**HTML 表单**

**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。

<form> 元素包含一个或多个如下的表单元素：

* [<input>](https://www.runoob.com/tags/tag-input.html)
* [<textarea>](https://www.runoob.com/tags/tag-textarea.html)
* [<button>](https://www.runoob.com/tags/tag-button.html)
* [<select>](https://www.runoob.com/tags/tag-select.html)
* [<option>](https://www.runoob.com/tags/tag-option.html)
* [<optgroup>](https://www.runoob.com/tags/tag-optgroup.html)
* [<fieldset>](https://www.runoob.com/tags/tag-fieldset.html)
* [<label>](https://www.runoob.com/tags/tag-label.html)

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） |

**注释：**您稍后将在本教程学到更多有关输入类型的知识。

## 文本输入

**<input type="text">** 定义用于**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>

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

在浏览器中看起来是这样的：

窗体顶端

First name:  
  
  
Last name:  


窗体底端

**注释：**表单本身并不可见。还要注意文本字段的默认宽度是 20 个字符。

## 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>

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

单选按钮在浏览器看起来是这样的：

窗体顶端

Male  
  
Female

窗体底端

## 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>

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

在浏览器中看起来是这样的：

窗体顶端

First name:  
  
  
Last name:  
  
  


窗体底端

## 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">

## 何时使用 GET？

您能够使用 GET（默认方法）：

如果表单提交是被动的（比如搜索引擎查询），并且没有敏感信息。

当您使用 GET 时，表单数据在页面地址栏中是可见的：

action\_page.php?firstname=Mickey&lastname=Mouse

**注释：**GET 最适合少量数据的提交。浏览器会设定容量限制。

## 何时使用 POST？

您应该使用 POST：

如果表单正在更新数据，或者包含敏感信息（例如密码）。

POST 的安全性更好，因为在页面地址栏中被提交的数据是不可见的。

## 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>

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

## 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>

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

以上 HTML 代码在浏览器中看起来是这样的：

窗体顶端

Personal information:  
First name:  
  
  
Last name:  
  
  


窗体底端

## 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>

下面是 <form> 属性的列表：

|  |  |
| --- | --- |
| **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）。 |

**注释：**您将在下面的章节学到更多关于属性的知识。

* [XHTML 属性](https://www.w3school.com.cn/html/html_xhtml_attributes.asp)
* [HTML 表单属性](https://www.w3school.com.cn/html/html_forms_attributes.asp)

## 注意：radio按钮等等的表单元素必须要有value值，否则提交的页面无法获取数据，文本类的表达元素的value值可以提供一个默认值。文本类的表单元素还可以有一个占位符placeholder属性，这个属性不能和value的值同时存在。占位符的内容是不会提交的。单选按钮有一个checked属性可以设置按钮默认被选中，复选框

## 和单选按钮类似只是它允许多选。单选按钮是按照name属性来分类的，同一的类别只能选中一个。

## 表单例子

|  |
| --- |
|  |

## HTML5 新的表单元素

HTML5 有以下新的表单元素:

* <datalist> 相当于combobox，当用户点击后会有一个下拉列表可以选择
* <keygen>
* <output>

**注意:**不是所有的浏览器都支持HTML5 新的表单元素，但是你可以在使用它们，即使浏览器不支持表单属性，仍然可以显示为常规的表单元素。

## HTML5 <datalist> 元素

<datalist> 元素规定输入域的选项列表。

<datalist> 属性规定 form 或 input 域应该拥有自动完成功能。当用户在自动完成域中开始输入时，浏览器应该在该域中显示填写的选项：

使用 <input> 元素的列表属性与 <datalist> 元素绑定.

OperaSafariChromeFirefoxInternet Explorer

## 实例

<input> 元素使用<datalist>预定义值:

<input list="browsers">

<datalist id="browsers">

<option value="Internet Explorer">

<option value="Firefox">

<option value="Chrome">

<option value="Opera">

<option value="Safari">

</datalist>

[**尝试一下 »**](https://www.runoob.com/try/try.php?filename=tryhtml5_datalist)

## HTML5 <keygen> 元素

<keygen> 元素的作用是提供一种验证用户的可靠方法。

<keygen>标签规定用于表单的密钥对生成器字段。

当提交表单时，会生成两个键，一个是私钥，一个公钥。

私钥（private key）存储于客户端，公钥（public key）则被发送到服务器。公钥可用于之后验证用户的客户端证书（client certificate）。

OperaSafariChromeFirefoxInternet Explorer

## 实例

带有keygen字段的表单:

<form action="demo\_keygen.asp" method="get"> 用户名: <input type="text" name="usr\_name"> 加密: <keygen name="security"> <input type="submit"> </form>

[**尝试一下 »**](https://www.runoob.com/try/try.php?filename=tryhtml5_keygen)

## HTML5 <output> 元素

<output> 元素用于不同类型的输出，比如计算或脚本输出：

OperaSafariChromeFirefoxInternet Explorer

## 实例

将计算结果显示在 <output> 元素:

<form oninput="x.value=parseInt(a.value)+parseInt(b.value)">0 <input type="range" id="a" value="50">100 + <input type="number" id="b" value="50">= <output name="x" for="a b"></output> </form>

[**尝试一下 »**](https://www.runoob.com/try/try.php?filename=tryhtml5_output)

## HTML5 新表单元素

|  |  |
| --- | --- |
| **标签** | **描述** |
| [<datalist>](https://www.runoob.com/tags/tag-datalist.html) | <input>标签定义选项列表。请与 input 元素配合使用该元素，来定义 input 可能的值。 |
| [<keygen>](https://www.runoob.com/tags/tag-keygen.html) | <keygen> 标签规定用于表单的密钥对生成器字段。 |
| [<output>](https://www.runoob.com/tags/tag-output.html) | <output> 标签定义不同类型的输出，比如脚本的输出。 |

## input标签也有新内容：

 <input type="color" name="color"> 给用户提供一个可以选择颜色的色块,需要有name属性否则颜色值提交不到

效果：

|  |
| --- |
|  |

# <input type="url">

**url** 类型的 [<input>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input) 元素用来让用户输入和编辑 URL。

## [尝试一下](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/url#%E5%B0%9D%E8%AF%95%E4%B8%80%E4%B8%8B)

所输入的值在表单提交前会自动经过验证，以确认它为空，或为一个合法的 URL 格式。[:valid](https://developer.mozilla.org/zh-CN/docs/Web/CSS/:valid) 和 [:invalid](https://developer.mozilla.org/zh-CN/docs/Web/CSS/:invalid) CSS 伪类会适当地自动应用，以在视觉上表示当前值是否为一个合法的 URL 值。

对于不支持类型为 url 的输入框的浏览器，它会回退为标准的 [text](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/text) 输入框。

|  |  |
| --- | --- |
| [**值**](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/url#%E5%80%BC) | 代表 URL 的字符串，或为空 |
| **事件** | [change](https://developer.mozilla.org/zh-CN/docs/Web/API/HTMLElement/change_event) 和 [input](https://developer.mozilla.org/zh-CN/docs/Web/API/HTMLElement/input_event) |
| **支持的通用属性** | [autocomplete](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#autocomplete)、[list](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#list)、[maxlength](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#maxlength)、[minlength](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#minlength)、[pattern](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#pattern)、[placeholder](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#placeholder)、[readonly](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#readonly)、[required](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#required) 和 [size](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#size) |
| **IDL 属性** | list、value、selectionEnd 和 selectionDirection |
| **DOM 接口** | [HTMLInputElement](https://developer.mozilla.org/zh-CN/docs/Web/API/HTMLInputElement) |
| **方法** | [select()](https://developer.mozilla.org/zh-CN/docs/Web/API/HTMLInputElement/select)、[setRangeText() (en-US)](https://developer.mozilla.org/en-US/docs/Web/API/HTMLInputElement/setRangeText) 和 [setSelectionRange()](https://developer.mozilla.org/zh-CN/docs/Web/API/HTMLInputElement/setSelectionRange)。 |

## [值](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/url#%E5%80%BC)

[<input>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input) 元素的 [value](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#value) 属性包含了自动验证为 URL 格式的字符串。具体来说，有两种可能的格式可以通过验证：

1. 一个空字符串（""），代表用户没有输入任何值，或者有值被移除。
2. 一个正确的绝对 URL 格式。这并不代表 URL 所指地址实际存在，但至少要符合格式。简单来说，要类似于 urlscheme://restofurl。

查看[验证](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/url#%E9%AA%8C%E8%AF%81)小节，以了解 URL 如何经过验证来确保格式正确性。

## [附加属性](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/url#%E9%99%84%E5%8A%A0%E5%B1%9E%E6%80%A7)

除了所有 [<input>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input) 元素上共有的可操作的属性外，url 输入框还支持以下属性：

### [list](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/url#list)

list 属性指定了一个 [<datalist>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/datalist) 元素的 id，用来提供输入建议。[<datalist>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/datalist) 提供了一个预定义的值列表，向用户建议这个输入。列表中任何与 [type](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#type) 不兼容的值都不包括在建议选项中。所提供的值是建议，不是要求：用户可以从这个预定义的列表中选择，或者提供不同的值。

### [maxlength](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/url#maxlength)

用户可以输入到 url 输入框中的最大字符数（以 UTF-16 代码单元为单位）。必须为大于等于 0 的整数。如果未指定 maxlength 或指定了无效的值，则 url 输入框将没有最大值。这个值也必须大于等于 minlength 的值。

如果文本框中的字符数大于 maxlength UTF-16 编码单元长度，则输入将无法通过[约束验证](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Constraint_validation)。约束验证仅作用于用户修改输入值的时候。

### [minlength](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/url#minlength)

用户可以输入到 url 输入框中的最小字符数（以 UTF-16 代码单元为单位）。该值必须是小于等于 maxlength 指定的值的非负整数值。如果未指定 minlength 或指定了无效的值，则 url 输入将没有最小值。

如果输入到字段中的文本的长度小于 minlength UTF-16 代码单元的长度，则输入将无法通过[约束验证](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Constraint_validation)。约束验证仅作用于用户修改输入值的时候。

### [pattern](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/url#pattern)

如果指定了 pattern 属性，为了使 value 通过[约束验证](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Constraint_validation)，必须满足该属性给定的正则表达式。它必须是 [RegExp](https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Global_Objects/RegExp) 类型的有效 JavaScript 正则表达式，并且已在我们的[正则表达式指南](https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Guide/Regular_expressions)中进行了说明；在编译正则表达式时指定了 'u' 标志，因此该模式被视为 Unicode 代码点的序列，而不是 ASCII。模式文本周围无需指定正斜杠。

如果模式未指定或无效，则不应用任何正则表达式，并且将完全忽略此属性。

**备注：** 使用 [title](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#title) 属性指定大多数浏览器将显示为文本的工具提示，以说明与模式匹配的要求。你还应该在附近添加其他说明性文字。

参见[模式验证](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/url#%E6%A8%A1%E5%BC%8F%E9%AA%8C%E8%AF%81)小节以获取细节和例子。

### [placeholder](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/url#placeholder)

placeholder 属性是一个字符串，可向用户提供有关该字段中需要什么样的信息的简短提示。它应该是一个单词或短语来说明预期的数据类型，而不是说明性消息。文本中不得包含回车符或换行符。

如果控件的内容具有方向（[LTR](https://developer.mozilla.org/zh-CN/docs/Glossary/LTR) 或 [RTL](https://developer.mozilla.org/zh-CN/docs/Glossary/RTL)），但需要以相反的方向显示占位符，则可以使用 Unicode 双向算法来格式化字符，从而覆盖原有占位符的方向；请参见[如何针对双向文本使用 Unicode 控制符](https://www.w3.org/International/questions/qa-bidi-unicode-controls)获取更多信息。

**备注：** 尽可能避免使用 placeholder 属性。它在语义上没有其他解释表单的方式有用，并且可能导致内容出现意外的问题。请参见 [<input> 标签](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#%E6%A0%87%E7%AD%BE)以获取更多信息。

### [readonly](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/url#readonly)

一个布尔属性，如果存在，则表示该字段不能由用户编辑。但是，仍可以通过 JavaScript 代码直接设置 [HTMLInputElement](https://developer.mozilla.org/zh-CN/docs/Web/API/HTMLInputElement) 的 value 属性来更改。

**备注：** 因为只读字段不能有值，所以 required 对指定了 readonly 属性的输入没有任何影响。

### [size](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/url#size)

size 属性是一个数字值，指示输入字段应有多少个字符宽。该值必须是一个大于零的数字，默认值是 20。由于字符宽度各不相同，这可能是也可能不是精确的，不应依赖于此；结果输入可能比指定的字符数更窄或更宽，这取决于字符和字体（[font](https://developer.mozilla.org/zh-CN/docs/Web/CSS/font) 使用中的设置）。

这并不限制用户可以在该字段中输入多少个字符，仅能指定一次可以看到多少个。要设置输入数据长度的上限，请使用 [maxlength](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/url#maxlength) 属性。

### [spellcheck](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/url#spellcheck)

spellcheck 是一个全局属性，用于指示是否启用元素的拼写检查。它可以用于任何可编辑的内容，但是这里我们考虑在 [<input>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input) 元素上使用 spellcheck 的细节。 spellcheck 的允许值为：

false

禁用此元素的拼写检查。

true

对此元素启用拼写检查。

""（空字符串）或空值

遵循元素的默认行为进行拼写检查。这可能基于父元素 spellcheck 设置或其他因素。

如果输入字段没有设置 [readonly](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/url#readonly) 属性且未禁用，则可以启用拼写检查。

如果[用户代理的](https://developer.mozilla.org/zh-CN/docs/Glossary/User_agent)首选项覆盖了设置，则通过读取 spellcheck 返回的值可能无法反映控件中拼写检查的实际状态。

# <input type="date">

**type="date"** 类型的 [<input>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input) 元素会创建一个让用户输入一个日期的输入区域，可以使用自动验证内容的文本框，也可以使用特殊的日期选择器界面。结果值包括年份，月份和日期，但不包括时间。[time](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/time) 和 [datetime-local](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/datetime-local) 类型支持时间和日期时间输入。

## [尝试一下](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/date#%E5%B0%9D%E8%AF%95%E4%B8%80%E4%B8%8B)

通常来说控件的 UI 界面因浏览器的不同而有变化，到目前为止此控件还不被所有浏览器支持，具体细节请参阅[浏览器兼容性](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/date#%E6%B5%8F%E8%A7%88%E5%99%A8%E5%85%BC%E5%AE%B9%E6%80%A7)。在不支持的浏览器当中，控件因此会被优雅的降级为普通的 [<input type="text">](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/text) 输入框。

|  |  |
| --- | --- |
| [**值**](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/date#%E5%80%BC) | 按照 YYYY-MM-DD 格式化过的代表日期的字符串，或者为空字符串 |
| **事件** | [change](https://developer.mozilla.org/zh-CN/docs/Web/API/HTMLElement/change_event) 事件和 [input](https://developer.mozilla.org/zh-CN/docs/Web/API/HTMLElement/input_event) 事件 |
| **支持的常用属性** | [autocomplete](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#autocomplete)、[list](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#list)、[readonly](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#readonly) 和 [step](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#step) |
| **IDL 属性** | list、value、valueAsDate 和 valueAsNumber |
| **DOM 接口** | [HTMLInputElement](https://developer.mozilla.org/zh-CN/docs/Web/API/HTMLInputElement) |
| **方法** | [select()](https://developer.mozilla.org/zh-CN/docs/Web/API/HTMLInputElement/select)、[stepDown() (en-US)](https://developer.mozilla.org/en-US/docs/Web/API/HTMLInputElement/stepDown) 和 [stepUp() (en-US)](https://developer.mozilla.org/en-US/docs/Web/API/HTMLInputElement/stepUp) |

## [值](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/date#%E5%80%BC)

一个字符串，代表着输入到输入框的日期值。输入的日期按照[日期字符串格式](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Date_and_time_formats#%E6%97%A5%E6%9C%9F%E5%AD%97%E7%AC%A6%E4%B8%B2)所描述的 ISO8601 标准进行格式化。

你可以通过在 [value](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#value) 属性中包含日期来为输入设置默认值，如下所示：

<input type="date" value="2017-06-01" />

Copy to Clipboard

**备注：** 有一点需要注意的是，在格式方面显示的日期与实际的 value 不一样——显示的日期格式取决于用户浏览器的区域设定，而经解析的 value 的格式始终为 yyyy-mm-dd。

当然你也可以通过 JavaScript 代码获取和设置 [HTMLInputElement](https://developer.mozilla.org/zh-CN/docs/Web/API/HTMLInputElement) 的 value 和 valueAsNumber 属性，例如：

const dateControl = document.querySelector('input[type="date"]');

dateControl.value = '2017-06-01';

console.log(dateControl.value); // prints "2017-06-01"

console.log(dateControl.valueAsNumber); // prints 1496275200000, a JavaScript timestamp (ms)

Copy to Clipboard

这行代码查找 type 为 date 的第一个 [<input>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input) 元素，并且将其值设置为 2017-06-01（2017 年 6 月 1 日）。稍后将以字符串和数字的形式读取出这个值。

## [其他属性](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/date#%E5%85%B6%E4%BB%96%E5%B1%9E%E6%80%A7)

对于所有 [<input>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input) 元素的大部分通用属性也对 date 输入起作用，但可能不会影响它的表现，如 size 和 placeholder 属性。除了共有属性外，date 输入还提供以下属性。

### [max](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/date#max)

所接受最新的日期。如果输入到该元素中的 [value](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#value) 发生在此之后，则元素将无法通过[约束验证](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Constraint_validation)。如果 max 属性的值不是格式为 yyyy-mm-dd 的有效日期星期字符串，则该元素没有最大日期值。

如果同时设置了 max 和 min 值，此值必须**晚于或等于** min 属性指定的日期值。

### [min](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/date#min)

所接受最早的日期。如果输入到该元素中的 [value](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#value) 发生在此之前，则元素将无法通过[约束验证](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Constraint_validation)。如果 min 属性的值不是格式为 yyyy-mm-dd 的有效日期星期字符串，则该元素没有最小日期值。

如果同时设置了 max 和 min 值，此值必须**早于或等于** max 属性指定的日期值。

### [step](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/date#step)

step 属性指定了值必须满足的粒度，或者是下文描述的特殊值 any。值必须满足基础的步进值，才有效。如果指定了 [min](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/date#min) 属性，则由 min 属性决定，否则，使用 [value](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#value) 属性的值，如果上述两个值都不存在，则提供适当的默认值。

字符串值 any 意味着不使用步进值，任意值都可以接受（除其他制约因素如 [min](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/date#min) 或 [max](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/date#max) 之外）。

**备注：** 当用户输入的数据不符合步进配置时，[用户代理](https://developer.mozilla.org/zh-CN/docs/Glossary/User_agent)可能会四舍五入到最近的有效值，当有两个同样接近的选项时，更倾向于正方向的数字。

对于 date 输入，step 的值以天为单位，并被视为等于 86,400,000 乘以 step 值的毫秒数（基础数值是毫秒）。step 的默认值为 1，表示 1 天。

**备注：** 将 date 输入的 step 值指定为 any 与指定为 1 的效果相同。

## [使用日期输入控件](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/date#%E4%BD%BF%E7%94%A8%E6%97%A5%E6%9C%9F%E8%BE%93%E5%85%A5%E6%8E%A7%E4%BB%B6)

日期控件提供了一个简单的日期选择 UI 界面，还标准化了发送到服务器的数据格式，而与用户的浏览器或区域设置无关。

在本章，我们将研究 <input type="date"> 基础的和更复杂的用法。

### [日期控件基本用法](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/date#%E6%97%A5%E6%9C%9F%E6%8E%A7%E4%BB%B6%E5%9F%BA%E6%9C%AC%E7%94%A8%E6%B3%95)

<input> 和 [<label>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/label) 元素组合是 <input type="date"> 最简单的使用方法，如下所示：

<form action="https://example.com">

<label>

Enter your birthday:

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

</label>

<p><button>Submit</button></p>

</form>

Copy to Clipboard

这段 HTML 代码将所输入的日期作为 bday 的参数发送至 https://example.com，获得了像这样的 URL https://example.com/?bday=1955-06-08 。

### [设置日期最大和最小值](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/date#%E8%AE%BE%E7%BD%AE%E6%97%A5%E6%9C%9F%E6%9C%80%E5%A4%A7%E5%92%8C%E6%9C%80%E5%B0%8F%E5%80%BC)

你可以通过 [min](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#min) 和 [max](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#max) 属性去限制用户的可选日期范围。在随后的例子中，我们将设定日期最小值为 2017-04-01，最大值为 2017-04-30：

<form>

<label>

Choose your preferred party date:

<input type="date" name="party" min="2017-04-01" max="2017-04-30" />

</label>

</form>

Copy to Clipboard

在结果中我们可以看到，只有 2017 年 4 月份的日期可选——输入框中可以编辑的部分只有“日”这部分，并且超出 4 月份以外的日期不能通过日期控件的选择组件选择。

**备注：** 你应该可以使用 [step](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#step) 属性来改变每次最佳日期时步进（增加值）的天数（例如：或许你只希望使周六可以选）。但是，在撰写这篇文章的时候，还没有可靠的实现。

### [控制输入框大小](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/date#%E6%8E%A7%E5%88%B6%E8%BE%93%E5%85%A5%E6%A1%86%E5%A4%A7%E5%B0%8F)

