



# 守卫者安全

信息安全爱好者

我的 HTML 学习笔记

## <!DOCTYPE> 声明

<!DOCTYPE>声明有助于浏览器中正确显示网页。

网络上有很多不同的文件，如果能够正确声明 HTML 的版本，浏览器就能正确显示网页内容。

doctype 声明是不区分大小写的，以下方式均可：

```
<!DOCTYPE html>

<!DOCTYPE HTML>

<!doctype html>

<!Doctype Html>
```

目前在大部分浏览器中，直接输出中文会出现中文乱码的情况，这时候我们就需要在头部将字符声明为 UTF-8 或 GBK。

```
<head>
<meta charset="UTF-8">
<title>页面标题</title>
</head>
```

## HTML 标题

HTML 标题（Heading）是通过<h1> - <h6> 标签来定义的，<h1> 定义最大的标题。<h6> 定义最小的标题。

```
1  <! doctype html>
2
3  <html>
4  <body>
5      <meta charset=utf-8></meta>
6      <h1>111</h1>
7      <h2>111</h2>
8      <h3>111</h3>
9      <h4>111</h4>
10     <h5>111</h5>
11     <h6>111</h6>
12     <h0>111</h0>
13     <h7>111</h7>
14     <h100>111</h100>
15 </body>
16 </html>
```

111

111

111

111

111

111

111 111 111

按照显示我们可以发现，<h1> - <h6>的标题都是自成一行的，且字体大小逐渐减小。但是最后面的 0、7、100 显示在了一行（也就是说这不属于标题），且字体大小都是最小号的。

注意：浏览器会自动地在标题的前后添加空行。

## HTML 段落

HTML 段落是通过标签 <p> 来定义的。

```
1 <!doctype html>
2
3 <html>
4 <body>
5 <meta charset=utf-8></meta>
6 <p>这是一个段落。</p>
7 <p>这是另外一个段落。</p>
8 </body>
9 </html>
```

这是一个段落。

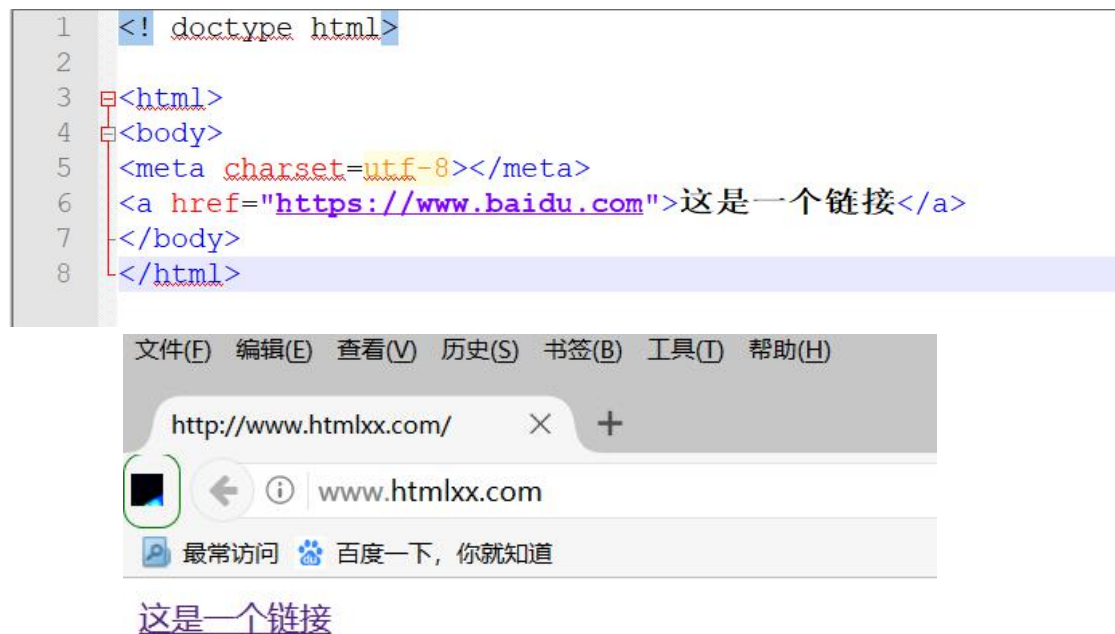
这是另外一个段落。

可以看到段落都是自成一行的。

注意：浏览器会自动地在段落的前后添加空行。

## HTML 链接

HTML 链接是通过标签 `<a>` 来定义的。

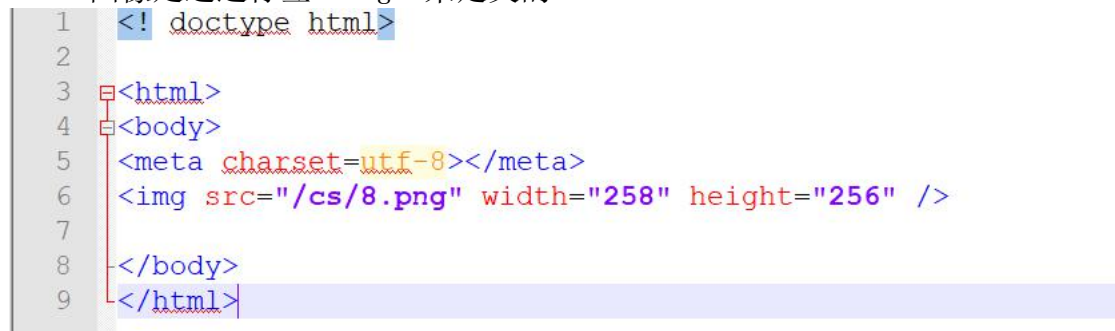


`<a></a>` 标签中间的内容就是显示在页面上的内容，`href` 属性定义了这个内容指向的链接。点击页面的文字就可以跳转到百度首页了。

**提示:**在 `href` 属性中指定链接的地址。

## HTML 图像（图片）

HTML 图像是通过标签 `<img>` 来定义的。





**注意：**图像的名称和尺寸是以属性的形式提供的。src 指明了图像的地址。

### 常用元素：

<br> 换行

<body> 定义了 HTML 文档的主体

<html> 定义了整个 HTML 文档

<title> 定义了网页文档的标题（网页显示名称）

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。<br> 就是没有关闭标签的空元素。

HTML 标签对大小写不敏感，但是推荐使用小写标签。

### 小常识：

在标题和段落等标签中输入内容，无论中间间隔了多少个空格，浏览器只会把它们缩为一个空格显示。如下

```
1  <!doctype html>
2
3  <html>
4  <body>
5    <meta charset=utf-8></meta>
6    <h1>标题</h1>
7    <hr/>
8    <p>测试一下    空格</p>
9
10   </body>
11  </html>
```

