HTML小记

1.<div class>和<div id>的区别

class是设置标签的类。  
id是设置标签的标识。  
  
class属性用于指定元素属于何种样式的类。  
如样式表可以加入：  
.baobao { color: lime; background: #ff80c0 }搜索   
使用方法：class="baobao"  
id属性用于定义一个元素的独特的样式。如一个CSS规则：  
#binbin { [font-size](https://www.baidu.com/s?wd=font-size&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y1rjIhmhwhPHwhnW6smhFW0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3En163PWnsn1fY): larger }   
使用方法：id="binbin"  
  
id是一个标签，用于区分不同的结构和内容，就象你的学号，如果一个班里有2个人相同学号，就会出现混淆；  
class是一个样式，可以套在任何结构和内容上，就象一件衣服；  
概念上说就是不一样的：id是先找到结构/内容，再给它定义样式；class是先定义好一种样式，再套给多个结构/内容。  
归纳成一句话就是：Class可以反复使用而ID在一个页面中仅能被使用一次。有可能在很大部分浏览器中反复使用同一个ID不会出现问题，但在标准上这绝对是错误的使用，而且很可能导致某些浏览器的现实问题。  
在实际应用的时候，Class可能对文字的排版等比较有用，而ID则对宏观布局和设计放置各种元素较有用。

2.HTML布局

[使用 <div> 元素的网页布局](http://www.runoob.com/try/try.php?filename=tryhtml_layout_divs)  
如何使用 <div> 元素添加布局。

[使用 <table> 元素的网页布局](http://www.runoob.com/try/try.php?filename=tryhtml_layout_tables)  
如何使用 <table> 元素添加布局。

## 网站布局

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。

|  |  |
| --- | --- |
| **lamp** | 虽然我们可以使用HTML table标签来设计出漂亮的布局，但是table标签是不建议作为布局工具使用的 - 表格不是布局工具。 |

## HTML 布局 - 使用<div> 元素

div 元素是用于分组 HTML 元素的块级元素。

下面的例子使用五个 div 元素来创建多列布局：

## 实例

<!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>

[尝试一下 »](http://www.runoob.com/try/try.php?filename=tryhtml_layout_divs)

上面的 HTML 代码会产生如下结果：



## HTML 布局 - 使用表格

使用 HTML <table> 标签是创建布局的一种简单的方式。

大多数站点可以使用 <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。

|  |  |
| --- | --- |
| **lamp** | 即使可以使用 HTML 表格来创建漂亮的布局，但设计表格的目的是呈现表格化数据 - 表格不是布局工具！ |

下面的例子使用三行两列的表格 - 第一和最后一行使用 colspan 属性来横跨两列：

## 实例

<!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>

[尝试一下 »](http://www.runoob.com/try/try.php?filename=tryhtml_layout_tables)

上面的 HTML 代码会产生以下结果：



## HTML 布局 - 有用的提示

**Tip:** 使用 CSS 最大的好处是，如果把 CSS 代码存放到外部样式表中，那么站点会更易于维护。通过编辑单一的文件，就可以改变所有页面的布局。如需学习更多有关 CSS 的知识，请访问我们的[CSS 教程](http://www.runoob.com/css/css-tutorial.html)。

**Tip:** 由于创建高级的布局非常耗时，使用模板是一个快速的选项。通过搜索引擎可以找到很多免费的网站模板（您可以使用这些预先构建好的网站布局，并优化它们）。

## HTML 布局标签

|  |  |
| --- | --- |
| **标签** | **描述** |
| [<div>](http://www.runoob.com/tags/tag-div.html) | 定义文档区块，块级(block-level) |
| [<span>](http://www.runoob.com/tags/tag-span.html) | 定义 span，用来组合文档中的行内元素。 |

3. **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。

**<input> 元素**

**<input>** 元素是最重要的**表单元素**。

<input> 元素有很多形态，根据不同的 **type** 属性。

这是本章中使用的类型：

|  |  |
| --- | --- |
| **类型** | **描述** |
| text | 定义常规文本输入。 |
| radio | 定义单选按钮输入（选择多个选择之一） |
| submit | 定义提交按钮（提交表单） |