<input type="date"> 不支持诸如 [size](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#size) 的表单大小属性。你必须依靠 [CSS](https://developer.mozilla.org/zh-CN/docs/Web/CSS) 来确定大小。

## [验证](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/date#%E9%AA%8C%E8%AF%81)

默认情况下，<input type="date"> 对输入的值不会做任何校验。接口实现通常不会让你输入任何非日期的东西（这一点很有帮助），但是你仍然可以留空或者（在被优雅降级为 text 类型的输入框时）输入一个不合法的值（例如 4 月 32 日）。

如果你使用 [min](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#min) 和 [min](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#min) 属性去限制可用日期（参见[设置日期最大和最小值](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/date#%E8%AE%BE%E7%BD%AE%E6%97%A5%E6%9C%9F%E6%9C%80%E5%A4%A7%E5%92%8C%E6%9C%80%E5%B0%8F%E5%80%BC)），对于支持的浏览器来说如果你尝试提交一个超出给定范围的日期，那么它将抛出一个错误。然而，你必须检查这些结果以确保他们在这些日期范围内，因为只有在用户设备上完全支持日期选择器的情况下，才能执行这些操作。

另外，你可以使用 [required](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#required) 属性强制填写日期，如果你尝试提交一个未填写日期的字段，将会抛出错误。即使优雅降级为 text 类型输入框，在大多数浏览器也是可以工作的。

让我们看一个例子——我们设置了日期的最大和最小值，并且将日期字段设定为必填：

<form>

<label>

Choose your preferred party date (required, April 1st to 20th):

<input

type="date"

name="party"

min="2017-04-01"

max="2017-04-20"

required />

<span class="validity"></span>

</label>

<p>

<button>Submit</button>

</p>

</form>

Copy to Clipboard

如果你尝试提交一个不完整日期的表单（或者超出日期选择设定范围），浏览器将会显示一个错误。尝试一下这个例子：

这是上面例子使用的 CSS。我们基于 [:valid](https://developer.mozilla.org/zh-CN/docs/Web/CSS/:valid) 和 [:invalid](https://developer.mozilla.org/zh-CN/docs/Web/CSS/:invalid) [伪类](https://developer.mozilla.org/zh-CN/docs/Web/CSS/Pseudo-elements)来在输入框旁边添加小图标。我们必须把图标放在 input 旁边的 [<span>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/span) 里面，而并不是它本身，因为在 Chrome 中被放置在表单中生成的内容不能有效的样式化或者显示。

label {

display: flex;

align-items: center;

}

span::after {

padding-left: 5px;

}

input:invalid + span::after {

content: "✖";

}

input:valid + span::after {

content: "✓";

}

Copy to Clipboard

**警告：** 客户端表单验证不能替代服务端验证。对于某人来说，对 HTML 进行调整以使其绕过验证或完全删除验证太容易了，甚至也可以完全绕开 HTML 并将数据直接提交到服务器。如果服务器端代码无法验证其接收到的数据，则在提交格式不正确的（或太大，类型错误……）的数据时，灾难可能会发生。

# <input type="email">

**"email"** 类型的 [<input>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input) 元素能够让用户输入或编辑一个电子邮箱地址，如果指定了 [multiple (en-US)](https://developer.mozilla.org/en-US/docs/Web/HTML/Attributes/multiple) 属性，则可以输入多个电子邮箱地址。

## [尝试一下](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/email#%E5%B0%9D%E8%AF%95%E4%B8%80%E4%B8%8B)

在表单提交前，输入框会自动验证输入值是否是一个或多个合法的电子邮箱地址（非空值且符合电子邮箱地址格式）。CSS 伪标签 [:valid](https://developer.mozilla.org/zh-CN/docs/Web/CSS/:valid) 和 [:invalid](https://developer.mozilla.org/zh-CN/docs/Web/CSS/:invalid) 能够在校验后自动应用，代表被校验的邮箱地址是否合法。

## [值](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/email#%E5%80%BC)

[<input>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input) 元素的 [value](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#value) 属性包含会自动验证是否为正确邮件地址形式的字符串。具体来说，此处有三种格式可以通过验证：

1. 一个空字符串（“”），表示用户未输入值或该值已被删除。
2. 单个符合格式的电子邮件地址。并不代表该邮件地址存在，而是说它至少在格式上是正确的，简单地讲就是 username@domain 或者 username@domain.tld，当然并不局限于此，关于匹配电子邮件地址验证算法的[正则表达式](https://developer.mozilla.org/zh-CN/docs/Glossary/Regular_expression)，参见[验证](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/email#%E9%AA%8C%E8%AF%81)一节。
3. 当且仅当指定了 [multiple](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#multiple) 属性时，值可以是一个列表，该列表包含一串符合格式的电子邮件地址，且每个地址之间用逗号分隔。列表中，位于每个地址之前和之后的任何空白字符都会被移除。

关于如何验证电子邮件地址以保证其格式正确的细节，参见[验证](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/email#%E9%AA%8C%E8%AF%81)一节。

## [额外属性](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/email#%E9%A2%9D%E5%A4%96%E5%B1%9E%E6%80%A7)

除了所有 [<input>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input) 元素支持的属性以外，email 输入还支持以下属性：

### [list](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/email#list)

list 属性指定了一个 [<datalist>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/datalist) 元素的 [id](https://developer.mozilla.org/zh-CN/docs/Web/API/Element/id)，用来提供输入建议。[<datalist>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/datalist) 提供了一个预定义的值列表，向用户建议这个输入。列表中任何与 [type](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#type) 不兼容的值都不包括在建议选项中。所提供的值是建议，不是要求：用户可以从这个预定义的列表中选择，或者提供不同的值。

### [maxlength](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/email#maxlength)

用户可以输入到 email 输入框中的最大字符数（以 UTF-16 代码单元为单位）。必须为大于等于 0 的整数。如果未指定 maxlength 或指定了无效的值，则 email 输入框将没有最大长度。这个值也必须大于等于 minlength 的值。

如果文本框中的字符数大于 maxlength UTF-16 编码单元长度，则输入将无法通过[约束验证](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Constraint_validation)。约束验证仅作用于用户修改输入值的时候。

### [minlength](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/email#minlength)

用户可以输入到 email 输入框中的最小字符数（以 UTF-16 代码单元为单位）。该值必须是小于等于 maxlength 指定的值的非负整数值。如果未指定 minlength 或指定了无效的值，则 email 输入将没有最小长度。

如果输入到字段中的文本的长度小于 minlength UTF-16 代码单元的长度，则输入将无法通过[约束验证](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Constraint_validation)。约束验证仅作用于用户修改输入值的时候。

### [multiple](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/email#multiple)

一个布尔属性，如果存在，代表用户可以输入多个由逗号和可选空白字符分开的电子邮件地址。参见示例[允许多个邮件地址](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/email#%E5%85%81%E8%AE%B8%E5%A4%9A%E4%B8%AA%E9%82%AE%E4%BB%B6%E5%9C%B0%E5%9D%80)，或 [HTML 属性: multiple (en-US)](https://developer.mozilla.org/en-US/docs/Web/HTML/Attributes/multiple) 一文以获取更多信息。

**备注：** 一般地，如果你指定了 [required](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#required) 属性，那么用户将必须输入一个合法的邮件地址。然而，如果你添加了 multiple 属性，一系列零地址（一个空字符串，或完全为空白字符的字符串）也是合法的。也就是说，当指定了 multiple 时，用户甚至不需要输入一个邮件地址，无论是否指定了 required。

### [pattern](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/email#pattern)

如果指定了 pattern 属性，为了使 value 通过[约束验证](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Constraint_validation)，必须满足该属性给定的正则表达式。它必须是 [RegExp](https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Global_Objects/RegExp) 类型的有效 JavaScript 正则表达式，并且已在我们的[正则表达式指南](https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Guide/Regular_expressions)中进行了说明；在编译正则表达式时指定了 'u' 标志，因此该模式被视为 Unicode 代码点的序列，而不是 ASCII。模式文本周围无需指定正斜杠。

如果没有指定模式，或指定了无效的模式，则不会应用任何正则表达式，且该属性将被完全忽略。

**备注：** 使用 [title](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#title) 属性来指定大多数浏览器将作为工具提示的文本，以解释匹配该模式的要求是什么。还应该在附近包括其他解释性的文本。

参阅[模式验证](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/email#%E6%A8%A1%E5%BC%8F%E9%AA%8C%E8%AF%81)一节以获取详细说明和示例。

### [placeholder](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/email#placeholder)

placeholder 属性是一个字符串，可向用户提供有关该字段中需要什么样的信息的简短提示。它应该是一个单词或短语来说明预期的数据类型，而不是说明性消息。文本中不得包含回车符或换行符。

如果控件的内容具有方向（[LTR](https://developer.mozilla.org/zh-CN/docs/Glossary/LTR) 或 [RTL](https://developer.mozilla.org/zh-CN/docs/Glossary/RTL)），但需要以相反的方向显示占位符，则可以使用 Unicode 双向算法来格式化字符，从而覆盖原有占位符的方向；请参见[如何针对双向文本使用 Unicode 控制符](https://www.w3.org/International/questions/qa-bidi-unicode-controls)获取更多信息。

**备注：** 尽可能避免使用 placeholder 属性。它在语义上没有其他解释表单的方式有用，并且可能导致内容出现意外的问题。请参见 [<input> 标签](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#%E6%A0%87%E7%AD%BE)以获取更多信息。

### [readonly](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/email#readonly)

一个布尔属性，如果存在，则表示该字段不能由用户编辑。但是，仍可以通过 JavaScript 代码直接设置 [HTMLInputElement](https://developer.mozilla.org/zh-CN/docs/Web/API/HTMLInputElement) 的 value 属性来更改。

**备注：** 因为只读字段不能有值，所以 required 对指定了 readonly 属性的输入没有任何影响。

### [size](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/email#size)

size 属性是一个数字值，指示输入字段应有多少个字符宽。该值必须是一个大于零的数字，默认值是 20。由于字符宽度各不相同，这可能是也可能不是精确的，不应依赖于此；结果输入可能比指定的字符数更窄或更宽，这取决于字符和字体（[font](https://developer.mozilla.org/zh-CN/docs/Web/CSS/font) 使用中的设置）。

这并不限制用户可以在该字段中输入多少个字符，仅能指定一次可以看到多少个。要设置输入数据长度的上限，请使用 [maxlength](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/email#maxlength) 属性。

## [使用 email 输入](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/email#%E4%BD%BF%E7%94%A8_email_%E8%BE%93%E5%85%A5)

电子邮件地址是网络上最经常输入的文本数据形式之一；它们被用于登录网站、请求信息、允许订单确认、网络邮件等等。因此，email 输入类型可以使你作为一个网络开发者的工作变得更加容易，因为它可以帮助你在建立用户界面和电子邮件地址的逻辑时简化工作。当你用适当的 type 值创建一个电子邮件输入时，你会得到自动验证，即输入的文本至少是正确的形式，可能是一个合法的电子邮件地址。这可以帮助避免用户错误输入他们的地址，或提供一个无效的地址。

然而，重要的是要注意，这不足以确保指定的文本是实际存在的电子邮件地址，对应于站点的用户，或者以任何其他方式可接受。它只是确保将字段的值正确格式化为电子邮件地址。

**备注：** 记住用户可以在幕后修改 HTML 也很重要，因此你的站点不得出于任何安全目的使用此验证。在任何事务中，如果所提供的文本可能有任何形式的安全影响，你必须在服务器端验证该电子邮件地址。

### [一个简单的 email 输入](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/email#%E4%B8%80%E4%B8%AA%E7%AE%80%E5%8D%95%E7%9A%84_email_%E8%BE%93%E5%85%A5)

目前，所有实现此元素的浏览器都将其实现为具有基本验证功能的标准文本输入字段。但是，该规范允许浏览器有一定的自由度。例如，该元素可以与用户设备的内置地址簿集成，以允许从该列表中挑选电子邮件地址。在最基本的形式中，电子邮件输入可以像这样实现：

<input id="emailAddress" type="email" />

Copy to Clipboard

请注意，在空的时候和输入一个有效格式的电子邮件地址的时候，它被认为是有效的，但在其他情况下，则不被认为是有效的。通过添加 [required](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#required) 属性，只允许输入有效格式的电子邮件地址；当输入为空时不再被视为有效。

### [允许多个邮件地址](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/email#%E5%85%81%E8%AE%B8%E5%A4%9A%E4%B8%AA%E9%82%AE%E4%BB%B6%E5%9C%B0%E5%9D%80)

通过添加 [multiple](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#multiple) 布尔属性，输入框可以接受多个邮件地址。

<input id="emailAddress" type="email" multiple />

Copy to Clipboard

现在，当输入一个电子邮件地址时，或当任何数量的电子邮件地址由逗号分隔，并可选地含有空白字符时，输入被认为是有效的。

**备注：** 当使用了 multiple 属性时，值允许为空。

当指定了 multiple 属性时，一些有效字符串示例如下：

* ""
* "me@example"
* "me@example.org"
* "me@example.org,you@example.org"
* "me@example.org, you@example.org"
* "me@example.org,you@example.org, us@example.org"

一些无效字符串示例如下：

* ","
* "me"
* "me@example.org you@example.org"

### [占位符](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/email#%E5%8D%A0%E4%BD%8D%E7%AC%A6)

有时，提供一个上下文提示，说明输入数据应该采取什么形式，是很有帮助的。如果页面设计没有为每个 [<input>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input) 提供描述性的标签，这一点就特别重要。这就是**占位符**的作用。占位符是一个值，它通过展示一个有效值的例子来证明 value 应该采取的形式，当元素的 value 为 "" 时，它将显示在编辑框内。一旦数据被输入框中，占位符就会消失；如果框被清空，占位符会重新出现。

这里，我们有一个含有占位符 sophie@example.com 的 email 输入框。请注意在改变编辑框的内容时占位符如何消失和重新出现。

<input type="email" placeholder="sophie@example.com" />

Copy to Clipboard

### [控制输入框大小](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/email#%E6%8E%A7%E5%88%B6%E8%BE%93%E5%85%A5%E6%A1%86%E5%A4%A7%E5%B0%8F)

不仅可以控制输入框的物理长度，还可以控制输入文本本身允许的最小和最大长度。

#### 输入框物理大小

输入框的物理尺寸可以用 [size](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#size) 属性来控制。通过它，你可以指定输入框一次可以显示的字符数。在这个例子中，email 编辑框有 15 个字符宽：

<input type="email" size="15" />

Copy to Clipboard

#### 元素值的长度

size 与输入的电子邮件地址本身的长度限制是分开的，所以你可以让字段在一个小空间里适应，同时仍然允许输入更长的电子邮件地址字符串。你可以使用 [minlength](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#minlength) 属性为输入的电子邮件地址指定一个最小长度，以字符为单位；同样，使用 [maxlength](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#maxlength) 来设置输入电子邮件地址的最大长度。

下面的例子创建了一个 32 个字符宽的电子邮件地址输入框，要求内容不能短于 3 个字符，不能长于 64 个字符。

<input type="email" size="32" minlength="3" maxlength="64" />

Copy to Clipboard

### [提供默认选项](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/email#%E6%8F%90%E4%BE%9B%E9%BB%98%E8%AE%A4%E9%80%89%E9%A1%B9)

#### 使用 value 属性提供单个默认值

像往常一样，你可以通过设置 [value](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#value) 属性，为 email 输入框提供一个默认值：

<input type="email" value="default@example.com" />

Copy to Clipboard

#### 提供推荐值列表

再进一步，你可以提供一个默认选项的列表，用户可以通过指定 [list](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#list) 属性从中选择。这并不限制用户使用这些选项，但可以让他们更快地选择常用的电子邮件地址。这也为 [autocomplete](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#autocomplete) 提供提示。list 属性指定了 [<datalist>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/datalist) 的 ID，它又包含了每个建议值的一个 [<option>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/option) 元素；每个 option 的 value 是电子邮件输入框的对应建议值。

<input type="email" size="40" list="defaultEmails" />

<datalist id="defaultEmails">

<option value="jbond007@mi6.defence.gov.uk"></option>

<option value="jbourne@unknown.net"></option>

<option value="nfury@shield.org"></option>

<option value="tony@starkindustries.com"></option>

<option value="hulk@grrrrrrrr.arg"></option>

</datalist>

Copy to Clipboard

有了 [<datalist>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/datalist) 元素及其 [<option>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/option)，浏览器将提供指定的值作为电子邮件地址的潜在值；这通常以弹出式或下拉式菜单的形式呈现，其中包含建议。虽然具体的用户体验可能因浏览器而异，但通常情况下，点击编辑框就会出现建议的电子邮件地址的下拉菜单。然后，当用户输入时，该列表被过滤，只显示匹配的值。每个键入的字符都会缩小列表的范围，直到用户做出选择或键入一个自定义值。

## [验证](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/email#%E9%AA%8C%E8%AF%81)

email 输入有两个级别的内容验证。首先，所有 [<input>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input) 都有标准的验证级别，它自动确保内容符合有效电子邮件地址的要求。但也可以选择添加额外的过滤功能，以确保满足你自己的特殊需求。

**警告：** HTML 表单验证不能替代确保输入的数据格式正确的脚本。对于某些人来说，调整 HTML 非常容易，因为它允许他们绕过验证，或者完全删除它。某人也可以完全绕过 HTML 并将数据直接提交给您的服务器。如果服务器端代码无法验证它收到的数据，则当数据格式不正确（或数据太大，类型错误等等）输入数据库时，灾难可能会发生。

### [基本验证](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/email#%E5%9F%BA%E6%9C%AC%E9%AA%8C%E8%AF%81)

浏览器自动提供验证，以确保只有符合互联网电子邮件地址标准格式的文本被输入到输入框。浏览器使用一种相当于以下正则表达式的算法：

/^[a-zA-Z0-9.!#$%&'\*+/=?^\_`{|}~-]+@[a-zA-Z0-9](?:[a-zA-Z0-9-]{0,61}[a-zA-Z0-9])?(?:\.[a-zA-Z0-9](?:[a-zA-Z0-9-]{0,61}[a-zA-Z0-9])?)\*$/

Copy to Clipboard

要了解更多关于表单验证的工作原理，以及如何利用 [:valid](https://developer.mozilla.org/zh-CN/docs/Web/CSS/:valid) 和 [:invalid](https://developer.mozilla.org/zh-CN/docs/Web/CSS/:invalid) 这两个 CSS 属性，根据当前值是否有效来设计输入的样式，请参阅[表单数据验证 (en-US)](https://developer.mozilla.org/en-US/docs/Learn/Forms/Form_validation)一文。

**备注：** 有一些已知的规范问题与国际域名和 HTML 中电子邮件地址的验证有关。参阅 [W3C bug 15489](https://www.w3.org/Bugs/Public/show_bug.cgi?id=15489) 以了解详细内容。

### [模式验证](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/email#%E6%A8%A1%E5%BC%8F%E9%AA%8C%E8%AF%81)

如果你需要进一步限制输入的邮件地址，不仅仅需要“任何看起来像邮件地址的字符串”，你可以使用 [pattern](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#pattern) 属性来指定一个[正则表达式](https://developer.mozilla.org/zh-CN/docs/Glossary/Regular_expression)，该值必须与之匹配才有效。如果指定了 [multiple](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#multiple) 属性，每个由逗号分开的值必须满足该[正则表达式](https://developer.mozilla.org/zh-CN/docs/Glossary/Regular_expression)。

例如，假设你正在为 Best Startup Ever, Inc. 公司的员工建立一个支持页面，如果他们的某个页面出现问题，他们可以联系 IT 部门寻求帮助。在我们的简化表单中，用户需要输入出现他们的邮箱地址，以及描述遇到的问题。为了安全起见，我们希望用户不仅提供了有效的邮箱地址，而且还是一个内部邮件地址。

由于 email 类型的输入会对标准邮箱地址和自定义 [pattern](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#pattern) 同时进行校验，你可以很方便地实现这些。我们来看看如何这样做：

<form>

<div class="emailBox">

<label for="emailAddress">你的邮箱地址</label><br />

<input

id="emailAddress"

type="email"

size="64"

maxlength="64"

required

placeholder="username@beststartupever.com"

pattern=".+@beststartupever\.com"

title="请仅提供 Best Startup Ever 公司邮件地址" />

</div>

<div class="messageBox">

<label for="message">请求</label><br />

<textarea

id="message"

cols="80"

rows="8"

required

placeholder="我的鞋子太紧了，而且我忘记如何跳舞了。"></textarea>

</div>

<input type="submit" value="发送请求" />

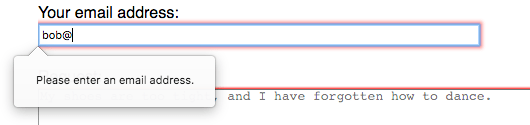
</form>

Copy to Clipboard

我们的 [<form>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/form) 表单包含一个 email 类型的 [<input>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input)，用于用户电子邮件地址，一个 [<textarea>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/textarea)，用来输入他们的 IT 信息，以及一个 <input> 类型为 ["submit"](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/submit) 的按钮，用来创建一个提交表单。每个文本输入框都有一个 [<label>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/label) 与之相关，让用户知道对他们的期望。

让我们仔细看看电子邮件地址输入框。它的 [size](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#size) 和 [maxlength](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#maxlength) 属性都被设置为 64，以便为 64 个字符的电子邮件地址提供空间，并将实际输入的字符数限制在最大 64 个。这里还指定了 [required](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#required) 属性，使得用户必须提供一个有效的电子邮件地址。