## 标题

---

测试一下 空格

## 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

属性总是以名称/值对的形式出现，比如：name="value"。

比如之前的例子，HTML 链接由 <a> 标签（元素）定义。链接的地址在 href 属性中指定。

```
1  <!doctype html>
2
3  <html>
4  <body>
5    <meta charset=utf-8></meta>
6    <a href="https://www.baidu.com">这是一个链接</a>
7  </body>
8  </html>
```

注意：

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。

属性和属性值对大小写不敏感。不过推荐使用小写（标准化）。

## HTML 水平线

`<hr>` 标签在 HTML 页面中创建水平线, 可用于分隔内容。

```
1  <!doctype html>
2
3  <html>
4  <body>
5      <meta charset=utf-8></meta>
6      <h1>标题</h1>
7      <hr/>
8      <p>段落</p>
9
10     </body>
11 </html>
```

# 标题

---

段落

## HTML 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。

注释写法如下:

```
<!-- 这是一个注释 -->
```

**注意:** 开始括号之后 (左边的括号) 需要紧跟一个感叹号, 结束括号之前 (右边的括号) 不需要感叹号, 但是也需要一紧贴。如果不按照标准书写注释, 会出现错误。(也就是说浏览器必须看到`<!--`这样的才知道是注释开始了, 看到`-->`这样的才知道注释应该结束了。)

## HTML 输出- 使用提醒

我们无法确定 HTML 被显示的确切效果。屏幕的大小, 以及对窗口的调整都可能导致不同的结果。

对于 HTML, 无法通过在 HTML 代码中添加额外的空格或换行来改变输出的效果。

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。

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>菜鸟教程(runoob.com)</title>
</head>

<body>

<h1>春晓</h1>

<p>
  春眠不觉晓，
    处处闻啼鸟。
    夜来风雨声，
    花落知多少。
</p>

<p>注意，浏览器忽略了源代码中的排版（省略了多余的空格和换行）。</p>

</body>

</html>
```

## 春晓

春眠不觉晓，处处闻啼鸟。夜来风雨声，花落知多少。

注意，浏览器忽略了源代码中的排版（省略了多余的空格和换行）。

## HTML 文本格式化

<b> 定义粗体文本

<i> 定义斜体文本

<big> 定义大号字体

<small> 定义小号字体

通常标签 <strong> 替换加粗标签 <b> 来使用，<em> 替换 <i> 标签使用（效果一样）。

但是这两个标签本来的意义不是用来显示粗体或斜体，而是意味着你要呈现的文本是重要的。

<strong> 定义加重语气

<em> 定义着重文字



```

1  <!--doctype html-->
2
3  <html>
4  <body>
5    <meta charset=utf-8></meta>
6    <p>测试</p>
7    <b>测试</b><br/>
8    <i>测试</i><br/>
9    <strong>测试</strong><br/>
10   <em>测试</em><br/>
11   <small>测试</small><br/>
12   <big>测试</big><br/>
13 </body>
14 </html>

```

效果:

测试

**测试**

*测试*

**测试**

*测试*

测试

测试

值得注意的是，这几个标签浏览器都不会自动换行，所以如果想要显示更明显，这里要自行加入换行符<br/>。可以看到正常显示如<p>里面的内容，小号字体要比正常小一点，大号稍大。

<sub> 定义下标字

<sup> 定义上标字

```

1  <!--doctype html-->
2  <html>
3  <body>
4    <meta charset=utf-8></meta>
5    <p>测试</p>
6    这个文本包含
7    <sub>下标</sub>
8
9    <br />
10   这个文本包含
11   <sup>上标</sup>
12 </body>
13 </html>

```

## 测试

这个文本包含 下标

这个文本包含 上标

这两个标签貌似需要紧跟着文本。如果是在<p>等标签外部则对文本不起作用。

<ins>        定义插入字

<del>        定义删除字

```
1 <!doctype html>
2 <html>
3 <body>
4 <meta charset=utf-8></meta>
5 <p>测试效果</p>
6 <p>看看删除字和插入字的效果：<del>删除字</del> <ins>插入字</ins>!</p>
7
8 </body>
9 </html>
```

效果：

## 测试效果

看看删除字和插入字的效果：删除字 插入字!

可以看到在<del>标签中的内容中间有一条短线，如同被划掉了一样；在<ins>标签中的内容会在内容下方显示短线。

# HTML “计算机输出” 标签

<code>&lt;code&gt;</code>	定义计算机代码
<code>&lt;kbd&gt;</code>	定义键盘码
<code>&lt;samp&gt;</code>	定义计算机代码样本
<code>&lt;var&gt;</code>	定义变量
<code>&lt;pre&gt;</code>	定义预格式文本

```
1 <!doctype html>
2 <html>
3 <body>
4 <meta charset=utf-8></meta>
5 <p>测试效果</p>
6 <code>计算机输出</code>
7 <br />
8 <kbd>键盘输入</kbd>
9 <br />
10 <tt>打字机文本</tt>
11 <br />
12 <samp>计算机代码样本</samp>
13 <br />
14 <var>计算机变量</var>
15 <br />
16 <pre>定义的格式    可以
17     正常显示
18     比如空格和换行
19 </pre>
20 <p>
21 <b>注释： </b>这些标签常用于显示计算机/编程代码。
22 </body>
23 </html>
```

效果：

## 测试效果

计算机输出  
键盘输入  
打字机文本  
计算机代码样本  
*计算机变量*

定义的格式 可以  
 正常显示  
 比如空格和换行

**注释：** 这些标签常用于显示计算机/编程代码。

在这里我只理解了<var>和<pre>标签的作用，其他的输出好像并没有什么区别。

## HTML 引文，引用，及标签定义

<a href="#"><u>&lt;abbr&gt;</u></a>	定义缩写
<a href="#"><u>&lt;address&gt;</u></a>	定义地址
<a href="#"><u>&lt;bdo&gt;</u></a>	定义文字方向
<a href="#"><u>&lt;blockquote&gt;</u></a>	定义长的引用
<a href="#"><u>&lt;q&gt;</u></a>	定义短的引用语
<a href="#"><u>&lt;cite&gt;</u></a>	定义引用、引证
<a href="#"><u>&lt;dfn&gt;</u></a>	定义一个定义项目。

### <address>标签

```
index.html
1  <!doctype html>
2  <html>
3  <body>
4    <meta charset=utf-8></meta>
5    <p>测试效果</p>
6
7    <address>
8      Written by <a href="mailto:webmaster@example.com">Jon Doe</a>.<br>
9      Visit us at:<br>
10     Example.com<br>
11     Box 564, Disneyland<br>
12     USA
13   </address>
14
15 </body>
16 </html>
```

### 测试效果

*Written by [Jon Doe](mailto:webmaster@example.com).*  
*Visit us at:*  
*Example.com*  
*Box 564, Disneyland*  
*USA*

## <abbr>标签