## 单选按钮输入

**<input type="radio">** 定义**单选按钮**。

单选按钮允许用户在有限数量的选项中选择其中之一：

### 实例

<form>

<input type="radio" name="sex" value="male" checked>Male

<br>

<input type="radio" name="sex" value="female">Female

</form>

## 提交按钮

**<input type="submit">** 定义用于向**表单处理程序**（form-handler）**提交**表单的按钮。

表单处理程序通常是包含用来处理输入数据的脚本的服务器页面。

表单处理程序在表单的 **action** 属性中指定：

### 实例

<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>

**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">

## Name 属性

如果要正确地被提交，每个输入字段必须设置一个 name 属性。

本例只会提交 "Last name" 输入字段：

### 实例

<form action="action\_page.php">

First name:<br>

<input type="text" value="Mickey">

<br>

Last name:<br>

<input type="text" name="lastname" value="Mouse">

<br><br>

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

</form>

## 用 <fieldset> 组合表单数据

**<fieldset>** 元素组合表单中的相关数据

**<legend>** 元素为 <fieldset> 元素定义标题。

### 实例

<form action="action\_page.php">

<fieldset>

<legend>Personal information:</legend>

First name:<br>

<input type="text" name="firstname" value="Mickey">

<br>

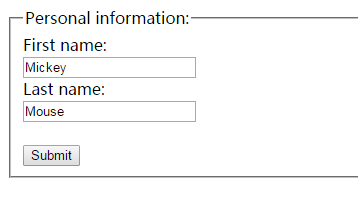
Last name:<br>

<input type="text" name="lastname" value="Mouse">

<br><br>

<input type="submit" value="Submit"></fieldset>

</form>



## HTML Form 属性

HTML <form> 元素，已设置所有可能的属性，是这样的：

### 实例

<form action="action\_page.php" method="GET" target="\_blank" accept-charset="UTF-8"

ectype="application/x-www-form-urlencoded" autocomplete="off" novalidate>

.

form elements

.

</form>

Here is the list of <form> attributes:

|  |  |
| --- | --- |
| **属性** | **描述** |
| accept-charset | 规定在被提交表单中使用的字符集（默认：页面字符集）。 |
| action | 规定向何处提交表单的地址（URL）（提交页面）。 |
| autocomplete | 规定浏览器应该自动完成表单（默认：开启）。 |
| enctype | 规定被提交数据的编码（默认：url-encoded）。 |
| method | 规定在提交表单时所用的 HTTP 方法（默认：GET）。 |
| name | 规定识别表单的名称（对于 DOM 使用：document.forms.name）。 |
| novalidate | 规定浏览器不验证表单。 |
| target | 规定 action 属性中地址的目标（默认：\_self）。 |

4.表单form的元素

**<input> 元素**

最重要的表单元素是 **<input>** 元素。

<input> 元素根据不同的 **type** 属性，可以变化为多种形态。

注释：下一章讲解所有 HTML 输入类型。

## <select> 元素（下拉列表）

**<select>** 元素定义**下拉列表**：

### 实例

<select name="cars">

<option value="volvo">Volvo</option>

<option value="saab">Saab</option>

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

<option value="audi">Audi</option>

</select>

其中**<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>

<html>

<body>

<form action="kajkfa.asp" method="get">

<textarea name="message" rows="10" cols="30">

The cat was playing in the garden.