表单中提供了一个适当的 [placeholder](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#placeholder)——username@beststartupever.com，以证明什么是有效的条目。这个字符串表明，应该输入一个电子邮件地址，并建议它应该是一个企业的 beststartupever.com 账户。此外，使用 email 类型将验证文本以确保它的格式像一个电子邮件地址。如果输入框中的文本不是电子邮件地址，你会得到一个错误信息，看起来像这样：

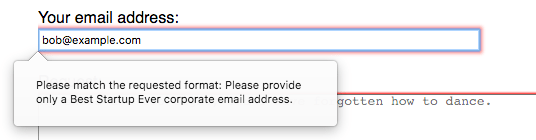


如果我们让事情停留在这个阶段，我们至少会在合法的电子邮件地址上进行验证。但我们想走得更远：我们想确保电子邮件地址实际上是“[\_username\_@beststartupever.com](mailto:_username_@beststartupever.com)”的形式。这就是我们要使用 [pattern](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#pattern) 的地方。我们将 pattern 设置为 .+@beststartupever.com。这个简单的正则表达式要求一个至少由一个任何类型的字符组成的字符串，然后是“@”，后面是域名“beststartupever.com”。

请注意，这根本不是一个有效的电子邮件地址过滤器；它允许诸如“ @beststartupever.com”（注意前面的空格）或“@@beststartupever.com”这样的内容，它们都是无效的。然而，浏览器对指定的文本同时运行标准的电子邮件地址过滤器和我们的自定义模式。结果，我们的验证结果是：“确保这类似于一个有效的电子邮件地址，如果是的话，确保它也是一个 beststartupever.com 的地址。”

建议在使用 pattern 属性的同时使用 [title](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Global_attributes#title)。如果你这样做，title 必须描述该模式。也就是说，它应该解释数据应该采取什么格式，而不是任何其他信息。这是因为 title 可以作为验证错误信息的一部分被显示或说出。例如，浏览器可能会显示“输入的文本不符合要求的模式。”，随后是你指定的 title。如果你的 title 是类似于“Email address”的字符串，结果将是“输入的文本不符合要求的模式。Email address”，这不太通顺。

这就是为什么我们指定字符串“请只提供 Best Startup Ever 公司的电子邮件地址”。通过这样做，所产生的完整错误信息可能是这样的：“输入的文本不符合要求的模式。请只提供 Best Startup Ever 公司的电子邮件地址”。



**备注：** 如果你在编写验证正则表达式时遇到麻烦，它们不能正常工作，请检查你的浏览器控制台；那里可能有有用的错误信息来帮助你解决问题。

## [示例](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/email#%E7%A4%BA%E4%BE%8B)

这里我们有一个 ID 为 emailAddress 的电子邮件输入框，它最多可以有 256 个字符长。输入框本身的宽度为 64 个字符，并在字段为空时显示文本 user@example.gov 作为占位符。此外，通过使用[multiple (en-US)](https://developer.mozilla.org/en-US/docs/Web/HTML/Attributes/multiple)属性，该框被配置为允许用户输入零个或多个电子邮件地址，用逗号分隔，如[允许多个电子邮件地址](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/email#%E5%85%81%E8%AE%B8%E5%A4%9A%E4%B8%AA%E7%94%B5%E5%AD%90%E9%82%AE%E4%BB%B6%E5%9C%B0%E5%9D%80)中所述。最后，[list](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Attributes/list) 属性包含一个 [<datalist>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/datalist) 的 ID，其 [<option>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/option) 指定一组建议值，供用户选择。

作为补充，[<label>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/label) 元素被用来为电子邮件输入框建立一个标签，其 [for](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/label#for) 属性引用了 [<input>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input) 元素的 emailAddress ID。通过这种方式将这两个元素联系起来，点击标签将聚焦输入元素。

<label for="emailAddress">电子邮件</label><br />

<input

id="emailAddress"

type="email"

placeholder="user@example.gov"

list="defaultEmails"

size="64"

maxlength="256"

multiple />

<datalist id="defaultEmails">

<option value="jbond007@mi6.defence.gov.uk"></option>

<option value="jbourne@unknown.net"></option>

<option value="nfury@shield.org"></option>

<option value="tony@starkindustries.com"></option>

<option value="hulk@grrrrrrrr.arg"></option>

</datalist>

# <input type="image">

**image** 类型的 [<input>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input) 元素用于创建图形化的提交按钮，即采用图像而非文本形式的提交按钮。

## [尝试一下](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/image#%E5%B0%9D%E8%AF%95%E4%B8%80%E4%B8%8B)

## [值](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/image#%E5%80%BC)

<input type="image"> 元素不接受 value 属性。要显示的图像的路径在 src 属性中指定。

## [额外属性](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/image#%E9%A2%9D%E5%A4%96%E5%B1%9E%E6%80%A7)

除了所有 [<input>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input) 元素的共有属性外，image 按钮还支持以下属性。

### [alt](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/image#alt)

alt 属性提供一个备用字符串，在图像无法加载的情况下作为按钮的标注（由于错误，一个不能或者被设置为不能显示图像的 [user agent](https://developer.mozilla.org/zh-CN/docs/Glossary/User_agent)，或用户正在使用屏幕阅读设备）。如果它被提供，它必须为非空字符串，且适合作为按钮的标签。

例如，如果你有一个按钮，显示一个带有图标和/或图像文本“现在登录”的图片，你也应该设置 alt 属性为类似于 现在登录 的东西。

**备注：** 虽然 alt 属性在技术上是可选的，但是你总应该包含它，以最大限度地提高你的内容的可用性。在功能上，<input type="image"> 元素的 alt 属性与 [<img>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/img) 元素的 [alt](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/img#alt) 属性相类似。

### [formaction](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/image#formaction)

一个字符串，指示要将数据提交到的 URL。这优先于拥有 [<input>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input) 的 [<form>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/form) 元素上的 [action](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/form#action) 属性。

该属性也在 [<input type="submit">](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/submit) 和 [<button>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/button) 元素上可用。

### [formenctype](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/image#formenctype)

一个字符串，标识在将表单数据提交到服务器时要使用的编码方法。有三个允许的值：

application/x-www-form-urlencoded

这是默认值，它将会将表单数据在使用 [encodeURI()](https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Global_Objects/encodeURI) 等算法进行 URL 编码后发送。

multipart/form-data

使用 [FormData](https://developer.mozilla.org/zh-CN/docs/Web/API/FormData) API 来管理数据，允许向服务器提交文件。你\_必须\_使用该编码方式，如果你的表单包含 [type](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#type) 为 file（[<input type="file">](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/file)）的任何 [<input>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input) 元素。

text/plain

纯文本；通常只在调试时有作用，你可以很容易看到要被提交的数据。

如果该属性被指定，formenctype 属性的值将会覆盖表单的 [action](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/form#action) 属性。

该属性也在 [<input type="submit">](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/submit) 和 [<button>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/button) 元素上可用。

### [formmethod](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/image#formmethod)

一个字符串，指定提交表单数据时使用的 HTTP 方法；该值将会覆盖所属表单上的任何 [method](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/form#method) 属性。允许的值如下：

get

以 formaction 或 [action](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/form#action) 属性给定的 URL 为前缀构造的一个 URL，在前缀后会附加一个问号（?）字符，然后附加表单数据，表单数据按照 formenctype 或表单的 [enctype](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/form#enctype) 属性的描述进行编码。构造的 URL 会通过 HTTP [get](https://developer.mozilla.org/zh-CN/docs/Web/HTTP/Methods/GET) 请求发送到服务器。此方法适用于仅包含 ASCII 字符，且没有副作用的简单表单。这是该属性的默认值。

post

表单数据包含在请求的主体（body）中，并使用 HTTP [post](https://developer.mozilla.org/zh-CN/docs/Web/HTTP/Methods/POST) 请求发送到 formaction 或 [action](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/form#action) 属性指定的 URL。此方法支持复杂数据和文件附件。

dialog

此方法用于指示关闭与该输入元素关联的对话框的一个按钮，其不会传输表单数据。

该属性也在 [<input type="submit">](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/submit) 和 [<button>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/button) 元素上可用。

### [formnovalidate](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/image#formnovalidate)

一个布尔值，如果存在，则指定在提交给服务器之前不应对表单进行验证。这将会覆盖元素所属表单上的 [novalidate](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/form#novalidate) 属性的值。

该属性也在 [<input type="submit">](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/submit) 和 [<button>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/button) 元素上可用。

### [formtarget](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/image#formtarget)

一个字符串，指定一个名字或关键词来表示在提交表单后在何处显示响应数据。该字符串必须是一个**浏览上下文**的名字（即一个标签页、窗口，或 [<iframe>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/iframe)）。在此处指定的值将会覆盖 [<form>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/form) 上拥有该输入的 [target](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/form#target) 属性所给出的任何目标。

除了标签页、窗口或内联框架的实际名字，还有部分特殊关键词可用：

\_self

将响应加载到与包含表单的浏览上下文相同的浏览上下文中。这将用接收到的数据替换当前文档。如果未指定，这将是默认值。

\_blank

将响应加载到新的未命名浏览上下文中。这通常是一个与当前文档相同的窗口中的新选项卡，但是可能会因为[用户代理](https://developer.mozilla.org/zh-CN/docs/Glossary/User_agent)的配置而有所不同。

\_parent

将响应加载到当前浏览上下文的父浏览上下文中。若不存在父浏览上下文，该选项的行为将与 \_self 相同。

\_top

将响应加载到顶级浏览上下文中；它是当前浏览上下文的最顶层祖先。如果当前浏览上下文为最顶层上下文，该选项的行为将与 \_self 相同。

该属性也在 [<input type="submit">](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/submit) 和 [<button>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/button) 元素上可用。

### [height](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/image#height)

一个表示绘制 src 属性指定图像的高度的数字，以 CSS 像素为单位。

### [src](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/image#src)

一个字符串，指定将要在提交按钮上显示的图像的 URL。当用户与该图像交互时，输入将会以和其他按钮输入相同的方式被处理。

### [width](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/image#width)

一个表示绘制图像宽度的数字，以 CSS 像素为单位。

## [已弃用的属性](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/image#%E5%B7%B2%E5%BC%83%E7%94%A8%E7%9A%84%E5%B1%9E%E6%80%A7)

下列属性是由 HTML 4 为 image 输入定义的，但未被所有浏览器实现，并且已被弃用。

### [usemap](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/image#usemap)

如果 usemap 被指定，它必须是一个图像映射元素的名字。[<map>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/map) 用于定义一个和图像一同使用的图像映射。该用法已被弃用；当你需要使用图像映射时，你应该切换到使用 [<img>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/img) 元素。

## [示例](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/image#%E7%A4%BA%E4%BE%8B)

### [一个登录表单](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/image#%E4%B8%80%E4%B8%AA%E7%99%BB%E5%BD%95%E8%A1%A8%E5%8D%95)

下面的例子显示和之前相同的按钮，但被包含在了一个典型登录表单的上下文中：

#### HTML

<form>

<p>登录到你的账户</p>

<div>

<label for="userId">用户 ID</label>

<input type="text" id="userId" name="userId" />

</div>

<div>

<label for="pwd">密码</label>

<input type="password" id="pwd" name="pwd" />

</div>

<div>

<input

id="image"

type="image"

src="https://raw.githubusercontent.com/mdn/learning-area/master/html/forms/image-type-example/login.png"

alt="Login"

width="100" />

</div>

</form>

Copy to Clipboard

#### CSS

下面是一些简单的 CSS 来让这些基本元素排列更整齐：

div {

margin-bottom: 10px;

}

label {

display: inline-block;

width: 70px;

text-align: right;

padding-right: 10px;

}

Copy to Clipboard

### [调整图像的位置和缩放](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/image#%E8%B0%83%E6%95%B4%E5%9B%BE%E5%83%8F%E7%9A%84%E4%BD%8D%E7%BD%AE%E5%92%8C%E7%BC%A9%E6%94%BE)

在这个例子里，我们调整了之前的例子，为图像留出了更多空间，并使用 [object-fit](https://developer.mozilla.org/zh-CN/docs/Web/CSS/object-fit) 和 [object-position](https://developer.mozilla.org/zh-CN/docs/Web/CSS/object-position) 来调节图像的大小和位置。

#### HTML

<form>

<p>登录到你的账户</p>

<div>

<label for="userId">用户 ID</label>

<input type="text" id="userId" name="userId" />

</div>

<div>

<label for="pwd">密码</label>

<input type="password" id="pwd" name="pwd" />

</div>

<div>

<input

id="image"

type="image"

src="https://raw.githubusercontent.com/mdn/learning-area/master/html/forms/image-type-example/login.png"

alt="Login"

width="200"

height="100" />

</div>

</form>

Copy to Clipboard

#### CSS

div {

margin-bottom: 10px;

}

label {

display: inline-block;

width: 70px;

text-align: right;

padding-right: 10px;

}

#image {

object-position: right top;

object-fit: contain;

background-color: #ddd;

}

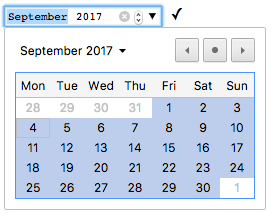
# <input type="month">

类型为 **month** 的 [<input>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input) 可以让你容易地创建一个方便输入年份或月份的一个 [<input>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input)。输入的值是一个经过“YYYY-MM”格式化的字符串，其中 YYYY 是四位数的年份，而 MM 是月份的数值表示。

## [尝试一下](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/month#%E5%B0%9D%E8%AF%95%E4%B8%80%E4%B8%8B)

通常来说控件的 UI 界面因浏览器的不同而有变化，到目前为止此控件还不被所有浏览器支持，在桌面浏览器中只有 Chrome/Opera 和 Edge 支持；在移动端被大部分现代浏览器所支持。在不支持的浏览器中，这个控件会被优雅的降级到 [<input type="text">](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/text)，不过对输入的文字会有自动的验证，保证它按照预期进行格式化。

对于那些使用不支持 month 控制器的浏览器用户，以下的截图呈现了 Chrome 和 Opera 的月份控制器。单击右侧的向下箭头会显示日期选择器，以便选择年份和月份。



Edge 的 month 控制器看起来像这样的：



|  |  |
| --- | --- |
| [**值**](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/month#%E5%80%BC) | 代表月份和年份的字符串，或为空字符串 |
| **事件** | [change](https://developer.mozilla.org/zh-CN/docs/Web/API/HTMLElement/change_event) 和 [input](https://developer.mozilla.org/zh-CN/docs/Web/API/HTMLElement/input_event) |
| **支持的共有属性** | [autocomplete](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#autocomplete)、[list](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#list)、[readonly](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#readonly) 和 [step](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#step) |
| **IDL 属性** | value |
| **DOM 接口** | [HTMLInputElement](https://developer.mozilla.org/zh-CN/docs/Web/API/HTMLInputElement) |
| **方法** | [select()](https://developer.mozilla.org/zh-CN/docs/Web/API/HTMLInputElement/select)、[stepDown() (en-US)](https://developer.mozilla.org/en-US/docs/Web/API/HTMLInputElement/stepDown) 和 [stepUp() (en-US)](https://developer.mozilla.org/en-US/docs/Web/API/HTMLInputElement/stepUp) |

## [值](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/month#%E5%80%BC)

一个以 YYYY-MM 形式（4 个或更多位数的年，接一个连字符 -，再接一个两位数月份）表示月份和年份的字符串。月份字符串的格式在[月份字符串](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Date_and_time_formats#%E6%9C%88%E4%BB%BD%E5%AD%97%E7%AC%A6%E4%B8%B2)中有描述。

### [设置默认值](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/month#%E8%AE%BE%E7%BD%AE%E9%BB%98%E8%AE%A4%E5%80%BC)

你可以通过在 [value](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#value) 属性中包含月份和年份来设置该输入控件的默认值，像这样：

<label for="bday-month">你在哪个月出生？</label>

<input id="bday-month" type="month" name="bday-month" value="2001-06" />

Copy to Clipboard

需要注意的是显示的如期格式不同于实际的 value ；大部分[用户代理](https://developer.mozilla.org/zh-CN/docs/Glossary/User_agent)基于操作系统，根据合适的本地化形式显示月份和年份，而日期的 value 总是会格式化为 yyyy-MM。

在向服务器提交上述值的时候它们看起来像这样：bday-month=1978-06。

### [通过 JavaScript 代码设置 value](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/month#%E9%80%9A%E8%BF%87_javascript_%E4%BB%A3%E7%A0%81%E8%AE%BE%E7%BD%AE_value)

当然你也可以使用 JavaScript 代码通过 HTMLInputElement.value 属性来获取或设置日期的值。例如：

<label for="bday-month">你在哪个月出生？</label>

<input id="bday-month" type="month" name="bday-month" />

Copy to Clipboard

const monthControl = document.querySelector('input[type="month"]');

monthControl.value = '2001-06';

Copy to Clipboard

## [其他属性](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/month#%E5%85%B6%E4%BB%96%E5%B1%9E%E6%80%A7)

除了 [<input>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input) 元素的共有属性外，month 输入还提供以下属性：

### [list](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/month#list)

列表属性的值是位于同一文档中的 [<datalist>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/datalist) 元素的 [id](https://developer.mozilla.org/zh-CN/docs/Web/API/Element/id)。[<datalist>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/datalist) 提供了一个预定义的值列表，向用户建议这个输入。列表中任何与 [type](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#type) 不兼容的值都不包括在建议选项中。所提供的值是建议，不是要求：用户可以从这个预定义的列表中选择，或者提供不同的值。

### [max](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/month#max)

使用在[值](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/month#%E5%80%BC)章节中讨论的字符串格式指定的所接受的最大年份和月份。如果输入到该元素的 [value](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#value) 超过了这个，则该元素无法通过[约束验证](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Constraint_validation)。如果 max 属性的值不是格式为 yyyy-MM 的有效字符串，则该元素没有最大值。

该值必须晚于或等于 min 属性所指定的年份-月份对。

### [min](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/month#min)

使用在[值](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/month#%E5%80%BC)章节中讨论的字符串格式指定的所接受的最小年份和月份。如果输入到该元素的 [value](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#value) 小于这个，则该元素无法通过[约束验证](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Constraint_validation)。如果 min 属性的值不是格式为 yyyy-MM 的有效字符串，则该元素没有最小值。

该值必须早于或等于 max 属性所指定的年份-月份对。

### [readonly](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/month#readonly)

一个布尔属性，如果存在，则表示该字段不能由用户编辑。但是，仍可以通过 JavaScript 代码直接设置 HTMLInputElement.value 属性来更改。

**备注：** 因为只读字段不能有值，所以 required 对指定了 readonly 属性的输入没有任何影响。

### [step](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/month#step)

step 属性指定了值必须满足的粒度，或者是下文描述的特殊值 any。值必须满足基础的步进值，才有效。如果指定了 [min](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/month#min) 属性，则由 min 属性决定，否则，使用 [value](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#value) 属性的值，如果上述两个值都不存在，则提供适当的默认值。

字符串值 any 意味着不使用步进值，任意值都可以接受（除其他制约因素如 [min](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/month#min) 或 [max](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/month#max) 之外）。

**备注：** 当用户输入的数据不符合步进配置时，[用户代理](https://developer.mozilla.org/zh-CN/docs/Glossary/User_agent)可能会四舍五入到最近的有效值，当有两个同样接近的选项时，更倾向于正方向的数字。

对于 month 输入，step 的值以月份为单位，缩放因子为 1（基础数值也是以月份为单位的）。step 的默认值为 1，表示 1 个月。

## [使用 month 输入](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/month#%E4%BD%BF%E7%94%A8_month_%E8%BE%93%E5%85%A5)

与日期相关的输入（包含 month）乍一看很方便：它们提供了一个简单的用户界面来选择日期，并且它们将发送到服务器的数据格式规范化，无论用户的本地化配置如何。但是，<input type="month"> 还是存在兼容性问题，大多数主流浏览器还没有支持它。

我们先看看 <input type="month"> 的基础和高级的用法，然后再讨论有关缓解浏览器支持问题的建议（请参阅[处理浏览器支持问题](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/month#%E5%A4%84%E7%90%86%E6%B5%8F%E8%A7%88%E5%99%A8%E6%94%AF%E6%8C%81%E9%97%AE%E9%A2%98)）。

### [基本使用方法](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/month#%E5%9F%BA%E6%9C%AC%E4%BD%BF%E7%94%A8%E6%96%B9%E6%B3%95)

最简单的 <input type="month"> 涉及到基础的 [<input>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input) 和 [<label>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/label) 的元素组合，像下面这样：

<form>

<label for="bday-month">你在哪个月出生？</label>

<input id="bday-month" type="month" name="bday-month" />

</form>

Copy to Clipboard

### [设置最大和最小日期](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/month#%E8%AE%BE%E7%BD%AE%E6%9C%80%E5%A4%A7%E5%92%8C%E6%9C%80%E5%B0%8F%E6%97%A5%E6%9C%9F)

你可以使用 [min](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#min) 和 [max](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#max) 属性来限制用户选择日期的范围。在下列的例子中我们指定最小月份 1900-01 和最大月份 2013-12：