```
1  <!doctype html>
2  <html>
3  <body>
4    <meta charset=utf-8></meta>
5    <p>测试效果</p>
6    <abbr title="html学习">html</abbr>
7
8  </body>
9  </html>
```

## 测试效果

### html

当把鼠标移到缩略词语上时，title 可用于展示表达完整版本。比如这里当鼠标放到 html 上时会显示 html 学习。

## <bdo>标签

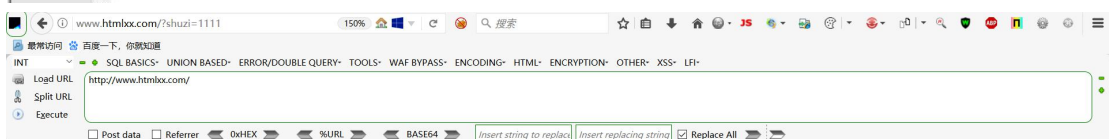
```
1  <!doctype html>
2  <html>
3  <body>
4    <meta charset=utf-8></meta>
5    <p>该段落文字从左到右显示。</p>
6    <p><bdo dir="rtl">该段落文字从右到左显示。</bdo></p>
7
8  </body>
9  </html>
```

该段落文字从左到右显示。

。示显左到右从字文落段该

dir 是属性，当值为 rtl 就以为着从右到左，ltr 意味着从左到右(right 是右边，left 是左边)。如果把这个属性添加到<p>标签中则表明段落是从浏览器页面右边开始输出，如下：

```
1 <!doctype html>
2 <html>
3 <body>
4 <meta charset=utf-8></meta>
5
6 <p dir="rtl">测试一下效果</p>
7
8 </body>
9 </html>
```



测试一下效果

## <q>标签

```
1 <!doctype html>
2 <html>
3 <body>
4 <meta charset=utf-8></meta>
5
6 <p>座右铭:
7 <q>生死看淡，不服就干。</q>
8 雷布斯.</p>
9
10 </body>
11 </html>
```

座右铭：“生死看淡，不服就干。” 雷布斯.

<q> 标签定义一个短的引用。浏览器经常会在这种引用的周围插入引号。

## HTML 超链接

HTML 使用标签 `<a>` 来设置超文本链接。

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，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。

在标签 `<a>` 中使用了 `href` 属性来描述链接的地址。

默认情况下，链接将以以下形式出现在浏览器中：

- 1、一个未访问过的链接显示为蓝色字体并带有下划线。
- 2、访问过的链接显示为紫色并带有下划线。
- 3、点击链接时，链接显示为红色并带有下划线。

```
1 <!doctype html>
2 <html>
3 <body>
4 <meta charset=utf-8></meta>
5 <a href="http://www.baidu.com">百度</a>
6 </body>
7 </html>
```

格式：

`<a href="url">链接文本</a>`

提示：“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。

`href` 属性：定义了超链接地址

`target` 属性：定义被链接的文档在何处显示

下面的这行会在新窗口打开文档

```
<a href="https://www.baidu.com/" target="_blank" rel="noopener norereferrer">访问百度!</a>
```

不添加该属性是默认在当前窗口打开文档

至于为什么要添加 `rel="noopener norereferrer"`

这是为了网站安全性考虑，详情请参考：

<https://www.cnblogs.com/ilinuxer/p/5245983.html>

<https://www.cnblogs.com/tangyuu/p/6912044.html>

id 属性:

id 属性可用于创建一个 HTML 文档书签标记。

提示：书签是不以任何特殊的方式显示，在 HTML 文档中是不显示的，所以对于访问者来说是隐藏的。

详情如下:

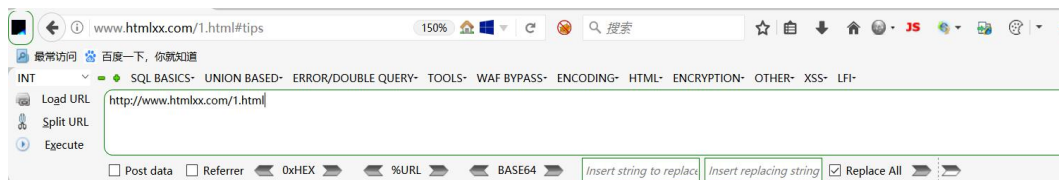
在同一个页面中，假如页面内容太多，对某些重要内容

`<a id="tips">这里显示重要内容</a>`这样处理，然后在显眼的地方加入一个超链接`<a href="#tips">查看重要内容</a>`。这样的话一旦点击页面就跳到了重要内容的地方。

```

1  <!doctype html>
2  <html>
3  <body>
4    <meta charset=utf-8></meta>
5    <a href="#tips">本页重点信息查看</a>
6  <p><pre>假设有很多内容
7    .....
8    .....
9    .....
10   .....
11   .....|
12   .....
13   .....
14   .....
15   .....
16   .....
17   .....
18   .....
19   .....
20   .....
21   .....
22   .....
23   .....
24   </pre></p>
25   <a id="tips">重点内容*****</a>
26 </body>
27 </html>

```

[本页重点信息查看](#)

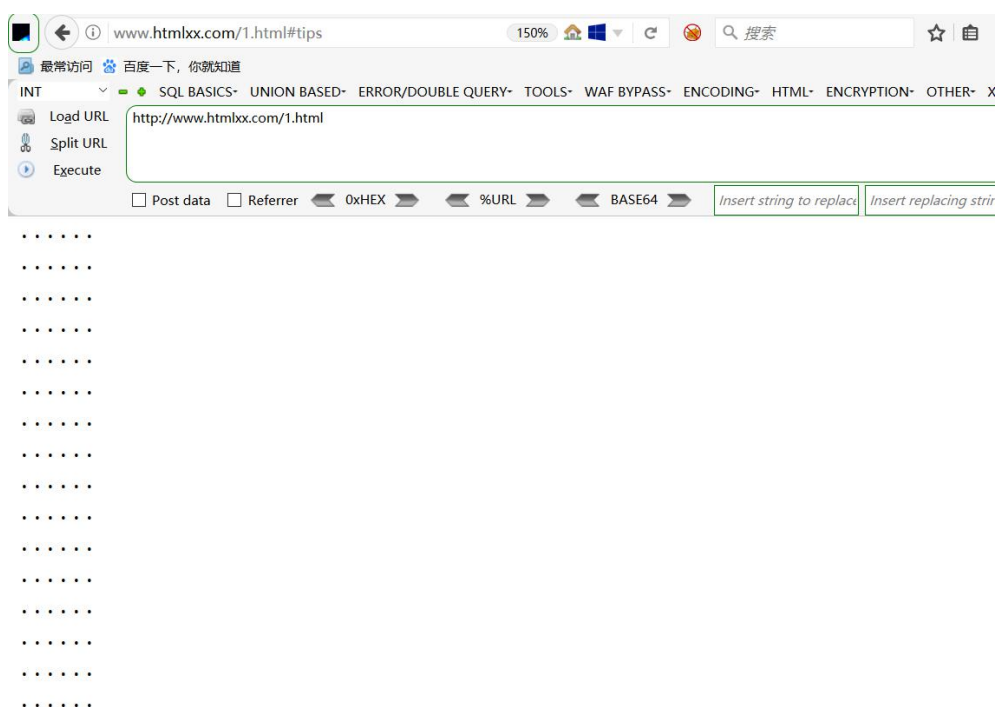
假设有很多内容

.....

可以看到在页面中还没有显示完全，重要内容还在页面下方，但是点击链接

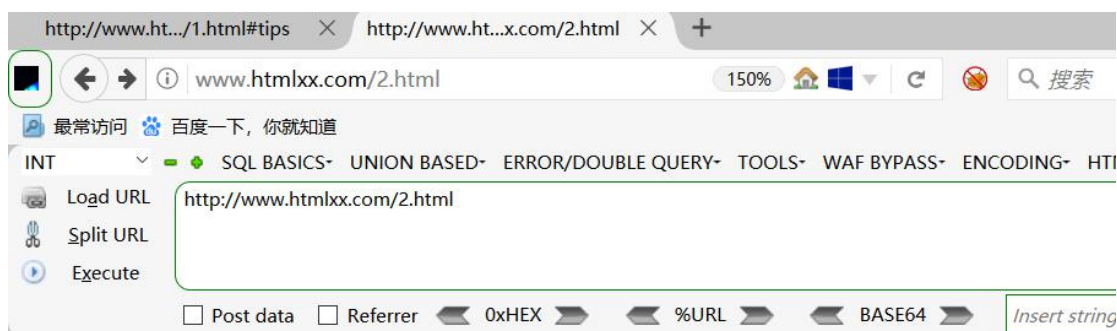


可以跳转到该处。



## 重点内容\*\*\*\*\*

不仅在同一个页面可以使用 id 这个属性，在不同的页面也可以跳转



## 查看内容

点击链接即可调到 1.html 页面中的重要内容处。

<img>标签属性:

border 定义图像的边框, 当值为 0 时就是无边框

alt 规定在图像无法显示时的替代文本