</textarea>

<br>

<input type="submit" value="草">

</form>

</body>

</html>

## <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>

**HTML5 表单元素**

HTML5 增加了如下表单元素：

* <datalist>
* <keygen>
* <output>

注释：默认地，浏览器不会显示未知元素。新元素不会破坏您的页面

## HTML5 <datalist> 元素

**<datalist>** 元素为 <input> 元素规定预定义选项列表。

用户会在他们输入数据时看到预定义选项的下拉列表。

<input> 元素的 **list** 属性必须引用 <datalist> 元素的 **id** 属性。

### 实例

通过 <datalist> 设置预定义值的 <input> 元素：

<form action="action\_page.php">

<input list="browsers">

<datalist id="browsers">

<option value="Internet Explorer">

<option value="Firefox">

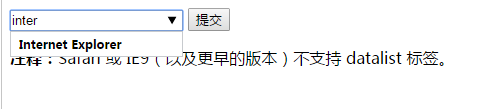
<option value="Chrome">

<option value="Opera">

<option value="Safari">

</datalist>

</form>



## 5. 输入类型

本章描述 <input> 元素的输入类型。

## 输入类型：text

**<input type="text">** 定义供**文本输入**的单行输入字段：

### 实例

<form>

First name:<br>

<input type="text" name="firstname">

<br>

Last name:<br>

<input type="text" name="lastname">

</form>

## 输入类型：password

**<input type="password">** 定义**密码字段**：

### 实例

<form>

User name:<br>

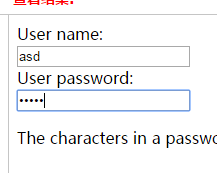
<input type="text" name="username">

<br>

User password:<br>

<input type="password" name="psw">

</form>



## 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）：

### 实例

<form action="action\_page.php">

First name:<br>

<input type="text" name="firstname" value="Mickey">

<br>

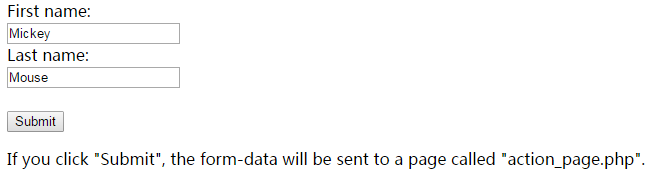
Last name:<br>

<input type="text" name="lastname" value="Mouse">

<br><br>

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

</form>



## 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:

### 实例

<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。

Checkboxes let a user select ZERO or MORE options of a limited number of choices.

### 实例

<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>

## Input Type: button（onclick事件通常用于绑定js函数）

**<input type="button>** 定义**按钮**。

### 实例

<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。

### 实例

<form>

Quantity (between 1 and 5):

<input type="number" name="quantity" **min="1" max="5"**>

</form>

**输入限制**

这里列出了一些常用的输入限制（其中一些是 HTML5 中新增的）：

|  |  |
| --- | --- |
| **属性** | **描述** |
| disabled | 规定输入字段应该被禁用。 |
| max | 规定输入字段的最大值。 |
| maxlength | 规定输入字段的最大字符数。 |
| min | 规定输入字段的最小值。 |
| pattern | 规定通过其检查输入值的正则表达式。 |
| readonly | 规定输入字段为只读（无法修改）。 |
| required | 规定输入字段是必需的（必需填写）。 |
| size | 规定输入字段的宽度（以字符计）。 |
| step | 规定输入字段的合法数字间隔。 |
| value | 规定输入字段的默认值。 |

## 输入类型：date

**<input type="date">** 用于应该包含日期的输入字段。

根据浏览器支持，**日期选择器**会出现输入字段中。(H5自带的，不用框架去实现，需要浏览器支持)

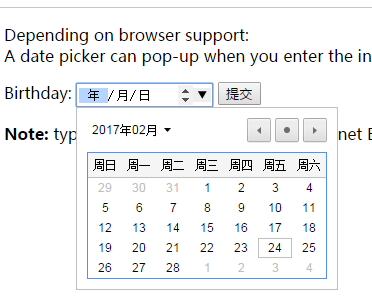
### 实例

<form>

Birthday:

<input type="date" name="bday">

</form>



注意：input type的month提供年月输入，week提供年周输入，time提供一个24小时输入

您可以向输入添加限制：

### 实例

<form>

Enter a date before 1980-01-01:

<input type="date" name="bday" max="1979-12-31"><br>

Enter a date after 2000-01-01:

<input type="date" name="bday" min="2000-01-02"><br>

</form>

## 输入类型：color

**<input type="color">** 用于应该包含颜色的输入字段。

根据浏览器支持，颜色选择器会出现输入字段中。

### 实例

<form>

Select your favorite color:

<input type="color" name="favcolor">

</form>

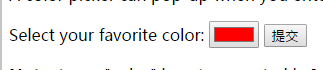
<form action="action\_page.php">

Select your favorite color:

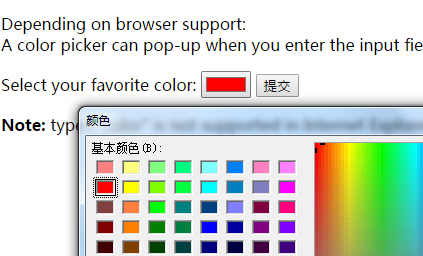
<input type="color" name="favcolor" value="#ff0000">