<form>

<label for="bday-month">你在哪个月出生？</label>

<input

id="bday-month"

type="month"

name="bday-month"

min="1900-01"

max="2013-12" />

</form>

Copy to Clipboard

结果是这样：

* 只有从 1900 年的 1 月到 2013 年的 12 月的月份才可以选择，在这个控件里这个范围以外的月份不能滚动选择。
* 取决于你使用的浏览器，你可能会发现不能在月份选择器中选择在所指定范围之外的月份（如 Edge），或显示不合法（参见[验证](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/month#%E9%AA%8C%E8%AF%81)小节）但仍然可选（例如 Chrome）。

### [控制输入大小](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/month#%E6%8E%A7%E5%88%B6%E8%BE%93%E5%85%A5%E5%A4%A7%E5%B0%8F)

<input type="month"> 不支持诸如 [size](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#size) 的表单大小属性，你必须依靠 [CSS](https://developer.mozilla.org/zh-CN/docs/Web/CSS) 来确定大小。

## [验证](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/month#%E9%AA%8C%E8%AF%81)

默认情况下，<input type="month"> 不会对输入的值应用任何验证，用户界面实现会屏蔽所有非日期值的输入，尽管这很有用，但是仍然无法完全依赖于该值的合法性。你仍然可以不填入任何值而提交，或输入不合法的日期值（如 4 月 32 日）。

为了避免这种情况，你可以使用 [min](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#min) 和 [max](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#max) 来限制可用的日期（参见[设置最大和最小日期](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/month#%E8%AE%BE%E7%BD%AE%E6%9C%80%E5%A4%A7%E5%92%8C%E6%9C%80%E5%B0%8F%E6%97%A5%E6%9C%9F)），并使用 [required](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#required) 属性令日期必填。在支持的浏览器中，当你尝试提交超出范围的日期，或空日期值时，会显示错误信息。

我们来看一个例子，这里我们设定了最小和最大的日期，并令该字段必填：

<form>

<div>

<label for="month">

你愿意在哪个月拜访（6 月至 9 月）？

</label>

<input

id="month"

type="month"

name="month"

min="2022-06"

max="2022-09"

required />

<span class="validity"></span>

</div>

<div>

<input type="submit" value="提交表单" />

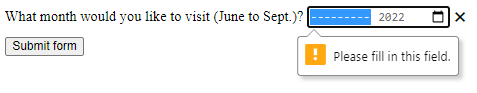
</div>

</form>

Copy to Clipboard

如果你尝试在未指定月份或年份，或指定超出给定范围的日期情况下提交表单，浏览器会显示错误。请试试以下实时演示：

对于那些没有使用支持的浏览器的人们来说，以下是该错误的屏幕截图：



下面是上述例子中使用的 CSS。这里我们使用了 [:valid](https://developer.mozilla.org/zh-CN/docs/Web/CSS/:valid) 和 [:invalid](https://developer.mozilla.org/zh-CN/docs/Web/CSS/:invalid) 这两个 CSS 属性，根据当前值是否有效来为输入添加样式。我们不得不把图标放在 input 旁边的 [<span>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/span) 上，而不是放在 input 框本身，因为在 Chrome 中，生成的内容被放在表单控件里面，无法有效地进行样式设计或显示。

div {

margin-bottom: 10px;

position: relative;

}

input[type="number"] {

width: 100px;

}

input + span {

padding-right: 30px;

}

input:invalid + span::after {

position: absolute;

content: "✖";

padding-left: 5px;

}

input:valid + span::after {

position: absolute;

content: "✓";

padding-left: 5px;

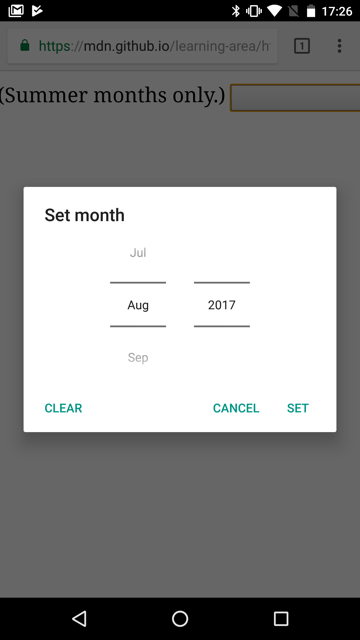
}

Copy to Clipboard

**警告：** HTML 表单验证并不能替代确保输入数据格式正确的脚本。很容易对 HTML 进行调整，使他们能够绕过验证，或完全删除验证。也有可能会完全绕过 HTML 代码，直接将数据提交给你的服务器。如果你的服务器端代码不能验证它所收到的数据，那么当提交的数据格式不当（或数据过大、类型错误等等）时，灾难就会降临。

## [处理浏览器支持](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/month#%E5%A4%84%E7%90%86%E6%B5%8F%E8%A7%88%E5%99%A8%E6%94%AF%E6%8C%81)

如前所述，使用 month 输入的最大问题是大部分主流浏览器还没有实现它们；只有桌面端的 Chrome/Opera、Edge 和大部分的移动端现代浏览器支持它们。例如，安卓版 Chrome 的 month 选择器看起来像这样：



不支持的浏览器会优雅地降级为文本输入，但这在用户界面的一致性（呈现的控件会有所不同）和数据处理方面都会产生问题。

第二个问题更为严重；如前所述，month 输入的值总是被规范为 yyyy-mm 的格式。另一方面，对于文本输入，默认情况下，浏览器不知道时间应该是什么格式，而且人们有多种写法，如：

* mmyyyy (072022)
* mm/yyyy (07/2022)
* mm-yyyy (07-2022)
* yyyy-mm (2022-07)
* Month yyyy (July 2022)
* 等等

一个办法是在你的 month 输入上添加 [pattern](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#pattern) 属性。即使 month 输入不使用它，text 输入回退也会使用。例如，试着在一个不支持 month 输入的浏览器中查看下面的演示：

<form>

<div>

<label for="month">

愿意在哪个月访问（6 月至 9 月）？

</label>

<input

id="month"

type="month"

name="month"

min="2022-06"

max="2022-09"

required

pattern="[0-9]{4}-[0-9]{2}" />

<span class="validity"></span>

</div>

<div>

<input type="submit" value="提交表单" />

</div>

</form>

Copy to Clipboard

如果你尝试提交，你会发现，如果你的输入不符合模式 nnnn-nn（其中 n 是 0 到 9 的数字），不支持的浏览器现在会显示一个错误信息（并突出显示输入无效）。当然，这并不能阻止人们输入无效的日期（如 0000-42），或者不正确但遵循格式的日期。

还有一个问题，就是用户不知道到底应该输入什么格式的日期。还有一些事情要做。

目前，以跨浏览器方式处理时间的最佳方法（至少在所有主流浏览器实现它们之前）是让用户在单独的控件中输入月份和年份（特别是在 [<select>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/select) 元素中，参见下面的示例），或使用 JavaScript 库（例如 [jQuery 日期选择器](https://jqueryui.com/datepicker/)插件）。

## [示例](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/month#%E7%A4%BA%E4%BE%8B)

在此示例中，我们创建了两组用于选择日期的接口元素：使用 <input type="month"> 创建的原生选择器，以及为不支持 month 输入类型的旧版浏览器准备的两个分别用于选择月份和年份的 [<select>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/select) 元素。

### [HTML](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/month#html)

用于请求月份和年份的表单看起来像这样：

<form>

<div class="nativeDatePicker">

<label for="month-visit">What month would you like to visit us?</label>

<input type="month" id="month-visit" name="month-visit" />

<span class="validity"></span>

</div>

<p class="fallbackLabel">What month would you like to visit us?</p>

<div class="fallbackDatePicker">

<div>

<span>

<label for="month">Month:</label>

<select id="month" name="month">

<option selected>January</option>

<option>February</option>

<option>March</option>

<option>April</option>

<option>May</option>

<option>June</option>

<option>July</option>

<option>August</option>

<option>September</option>

<option>October</option>

<option>November</option>

<option>December</option>

</select>

</span>

<span>

<label for="year">Year:</label>

<select id="year" name="year"></select>

</span>

</div>

</div>

</form>

Copy to Clipboard

ID 为 nativeDatePicker 的 [<div>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/div) 使用 month 输入类型来请求月份和年份，而 ID 为 fallbackDatePicker 的 <div> 则使用一对 <select> 元素。第一个元素请求月份，第二个元素请求年份。

用于选择月份的 <select> 是硬编码的，因为月份的名称不会改变（不考虑本地化）。可用的年份值列表是根据当前年份动态生成的（关于这些函数如何工作的详细解释，见下面的代码注释）。

### [JavaScript](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/month#javascript)

处理选择哪种方案并设定一系列包含于非原生的 <select> 的年份列表的 JavaScript 代码如下所示。

该代码中可能有趣的另一部分是特性检测代码。要检测浏览器是否支持 <input type="month">，我们创建一个新的 [<input>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input) 元素，尝试将其 type 设置为 month，然后立即检查其 type 值。不支持的浏览器将返回 text，与 month 的回退行为相符。如果不支持 <input type="month">，我们将隐藏原生选择器并显示作为回退的选择器 UI。

// 获取 UI 元素

const nativePicker = document.querySelector('.nativeDatePicker');

const fallbackPicker = document.querySelector('.fallbackDatePicker');

const fallbackLabel = document.querySelector('.fallbackLabel');

const yearSelect = document.querySelector('#year');

const monthSelect = document.querySelector('#month');

// 最初，隐藏回退元素

fallbackPicker.style.display = 'none';

fallbackLabel.style.display = 'none';

// 测试一个新的 date 输入框是否会回退至 text 输入框

const test = document.createElement('input');

try {

test.type = 'month';

} catch (e) {

console.log(e.description);

}

// 如果回退了，运行 if 代码块中的代码

if (test.type === 'text') {

// 隐藏原生选择器，显示回退元素

nativePicker.style.display = 'none';

fallbackPicker.style.display = 'block';

fallbackLabel.style.display = 'block';

// 动态生成年份

// 月份总是相同的，故将它们硬编码

populateYears();

}

function populateYears() {

// 获取当前年份的数值表示

const date = new Date();

const year = date.getFullYear();

// 在年份 <select> 中，令该年和之前的 100 年可选

for (let i = 0; i <= 100; i++) {

const option = document.createElement('option');

option.textContent = year - i;

yearSelect.appendChild(option);

}

}

**备注：** 请记住有些年份有 53 周（见[每年的周数](https://zh.wikipedia.org/wiki/ISO%E9%80%B1%E6%97%A5%E6%9B%86#%E6%AF%8F%E5%B9%B4%E7%9A%84%E6%98%9F%E6%9C%9F)）！当你在开发产品应用时应当考虑这个问题。

# <input type="number">

**number** 类型的 [<input>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input) 元素用于让用户输入一个数字，其包括内置验证以拒绝非数字输入。

浏览器可能会选择提供步进箭头，让用户可以使用鼠标增加和减少输入的值，或者只需用指尖敲击即可。

## [尝试一下](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/number#%E5%B0%9D%E8%AF%95%E4%B8%80%E4%B8%8B)

不支持 number 类型的浏览器会回退为标准的 text 输入框。

|  |  |
| --- | --- |
| [**值**](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/number#%E5%80%BC) | 表示一个数字的 [Number](https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Global_Objects/Number)，或为空 |
| **事件** | [change](https://developer.mozilla.org/zh-CN/docs/Web/API/HTMLElement/change_event) 和 [input](https://developer.mozilla.org/zh-CN/docs/Web/API/HTMLElement/input_event) |
| **支持的公共属性** | [autocomplete](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#autocomplete)、[list](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#list)、[placeholder](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#placeholder) 和 [readonly](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#readonly) |
| **IDL 属性** | list、value 和 valueAsNumber |
| **DOM 接口** | [HTMLInputElement](https://developer.mozilla.org/zh-CN/docs/Web/API/HTMLInputElement) |
| **方法** | [select()](https://developer.mozilla.org/zh-CN/docs/Web/API/HTMLInputElement/select)、[stepUp() (en-US)](https://developer.mozilla.org/en-US/docs/Web/API/HTMLInputElement/stepUp) 和 [stepDown() (en-US)](https://developer.mozilla.org/en-US/docs/Web/API/HTMLInputElement/stepDown) |

## [值](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/number#%E5%80%BC)

填写到输入框中的数值文字的 [Number](https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Global_Objects/Number) 表示。你可以通过在 [value](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#value) 属性中包含一个数字来设置输入框的默认值，如下所示：

<input id="number" type="number" value="42" />

Copy to Clipboard

## [附加属性](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/number#%E9%99%84%E5%8A%A0%E5%B1%9E%E6%80%A7)

除了 [<input>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input) 元素共有的属性外，数字输入还提供以下属性。

### [list](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/number#list)

list 属性的值是位于同一文档中的 [<datalist>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/datalist) 元素的 [id](https://developer.mozilla.org/zh-CN/docs/Web/API/Element/id)。[<datalist>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/datalist) 提供了可输入到当前输入框的一个预定义的值列表。列表中任何与 [type](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#type) 不兼容的值都不包括在建议选项中。所提供的值是建议，不是要求：用户可以从这个预定义的列表中选择，或者提供不同的值。

### [max](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/number#max)

允许值范围内的最大值。如果输入到元素中的 [value](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#value) 超过此值，则元素将无法通过[约束验证](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Constraint_validation)。如果 max 属性的值不是数字，则元素没有最大值。

此值必须大于或等于 min 属性的值。

### [min](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/number#min)

允许值范围内的最小值。如果元素的 [value](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#value) 小于此值，则该元素将无法通过[约束验证](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Constraint_validation)。如果为 min 指定的值不是有效数字，则输入没有最小值。

该值必须小于或等于 max 属性的值。

### [placeholder](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/number#placeholder)

placeholder 属性是一个字符串，可向用户提供有关该字段中需要什么样的信息的简短提示。它应该是说明预期的数据类型的单词或短语，而不是说明性消息。文本中不得包含回车符或换行符。

如果控件的内容具有方向（[LTR](https://developer.mozilla.org/zh-CN/docs/Glossary/LTR) 或 [RTL](https://developer.mozilla.org/zh-CN/docs/Glossary/RTL)），但需要以相反的方向显示占位符，则可以使用 Unicode 双向算法来格式化字符，从而覆盖原有占位符的方向；请参见[如何针对双向文本使用 Unicode 控制符](https://www.w3.org/International/questions/qa-bidi-unicode-controls)获取更多信息。

**备注：** 请尽可能避免使用 placeholder 属性，它在语义上没有其他解释表单的方式有用，并且可能会导致额外的问题。参见 [<input> 标签](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#%E6%A0%87%E7%AD%BE)以获取更多信息。

### [readonly](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/number#readonly)

如果该布尔属性存在，意味着用户将不能编辑此字段。然而其 value 值仍然可以直接通过 JavaScript 代码设置 [HTMLInputElement](https://developer.mozilla.org/zh-CN/docs/Web/API/HTMLInputElement) 的 value 属性改变。

**备注：** 由于只读字段不可以拥有值，required 不会对指定了 readonly 属性的字段起作用。

### [step](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/number#step)

step 属性指定了值必须满足的粒度，或者是下文描述的特殊值 any。值必须满足基础的步进值，才有效。如果指定了 [min](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/number#min) 属性，则由 min 属性决定，否则，使用 [value](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#value) 属性的值，如果上述两个值都不存在，则提供适当的默认值。

字符串值 any 意味着不使用步进值，任意值都可以接受（除其他制约因素如 [min](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/number#min) 或 [max](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/number#max) 之外）。

**备注：** 当用户输入的数据不符合步进配置时，[用户代理](https://developer.mozilla.org/zh-CN/docs/Glossary/User_agent)可能会四舍五入到最近的有效值，当有两个同样接近的选项时，更倾向于正方向的数字。

number 输入的默认步进值为 1，且只允许数字输入——除非步进值不是数字。

## [使用 number 输入框](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/number#%E4%BD%BF%E7%94%A8_number_%E8%BE%93%E5%85%A5%E6%A1%86)

number 输入类型只能用于可增长的数字，特别是在使用按钮增减值有助于用户体验的时候。number 输入类型对于显示那些不能严格表示数字（可能只是碰巧包含数字）的值（例如邮政编码、信用卡号等）并不合适。对于非数字输入，请考虑使用不同的 input 类型，例如 [<input type="tel">](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/tel) 或其他包含 [inputmode](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Global_attributes#inputmode) 属性的 [<input>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input) 类型。

<input type="text" inputmode="numeric" pattern="\d\*" />

Copy to Clipboard

<input type ="number"> 元素可以帮助简化构建在表单中输入数字的用户界面和逻辑。当你使用正确的 type 值 number 创建数字输入时，会自动验证你输入的文本是否为数字，而且通常会一同设置一组用于步进数字值的向上向下按钮。

**警告：** 逻辑上，不应该向数字输入框输入数字以外的字符。某些浏览器允许这些不合法字符，另外一些浏览器不允许；请查看 [Firefox bug 1398528](https://bugzil.la/1398528)。

**备注：** 记住，用户可以在幕后修改 HTML ，因此网站不得使用简单的客户端验证来实现任何安全目的。你必须在服务器端验证所提供的值，以确保这些值没有任何安全问题。

此外，移动浏览器通过在用户尝试输入值时显示更适合输入数字的特殊键盘，进一步帮助用户体验。

### [一个简单的 number 输入框](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/number#%E4%B8%80%E4%B8%AA%E7%AE%80%E5%8D%95%E7%9A%84_number_%E8%BE%93%E5%85%A5%E6%A1%86)

在最基本的形式中，数字输入可以像这样实现：

<label for="ticketNum">需要买票的数量：</label>

<input id="ticketNum" type="number" name="ticketNum" value="0" />

Copy to Clipboard

数字输入在为空的时候以及输入单个数字时被认为是有效的，否则是无效的。如果使用了 [required](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#required) 属性，则输入在空时不再被视为有效。

**备注：** 任何数字都是可接受的值，只要它是[有效的浮点数](https://html.spec.whatwg.org/multipage/infrastructure.html#valid-floating-point-number)（即不是 [NaN](https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Global_Objects/NaN) 或 [Infinity](https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Global_Objects/Infinity)）。

### [占位符](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/number#%E5%8D%A0%E4%BD%8D%E7%AC%A6)

有时，提供一个如何输入表单数据的上下文提示是非常友好的。尤其当页面所有[<input>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input) 元素都没有说明标签时，会非常重要。这就是使用**占位符**的原因。占位符是一个通常用于说明一个输入框的 value 应该如何输入的提示值，当元素的 value 为 "" 时，该提示将出现在编辑框里面。一旦有数据输入到编辑框中，占位符就会消失；如果该框被清空，占位符将重新出现。

这里，我们有一个占位符为 "Multiple of 10" 的 number 输入框。请注意在编辑字段内容时占位符是如何消失的，又是如何重现的。

<input type="number" placeholder="Multiple of 10" />

Copy to Clipboard

### [控制步进大小](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/number#%E6%8E%A7%E5%88%B6%E6%AD%A5%E8%BF%9B%E5%A4%A7%E5%B0%8F)

默认情况下，向上和向下按钮可以将值增加或减小 1。你可以使用 [step](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#step) 属性来更改此步长值。我们上面的例子中的占位符提示我们其值是 10 的倍数，所以设置 step 的值为 10 是合理的：

<input type="number" placeholder="multiple of 10" step="10" />

Copy to Clipboard

在这个例子中，你会发现上下箭头每次会将其值增加和减少 10，而不是 1。你仍然可以手动输入一个不是 10 的倍数的数字，但它会被认为是无效的。

### [指定最小和最大值](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/number#%E6%8C%87%E5%AE%9A%E6%9C%80%E5%B0%8F%E5%92%8C%E6%9C%80%E5%A4%A7%E5%80%BC)

可以使用 [min](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#min) 和 [max](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#max) 属性指定该字段可以具有的最小值和最大值。例如，可以给我们的示例指定最小值为 0，最大值为 100：

<input type="number" placeholder="multiple of 10" step="10" min="0" max="100" />

Copy to Clipboard

修改过后，你会发现点击上下按钮不会让你低于 0 或高于 100 . 可以在这些界限之外手动输入数字，但它将被视为无效。

### [允许小数值](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/number#%E5%85%81%E8%AE%B8%E5%B0%8F%E6%95%B0%E5%80%BC)

数字输入的一个问题是，步长默认为 1——如果你尝试输入带小数的数字（例如“1.0”），则它将被视为无效。如果要输入一个需要小数的值，则需要修改 step 值（例如，step="0.01" 以允许 2 位小数）。这里有一个简单的例子：

<input type="number" placeholder="1.0" step="0.01" min="0" max="10" />

Copy to Clipboard

你可以看到这个例子允许 0.0 到 10.0 之间的任何值，小数点为 2 位。本例中，“9.52”是有效的，但“9.521”无效。

### [控制输入框大小](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/number#%E6%8E%A7%E5%88%B6%E8%BE%93%E5%85%A5%E6%A1%86%E5%A4%A7%E5%B0%8F)

类型为 number 的 [<input>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input) 元素不支持像 [size](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#size) 之类的调整大小属性。你必须通过借助 [CSS](https://developer.mozilla.org/zh-CN/docs/Web/CSS) 来改变它们的尺寸大小。

例如，要调整输入框的宽度为仅可输入 3 位数字的宽度，我们可以在 HTML 中添加一个 ID 并缩短很长的占位符文本（这个字段对于目前的示例来说太窄了），如下：

<input

type="number"

placeholder="x10"

step="10"

min="0"

max="100"

id="number" />

Copy to Clipboard

然后我们使用 id 选择器 #number 缩小元素的宽度：

#number {

width: 3em;