```

```

代码引入了一个图像, 无边框。当图像出现问题无法显示时, 在图像位置显示 alt 属性的值。

## HTML 头部

<head>

<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>, <base>.

<title> - 定义了 HTML 文档的标题

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

<base>

<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

<head>

```
<base href="http://www.runoob.com/images/" target="_blank">
```

</head>

这里要特别说明一下, 在网页设计时, 添加<base>标签其实是特别方便的, 它可以对往下所有的<a>标签定义默认的属性, 比如<base>中属性 target 是 \_blank 的话, 即使后续<a>中不单独定义, 浏览器也会默认 target 属性值为 \_blank。而且在<base>标签中定义了 href 属性的值, 后续<a>中就可以直接写相对的地址, 不再需要每一个都写一大串地址。

详情参考:

<https://baijiahao.baidu.com/s?id=1630884615185177419&wfr=spider&for=pc>

## <link>

<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 (CSS)

<head>

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

</head>

## <style>

<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

<head>

```
<style type="text/css">
```

```
body {background-color:yellow}
```

```
p {color:blue}
```

```
</style>
```

</head>

这一段样式是把网页背景颜色定义为了黄色,所有<p>标签的内容都是蓝色。

## <meta>

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。

<meta> 一般放置于 <head> 区域

为搜索引擎定义**关键词**:

```
<meta name="keywords" content="HTML, CSS, XML, XHTML, JavaScript">
```

为网页定义描述内容:

```
<meta name="description" content="免费 Web & 编程 教程">
```

定义网页作者:

```
<meta name="author" content="Runoob">
```

每 30 秒钟刷新当前页面:

```
<meta http-equiv="refresh" content="30">
```

## <script>

<script>标签用于加载脚本文件, 如: JavaScript。

## HTML 样式- 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。

不管是内联样式还是内联样式表还是外部引用，形式都是**属性：值；**。

背景颜色： background-color:yellow;

字体，字体颜色，字体大小： font-family:arial;color:red;font-size:20px;

文本对齐方式： text-align:center;

内联样式格式：

```
<h1 style="font-family:verdana;">一个标题</h1>
```

```
<p style="font-family:arial;color:red;font-size:20px;">一个段落。</p>
```

内联样式表格式：

```
<head>
```

```
<style type="text/css">
```

```
body {background-color:yellow;}
```

```
p {color:blue;}
```

```
</style>
```

```
</head>
```

外部引用格式：

```
<head>
```

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

```
</head>
```

## HTML 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：

```

```

alt 属性

alt 属性用来为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，替换文本属性告诉用户他们失去的信息。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。

注意：加载页面时，要注意插入页面图像的路径，如果不能正确设置图像的位置，浏览器无法加载图片，图像标签就会显示一个破碎的图片。



设置图像的高度与宽度

height（高度）与 width（宽度）属性用于设置图像的高度与宽度。

属性值默认单位为像素：

```

```

这里还要介绍一个好玩的，图像地图

<u>&lt;map&gt;</u>	定义图像地图
<u>&lt;area&gt;</u>	定义图像地图中的可点击区域

```

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>菜鸟教程(runoob.com)</title>
</head>
<body>

<p>点击太阳或其他行星，注意变化： </p>

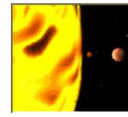


<map name="planetmap">
  <area shape="rect" coords="0,0,82,126" alt="Sun" href="sun.htm">
  <area shape="circle" coords="90,58,3" alt="Mercury" href="mercur.htm">
  <area shape="circle" coords="124,58,8" alt="Venus" href="venus.htm">
</map>

</body>
</html>

```

点击太阳或其他行星，注意变化：



在图像中点击不同区域会跳转到不同的页面

```

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

<p>点击太阳或其他行星，注意变化： </p>



<map name="planetmap">
  <area shape="rect" coords="0,0,82,126" alt="Sun" href="sun.htm">
  <area shape="circle" coords="90,58,3" alt="Mercury" href="mercur.htm">
  <area shape="circle" coords="124,58,8" alt="Venus" href="venus.htm">
</map>

