式。

2. 脱离文档流的三种方式

摘自[CSDN](#)

(1). float

使用 float 脱离文档流时，虽然其他盒子会无视这个元素，但其他盒子内的文本依然会在这个元素让出位置，环绕在该元素的周围。

(2). position:absolute

使用 absolute 脱离文档流后的元素，是相对于该元素的父类（及以上，如果直系父类元素不满足条件则继续向上查询）元素进行定位的，并且这个父类元素的 position 必须是非 static 定位的（static 是默认定位方式）。

(3). position:fixed

完全脱离文档流，相对于浏览器窗口进行定位，也就是这个 div 固定在浏览器窗口上了，不论我们怎么拖动滚动条都无法改变它在浏览器窗口的位置。

3. 相对定位与绝对定位

(1). 相对定位 position:relative

相对于原来位置移动，元素设置此属性之后仍然处在文档流中，不影响其他元素的布局。

(2). 绝对定位 `position: absolute`

元素会脱离文档流，如果设置偏移量，会影响其他元素的位置定位。

在父元素没有设置相对定位或绝对定位的情况下，元素相对于根元素定位(即 `HTML` 元素)(是父元素没有)。

父元素设置了相对定位或绝对定位，元素会相对于离自己最近的设置了相对或绝对定位的父元素进行定位(或者说离自己最近的不是 `static` 的父元素进行定位，因为元素默认是 `static`)。

五、注意事项

1. 当显示页面时，浏览器会移除源代码中**多余的空格和空行**。所有连续的空格或空行都会被算作一个空格。需要注意的是，HTML 代码中的所有连续的空行（换行）也被**显示为一个空格**。