}

Copy to Clipboard

效果如下：

### [提供建议值](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/number#%E6%8F%90%E4%BE%9B%E5%BB%BA%E8%AE%AE%E5%80%BC)

你可以通过指定 [list](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#list) 属性来提供一个默认选项列表，用户可以从中选择，该属性包含一个 [<datalist>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/datalist) 的 ID 作为其值，而该 ID 又包含一个 [<option>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/option) 元素的每个建议值；每个 option 的 value 就是数字输入框的相应建议值。

<input id="ticketNum" type="number" name="ticketNum" list="defaultNumbers" />

<span class="validity"></span>

<datalist id="defaultNumbers">

<option value="10045678"></option>

<option value="103421"></option>

<option value="11111111"></option>

<option value="12345678"></option>

<option value="12999922"></option>

</datalist>

Copy to Clipboard

## [验证](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/number#%E9%AA%8C%E8%AF%81)

我们已经提及了一些数字输入的验证特性，来做个回忆：

* <input type="number"> 元素自动拒绝任何非数字输入（如果指定了 required，也会拒绝空的输入）。
* 你可以使用 [required](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#required) 属性来使得空的输入无效，也就是说，该输入框必须有值填入。
* 你可以使用 [step](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#step) 属性来将有效值限制在一系列步进值中（如 10 的倍数）。
* 你可以使用 [min](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#min) 和 [max](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#max) 属性来限制输入的最大与最小值。

以下示例展示了上面所述的所有特性，也使用了 CSS 来根据 input 的不同值，显示合法及不合法的图标：

<form>

<div>

<label for="balloons">要购买气球的数量（10 的倍数）：</label>

<input

id="balloons"

type="number"

name="balloons"

step="10"

min="0"

max="100"

required />

<span class="validity"></span>

</div>

<div>

<input type="submit" />

</div>

</form>

Copy to Clipboard

请尝试以非法的值提交该表单，如空值、小于 0 或大于 100 的值、不是 10 的倍数的值、或非数字值——来观察一下浏览器如何提示这些错误信息。

该示例应用的 CSS 如下所示：

div {

margin-bottom: 10px;

}

input:invalid + span::after {

content: "✖";

padding-left: 5px;

}

input:valid + span::after {

content: "✓";

padding-left: 5px;

}

Copy to Clipboard

我们使用了 [:invalid](https://developer.mozilla.org/zh-CN/docs/Web/CSS/:invalid) 和 [:valid](https://developer.mozilla.org/zh-CN/docs/Web/CSS/:valid) 伪类，在相邻的 [<span>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/span) 元素上显示一个适当的无效或有效图标作为生成内容，作为有效性的视觉指标。

我们把它放在一个单独的 <span>元素上，以增加灵活性；一些浏览器在某些类型的表单输入上不能很有效地显示生成的内容（请阅读 [<input type="date"> 验证](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/date#validation)一节的示例以了解）。

**警告：** 客户端表单验证不能替代服务端验证。对于某人来说，对 HTML 进行调整以使其绕过验证或完全删除验证太容易了，甚至也可以完全绕开 HTML 并将数据直接提交到服务器。如果服务器端代码无法验证其接收到的数据，则在提交格式不正确的（或太大，类型错误……）的数据时，灾难可能会发生。

### [模式验证](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/number#%E6%A8%A1%E5%BC%8F%E9%AA%8C%E8%AF%81)

<input type="number"> 不支持 [pattern](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#pattern) 属性以验证输入值是否满足特定的正则表达式。

这样做的理由是，如果数字输入中包含除数字以外的任何内容，则不会有效，你可以使用 [min](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#min) 和 [max](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#max) 属性来限制有效数字的最小和最大数量（如上所述）。

## [示例](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/number#%E7%A4%BA%E4%BE%8B)

我们已经介绍过，默认情况下，增量是 1，你可以使用 [step](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#step) 属性来允许小数输入。

在下面的例子中，我们设置了一个用于输入用户身高的表单；它默认接受以米为单位的身高，但你可以点击相关按钮来改变表单，使其接受英尺和英寸。输入米的身高可以接受小数点后两位。

HTML 代码看起来像这样：

<form>

<div class="metersInputGroup">

<label for="meters">Enter your height — meters:</label>

<input

id="meters"

type="number"

name="meters"

step="0.01"

min="0"

placeholder="e.g. 1.78"

required />

<span class="validity"></span>

</div>

<div class="feetInputGroup" style="display: none;">

<span>Enter your height — </span>

<label for="feet">feet:</label>

<input id="feet" type="number" name="feet" min="0" step="1" />

<span class="validity"></span>

<label for="inches">inches:</label>

<input id="inches" type="number" name="inches" min="0" max="11" step="1" />

<span class="validity"></span>

</div>

<div>

<input

type="button"

class="meters"

value="Enter height in feet and inches" />

</div>

<div>

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

</div>

</form>

Copy to Clipboard

你会看到，我们正在使用许多我们已经在前面的文章中看过的属性。由于我们想接受以厘米为单位的米值，我们将 step 值设置为 0.01，这样像 1.78 这样的值就不会被视为无效。我们还为这个输入提供了一个占位符。

我们使用 style="display: none;" 隐藏了英尺和英寸的输入，所以米是默认的输入类型。

现在来看看 CSS 代码——跟之前的验证部分样式看起来是一样的，没什么可说的：

div {

margin-bottom: 10px;

position: relative;

}

input[type="number"] {

width: 100px;

}

input + span {

padding-right: 30px;

}

input:invalid + span::after {

position: absolute;

content: "✖";

padding-left: 5px;

}

input:valid + span::after {

position: absolute;

content: "✓";

padding-left: 5px;

}

Copy to Clipboard

最后是一些 JavaScript 代码：

const metersInputGroup = document.querySelector('.metersInputGroup');

const feetInputGroup = document.querySelector('.feetInputGroup');

const metersInput = document.querySelector('#meters');

const feetInput = document.querySelector('#feet');

const inchesInput = document.querySelector('#inches');

const switchBtn = document.querySelector('input[type="button"]');

switchBtn.addEventListener('click', () => {

if (switchBtn.getAttribute('class') === 'meters') {

switchBtn.setAttribute('class', 'feet');

switchBtn.value = 'Enter height in meters';

metersInputGroup.style.display = 'none';

feetInputGroup.style.display = 'block';

feetInput.setAttribute('required', '');

inchesInput.setAttribute('required', '');

metersInput.removeAttribute('required');

metersInput.value = '';

} else {

switchBtn.setAttribute('class', 'meters');

switchBtn.value = 'Enter height in feet and inches';

metersInputGroup.style.display = 'block';

feetInputGroup.style.display = 'none';

feetInput.removeAttribute('required');

inchesInput.removeAttribute('required');

metersInput.setAttribute('required', '');

feetInput.value = '';

inchesInput.value = '';

}

});

Copy to Clipboard

在声明了一些变量之后，我们为按钮添加一个事件监听器来控制切换机制。这很简单，主要是改变按钮的类别和标签，并在按钮被按下时更新两组输入的显示值。请注意，我们不是在米和英尺/英寸之间来回转换，现实生活中的网络应用可能会这么做。

**备注：** 请注意，当用户点击按钮时，我们从隐藏的输入中移除 required 属性，并清空 value 属性。这样我们就可以在两个输入组都没有填写的情况下提交表单，而不会提交那些我们不想提交的数据。如果我们不这样做，你就必须同时填入英尺/英寸**和**米来提交表单了！

## [无障碍](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/number#%E6%97%A0%E9%9A%9C%E7%A2%8D)

<input type="number"> 元素的隐含 [role](https://developer.mozilla.org/zh-CN/docs/Web/Accessibility/ARIA/Roles) 是 [spinbutton (en-US)](https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/Roles/spinbutton_role) 。如果 spinbutton 对你的表单控件来说不是一个重要的功能，那就不要使用 type="number"；此时，请使用带有 [pattern](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Attributes/pattern) 属性的 [inputmode="numeric"](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Global_attributes/inputmode)，将字符限制在数字和相关字符。如果坚持使用 <input type="number">，用户在试图做其他事情时，有可能意外地增加一个数字。此外，如果用户试图输入不是数字的东西，也没有明确的反馈说明他们做错了什么。

也可以考虑使用 [autocomplete](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Attributes/autocomplete) 属性，帮助用户更快完成表单，减少出错的机会。例如，要在一个邮编字段上启用自动填写功能，请设置 autocomplete="postal-code"。

# <input type="range">

**range** 类型的 [<input>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input) 元素允许用户指定一个数值，该数值必须不小于给定值，并且不得大于另一个给定值。但是，其精确值并不重要。通常使用滑块或拨号控件而不是像 [number](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/number) 输入类型这样的文本输入框来表示。

由于这种小部件不精确，因此除非控件的确切值不重要，否则通常不应使用它。

## [尝试一下](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/range#%E5%B0%9D%E8%AF%95%E4%B8%80%E4%B8%8B)

如果用户的浏览器不支持 range 类型，它将回退并将其视为 [text](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/text) 输入。

|  |  |
| --- | --- |
| [**值**](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/range#%E5%80%BC) | 一个代表已选择数值的字符串表示，使用 valueAsNumber 来将此值作为数值获取。 |
| **事件** | [change](https://developer.mozilla.org/zh-CN/docs/Web/API/HTMLElement/change_event) 和 [input](https://developer.mozilla.org/zh-CN/docs/Web/API/HTMLElement/input_event) |
| **支持的常用属性** | [autocomplete](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#autocomplete)、[list](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#list)、[max](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#max)、[min](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#min) 和 [step](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#step) |
| **IDL 属性** | list、value 和 valueAsNumber |
| **DOM 接口** | [HTMLInputElement](https://developer.mozilla.org/zh-CN/docs/Web/API/HTMLInputElement) |
| **方法** | [stepDown() (en-US)](https://developer.mozilla.org/en-US/docs/Web/API/HTMLInputElement/stepDown) 和 [stepUp() (en-US)](https://developer.mozilla.org/en-US/docs/Web/API/HTMLInputElement/stepUp) |

### [验证方式](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/range#%E9%AA%8C%E8%AF%81%E6%96%B9%E5%BC%8F)

没有可用的模式验证。但是，执行以下形式的自动验证：

* 如果将 [value](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#value) 设置为无法转换为有效浮点数的值，则验证将失败，因为输入框的值是错误的。
* 该值不得小于 [min](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#min)，默认值为 0。
* 该值将不大于 [max](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#max)，默认值为 100。
* 该值将是 [step](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#step) 的整倍数，默认值为 1。

### [值](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/range#%E5%80%BC)

[value](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#value) 属性包含所选数字的字符串表示形式。该值绝不能为空字符串（""）。默认值介于指定的最小值和最大值之间，除非最大值实际上小于最小值，在这种情况下，默认值将设置 min 属性的值。确定默认值的算法是：

defaultValue =

rangeElem.max < rangeElem.min

? rangeElem.min

: rangeElem.min + (rangeElem.max - rangeElem.min) / 2;

Copy to Clipboard

如果尝试将值设置为小于最小值，则将其设置为最小值。类似地，尝试将值设置为大于最大值会导致将其设置为最大值。

## [其他属性](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/range#%E5%85%B6%E4%BB%96%E5%B1%9E%E6%80%A7)

除了所有 [<input>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input) 元素共享的属性之外，range 输入还提供以下属性。

### [list](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/range#list)

list 属性指定了一个 [<datalist>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/datalist) 元素的 [id](https://developer.mozilla.org/zh-CN/docs/Web/API/Element/id)，用来提供输入建议。[<datalist>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/datalist) 提供了一个预定义的值列表，向用户建议这个输入。列表中任何与 type 不兼容的值都不包括在建议选项中。所提供的值是建议，不是要求：用户可以从这个预定义的列表中选择，或者提供不同的值。

有关在支持的浏览器中如何表示范围中的选项的示例，请参见下面的[添加井号标记](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/range#%E6%B7%BB%E5%8A%A0%E4%BA%95%E5%8F%B7%E6%A0%87%E8%AE%B0)。

### [max](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/range#max)

允许值范围内的最大值。如果输入到元素中的 [value](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#value) 超过此值，则该元素将无法通过[约束验证](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Constraint_validation)。如果 [max (en-US)](https://developer.mozilla.org/en-US/docs/Web/HTML/Attributes/max) 属性的值不是数字，则元素没有最大值。

此值必须大于或等于 min 属性的值。请参见 HTML [max (en-US)](https://developer.mozilla.org/en-US/docs/Web/HTML/Attributes/max) 属性。

### [min](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/range#min)

允许值范围内的最小值。如果元素的[value](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#value) 小于此值，则该元素将无法通过[约束验证](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Constraint_validation)。如果 min 指定的值不是有效数字，则输入没有最小值。

该值必须小于或等于 [max (en-US)](https://developer.mozilla.org/en-US/docs/Web/HTML/Attributes/max) 属性的值。请参见 HTML [min (en-US)](https://developer.mozilla.org/en-US/docs/Web/HTML/Attributes/min) 属性。

### [step](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/range#step)

step 属性指定了值必须满足的粒度，或者是下文描述的特殊值 any。值必须满足基础的步进值，才有效。如果指定了 [min](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/range#min) 属性，则由 min 属性决定，否则，使用 [value](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#value) 属性的值，如果上述两个值都不存在，则提供适当的默认值。

字符串值 any 意味着不使用步进值，任意值都可以接受（除其他制约因素如 [min](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/range#min) 或 [max](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/range#min) 之外）。

**备注：** 当用户输入的数据不符合步进配置时，[用户代理](https://developer.mozilla.org/zh-CN/docs/Glossary/User_agent)可能会四舍五入到最近的有效值，当有两个同样接近的选项时，更倾向于正方向的数字。

range 输入的默认步进值为 1，除非步进基数不是整数，否则仅允许输入整数；否则，默认值为 1。例如，如果将 min 设置为 -10 并将 value 设置为 1.5，则 1 的 step 将只允许正方向上的值为 1.5、2.5、3.5 等，以及负方向上的值为 -0.5，-1.5，-2.5 等。请参阅 [HTML step 属性 (en-US)](https://developer.mozilla.org/en-US/docs/Web/HTML/Attributes/step)。

## [非标准属性](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/range#%E9%9D%9E%E6%A0%87%E5%87%86%E5%B1%9E%E6%80%A7)

### [orient](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/range#orient)

类似于影响 [<progress>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/progress) 和 [<meter>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/meter) 元素的非标准 CSS 属性 -moz-orient，orient 属性定义范围滑块的方向。值包括 horizontal 和 vertical，分别代表范围的水平呈现和垂直呈现。

**备注：** 以下输入属性不适用于 range 输入：accept、alt、checked、dirname、formaction、formenctype、formmethod、formnovalidate、formtarget、height、maxlength、minlength、multiple、pattern、placeholder、readonly、required、size、src 和 width。如果包含了这些属性中的任何一个，都将被忽略。

## [示例](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/range#%E7%A4%BA%E4%BE%8B)

虽然 number 类型允许用户输入带有可选约束的数字，以强制其值介于最小值和最大值之间，但它确实要求输入特定值。range 输入类型可以在用户甚至不关心或不知道所选的特定数字值是什么的情况下，向用户询问一个值。

常用范围输入的一些情况示例：

* 音频控件（例如音量和平衡）或过滤器控件。
* 颜色配置控件，例如颜色通道、透明度、亮度等。
* 游戏配置控件，例如难度、可见性距离、世界范围等等。
* 密码管理员生成的密码的密码长度。

通常，如果用户对最小值和最大值之间的距离的百分比比实际数字本身更感兴趣，则 range 输入是一个不错的选择。例如，在家庭立体声音量控制的情况下，用户通常会想到“将音量设置为最大音量的一半”而不是“将音量设置为 0.5”。

### [指定最小值和最大值](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/range#%E6%8C%87%E5%AE%9A%E6%9C%80%E5%B0%8F%E5%80%BC%E5%92%8C%E6%9C%80%E5%A4%A7%E5%80%BC)

默认情况下，最小值为 0，最大值为 100。如果这不是想要的值，则可以通过更改 [min](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#min) 和/或 [max](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#max) 属性。这些可以是任何浮点值。

例如，要要求用户输入介于 -10 和 10 之间的值，可以使用：

<input type="range" min="-10" max="10" />

Copy to Clipboard

### [设置值的粒度](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/range#%E8%AE%BE%E7%BD%AE%E5%80%BC%E7%9A%84%E7%B2%92%E5%BA%A6)

默认情况下，粒度为 1，表示该值始终是整数。可以通过更改 [step](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Global_attributes#step) 属性控制粒度。例如，如果需要一个介于 5 到 10 之间的值（精确到两位小数），则应将 step 的值设置为 0.01：

#### 设置 step 属性

<input type="range" min="5" max="10" step="0.01" />

Copy to Clipboard

#### 设置 step 为“any”

如果要接受任何值而不论扩展到小数点后多少位，可以为 [step](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#step) 属性指定值 any。

<input type="range" min="0" max="3.14" step="any" />

Copy to Clipboard

该示例使用户可以选择 0 到 π 之间的任何值，而对所选值的小数部分没有任何限制。

### [添加井号标记](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/range#%E6%B7%BB%E5%8A%A0%E4%BA%95%E5%8F%B7%E6%A0%87%E8%AE%B0)

为了在 range 控制器中添加井号标记，要包含 list 属性，赋予它某个定义了一系列控件上的井号标记 [<datalist>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/datalist) 元素的 id。每个点使用带有 [value](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/option#value) 属性的 [<option>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/option) 元素设置需要添加标记的点。

#### HTML

<label for="temp">选择一个适宜的温度：</label><br />

<input type="range" id="temp" name="temp" list="tickmarks" />

<datalist id="tickmarks">

<option value="0"></option>

<option value="25"></option>

<option value="50"></option>

<option value="75"></option>

<option value="100"></option>

</datalist>

Copy to Clipboard

#### 结果

### [添加标签](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/range#%E6%B7%BB%E5%8A%A0%E6%A0%87%E7%AD%BE)

你可以通过给 <option> 元素赋予 label 属性来标记这些井号标记，然而，你需要使用 CSS 令标签显示且定位正确。以下是可以实现这点的方法之一。

#### HTML

<label for="temp">选择一个适宜的温度：</label><br />

<input type="range" id="temp" name="temp" list="tickmarks" />

<datalist id="tickmarks">

<option value="0" label="很冷！"></option>

<option value="25" label="凉爽"></option>

<option value="50" label="适中"></option>

<option value="75" label="变得暖和了！"></option>

<option value="100" label="很热！"></option>

</datalist>

Copy to Clipboard

#### CSS

datalist {

display: flex;

flex-direction: column;

justify-content: space-between;

writing-mode: vertical-lr;

width: 200px;

}

option {

padding: 0;

}

input[type="range"] {

width: 200px;

margin: 0;

}

Copy to Clipboard

#### 结果

### [创建垂直 range 控件](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/range#%E5%88%9B%E5%BB%BA%E5%9E%82%E7%9B%B4_range_%E6%8E%A7%E4%BB%B6)

默认情况下，浏览器将 range 输入渲染为可左右滑动的旋钮。

要创建一个垂直滑动的 range 输入，请将 CSS [appearance](https://developer.mozilla.org/zh-CN/docs/Web/CSS/appearance) 属性设置为 slider-vertical，并包含 Firefox 的非标准 orient 属性。

#### 水平 range 控件

考虑以下 range 控件：

<input type="range" id="volume" min="0" max="11" value="7" step="1" />

Copy to Clipboard

该控件是水平的（至少在大部分主流浏览器中，其他的根据情况会有所不同）。

#### 使用 appearance 属性

[appearance](https://developer.mozilla.org/zh-CN/docs/Web/CSS/appearance) 属性包含了一个非标准值 slider-vertical 使得滑块垂直。

我们使用与前例相同的 HTML 代码：

<input type="range" min="0" max="11" value="7" step="1" />

Copy to Clipboard

我们以 range 类型的 input 作为目标：

input[type="range"] {

appearance: slider-vertical;

}

Copy to Clipboard

#### 使用 orient 属性

仅在 Firefox 中，有一个非标准的 orient 属性。

使用与前例类似的代码，只是添加一个值为 vertical 的该属性：

<input type="range" min="0" max="11" value="7" step="1" orient="vertical" />

Copy to Clipboard

#### writing-mode: bt-lr

[writing-mode](https://developer.mozilla.org/zh-CN/docs/Web/CSS/writing-mode) 属性一般不应用于改变文本方向以达到国际化或本地化的目的，但可用于特殊效果。

我们使用与前例类似的 HTML 代码：

<input type="range" min="0" max="11" value="7" step="1" />

Copy to Clipboard

我们以 range 类型的 input 作为目标，将写作模式从默认改为 bt-lr（从底部向上，从左到右）：

input[type="range"] {

writing-mode: bt-lr;

}

Copy to Clipboard

#### 将它们放在一起

由于以上这些示例会在不同的浏览器中奏效，你可以将它们放在一起，使得纵向 range 输入可以跨浏览器工作：

我们会在 Firefox 浏览器中保持使用 orient 属性的 vertical 值：

<input type="range" min="0" max="11" value="7" step="1" orient="vertical" />

Copy to Clipboard

我们只对 range 类型的 input 施加 CSS 样式，将写作模式从默认改为 bt-lr，或 bottom-to-top 和 left-to-right（对于 Edge 和 Internet Explorer 浏览器）。对 Blink 和 Webkit 浏览器可以添加 appearance: slider-vertical 配置：

input[type="range"] {

writing-mode: bt-lr;

appearance: slider-vertical;