</body>
</html>

```

<map>标签必须要有一个 name 属性

<area>标签中 coords 属性规定区域的 x 和 y 坐标。

coords 属性与 shape 属性配合使用，来规定区域的尺寸、形状和位置。

图像左上角的坐标是 "0,0"。

详细解释：

<area> 标签的 coords 属性定义了客户端图像映射中对鼠标敏感的区域坐标。坐标的数字及其含义取决于 shape 属性中决定的区域形状。可以将客户端图像映射中的超链接区域定义为矩形、圆形或多边形等。

下面列出了每种形状的适当值：

**圆形**：shape="circle"，coords="x,y,z"

这里的 x 和 y 定义了圆心的位置（"0,0" 是图像左上角的坐标），r 是以像素为单位的圆形半径。

**多边形**：shape="polygon"，coords="x1,y1,x2,y2,x3,y3,..."

每一对 "x,y" 坐标都定义了多边形的一个顶点（"0,0" 是图像左上角的坐标）。定义三角形至少需要三组坐标；高维多边形则需要更多数量的顶点。

多边形会自动封闭，因此在列表的结尾不需要重复第一个坐标来闭合整个区域。

**矩形**：shape="rectangle"，coords="x1,y1,x2,y2"

第一个坐标是矩形的一个角的顶点坐标，另一对坐标是对角的顶点坐标，"0,0" 是图像左上角的坐标。请注意，定义矩形实际上是定义带有四个顶点的多边形的一种简化方法。

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>
```

小技巧：需要知道图像中的坐标，可以用 QQ 截图。

## HTML 表格

表格由 `<table>` 标签来定义。每个表格均有若干行（由 `<tr>` 标签定义），每行被分割为若干单元格（由 `<td>` 标签定义）。字母 `td` 指表格数据（table data），即数据单元格的内容。数据单元格可以包含文本、图片、列表、段落、表单、水平线、表格等等。

```
1 <!doctype html>
2 <html>
3 <body>
4 <meta charset=utf-8></meta>
5 <table border="1">
6   <tr>
7     <td>row 1, cell 1</td>
8     <td>row 1, cell 2</td>
9   </tr>
10  <tr>
11    <td>row 2, cell 1</td>
12    <td>row 2, cell 2</td>
13  </tr>
14 </table>
15 </body>
16 </html>
```

row 1, cell 1	row 1, cell 2
row 2, cell 1	row 2, cell 2

`<table>` 标签用来定义生成一个表格，而 `<tr>` 标签就代表表格中的一行，一行中有几列数据就用 `<td>` 标签来定义。

表格的边框属性是 `border`，如果不定义边框属性，表格将不显示边框。

### `<th>` 标签

表格的表头使用 `<th>` 标签进行定义。这里的表头就是相当于数据库中表格的列名。大多数浏览器会把表头显示为粗体居中的文本：

```
1 <!doctype html>
2 <html>
3 <body>
4 <meta charset=utf-8></meta>
5 <table border="1">
6   <tr>
7     <th>Header 1</th>
8     <th>Header 2</th>
9   </tr>
10  <tr>
11    <td>row 1, cell 1</td>
12    <td>row 1, cell 2</td>
13  </tr>
14  <tr>
15    <td>row 2, cell 1</td>
16    <td>row 2, cell 2</td>
17  </tr>
18 </table>
19 </table>
20 </body>
21 </html>
```



---

Header 1	Header 2
row 1, cell 1	row 1, cell 2
row 2, cell 1	row 2, cell 2

可以看到表头也是嵌套在<tr>标签中使用的。

<caption>标签定义表格的标题

---

```
1 <!doctype html>
2 <html>
3 <body>
4 <meta charset=utf-8></meta>
5 <table border="1">
6   <caption>Monthly savings</caption>
7   <tr>
8     <th>Month</th>
9     <th>Savings</th>
10  </tr>
11  <tr>
12    <td>January</td>
13    <td>$100</td>
14  </tr>
15  <tr>
16    <td>February</td>
17    <td>$50</td>
18  </tr>
19 </table>
20 </body>
21 </html>
```

Monthly savings

Month	Savings
January	\$100
February	\$50

注意<caption>标签的使用位置。

## 关于跨行或跨列的表格单元格

一般都是使用属性 colspan 和 rowspan 来实现。例如：

```
1 <!doctype html>
2 <html>
3 <body>
4 <meta charset=utf-8></meta>
5 <h4>单元格跨两列:</h4>
6 <table border="1">
7 <tr>
8 <th>Name</th>
9 <th colspan="2">Telephone</th>
10 </tr>
11 <tr>
12 <td>Bill Gates</td>
13 <td>555 77 854</td>
14 <td>555 77 855</td>
15 </tr>
16 </table>
17
18 <h4>单元格跨两行:</h4>
19 <table border="1">
20 <tr>
21 <th>First Name:</th>
22 <td>Bill Gates</td>
23 </tr>
24 <tr>
25 <th rowspan="2">Telephone:</th>
26 <td>555 77 854</td>
27 </tr>
28 <tr>
29 <td>555 77 855</td>
30 </tr>
31 </table>
32 </body>
33 </html>
```

这里需要注意 colspan 表明该表头跨了两列，rowspan 表明该表头跨了两行

### 单元格跨两列:

Name	Telephone	
Bill Gates	555 77 854	555 77 855

### 单元格跨两行:

First Name:	Bill Gates
Telephone:	555 77 854
	555 77 855

在表格中表格单元需要跨行或者跨列，就在哪个标签中添加属性，赋值就是所跨的长度。

下面向大家展示一下表格内的标签：

```
5 <table border="1">
6 <tr>
7   <td>
8     <p>这是一个段落</p>
9     <p>这是另一个段落</p>
10  </td>
11  <td>这个单元格包含一个表格:
12    <table border="1">
13      <tr>
14        <td>A</td>
15        <td>B</td>
16      </tr>
17      <tr>
18        <td>C</td>
19        <td>D</td>
20      </tr>
21    </table>
22  </td>
23 </tr>
24 <tr>
25  <td>这个单元格包含一个列表
26    <ul>
27      <li>apples</li>
28      <li>bananas</li>
29      <li>pineapples</li>
30    </ul>
31  </td>
32  <td>HELLO</td>
33 </tr>
34 </table>
```

这是一个段落 这是另一个段落	这个单元格包含一个表格: <table><tr><td>A</td><td>B</td></tr><tr><td>C</td><td>D</td></tr></table>	A	B	C	D
A	B				
C	D				
这个单元格包含一个列表 <ul style="list-style-type: none"><li>apples</li><li>bananas</li><li>pineapples</li></ul>	HELLO				

单元格边距：单元格内容与其边框之间的空白

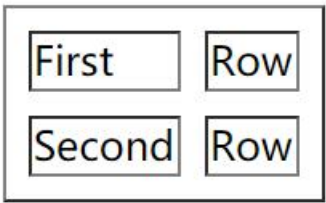
```
<table border="1" cellpadding="10">
```

First	Row
Second	Row

单元格间距：单元格之间的距离

```
<table border="1" cellspacing="10">
```

单元格间距="10":