<input type="submit">

</form>



将提交颜色的RGB文本值，但是调用了系统的取色器



## 输入类型：range

**<input type="range">** 用于应该包含一定范围内的值的输入字段。

根据浏览器支持，输入字段能够显示为滑块控件。

### 实例

<form>

<input type="range" name="points" min="0" max="10">

</form>

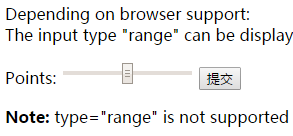
<form action="action\_page.php" method="get">

Points:

<input type="range" name="points" min="0" max="10">

<input type="submit">

</form>



通过滑块控件来设置input的值

**输入类型：email**

**<input type="email">** 用于应该包含电子邮件地址的输入字段。

根据浏览器支持，能够在被提交时自动对电子邮件地址进行验证。

某些智能手机会识别 email 类型，并在键盘增加 ".com" 以匹配电子邮件输入。

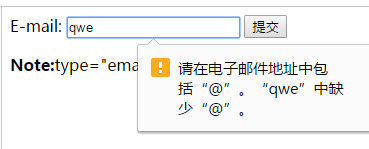
<form action="action\_page.php">

E-mail:

<input type="email" name="email">

<input type="submit">

</form>



**输入类型：url**

**<input type="url">** 用于应该包含 URL 地址的输入字段。

根据浏览器支持，在提交时能够自动验证 url 字段。

某些智能手机识别 url 类型，并向键盘添加 ".com" 以匹配 url 输入。

<form action="action\_page.php">

Add your homepage:

<input type="url" name="homepage">

<input type="submit">

</form>



# 6. HTML Input 属性

## value 属性

**value** 属性规定输入字段的初始值：

### 实例

<form action="">

First name:<br>

<input type="text" name="firstname" value="John">

<br>

Last name:<br>

<input type="text" name="lastname">

</form>

## readonly 属性

**readonly** 属性规定输入字段为只读（不能修改）：

### readonly 属性不需要值。它等同于 readonly="readonly"

### 实例

<form action="">

First name:<br>

<input type="text" name="firstname" value="John" readonly>

<br>

Last name:<br>

<input type="text" name="lastname">

</form>

## disabled 属性

**disabled** 属性规定输入字段是禁用的。

被禁用的元素是不可用和不可点击的。

被禁用的元素不会被提交。

### 实例

<form action="">

First name:<br>

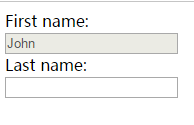
<input type="text" name="firstname" value="John" disabled>

<br>

Last name:<br>

<input type="text" name="lastname">

</form>



## size 属性

**size** 属性规定输入字段的尺寸（以字符计）：

### 实例

<form action="">

First name:<br>

<input type="text" name="firstname" value="John" size="40">

<br>

Last name:<br>

<input type="text" name="lastname">

</form>



## maxlength 属性

**maxlength** 属性规定输入字段允许的最大长度：

### 实例

<form action="">

First name:<br>

<input type="text" name="firstname" maxlength="10">

<br>

Last name:<br>

<input type="text" name="lastname">

</form>

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=html_input_attributes_maxlength)

如设置 maxlength 属性，则输入控件不会接受超过所允许数的字符。

该属性不会提供任何反馈。如果需要提醒用户，则必须编写 JavaScript 代码。

注释：输入限制并非万无一失。JavaScript 提供了很多方法来增加非法输入。如需安全地限制输入，则接受者（服务器）必须同时对限制进行检查。

**HTML5 属性**

HTML5 为 <input> 增加了如下属性：

* autocomplete
* autofocus
* form
* formaction
* formenctype
* formmethod
* formnovalidate
* formtarget
* height 和 width
* list
* min 和 max
* multiple
* pattern (regexp)
* placeholder
* required
* step

并为 <form> 增加如需属性：

* autocomplete
* novalidate

# 7. HTML <thead> 标签，代表表头

**实例**

带有 thead、tbody 以及 tfoot 元素的 HTML 表格：

<table border="1">

<thead>

<tr>

<th>Month</th>

<th>Savings</th>

</tr>

</thead>

<tfoot>

<tr>

<td>Sum</td>

<td>$180</td>

</tr>

</tfoot>

<tbody>

<tr>

<td>January</td>

<td>$100</td>

</tr>

<tr>

<td>February</td>

<td>$80</td>

</tr>

</tbody>

</table>