}

# <input type="search">

**search** 类型的 [<input>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input) 是专为用户输入查询文本而设计的字段。功能上与 [text](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/text) 输入相同，但是根据[用户代理](https://developer.mozilla.org/zh-CN/docs/Glossary/User_agent)不同，可能会有不同的样式表现。

## [尝试一下](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/search#%E5%B0%9D%E8%AF%95%E4%B8%80%E4%B8%8B)

|  |  |
| --- | --- |
| [**值**](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/search#%E5%80%BC) | 代表搜索字段中值的字符串。 |
| **事件** | [change](https://developer.mozilla.org/zh-CN/docs/Web/API/HTMLElement/change_event) 和 [input](https://developer.mozilla.org/zh-CN/docs/Web/API/HTMLElement/input_event) |
| **支持的通用属性** | [autocomplete](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#autocomplete)、[list](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#list)、[maxlength](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#maxlength)、[minlength](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#minlength)、[pattern](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#pattern)、[placeholder](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#placeholder)、[required](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#required)、[size](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#size)。 |
| **IDL 属性** | value |
| **DOM 接口** | [HTMLInputElement](https://developer.mozilla.org/zh-CN/docs/Web/API/HTMLInputElement) |
| **方法** | [select()](https://developer.mozilla.org/zh-CN/docs/Web/API/HTMLInputElement/select)、[setRangeText() (en-US)](https://developer.mozilla.org/en-US/docs/Web/API/HTMLInputElement/setRangeText)、[setSelectionRange()](https://developer.mozilla.org/zh-CN/docs/Web/API/HTMLInputElement/setSelectionRange)。 |

## [值](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/search#%E5%80%BC)

[value](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#value) 属性包含一个代表搜索字段中含有值的字符串。你可以使用 JavaScript 的 HTMLInputElement.value 属性获取该值。

searchTerms = mySearch.value;

Copy to Clipboard

如果没有针对该 input 的验证约束（请参见[验证方式](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/search#%E9%AA%8C%E8%AF%81%E6%96%B9%E5%BC%8F)），该值可以是任何文本字符串或空字符串（""）。

## [其他属性](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/search#%E5%85%B6%E4%BB%96%E5%B1%9E%E6%80%A7)

除了可在所有 [<input>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input) 上应用的属性（无论其类型）之外，搜索字段输入还支持以下属性。

### [list](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/search#list)

list 属性指定了一个 [<datalist>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/datalist) 元素的 [id](https://developer.mozilla.org/zh-CN/docs/Web/API/Element/id)，用来提供输入建议。[<datalist>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/datalist) 提供了一个预定义的值列表，向用户建议这个输入。列表中任何与 [type](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#type) 不兼容的值都不包括在建议选项中。所提供的值是建议，不是要求：用户可以从这个预定义的列表中选择，或者提供不同的值。

### [maxlength](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/search#maxlength)

用户可以在搜索字段中输入的最大字符数（UTF-16 代码单元）。必须为 0 或更高的整数。如果未指定 maxlength 或指定了无效值，则搜索字段没有最大长度。此值还必须大于或等于 minlength 的值。

如果输入到字段中的文本的长度大于 maxlength UTF-16 代码单元的长度，则输入无法通过[约束验证](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Constraint_validation)。

### [minlength](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/search#minlength)

用户可以在搜索字段中输入的最小字符数（UTF-16 代码单元）。该值必须是小于或等于 maxlength 指定的值的非负整数值。如果未指定 minlength 或指定了无效值，则搜索输入没有最小长度。

如果在字段中输入的文本的长度小于 minlength UTF-16 代码单元的长度，则搜索字段将无法通过[约束验证](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Constraint_validation)。

### [pattern](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/search#pattern)

如果指定了 pattern 属性，为了使 value 通过[约束验证](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Constraint_validation)，必须满足该属性给定的正则表达式。它必须是 [RegExp](https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Global_Objects/RegExp) 类型的有效 JavaScript 正则表达式，并且已在我们的[正则表达式指南](https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Guide/Regular_expressions)中进行了说明；在编译正则表达式时指定了 'u' 标志，因此该模式被视为 Unicode 代码点的序列，而不是 ASCII。模式文本周围无需指定正斜杠。

如果模式未指定或无效，则不应用任何正则表达式，并且将完全忽略此属性。

**备注：** 使用 [title](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#title) 属性指定大多数浏览器将显示为文本的工具提示，以说明与模式匹配的要求。你还应该在附近添加其他说明性文字。

请参照[指定模式](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/search#%E6%8C%87%E5%AE%9A%E6%A8%A1%E5%BC%8F)更多内容和例子。

### [placeholder](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/search#placeholder)

placeholder 属性是一个字符串，可向用户提供有关该字段中需要什么样的信息的简短提示。它应该是一个单词或短语来说明预期的数据类型，而不是说明性消息。文本中不得包含回车符或换行符。

如果控件的内容具有方向（[LTR](https://developer.mozilla.org/zh-CN/docs/Glossary/LTR) 或 [RTL](https://developer.mozilla.org/zh-CN/docs/Glossary/RTL)），但需要以相反的方向显示占位符，则可以使用 Unicode 双向算法来格式化字符，从而覆盖原有占位符的方向；请参见[如何针对双向文本使用 Unicode 控制符](https://www.w3.org/International/questions/qa-bidi-unicode-controls)获取更多信息。

**备注：** 尽可能避免使用 placeholder 属性。它在语义上没有其他解释表单的方式有用，并且可能导致内容出现意外的问题。请参见 [<input> 标签](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#%E6%A0%87%E7%AD%BE)以获取更多信息。

### [readonly](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/search#readonly)

一个布尔属性，如果存在，则表示该字段不能由用户编辑。但是，仍可以通过 JavaScript 代码直接设置 [HTMLInputElement](https://developer.mozilla.org/zh-CN/docs/Web/API/HTMLInputElement) 的 value 属性来更改。

**备注：** 因为只读字段不能有值，所以 required 对指定了 readonly 属性的输入没有任何影响。

### [size](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/search#size)

size 属性是一个数字值，指示输入字段应有多少个字符宽。该值必须是一个大于零的数字，默认值是 20。由于字符宽度各不相同，这可能是也可能不是精确的，不应依赖于此；结果输入可能比指定的字符数更窄或更宽，这取决于字符和字体（正在使用的 [font](https://developer.mozilla.org/zh-CN/docs/Web/CSS/font) 设置）。

这并不限制用户可以在该字段中输入多少个字符，仅能指定一次可以看到多少个。要设置输入数据长度的上限，请使用 [maxlength](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/search#maxlength) 属性。

### [spellcheck](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/search#spellcheck)

spellcheck 是一个全局属性，用于指示是否启用元素的拼写检查。它可以用于任何可编辑的内容，但是这里我们考虑在 [<input>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input) 元素上使用 spellcheck 的细节。 spellcheck 的允许值为：

false

禁用此元素的拼写检查。

true

对此元素启用拼写检查。

""（空字符串）或空值

遵循元素的默认行为进行拼写检查。这可能基于父元素 spellcheck 设置或其他因素。

如果输入字段没有设置 [readonly](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/search#readonly) 属性且未禁用，则可以启用拼写检查。

如果[用户代理的](https://developer.mozilla.org/zh-CN/docs/Glossary/User_agent)首选项覆盖了设置，则通过读取 spellcheck 返回的值可能无法反映控件中拼写检查的实际状态。

## [非标准属性](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/search#%E9%9D%9E%E6%A0%87%E5%87%86%E5%B1%9E%E6%80%A7)

以下非标准属性在某些浏览器上对搜索输入字段也可用。一般来说，除非万不得已，否则你应该避免使用它们。

### [autocorrect](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/search#autocorrect)

Safari 扩展，autocorrect 属性是一个字符串，它指示在用户编辑此字段时是否激活自动更正。允许的值为：

on

启用拼写错误的自动更正，以及配置文本替换（如果已配置）的处理。

off

禁用自动更正和文本替换。

### [incremental](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/search#incremental)

布尔值属性 incremental 是 WebKit 和 Blink 的扩展（因此被 Safari、Opera、Chrome 等支持）。如果制定了它，它会告诉[用户代理](https://developer.mozilla.org/zh-CN/docs/Glossary/User_agent)将输入作为实时搜索进行处理。当用户编辑字段的值时，用户代理将 [search (en-US)](https://developer.mozilla.org/en-US/docs/Web/API/HTMLInputElement/search_event) 事件发送到代表搜索框的 [HTMLInputElement](https://developer.mozilla.org/zh-CN/docs/Web/API/HTMLInputElement) 对象。这允许你的代码在用户编辑搜索时实时更新搜索结果。

如果没有指定 incremental ，则仅当用户显式启动搜索时（例如，在编辑字段时按 Enter 或 Return ）才发送 [search (en-US)](https://developer.mozilla.org/en-US/docs/Web/API/HTMLInputElement/search_event) 事件。

search 事件受速率限制，因此发送事件的频率不会超过实现定义的间隔。

### [mozactionhint](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/search#mozactionhint)

Mozilla 扩展，它提供了一个提示，提示用户在编辑字段时按 Enter 或 Return 键将采取何种操作。

此属性已经废弃：请使用全局属性 [enterkeyhint](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Global_attributes#enterkeyhint) 作为替代。

### [results](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/search#results)

results 属性是一个数字值（仅受 Safari 支持），可让你覆盖要在 [<input>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input) 元素原生提供的先前搜索查询的下拉菜单中所显示的最大条目数。

该值必须是非负十进制数字。如果未提供或给出了无效值，则使用浏览器的默认最大条目数。

## [使用搜索输入](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/search#%E4%BD%BF%E7%94%A8%E6%90%9C%E7%B4%A2%E8%BE%93%E5%85%A5)

search 类型的 <input> 元素与 text 类型的元素非常相似，不同之处在于它们专门用于处理搜索条目。它们的行为基本相同，但是用户代理可以选择与默认样式不同的样式（当然，站点可以使用样式表向其应用自定义样式）。

### [基本示例](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/search#%E5%9F%BA%E6%9C%AC%E7%A4%BA%E4%BE%8B)

<form>

<div>

<input type="search" id="mySearch" name="q" />

<button>搜索</button>

</div>

</form>

Copy to Clipboard

如下：

q 是搜索输入中最常用的 name 值，尽管不是强制性的。提交后，发送到服务器的键值对数据将为 q=searchterm。

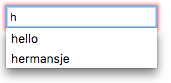
**备注：** 一定要记住为你的 input 设置 [name](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#name)，否则不会提交任何内容。

### [搜索类型和文本类型之间的差异](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/search#%E6%90%9C%E7%B4%A2%E7%B1%BB%E5%9E%8B%E5%92%8C%E6%96%87%E6%9C%AC%E7%B1%BB%E5%9E%8B%E4%B9%8B%E9%97%B4%E7%9A%84%E5%B7%AE%E5%BC%82)

主要的基本区别在于浏览器处理它们的方式。首先要注意的是，某些浏览器显示一个十字图标，如果需要，可以单击该十字图标以立即删除搜索词。在 Chrome 中当按下 escape 键时也会触发这个动作。以下屏幕截图来自 Chrome：

Focused search input, with focus ring, with the text 'cats'. There is an x icon in the input abutting the right side.

此外，现代浏览器还倾向于自动存储先前在各个域中输入的搜索词，在该域的搜索输入中执行后续搜索时，这些搜索词会作为自动完成选项出现。这有助于随着时间的流逝倾向于在相同或相似搜索查询上进行搜索的用户。此屏幕截图来自 Firefox：

现在，让我们看看可以应用于搜索表单的一些有用的技巧。

### [设置占位符](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/search#%E8%AE%BE%E7%BD%AE%E5%8D%A0%E4%BD%8D%E7%AC%A6)

可以在搜索输入中提供一个有用的占位符，使用 [placeholder](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#placeholder) 属性来提示要做什么。看下面的示例：

<form>

<div>

<input

type="search"

id="mySearch"

name="q"

placeholder="站内搜索..." />

<button>搜索</button>

</div>

</form>

Copy to Clipboard

可以在下面看到占位符的呈现方式：

### [搜索表单标签和无障碍](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/search#%E6%90%9C%E7%B4%A2%E8%A1%A8%E5%8D%95%E6%A0%87%E7%AD%BE%E5%92%8C%E6%97%A0%E9%9A%9C%E7%A2%8D)

搜索表单的一个问题是它们的无障碍性。常见的设计惯例是不为搜索字段提供标签（尽管可能会有放大镜或类似图标），因为由于放置位置的原因，搜索表单的目的通常对于视力正常的用户而言相当明显（[这个例子展示了一个典型的模式](https://mdn.github.io/learning-area/accessibility/aria/website-aria-roles/)）。

但是，这可能会使屏幕阅读器用户感到困惑，因为它们不会对搜索输入内容有任何口头指示。解决此问题而不会影响您的视觉设计的一种方法是使用 [WAI-ARIA](https://developer.mozilla.org/zh-CN/docs/Learn/Accessibility/WAI-ARIA_basics) 功能：

* <form> 元素上， search 值的 role 属性向屏幕阅读器用户声明该表单是搜索表单。
* 还可以在 [<input>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input) 本身使用 [aria-label](https://developer.mozilla.org/zh-CN/docs/Web/Accessibility/ARIA/Attributes/aria-label) 属性。这应该是一个描述性的文本标签，屏幕阅读器会读出该标签；它用作 <label> 的非可视等效项。

让我们看一个例子：

<form role="search">

<div>

<input

type="search"

id="mySearch"

name="q"

placeholder="站内搜索..."

aria-label="在全站内容中搜索" />

<button>搜索</button>

</div>

</form>

Copy to Clipboard

您可以在下面看到它的呈现方式：

与上一个示例没有视觉上的区别，但是屏幕阅读器用户可以获得更多可用信息。

**备注：** 有关此类辅助功能的更多信息，请参见 [Signposts/Landmarks](https://developer.mozilla.org/zh-CN/docs/Learn/Accessibility/WAI-ARIA_basics#signpostslandmarks) 。

### [输入框元素物理大小](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/search#%E8%BE%93%E5%85%A5%E6%A1%86%E5%85%83%E7%B4%A0%E7%89%A9%E7%90%86%E5%A4%A7%E5%B0%8F)

可以使用 [size](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#size) 属性来控制输入框的尺寸。使用它，可以指定输入框一次可以显示的字符数。例如，在此示例中，搜索框为 30 个字符宽：

<form>

<div>

<input

type="search"

id="mySearch"

name="q"

placeholder="站内搜索..."

size="30" />

<button>搜索</button>

</div>

</form>

Copy to Clipboard

结果是这个更宽的输入框：

## [验证方式](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/search#%E9%AA%8C%E8%AF%81%E6%96%B9%E5%BC%8F)

<input> 元素的 search 类型具有与常规文本 text 输入相同的验证功能。通常，不太可能希望对搜索框使用验证功能。在许多情况下，应该允许用户搜索任何内容，但是有几种情况需要考虑，例如搜索已知格式的数据。

**备注：** HTML 表单验证不能替代服务器脚本，以确保输入的数据格式正确。对于某人来说，对 HTML 进行调整以使其绕过验证或完全删除验证太容易了。有人也可以完全绕开您的 HTML 并将数据直接提交到您的服务器。如果服务器端代码无法验证接收到的数据，则当将格式不正确的数据（或太大，类型错误的数据等等）输入数据库时，灾难可能会发生。

### [样式注意事项](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/search#%E6%A0%B7%E5%BC%8F%E6%B3%A8%E6%84%8F%E4%BA%8B%E9%A1%B9)

有一些有用的伪类可用于设置有效/无效表单元素的样式：[:valid](https://developer.mozilla.org/zh-CN/docs/Web/CSS/:valid) 和 [:invalid](https://developer.mozilla.org/zh-CN/docs/Web/CSS/:invalid)。在本节中，我们将使用以下 CSS，它将在包含有效值的输入旁边放置一个复选标记（勾号），并在包含无效值的输入旁边放置一个叉号。

input:invalid ~ span::after {

content: "✖";

padding-left: 5px;

position: absolute;

}

input:valid ~ span::after {

content: "✓";

padding-left: 5px;

position: absolute;

}

Copy to Clipboard

该技巧还需要将 [<span>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/span) 元素放置在 form 元素之后，该 form 元素充当图标的持有者。这是必要的，因为某些浏览器上的某些输入类型不能很好地显示直接位于其后的图标。

### [让输入框必填](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/search#%E8%AE%A9%E8%BE%93%E5%85%A5%E6%A1%86%E5%BF%85%E5%A1%AB)

你可以使用 [required](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#required) 属性令输入一个值成为允许提交表单前的必要条件：

<form>

<div>

<input

type="search"

id="mySearch"

name="q"

placeholder="站内搜索..."

required />

<button>搜索</button>

<span class="validity"></span>

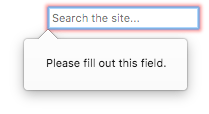
</div>

</form>

Copy to Clipboard

如下所示：

另外，如果尝试在未输入搜索词的情况下提交表单，浏览器将显示一条消息。以下示例来自 Firefox：



当尝试使用输入中包含的不同类型的无效数据提交表单时，将显示不同的消息；请参阅以下示例。

### [输入值长度](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/search#%E8%BE%93%E5%85%A5%E5%80%BC%E9%95%BF%E5%BA%A6)

你可以使用 [minlength](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#minlength) 属性为输入的值指定最小长度（以字符为单位）；同样，使用 [maxlength](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#maxlength) 设置输入值的最大长度。

下面的示例要求输入的值的长度为 4–8 个字符。

<form>

<div>

<label for="mySearch">查找用户</label>

<input

type="search"

id="mySearch"

name="q"

placeholder="用户 ID 的长度为 4-8 个字符"

required

size="30"

minlength="4"

maxlength="8" />

<button>查找</button>

<span class="validity"></span>

</div>

</form>

Copy to Clipboard

如下所示：

如果尝试提交少于 4 个字符的表单，则会收到相应的错误消息（不同的浏览器会有所不同）。如果尝试输入超过 8 个字符，浏览器将不允许你这样做。

### [指定模式](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/search#%E6%8C%87%E5%AE%9A%E6%A8%A1%E5%BC%8F)

你可以使用 [pattern](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#pattern) 属性指定一个正则表达式，输入值必须遵循该正则表达式才能被视为有效（请参照[使用正则表达式进行验证](https://developer.mozilla.org/zh-CN/docs/Learn/Forms/Form_validation#%E4%BD%BF%E7%94%A8%E6%AD%A3%E5%88%99%E8%A1%A8%E8%BE%BE%E5%BC%8F%E6%A0%A1%E9%AA%8C)）。

让我们看一个例子。假设我们想提供一个产品 ID 搜索表格，这些 ID 都是由两个字母和四个数字组成的代码。以下示例对此进行了介绍：

<form>

<div>

<label for="mySearch">根据 ID 查找产品：</label>

<input

type="search"

id="mySearch"

name="q"

placeholder="两个字母接四位数字"

required

size="30"

pattern="[A-z]{2}[0-9]{4}" />

<button>查找</button>

<span class="validity"></span>

</div>

</form>

Copy to Clipboard

如下所示：

## [例子](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/search#%E4%BE%8B%E5%AD%90)

可以在我们的网站 [website-aria-roles](https://github.com/mdn/learning-area/tree/master/accessibility/aria/website-aria-roles) 示例中看到一个在上下文中使用搜索表单的一个好例子（[试试在线运行它](https://mdn.github.io/learning-area/accessibility/aria/website-aria-roles/)）。

# <input type="tel">

**tel** 类型的 [<input>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input) 元素用于让用户输入和编辑电话号码。不同于 [<input type="email">](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/email) 和 [<input type="url">](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/url)，在提交表单之前，输入值不会被自动验证为特定格式，因为世界各地的电话号码格式差别很大。

## [尝试一下](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/tel#%E5%B0%9D%E8%AF%95%E4%B8%80%E4%B8%8B)

尽管 tel 类型的输入在功能上和 text 输入一致，但它们确实有用；其中最明显的就是移动浏览器（特别是在手机上），可能会选择提供为输入电话号码而优化的自定义键盘。使用电话号码的特定输入类型也使添加自定义验证和处理电话号码更方便。

**备注：** 不支持 tel 类型的浏览器会回退为标准 [text](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/text) 类型输入。

<input id="telNo" type="tel">

Copy to Clipboard

|  |  |
| --- | --- |
| [**值**](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/tel#%E5%80%BC) | 代表电话号码的字符串，或为空 |
| **事件** | [change](https://developer.mozilla.org/zh-CN/docs/Web/API/HTMLElement/change_event) 和 [input](https://developer.mozilla.org/zh-CN/docs/Web/API/HTMLElement/input_event) |
| **支持的通用属性** | [autocomplete](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#autocomplete)、[list](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#list)、[maxlength](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#maxlength)、[minlength](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#minlength)、[pattern](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#pattern)、[placeholder](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#placeholder)、[readonly](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#readonly) 和 [size](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#size) |
| **IDL 属性** | list、selectionStart、selectionEnd、selectionDirection 和 value |
| **DOM 接口** | [HTMLInputElement](https://developer.mozilla.org/zh-CN/docs/Web/API/HTMLInputElement) |
| **方法** | [select()](https://developer.mozilla.org/zh-CN/docs/Web/API/HTMLInputElement/select)、[setRangeText() (en-US)](https://developer.mozilla.org/en-US/docs/Web/API/HTMLInputElement/setRangeText) 和 [setSelectionRange()](https://developer.mozilla.org/zh-CN/docs/Web/API/HTMLInputElement/setSelectionRange) |