HTML 表格标签：

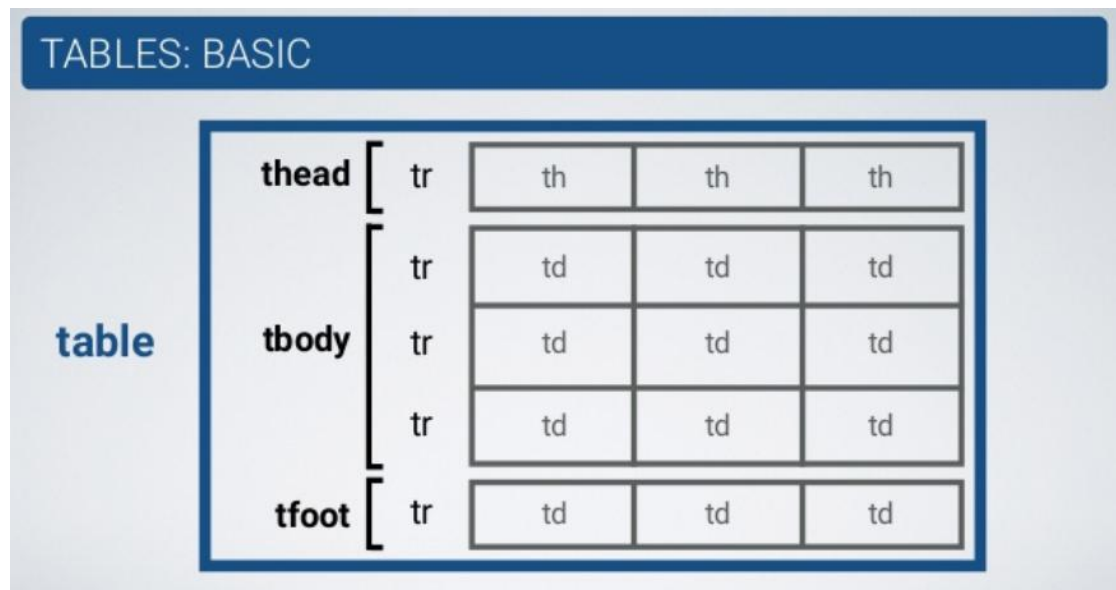
<a href="#"><u>&lt;table&gt;</u></a>	定义表格
<a href="#"><u>&lt;th&gt;</u></a>	定义表格的表头
<a href="#"><u>&lt;tr&gt;</u></a>	定义表格的行
<a href="#"><u>&lt;td&gt;</u></a>	定义表格单元
<a href="#"><u>&lt;caption&gt;</u></a>	定义表格标题
<a href="#"><u>&lt;colgroup&gt;</u></a>	定义表格列的组
<a href="#"><u>&lt;col&gt;</u></a>	定义用于表格列的属性
<a href="#"><u>&lt;thead&gt;</u></a>	定义表格的页眉
<a href="#"><u>&lt;tbody&gt;</u></a>	定义表格的主体
<a href="#"><u>&lt;tfoot&gt;</u></a>	定义表格的页脚

通常<colgroup>和<col>联合使用对表格的列作出一些设置。  
<colgroup> 和 <col> 标签为表格中的三个列设置了背景色

```
<table border="1">
  <colgroup>
    <col span="2" style="background-color:red">
    <col style="background-color:yellow">
  </colgroup>
  <tr>
    <th>ISBN</th>
    <th>Title</th>
    <th>Price</th>
  </tr>
  <tr>
    <td>3476896</td>
    <td>My first HTML</td>
    <td>$53</td>
  </tr>
  <tr>
    <td>5869207</td>
    <td>My first CSS</td>
    <td>$49</td>
  </tr>
</table>
```

ISBN	Title	Price
3476896	My first HTML	\$53
5869207	My first CSS	\$49

关于表格的页眉、主体、页脚



Title	Price	Available
CSS Demystified	\$29	16
Mastering JavaScript	\$35	10
HTML5: An Introduction	\$15	6
Total		32

```
<table>
  <thead>
    <tr>
      <th> Title </th>
      <th> Price </th>
      <th> Available </th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td> CSS Demystified </td>
      <td> $29 </td>
      <td> 16 </td>
    </tr>
    <tr>
      <td> Mastering JavaScript </td>
      <td> $35 </td>
      <td> 10 </td>
    </tr>
    <tr>
      <td> HTML5: An Introduction </td>
      <td> $15 </td>
      <td> 6 </td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <td> Total </td>
      <td> </td>
      <td> 32 </td>
    </tr>
  </tfoot>
</table>
```

<table> 标签常用属性:

`border="1"` 表格边框的宽度

`bordercolor="#fff"` 表格边框的颜色

`cellspacing="5"` 单元格之间的间距

`width="500"` 表格的总宽度

`height="100"` 表格的总高度

`align="right"` 表格整体对齐方式 (参数有 left、center、right)

`bgcolor="#fff"` 表格整体的背景色

<tr> 标签的常用属性:

<code>bgcolor="#fff"</code>	行的颜色
<code>align="right"</code>	行内文字的水平对齐方式 (参数有 left、center、right)
<code>valign="top"</code>	行内文字的垂直对齐方式 (参数有 top、middle、bottom)

<td>、<th> 标签的常用属性:

<code>width="500"</code>	单元格的宽度, 设置后对当前一列的单元格都有影响
<code>height="100"</code>	单元格的高度, 设置后对当前一行的单元格都有影响
<code>bgcolor="fff"</code>	单元格的背景色
<code>align="right"</code>	单元格文字的水平对齐方式 (参数 left、center、right)
<code>rowspan="3"</code>	合并垂直水平方向的单元格
<code>colspan="3"</code>	合并水平方向单元格
<code>valign="top"</code>	单元格文字的垂直对齐方式 (参数 middle、bottom、top)

## HTML 列表

HTML 支持有序、无序和定义列表:

有序列表	无序列表
1. 第一个列表项	● 列表项
2. 第二个列表项	● 列表项
3. 第三个列表项	● 列表项

### HTML 列表标签

<code>&lt;ol&gt;</code>	定义有序列表
<code>&lt;ul&gt;</code>	定义无序列表
<code>&lt;li&gt;</code>	定义列表项
<code>&lt;dl&gt;</code>	定义列表
<code>&lt;dt&gt;</code>	自定义列表项目
<code>&lt;dd&gt;</code>	定义自定义列表项的描述

## HTML 无序列表

无序列表是一个项目的列表，此列项目使用粗体圆点（典型的小黑圆圈）进行标记。

无序列表使用 `<ul>` 标签

```
<ul>
<li>Coffee</li>
<li>Milk</li>
</ul>
```

浏览器显示如下：

- Coffee
- Milk

## HTML 有序列表

同样，有序列表也是一列项目，列表项目使用数字进行标记。有序列表始于 `<ol>` 标签。每个列表项始于 `<li>` 标签。

列表项使用数字来标记。

```
<ol>
<li>Coffee</li>
<li>Milk</li>
</ol>
```

浏览器中显示如下：

1. Coffee
2. Milk

## HTML 自定义列表

自定义列表不仅仅是一列项目，而是项目及其注释的组合。

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。

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

浏览器显示如下：



## Coffee

- black hot drink

## Milk

- white cold drink

<ol>标签的 type 属性:

<ol>标签中的列表项默认是以数字排序的,如前面的例子。但是也可以自己定义。

编号列表:

<ol>	
<li>Apples</li>	1. Apples
<li>Bananas</li>	2. Bananas
<li>Lemons</li>	3. Lemons
<li>Oranges</li>	4. Oranges
</ol>	

大写字母列表:

<ol type="A">	
<li>Apples</li>	A. Apples
<li>Bananas</li>	B. Bananas
<li>Lemons</li>	C. Lemons
<li>Oranges</li>	D. Oranges
</ol>	

小写字母列表:

<ol type="a">	
<li>Apples</li>	a. Apples
<li>Bananas</li>	b. Bananas
<li>Lemons</li>	c. Lemons
<li>Oranges</li>	d. Oranges
</ol>	

罗马数字列表:

<ol type="I">	
<li>Apples</li>	I. Apples
<li>Bananas</li>	II. Bananas
<li>Lemons</li>	III. Lemons
<li>Oranges</li>	IV. Oranges
</ol>	

在 HTML 4 中 ul 属性已废弃，HTML5 已不支持该属性，因此我们使用 CSS 代替来定义不同类型的无序列表如下：

<h4>圆点列表: </h4>

<ul style="list-style-type:disc">

<li>Apples</li>

<li>Bananas</li>

<li>Lemons</li>

<li>Oranges</li>

</ul>

**圆点列表:**

- Apples
- Bananas
- Lemons
- Oranges

<h4>圆圈列表: </h4>

<ul style="list-style-type:circle">

<li>Apples</li>

<li>Bananas</li>

<li>Lemons</li>

<li>Oranges</li>

</ul>

**圆圈列表:**

- Apples
- Bananas
- Lemons
- Oranges

<h4>正方形列表: </h4>

<ul style="list-style-type:square">

<li>Apples</li>

<li>Bananas</li>

<li>Lemons</li>

<li>Oranges</li>

</ul>

**正方形列表:**

- Apples
- Bananas
- Lemons
- Oranges

## HTML 区块

HTML <div> 和<span>

HTML 可以通过 <div> 和 <span>将元素组合起来。

HTML 区块元素

大多数 HTML 元素被定义为**块级元素**或**内联元素**。块级元素在浏览器显示时，通常会以**新行来开始（和结束）**。

实例：<h1>, <p>, <ul>, <table>

HTML 内联元素

内联元素在显示时通常不会以新行开始。

实例：<b>，<td>，<a>，<img>

## HTML <div> 元素

HTML <div> 元素是块级元素，它是一个可用于组合其他 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。

## HTML <span> 元素

HTML <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。

## HTML 分组标签

<u>&lt;div&gt;</u>	定义了文档的区域，块级 (block-level)
<u>&lt;span&gt;</u>	用来组合文档中的行内元素， 内联元素(inline)

## HTML 布局

网页布局对改善网站的外观非常重要。

大多数网站会把内容安排到多个列中（就像杂志或报纸那样）。

大多数网站可以使用 <div> 或者 <table> 元素来创建多列。CSS 用于对元素进行定位，或者为页面创建背景以及色彩丰富的外观。

注意：不推荐用<table>标签来实现布局效果。

## 使用<div> 元素

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>菜鸟教程(runoob.com)</title>
</head>
<body>

<div id="container" style="width:500px">

<div id="header" style="background-color:#FFA500;">
<h1 style="margin-bottom:0;">主要的网页标题</h1></div>

<div id="menu" style="background-color:#FFD700;height:200px;width:100px;float:left;">
<b>菜单</b><br>
HTML<br>
CSS<br>
JavaScript</div>

<div id="content" style="background-color:#EEEEEE;height:200px;width:400px;float:left;">
内容在这里</div>

<div id="footer" style="background-color:#FFA500;clear:both;text-align:center;">
版权 © runoob.com</div>

</div>

</body>
</html>
```



## 使用表格

使用 HTML <table> 标签是创建布局的一种简单的方式。

```

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>菜鸟教程(runoob.com)</title>
</head>
<body>

<table width="500" border="0">
<tr>
<td colspan="2" style="background-color:#FFA500;">
<h1>主要的网页标题</h1>
</td>
</tr>

<tr>
<td style="background-color:#FFD700;width:100px;">
<b>菜单</b><br>
HTML<br>
CSS<br>
JavaScript
</td>
<td style="background-color:#eeeeee;height:200px;width:400px;">
内容在这里</td>
</tr>

<tr>
<td colspan="2" style="background-color:#FFA500;text-align:center;">
版权 © runoob.com</td>
</tr>
</table>

</body>
</html>

```

# 主要的网页标题

## 菜单

HTML

CSS

JavaScript

内容在这里

版权 © runoob.com

# HTML 表单

HTML 表单用于收集不同类型的用户输入。  
表单是一个包含表单元素的区域。  
表单元素是允许用户在表单中输入内容, 比如: 文本域(textarea)、下拉列表、单选框(radio-buttons)、复选框(checkboxes)等等。  
表单使用表单标签 <form> 来设置

HTML 表单标签:

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

## HTML 表单 - 输入元素

多数情况下被用到的表单标签是输入标签（<input>）。

输入类型是由类型属性（type）定义的。大多数经常被用到的输入类型如下：

### 文本域（Text Fields）

文本域通过<input type="text"> 标签来设定，当用户要在表单中键入字母、数字等内容时，就会用到文本域。

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

First name:

Last name:

注意:表单本身并不可见。同时，在大多数浏览器中，文本域的默认宽度是 20 个字符。

### 密码字段

密码字段通过标签<input type="password"> 来定义：

```
<form>
Password: <input type="password" name="pwd">
</form>
```

Password:

注意:密码字段字符不会明文显示，而是以星号或圆点替代。

### 单选按钮

<input type="radio"> 标签定义了表单单选框选项

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

☒ Male

☐ Female

一次只能选中一个选项。

## 复选框

`<input type="checkbox">` 定义了复选框。用户需要从若干给定的选择中选取一个或若干选项。

```
<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>
```

☒ I have a bike

☒ I have a car

可以选择单个也可以多选

## 提交按钮

`<input type="submit">` 定义了提交按钮。

当用户单击确认按钮时，表单的内容会被传送到另一个文件。表单的动作属性(action)定义了目的文件的文件名。由动作属性定义的这个文件通常会对接收到的输入数据进行相关的处理。

```
<form name="input" action="html_form_action.php" method="get">
```

```
Username: <input type="text" name="user">
```

```
<input type="submit" value="Submit">
```

```
</form>
```

Username:

假如您在上面的文本框内键入几个字母，然后点击确认按钮，那么输入数据会传送到“html\_form\_action.php”的页面。



实例：