## [值](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/tel#%E5%80%BC)

[<input>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input) 元素的 [value](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#value) 属性包含表示一个电话号码的字符串或者一个空字符串（""）。

## [额外属性](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/tel#%E9%A2%9D%E5%A4%96%E5%B1%9E%E6%80%A7)

除了对 [<input>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input) 元素生效的通用属性外，电话号码输入框还支持以下属性。

### [list](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/tel#list)

list 属性的值是位于同一文档中的 [<datalist>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/datalist) 元素的 [id](https://developer.mozilla.org/zh-CN/docs/Web/API/Element/id)。[<datalist>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/datalist) 提供了一个预定义的值列表，向用户建议向这个输入框输入的值。列表中任何与 [type](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#type) 不兼容的值都不包括在建议选项中。所提供的值是建议，并不是必需的：用户可以从这个预定义的列表中选择，或者提供不同的值。

### [maxlength](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/tel#maxlength)

用户可以在电话号码字段中输入的最大字符数（作为 UTF-16 编码单位）。这必须是一个不低于 0 的整数值。如果没有指定 maxlength，或者指定了一个无效的值，则电话号码字段没有最大长度。这个值也必须大于或等于 minlength 的值。

如果输入字段的文本长度大于 maxlength UTF-16 编码单位长度，则输入将无法通过[约束验证](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Constraint_validation)。

### [minlength](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/tel#minlength)

用户可以在电话号码字段中输入的最小字符数（作为 UTF-16 编码单位）。这必须是一个小于或等于 maxlength 所指定的值的非负整数。如果没有指定 minlength，或者指定了一个无效的值，电话号码的输入就没有最小长度。

如果输入字段的文本长度小于 maxlength UTF-16 编码单位长度，则输入将无法通过[约束验证](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Constraint_validation)。

### [pattern](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/tel#pattern)

如果指定了 pattern 属性，则该 input 的 [value](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Global_attributes#value) 将必须要满足其所指定的正则表达式，以通过[约束验证](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Constraint_validation)。它必须是一个合法的 [RegExp](https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Global_Objects/RegExp) 类型 JavaScript 正则表达式，其具体使用方法记载到了我们的[正则表达式指南](https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Guide/Regular_expressions)中。在编译正则表达式时，使用了 'u' 标志，故所指定模式将视为 Unicode 码点，而非 ASCII。在模式文本中，不可以指定正斜杠符号。

具体介绍和示例请参阅[模式验证](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/tel#%E6%A8%A1%E5%BC%8F%E9%AA%8C%E8%AF%81)部分。

### [placeholder](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/tel#placeholder)

placeholoder 属性是一个字符串，它向用户提供一个简短的提示，说明该字段预计会有什么样的信息。它应该是一个显示预期数据类型的单词或短语，而不是一个解释性的信息。该文本不得包括回车或换行。

如果控件的内容具有方向（[LTR](https://developer.mozilla.org/zh-CN/docs/Glossary/LTR) 或 [RTL](https://developer.mozilla.org/zh-CN/docs/Glossary/RTL)），但需要以相反的方向性呈现占位符，可以使用 Unicode 双向算法格式化字符来覆盖占位符内的方向性；更多信息请参见[如何使用 Unicode 控件处理 bidi 文本](https://www.w3.org/International/questions/qa-bidi-unicode-controls)。

**备注：** 如果可以的话，避免使用 placeholder 属性。它在语义上不如其他解释表单的方式有用，而且会对你的内容造成意想不到的技术问题。参见 [<input> 标签](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#%E6%A0%87%E7%AD%BE)获取更多信息。

### [readonly](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/tel#readonly)

一个布尔属性，如果存在，意味着这个字段不能被用户编辑。但是，它的 value 仍然可以由 JavaScript 代码通过设置 [HTMLInputElement](https://developer.mozilla.org/zh-CN/docs/Web/API/HTMLInputElement) 的 value 属性来改变。

**备注：** 由于只读字段不可以拥有值，required 对指定了 readonly 属性的输入字段不起作用。

### [size](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/tel#size)

size 属性是一个数值，表示输入字段应该有多少个字符宽。该值必须是一个大于 0 的数字，默认值是 20。由于字符的宽度各不相同，该值可能并不精确，而且不应该依赖于此；结果输入可能比指定的字符数更窄或更宽，这取决于使用中的字符和字体（[font](https://developer.mozilla.org/zh-CN/docs/Web/CSS/font)）设置。

这并不是对用户可以在该字段中输入多少个字符的限制。它只是指定了一次大约可以看到多少个。要设置输入数据的长度上限，请使用 [maxlength](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/tel#maxlength) 属性。

## [非标准属性](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/tel#%E9%9D%9E%E6%A0%87%E5%87%86%E5%B1%9E%E6%80%A7)

以下非标准属性可用于电话号码输入字段。一般来说，你应该避免使用它们，除非没有办法。

### [autocorrect](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/tel#autocorrect)

Safari 的扩展，autocorrect 属性是一个字符串，表示在用户编辑这个字段时是否要激活自动更正。允许的值有：

on

启用自动更正和任何配置的文本替换处理。

off

禁用自动更正和文本替换。

### [mozactionhint](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/tel#mozactionhint)

Mozilla 的扩展，当用户编辑该字段时按下键盘的 Enter 或 Return 键时，提供的操作帮助。

该属性已经废弃：作为替代，请使用全局属性 [enterkeyhint](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Global_attributes#enterkeyhint)。

## [使用 tel 输入](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/tel#%E4%BD%BF%E7%94%A8_tel_%E8%BE%93%E5%85%A5)

电话号码是网络上非常普遍收集的数据类型。例如，在创建任何类型的注册或电子商务网站时，无论出于商业目的还是出于紧急联系目的，您都可能需要向用户索要电话号码。鉴于通常输入的电话号码是多少，不幸的是，用于验证电话号码的“一刀切”解决方案是不实际的。

幸运的是，你可以考虑自己网站的要求，并自行实施适当的验证级别。有关详细信息，请参阅下面的[验证](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/tel#%E9%AA%8C%E8%AF%81)部分。

### [自定义键盘](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/tel#%E8%87%AA%E5%AE%9A%E4%B9%89%E9%94%AE%E7%9B%98)

<input type="tel"> 主要优势是它可以在移动浏览器显示一个特殊的电话号码输入键盘。以下是一些设备上键盘的外观。

| **Android 上的 Firefox 浏览器** | **WebKit iOS (Safari/Chrome/Firefox)** |
| --- | --- |
| Firefox for Android screen shot | Firefox for iOS screenshot |

### [一个简单的 tel 输入](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/tel#%E4%B8%80%E4%B8%AA%E7%AE%80%E5%8D%95%E7%9A%84_tel_%E8%BE%93%E5%85%A5)

在最基本的形式中，tel 输入可以这样实现：

<label for="telNo">电话号码：</label>

<input id="telNo" name="telNo" type="tel" />

Copy to Clipboard

这里没有什么神奇的事情发生。当提交给服务器时，上述输入的数据将被表示为 telNo=+12125553151。

### [占位符](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/tel#%E5%8D%A0%E4%BD%8D%E7%AC%A6)

有时候提供关于输入数据应该采用什么形式的上下文提示是很有帮助的。如果页面设计没有为每个 [<input>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input) 页面提供描述性标签，这可能是特别重要的。所以需要**占位符**。一个占位符是一个值，它通过提供一个有效值的例子来演示值的形式，当元素的值是""时，它显示在编辑框中。一旦数据输入框中，占位符就会消失；如果该框被清空，占位符会重新出现。

在这里，我们的 tel 输入的占位符是 123-4567-8901。请注意占位符如何消失并在编辑字段内容时重新出现。

<input id="telNo" name="telNo" type="tel" placeholder="123-4567-8901" />

Copy to Clipboard

### [控制输入长度](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/tel#%E6%8E%A7%E5%88%B6%E8%BE%93%E5%85%A5%E9%95%BF%E5%BA%A6)

不仅可以控制输入框的物理长度，还可以控制输入文本自身允许的最小和最大长度。

#### 输入元素的物理长度

可以使用 [size](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#size) 属性来控制输入框的物理大小，通过它，你可以指定输入框依次可以显示多少个字符。在以下示例中，tel 编辑框具有 20 个字符的宽度：

<input id="telNo" name="telNo" type="tel" size="20" />

Copy to Clipboard

#### 元素值的长度

size 和电话号码的长度限制是分开的，你可以使用 [minlength](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#minlength) 属性为输入的电话号码指定一个最小长度，以字符为单位；同样，使用 [maxlength](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#maxlength) 来设置输入电话号码的最大长度。

下面的示例创建了一个 20 个字符的电话号码输入框，要求内容不少于 9 个字符且不超过 14 个字符。

<input

id="telNo"

name="telNo"

type="tel"

size="20"

minlength="9"

maxlength="14" />

Copy to Clipboard

**备注：** 上述属性确实会影响[验证](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/tel#%E9%AA%8C%E8%AF%81)——如果值的长度小于 9 个字符，或者大于 14 个，上述示例的输入将被视为无效。大多数浏览器甚至不会让你输入超过最大长度的值。

### [提供默认选项](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/tel#%E6%8F%90%E4%BE%9B%E9%BB%98%E8%AE%A4%E9%80%89%E9%A1%B9)

#### 使用 value 属性提供简单默认选项

与往常一样，你可以通过设置其 [value](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#value) 属性为 tel 输入框提供默认值：

<input id="telNo" name="telNo" type="tel" value="333-4444-4444" />

Copy to Clipboard

#### 提供建议值

你可以更进一步，提供一个默认电话号码的列表，让用户从中选择。要做到这一点，使用 [list](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#list) 属性。这不会将用户限制在这些选项中，但确实能让他们更快选择常用的电话号码。这也为 [autocomplete](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#autocomplete) 提供提示。list 属性指定了一个 [<datalist>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/datalist) 元素的 ID，该元素又包含了每个建议值的一个 [<option>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/option) 元素；每个 option 的 value 是电话号码输入框的相应建议值。

<label for="telNo">电话号码：</label>

<input id="telNo" name="telNo" type="tel" list="defaultTels" />

<datalist id="defaultTels">

<option value="111-1111-1111"></option>

<option value="122-2222-2222"></option>

<option value="333-3333-3333"></option>

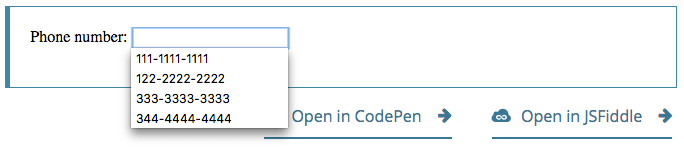
<option value="344-4444-4444"></option>

</datalist>

Copy to Clipboard

有了 [<datalist>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/datalist) 元素及其 [<option>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/option) 子元素，浏览器将提供指定的值作为电话号码的潜在值，通常以弹出式或下拉式菜单的形式呈现，其中包含建议。虽然具体的用户体验可能因浏览器而异，但通常情况下，点击编辑框就会出现下拉式的建议电话号码。然后，随着用户的输入，列表被调整为只显示过滤后的匹配值。每一个键入的字符都会缩小列表的范围，直到用户做出选择或键入一个自定义值。

以下屏幕截图是可能的外观表现：



## [验证](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/tel#%E9%AA%8C%E8%AF%81)

正如我们之前谈到的那样，为电话号码提供一种通用的客户端验证解决方案是相当困难的。所以，我们能做些什么？让我们考虑一些选项。

**警告：** HTML 表单验证不能替代服务器端脚本，以确保输入的数据在被允许进入数据库之前是正确的格式。对于有些人来说，调整 HTML 代码是非常容易的，这样他们就可以绕过验证，或者完全删除它。也有人可能完全绕过你的 HTML，直接提交数据到你的服务器。如果你的服务器端代码无法验证接收到的数据，那么当格式不正确的数据（或数据太大，类型错误等等）输入到数据库时，可能会导致灾难。

### [要求电话号码必填](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/tel#%E8%A6%81%E6%B1%82%E7%94%B5%E8%AF%9D%E5%8F%B7%E7%A0%81%E5%BF%85%E5%A1%AB)

你可以利用 [required](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#required) 属性令空的输入无效，并阻止其提交到你的服务器。例如，使用以下 HTML 代码：

<form>

<div>

<label for="telNo">输入一个电话号码（必填）：</label>

<input id="telNo" name="telNo" type="tel" required />

<span class="validity"></span>

</div>

<div>

<button>提交</button>

</div>

</form>

Copy to Clipboard

我们包含以下 CSS 代码，将有效输入显示为对勾，无效输入显示错叉：

div {

margin-bottom: 10px;

position: relative;

}

input[type="number"] {

width: 100px;

}

input + span {

padding-right: 30px;

}

input:invalid + span::after {

position: absolute;

content: "✖";

padding-left: 5px;

color: #8b0000;

}

input:valid + span::after {

position: absolute;

content: "✓";

padding-left: 5px;

color: #009000;

}

Copy to Clipboard

以上代码的输出如下图所示：

### [模式验证](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/tel#%E6%A8%A1%E5%BC%8F%E9%AA%8C%E8%AF%81)

如果你想进一步限制输入的数字，使它们也必须符合特定的模式，你可以使用 [pattern](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#pattern) 属性，它的值是[正则表达式](https://developer.mozilla.org/zh-CN/docs/Glossary/Regular_expression)，输入的值必须符合这个正则表达式。

在这个示例中我们将使用同样的 CSS 代码，但是 HTML 代码将做一些修改：

<form>

<div>

<label for="telNo">

以 xxx-xxx-xxxx 格式输入一个电话号码：

</label>

<input

id="telNo"

name="telNo"

type="tel"

required

pattern="[0-9]{3}-[0-9]{3}-[0-9]{4}" />

<span class="validity"></span>

</div>

<div>

<button>提交</button>

</div>

</form>

Copy to Clipboard

请注意，除非与模式 xxx-xxx-xxxx 相匹配，否则输入的值将被报告为无效；例如，41-323-421 将不被接受。800-MDN-ROCKS 也不被接受。然而，865-555-6502 将被接受。这个特定的模式显然只对某些地区有用——在实际应用中，你可能需要根据用户的地区来改变使用的模式。

## [示例](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/tel#%E7%A4%BA%E4%BE%8B)

在这个示例中，我们展示了一个简单的界面，其中有一个 [<select>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/select) 元素，让用户选择他们所在的国家，还有一组 <input type="tel"> 元素，让他们输入电话号码的每一部分；显然你应该有多个 tel 输入。

每个输入都有一个 [placeholder](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#placeholder) 属性，以向视力正常的用户显示一个关于输入内容的提示，一个 [pattern](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#pattern) 来执行所需部分的特定字符数。和一个 [aria-label](https://developer.mozilla.org/zh-CN/docs/Web/Accessibility/ARIA/Attributes/aria-label)属性，以包含一个提示，向屏幕阅读器用户宣读要输入的信息。

<form>

<div>

<label for="country">Choose your country:</label>

<select id="country" name="country">

<option>UK</option>

<option selected>US</option>

<option>Germany</option>

</select>

</div>

<div>

<p>输入你的电话号码：</p>

<span class="areaDiv">

<input

id="areaNo"

name="areaNo"

type="tel"

required

placeholder="Area code"

pattern="[0-9]{3}"

aria-label="Area code" />

<span class="validity"></span>

</span>

<span class="number1Div">

<input

id="number1"

name="number1"

type="tel"

required

placeholder="First part"

pattern="[0-9]{3}"

aria-label="First part of number" />

<span class="validity"></span>

</span>

<span class="number2Div">

<input

id="number2"

name="number2"

type="tel"

required

placeholder="Second part"

pattern="[0-9]{4}"

aria-label="Second part of number" />

<span class="validity"></span>

</span>

</div>

<div>

<button>Submit</button>

</div>

</form>

Copy to Clipboard

JavaScript 代码相对简单——它包含一个 [onchange](https://developer.mozilla.org/zh-CN/docs/Web/API/HTMLElement/change_event) 事件处理程序，当 <select> 值改变时，更新 <input> 元素的 pattern、placeholder 和 aria-label 以适应那个国家或地区的电话号码格式。

const selectElem = document.querySelector("select");

const inputElems = document.querySelectorAll("input");

selectElem.onchange = () => {

for (let i = 0; i < inputElems.length; i++) {

inputElems[i].value = "";

}

if (selectElem.value === "US") {

inputElems[2].parentNode.style.display = "inline";

inputElems[0].placeholder = "Area code";

inputElems[0].pattern = "[0-9]{3}";

inputElems[1].placeholder = "First part";

inputElems[1].pattern = "[0-9]{3}";

inputElems[1].setAttribute("aria-label","First part of number");

inputElems[2].placeholder = "Second part";

inputElems[2].pattern = "[0-9]{4}";

inputElems[2].setAttribute("aria-label", "Second part of number");

} else if (selectElem.value === "UK") {

inputElems[2].parentNode.style.display = "none";

inputElems[0].placeholder = "Area code";

inputElems[0].pattern = "[0-9]{3,6}";

inputElems[1].placeholder = "Local number";

inputElems[1].pattern = "[0-9]{4,8}";

inputElems[1].setAttribute("aria-label", "Local number");

} else if (selectElem.value === "Germany") {

inputElems[2].parentNode.style.display = "inline";

inputElems[0].placeholder = "Area code";

inputElems[0].pattern = "[0-9]{3,5}";

inputElems[1].placeholder = "First part";

inputElems[1].pattern = "[0-9]{2,4}";

inputElems[1].setAttribute("aria-label","First part of number");

inputElems[2].placeholder = "Second part";

inputElems[2].pattern = "[0-9]{4}";

inputElems[2].setAttribute("aria-label","Second part of number");

}

}

Copy to Clipboard

示例的显示效果如下所示：

这是一个有趣的想法，它显示了处理国际电话号码问题的潜在解决方案。你将不得不扩大这个范例，为潜在的每个国家提供正确的模式，工作量会相当繁重，且仍然不可能全部保证用户所输入的号码值是正确的。

值得在客户端上这么麻烦吗？换个思路，就让用户在客户端随便输入任何它们想要的格式，然后让服务器进行处理，看你怎么想了。

# <input type="time">

类型为 **time** 的 [<input>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input) 元素，旨在让用户轻松输入时间（小时和分钟，以及可选的秒）。

控件的用户界面因浏览器而异，请查阅[浏览器兼容性](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/time#%E6%B5%8F%E8%A7%88%E5%99%A8%E5%85%BC%E5%AE%B9%E6%80%A7)以了解更多细节。在不支持该类型的浏览器中，它会优雅地降级为 [<input type="text">](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/text)。

## [尝试一下](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/time#%E5%B0%9D%E8%AF%95%E4%B8%80%E4%B8%8B)

## [外观](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/time#%E5%A4%96%E8%A7%82)

### [Chrome 和 Opera](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/time#chrome_%E5%92%8C_opera)

在 Chrome 或 Opera 中，time 选择控件很简单，根据操作系统区域设置，插槽以 12 或 24 小时格式输入小时和分钟，上下箭头用于递增和递减当前选定的组件。在某些版本中，会提供一个“x”按钮来清除控件的值。

12 小时制输入格式12 小时制

24 小时制输入格式24 小时制

### [Firefox](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/time#firefox)

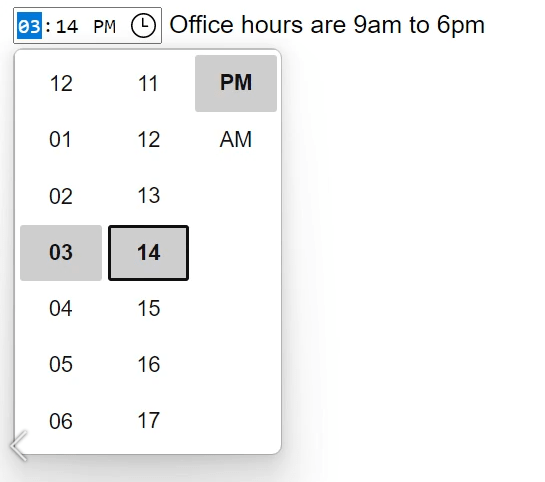
Firefox 的 time 选择控件和 Chrome 非常相似，只是它没有上下箭头。它还基于系统区域设置使用 12 或 24 小时格式输入时间，也提供“X”按钮以清除控件的值。