```
index.html x  html_form_action.php x
1  <!doctype html>
2  <html>
3  <body>
4    <meta charset=utf-8></meta>
5    <form name="input" action="html_form_action.php" method="get">
6      Username: <input type="text" name="user"><br>
7      性别: <br>
8      <input type="radio" name="sex" value="男">男</input>
9      <input type="radio" name="sex" value="女">女</input><br>
10     <input type="submit" value="Submit">
11   </form>
12 </body>
13 </html>
```

```
index.html x  html_form_action.php x
1  <?php
2    $a=$_GET['user'];
3    $b=$_GET['sex'];
4    echo $a."是".$b."性";
5
6  ?>
```

Username:

性别:

☒男 ☐女

守卫者是男性

## 简单的下拉列表

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



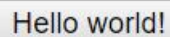
`<select>`标签定义了下拉列表，`<option>`则定义了下拉列表中的选项。

如果没有对下拉列表设定预选值，则默认显示第一个。如果在`<option>`标签中设定了 `selected`, 则该项就是预选值，框中显示该项。

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

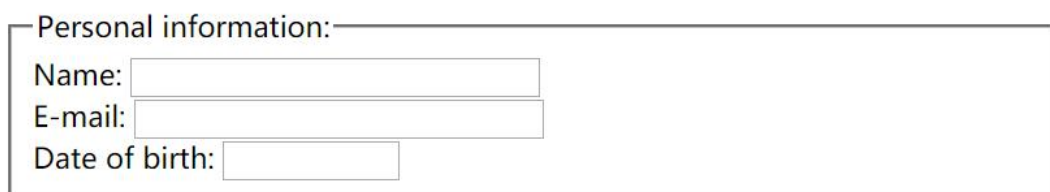
## 创建按钮

```
<form action="">
<input type="button" value="Hello world!">
</form>
```



## `<fieldset>`和`<legend>`联合使用创建带边框的表单

```
<form action="">
<fieldset>
<legend>Personal information:</legend>
Name: <input type="text" size="30"><br>
E-mail: <input type="text" size="30"><br>
Date of birth: <input type="text" size="10">
</fieldset>
</form>
```



`<legend>`标签是定义了 `<fieldset>` 元素的标题

带有输入框和确认按钮的表单

```
<form action="demo-form.php">
First name: <input type="text" name="FirstName" value="Mickey"><br>
Last name: <input type="text" name="LastName" value="Mouse"><br>
<input type="submit" value="提交">
</form>
```

First name:

Last name:

点击"提交"按钮，表单数据将被发送到服务器上的 "demo-form.php" 。

这里值得注意的是，在<input>标签中给 value 属性给出了默认值，name 属性的值就是 url 传输时的参数名，value 就是传输时的值。如果不输入别的内容，就默认传输 value 的值，输入了就传输输入的内容。

注意：<input type="reset">定义重置按钮

```
<input type="reset" name="button" id="button" value="重置">
```

点击之后会将重置按钮所在的表单中填写的内容重新设置为默认值。

对于单选按钮和复选框来说，可以设置以下几个属性：value、name、checked

value：提交数据到服务器的值（后台程序 PHP 使用）

name：为控件命名，以备后台程序 ASP、PHP 使用

checked：当设置 checked="checked" 时，该选项被默认选中

```
<form>
<p>你生活在哪个国家？ </p>
<input type="radio" name="country" value="China" checked="checked">中
国<br />
<input type="radio" name="country" value="the USA">美国
</form>
```

注意：同一组的单选按钮，name 取值一定要一致，比如上面例子为同一个名称 "country"，这样同一组的单选按钮才可以起到单选的作用。

## HTML 框架

通过使用框架，你可以在同一个浏览器窗口中显示不止一个页面。

iframe 语法：

```
<iframe src="URL"></iframe>
```

该 URL 指向不同的网页。

### iframe - 设置高度与宽度

height 和 width 属性用来定义 iframe 标签的高度与宽度。属性默认以像素为单位，但是你可以指定其按比例显示（如：“80%”）。

```
<iframe src="demo_iframe.htm" width="200" height="200"></iframe>
```

### iframe - 移除边框

frameborder 属性用于定义 iframe 表示是否显示边框。

设置属性值为 “0” 移除 iframe 的边框：

```
<iframe src="demo_iframe.htm" frameborder="0"></iframe>
```

### 使用 iframe 来显示目标链接页面

iframe 可以显示一个目标链接的页面

目标链接的属性必须使用 iframe 的属性，如下实例：

```
<iframe src="demo_iframe.htm" name="iframe_a"></iframe>
```

```
<p><a href="http://www.runoob.com" target="iframe_a">RUNOOB.COM</a></p>
```

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>菜鸟教程(runoob.com)</title>
</head>
<body>

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

<p><b>注意：</b> 因为 a 标签的 target 属性是名为 iframe_a 的 iframe 框架，所以在点
</body>
</html>
```

点击运行

运行结果



[RUNOOB.COM](http://www.runoob.com)

**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。

代码的意思就是定义了一个框架（名字为 iframe\_a），然后又定义了一个超链接，目标地址是 <http://www.runoob.com>，target 定义了超链接在目标名为 iframe\_a 的窗口打开，也就是框架内打开。

## HTML 脚本

JavaScript 使 HTML 页面具有更强的动态和交互性。

### HTML <script> 标签

<script> 标签用于定义客户端脚本，比如 JavaScript（也不一定非得 javascript 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。JavaScript 最常用于图片操作、表单验证以及内容动态更新。

下面的脚本会向浏览器输出“Hello World!”：

```
<script>
document.write("Hello World!");
</script>
```

### 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：

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

## HTML 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。

### HTML 实体

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

&#entity\_number;

如需显示小于号，我们必须这样写：&lt; 或 &#60; 或 &#060;

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(&nbsp; )。

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。

HTML 字符实体

| 显示结果 | 描述     | 实体名称            | 实体编号    |
|------|--------|-----------------|---------|
|      | 空格     | &nbsp;          | &#160;  |
| <    | 小于号    | &lt;            | &#60;   |
| >    | 大于号    | &gt;            | &#62;   |
| &    | 和号     | &amp;           | &#38;   |
| "    | 引号     | &quot;          | &#34;   |
| '    | 撇号     | &apos; (IE 不支持) | &#39;   |
| ¢    | 分      | &cent;          | &#162;  |
| £    | 镑      | &pound;         | &#163;  |
| ¥    | 人民币/日元 | &yen;           | &#165;  |
| €    | 欧元     | &euro;          | &#8364; |
| §    | 小节     | &sect;          | &#167;  |
| ©    | 版权     | &copy;          | &#169;  |
| ®    | 注册商标   | &reg;           | &#174;  |
| ™    | 商标     | &trade;         | &#8482; |
| ×    | 乘号     | &times;         | &#215;  |
| ÷    | 除号     | &divide;        | &#247;  |

虽然 html 不区分大小写，但实体字符对大小写敏感。