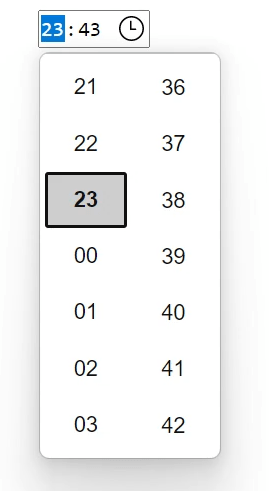
Firefox 的 12 小时制输入格式12 小时制

Firefox 的 24 小时制输入格式24 小时制

### [Edge](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/time#edge)

Edge 的 time 选择控件有点复杂，打开一个小时和分钟的滚动区域。它和 Chrome 一样，基于系统区域设置使用 12 或 24 小时格式输入时间：

12 小时制

24 小时制

|  |  |
| --- | --- |
| [**值**](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/time#%E5%80%BC) | 一个代表时间的字符串，或为空字符串。 |
| **事件** | [change](https://developer.mozilla.org/zh-CN/docs/Web/API/HTMLElement/change_event) 和 [input](https://developer.mozilla.org/zh-CN/docs/Web/API/HTMLElement/input_event) |
| **支持的常用属性** | [autocomplete](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#autocomplete)、[list](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#list)、[readonly](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#readonly) 和 [step](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#step) |
| **IDL 属性** | value、valueAsDate、valueAsNumber 和 list |
| **DOM 接口** | [HTMLInputElement](https://developer.mozilla.org/zh-CN/docs/Web/API/HTMLInputElement) |
| **方法** | [select()](https://developer.mozilla.org/zh-CN/docs/Web/API/HTMLInputElement/select)、[stepDown() (en-US)](https://developer.mozilla.org/en-US/docs/Web/API/HTMLInputElement/stepDown) 和 [stepUp() (en-US)](https://developer.mozilla.org/en-US/docs/Web/API/HTMLInputElement/stepUp) |

## [值](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/time#%E5%80%BC)

输入到该控件中，包含时间值的一个字符串。

### [设置 value 属性](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/time#%E8%AE%BE%E7%BD%AE_value_%E5%B1%9E%E6%80%A7)

你可以通过在创建 <input> 元素时在 [value](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#value) 属性包含一个合法的时间来设置默认值，比如：

<label for="appt-time">选择约定的时间：</label>

<input id="appt-time" type="time" name="appt-time" value="13:30" />

Copy to Clipboard

### [使用 JavaScript 设置 value](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/time#%E4%BD%BF%E7%94%A8_javascript_%E8%AE%BE%E7%BD%AE_value)

你也可以在 JavaScript 代码中通过 [HTMLInputElement](https://developer.mozilla.org/zh-CN/docs/Web/API/HTMLInputElement) 的 value 属性获取和设置时间的值，比如：

const timeControl = document.querySelector('input[type="time"]');

timeControl.value = '15:30';

Copy to Clipboard

### [时间格式](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/time#%E6%97%B6%E9%97%B4%E6%A0%BC%E5%BC%8F)

time 输入框无论输入何种格式的值（这可能由用户的区域设置，或用户代理决定），其 value 值总是带有前导零的 24 小时制时间：hh:mm。如果输入的时间包含秒数（参见[使用 step 属性](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/time#%E4%BD%BF%E7%94%A8_step_%E5%B1%9E%E6%80%A7)），则格式总是为 hh:mm:ss。你可以在[时间字符串](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Date_and_time_formats#%E6%97%B6%E9%97%B4%E5%AD%97%E7%AC%A6%E4%B8%B2)中了解到更多有关时间值的格式。

在此示例中，你可以通过输入时间并查看之后如何更改来查看时间输入的值。

首先，看一下 HTML。这很简单，带有我们之前看到的标签和输入控件，但添加了额外的元素 [<p>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/p) 和 [<span>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/span) 用来显示 time 控件的值：

<form>

<label for="startTime">开始时间：</label>

<input type="time" id="startTime" />

<p>

<code>time</code> 输入框的值：

<code> "<span id="value">n/a</span>"</code>.

</p>

</form>

Copy to Clipboard

这段 JavaScript 代码将代码添加到时间 input 中，以监听 [input](https://developer.mozilla.org/zh-CN/docs/Web/API/HTMLElement/input_event) 事件，每次 input 元素的内容发生变化时都会触发该事件。每次事件触发时，<span> 元素的内容都会替换为 input 元素的最新值。

const startTime = document.getElementById("startTime");

const valueSpan = document.getElementById("value");

startTime.addEventListener("input", () => {

valueSpan.innerText = startTime.value;

}, false);

Copy to Clipboard

当包含 time 输入框的表单被提交后，该值在被包含在表单数据中之前被编码。表单的时间输入数据总是以 name=hh%3Amm 的形式出现，如果包含秒数，则以 name=hh%3Amm%3Ass 的形式出现（参见[使用 step 属性](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/time#%E4%BD%BF%E7%94%A8_step_%E5%B1%9E%E6%80%A7)）。

## [其他属性](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/time#%E5%85%B6%E4%BB%96%E5%B1%9E%E6%80%A7)

除了 [<input>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input) 元素共有的属性外，时间输入还提供以下属性：

**备注：** 与许多数据类型不同，时间值有一个**周期域**，意味着数值达到可能的最高值之后，就会绕回起点。例如，指定 min 为 14:00，max 为 2:00 意味着允许的时间值从下午 2:00 开始，通过午夜运行到第二天，结束于凌晨 2:00。更多信息请参见本文的[使最小值和最大值跨越午夜](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/time#%E4%BD%BF%E6%9C%80%E5%B0%8F%E5%80%BC%E5%92%8C%E6%9C%80%E5%A4%A7%E5%80%BC%E8%B7%A8%E8%B6%8A%E5%8D%88%E5%A4%9C)部分。

### [list](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/time#list)

列表属性的值是位于同一文档中的 [<datalist>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/datalist) 元素的 [id](https://developer.mozilla.org/zh-CN/docs/Web/API/Element/id)。[<datalist>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/datalist) 提供了一个预定义的值列表，向用户建议这个输入。列表中任何与 [type](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#type) 不兼容的值都不包括在建议选项中。所提供的值是建议，不是要求：用户可以从这个预定义的列表中选择，或者提供不同的值。

### [max](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/time#max)

一个代表所接受的最迟时间的字符串，使用上述的[时间格式](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/time#%E6%97%B6%E9%97%B4%E6%A0%BC%E5%BC%8F)指定。如果指定的字符串不是合法的时间表示，则不会设置最大值。

### [min](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/time#min)

一个代表所接受的最早时间的字符串，使用上述的[时间格式](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/time#%E6%97%B6%E9%97%B4%E6%A0%BC%E5%BC%8F)指定。如果指定的字符串不是合法的时间表示，则不会设置最小值。

### [readonly](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/time#readonly)

如果该布尔属性存在，意味着用户将不能编辑此字段。然而其 value 值仍然可以直接通过 JavaScript 代码设置 [HTMLInputElement](https://developer.mozilla.org/zh-CN/docs/Web/API/HTMLInputElement) 的 value 属性改变。

**备注：** 由于只读字段不可以拥有值，required 不会对指定了 readonly 属性的字段起作用。

### [step](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/time#step)

step 属性指定了值必须满足的粒度，或者是下文描述的特殊值 any。值必须满足基础的步进值，才有效。如果指定了 [min](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/time#min) 属性，则由 min 属性决定，否则，使用 [value](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#value) 属性的值，如果上述两个值都不存在，则提供适当的默认值。

字符串值 any 意味着不使用步进值，任意值都可以接受（除其他制约因素如 [min](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/time#min) 或 [max](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/time#max) 之外）。

**备注：** 当用户输入的数据不符合步进配置时，[用户代理](https://developer.mozilla.org/zh-CN/docs/Glossary/User_agent)可能会四舍五入到最近的有效值，当有两个同样接近的选项时，更倾向于正方向的数字。

对于 time 输入， step 的值以秒为单位，比例因子为 1000（因为基础数值以毫秒为单位）。step 的默认值为 60，表示 60 秒（或 1 分钟、60000 毫秒）。

目前，尚不清楚当与 *time* 输入一起使用时，*any* 的值对 *step* 意味着什么。确定该信息后，将对其进行更新。

## [使用 time 输入](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/time#%E4%BD%BF%E7%94%A8_time_%E8%BE%93%E5%85%A5)

### [time 的基本用法](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/time#time_%E7%9A%84%E5%9F%BA%E6%9C%AC%E7%94%A8%E6%B3%95)

<input type="time"> 的最简单用法涉及基本的 <input> 和 [<label>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/label) 元素组合，如下所示：

<form>

<label for="appt-time">选择约定的时间：</label>

<input id="appt-time" type="time" name="appt-time" />

</form>

Copy to Clipboard

### [控制输入框大小](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/time#%E6%8E%A7%E5%88%B6%E8%BE%93%E5%85%A5%E6%A1%86%E5%A4%A7%E5%B0%8F)

<input type="time"> 不支持诸如 [size](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#size) 的表单大小属性，因为时间字符串的长度总是相同的。你必须依靠 [CSS](https://developer.mozilla.org/zh-CN/docs/Web/CSS) 来确定大小。

### [使用 step 属性](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/time#%E4%BD%BF%E7%94%A8_step_%E5%B1%9E%E6%80%A7)

你可以使用 [step](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#step) 属性来更改每次递增或递减的时间数（例如，点击箭头控件时时间值会以 10 分钟为单位变动）。

**备注：** 此属性在不同浏览器之间有一些怪异的行为，并不可靠。

它需要一个整数，相当于你要增加的秒数；默认值是 60 秒，或 1 分钟。如果你指定一个小于 60 秒（1 分钟）的值，time 输入将显示秒的输入区，同时显示小时和分钟。

<form>

<label for="appt-time">选择约定的时间：</label>

<input id="appt-time" type="time" name="appt-time" step="2" />

</form>

Copy to Clipboard

在 Chrome 和 Opera 这两个为数不多显示上下迭代箭头的浏览器中，点击箭头可以将秒的数值改变两秒，但不会影响小时或分钟。分（或小时）只有在你指定了以秒为单位的分钟（或小时）数时，才能用于步进，比如 120 代表 2 分钟，7200 代表 2 小时）。

在 Firefox 浏览器中，没有箭头，所以没有使用 step 值。然而，提供它会在分钟部分的旁边增加秒数输入区。

在 Edge 浏览器中，step 值不起作用。

**备注：** 使用 step 可能会导致验证工作不正常（如下一节所见）。

## [验证方式](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/time#%E9%AA%8C%E8%AF%81%E6%96%B9%E5%BC%8F)

默认情况下，<input type="time"> 不会对输入的值应用任何验证，用户代理接口会屏蔽所有非时间值的输入，尽管这很有用（假设用户代理完全支持 time 输入），但是仍然无法完全依赖于该值的合法性。它仍然允许空字符串（""），且可能允许那些看起来像，但并不正确的时间值（如 25:05）。

### [设置最大和最小时间](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/time#%E8%AE%BE%E7%BD%AE%E6%9C%80%E5%A4%A7%E5%92%8C%E6%9C%80%E5%B0%8F%E6%97%B6%E9%97%B4)

你可以使用 [min](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#min) 和 [max](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#max) 属性来限制用户可以选择的有效时间。在以下示例中，我们设置了可供选择的最小值 12:00 和最大值 18:00：

<form>

<label for="appt-time">

选择约定的时间（开放时间为 12:00 到 18:00）：

</label>

<input id="appt-time" type="time" name="appt-time" min="12:00" max="18:00" />

<span class="validity"></span>

</form>

Copy to Clipboard

下面是上述例子中使用的 CSS。这里我们使用了 [:valid](https://developer.mozilla.org/zh-CN/docs/Web/CSS/:valid) 和 [:invalid](https://developer.mozilla.org/zh-CN/docs/Web/CSS/:invalid) 这两个 CSS 属性，根据当前值是否有效来为输入添加样式。我们不得不把图标放在 input 旁边的 [<span>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/span) 上，而不是放在 input 框本身，因为在 Chrome 中，生成的内容被放在表单控件里面，无法有效地进行样式设计或显示。

div {

margin-bottom: 10px;

position: relative;

}

input[type="number"] {

width: 100px;

}

input + span {

padding-right: 30px;

}

input:invalid + span::after {

position: absolute;

content: "✖";

padding-left: 5px;

}

input:valid + span::after {

position: absolute;

content: "✓";

padding-left: 5px;

}

Copy to Clipboard

运行的结果可以说明：

* 只有 12:00 和 18:00 之间的时间才会被视为有效，超出这个范围的时间将被表示为无效。
* 根据你使用的浏览器，你可能会发现指定范围之外的时间甚至无法在时间选择器中选择（例如 Edge）。

#### 使最小值和最大值跨越午夜

如果设置的 [min](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#min) 属性大于 [max](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#max) 属性，有效时间范围将环绕午夜，产生一个跨越午夜的有效时间范围。这个功能不被任何其他 input 类型所支持。虽然这个功能[在 HTML 规范中](https://html.spec.whatwg.org/multipage/input.html#has-a-reversed-range)，但它还没有被普遍支持。基于 Chrome 的浏览器从 82 版开始支持该功能，Firefox 在 76 版中加入了该功能。Safari 从 14.1 版开始不支持这个功能。要做好出现这种情况的准备。

const input = document.createElement('input');

input.type = 'time';

input.min = '23:00';

input.max = '01:00';

input.value = '23:59';

if (input.validity.valid && input.type === 'time') {

// <input type=time> reversed range supported

} else {

// <input type=time> reversed range unsupported

}

Copy to Clipboard

### [使时间成为必需值](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/time#%E4%BD%BF%E6%97%B6%E9%97%B4%E6%88%90%E4%B8%BA%E5%BF%85%E9%9C%80%E5%80%BC)

此外，你可以使用 [required](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#required) 属性来强制填写时间。因此，如果你试图提交一个超出设定范围的时间，或一个空的时间字段，支持的浏览器将显示一个错误。

让我们看一个例子；在这里，我们设置了最小和最大的时间，也使这个字段成为必需的：

<form>

<div>

<label for="appt-time">

选择约定的时间（开放时间为 12:00 到 18:00）：

</label>

<input

id="appt-time"

type="time"

name="appt-time"

min="12:00"

max="18:00"

required />

<span class="validity"></span>

</div>

<div>

<input type="submit" value="提交表单" />

</div>

</form>

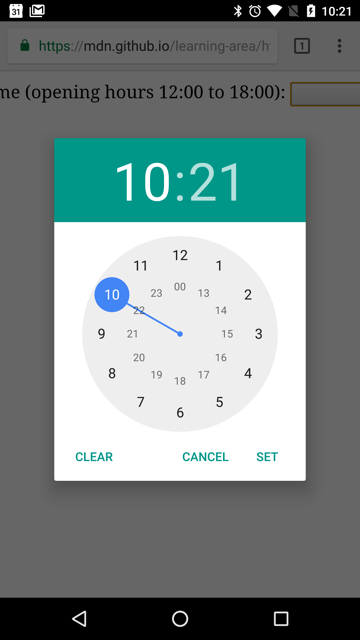
Copy to Clipboard

如果你试图提交含有不完整的时间（或用超出设定范围的时间）的表单，浏览器会显示一个错误。现在试一试下面这个例子。

**警告：** HTML 表单验证并不能替代确保输入数据格式正确的脚本。很容易对 HTML 进行调整，使他们能够绕过验证，或完全删除验证。也有可能会完全绕过 HTML 代码，直接将数据提交给你的服务器。如果你的服务器端代码不能验证它所收到的数据，那么当提交的数据格式不当（或数据过大、类型错误等等）时，灾难就会降临。

## [处理浏览器支持](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/time#%E5%A4%84%E7%90%86%E6%B5%8F%E8%A7%88%E5%99%A8%E6%94%AF%E6%8C%81)

如前所述，老版本的 Safari 和其他一些不太常见的浏览器并不支持时间输入。总的来说，除此之外，支持情况良好——特别是在移动平台上，这些平台往往有非常好的用户界面来指定时间值。例如，安卓版 Chrome 浏览器上的 time 选择器看起来是这样的：



不支持时间输入的浏览器会优雅地降级为文本输入，但这在用户界面的一致性（呈现的控件会有所不同）和数据处理方面都会产生问题。

第二个问题更为严重；如前所述，time 输入的值总是被规范为 hh:mm 或 hh:mm:ss 的格式。另一方面，对于文本输入，默认情况下，浏览器不知道时间应该是什么格式，而且人们有多种写法，如：

* 3.00 pm
* 3:00pm
* 15:00
* 3 o'clock in the afternoon
* 等等

一个办法是在你的 time 输入上添加 [pattern](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#pattern) 属性。即使 time 输入不使用它，text 输入回退也会使用。例如，试着在一个不支持时间输入的浏览器中查看下面的演示：

<form>

<div>

<label for="appt-time">

选择约定的时间（开放时间为 12:00 到 18:00）：

</label>

<input

id="appt-time"

type="time"

name="appt-time"

min="12:00"

max="18:00"

required

pattern="[0-9]{2}:[0-9]{2}" />

<span class="validity"></span>

</div>

<div>

<input type="submit" value="提交表单" />

</div>

</form>

Copy to Clipboard

如果你尝试提交，你会发现，如果你的输入不符合模式 nn:nn（其中 n 是 0 到 9 的数字），不支持的浏览器现在会显示一个错误信息（并突出显示输入无效）。当然，这并不能阻止人们输入无效的时间，或者不正确的遵循时间格式的时间。

然后还有一个问题，就是用户不知道到底应该输入什么格式的时间。

目前，以跨浏览器方式处理时间的最佳方法是让用户在单独的控件中输入小时和分钟（如果需要，也有秒数）（特别是在 [<select>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/select) 元素中，参见下面的示例），或使用 JavaScript 库（例如 [jQuery 日期选择器](https://timepicker.co/)）。

## [示例](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/time#%E7%A4%BA%E4%BE%8B)

在此示例中，我们创建了两组用于选择时间的接口元素：使用 <input type="time"> 创建的原生选择器，以及为不支持 time 输入类型的旧版浏览器准备的两个用于选择小时/分钟的 [<select>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/select) 元素。

HTML 看起来像这样：

<form>

<div class="nativeTimePicker">

<label for="appt-time">

选择约定的时间（开放时间为 12:00 到 18:00）：

</label>

<input

id="appt-time"

type="time"

name="appt-time"

min="12:00"

max="18:00"

required />

<span class="validity"></span>

</div>

<p class="fallbackLabel">

选择约定的时间（开放时间为 12:00 到 18:00）：

</p>

<div class="fallbackTimePicker">

<div>

<span>

<label for="hour">时：</label>

<select id="hour" name="hour"></select>

</span>

<span>

<label for="minute">分：</label>

<select id="minute" name="minute"></select>

</span>

</div>

</div>

</form>

Copy to Clipboard

其 <select> 元素的小时和分钟值是动态生成的。

该代码中可能有趣的另一部分是特性检测代码。要检测浏览器是否支持 <input type="time">，我们创建一个新的 [<input>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input) 元素，尝试将其 type 设置为 time，然后立即检查其 type 值。不支持的浏览器将返回 text，因为 time 类型将回退为 text 类型。如果不支持 <input type="time">，我们将隐藏原生选择器并显示后备选择器 UI（[<select>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/select) 元素）。

// 获取 UI 元素

const nativePicker = document.querySelector('.nativeTimePicker');

const fallbackPicker = document.querySelector('.fallbackTimePicker');

const fallbackLabel = document.querySelector('.fallbackLabel');

const hourSelect = document.querySelector('#hour');

const minuteSelect = document.querySelector('#minute');

// 最初，隐藏回退元素

fallbackPicker.style.display = 'none';

fallbackLabel.style.display = 'none';

// 测试一个新的 time 输入框是否会回退至 text 输入框

const test = document.createElement('input');

try {

test.type = 'time';

} catch (e) {

console.log(e.description);

}

// 如果回退了，运行 if 代码块中的代码

if (test.type === 'text') {

// 隐藏原生选择器，显示回退元素

nativePicker.style.display = 'none';

fallbackPicker.style.display = 'block';

fallbackLabel.style.display = 'block';

// 动态生成小时和分钟数

populateHours();

populateMinutes();

}

function populateHours() {

// 生成含有 6 小时开放时间的 <select>

for (let i = 12; i <= 18; i++) {

const option = document.createElement('option');

option.textContent = i;

hourSelect.appendChild(option);

}

}

function populateMinutes() {

// 对每一小时动态生成 60 个代表分钟数的 <select>

for (let i = 0; i <= 59; i++) {

const option = document.createElement('option');

option.textContent = (i < 10) ? `0${i}` : i;

minuteSelect.appendChild(option);

}

}

// 如果小时数为 18，分钟数必须设置为 00

// 你不可以选择超过 18:00 的时间

function setMinutesToZero() {

if (hourSelect.value === '18') {

minuteSelect.value = '00';

}

}

hourSelect.onchange = setMinutesToZero;

minuteSelect.onchange = setMinutesToZero;

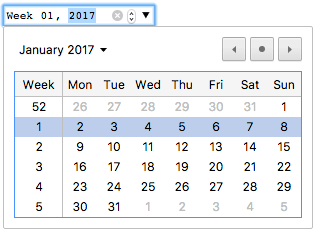
# <input type="week">

[<input>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input) 类型为 **week** 的元素会创建输入字段，以便轻松输入年份以及该年（即第 1 周到第 [52 或 53](https://zh.wikipedia.org/zh-cn/ISO_8601#%E6%97%A5%E5%8E%86%E6%98%9F%E6%9C%9F%E8%A1%A8%E7%A4%BA%E6%B3%95) 周）的 [ISO 8601 星期数](https://zh.wikipedia.org/zh-cn/ISO_8601#%E6%97%A5%E5%8E%86%E6%98%9F%E6%9C%9F%E8%A1%A8%E7%A4%BA%E6%B3%95)

## [尝试一下](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/week#%E5%B0%9D%E8%AF%95%E4%B8%80%E4%B8%8B)

控件的用户界面因浏览器而异；跨浏览器的支持目前受到限制，目前只有 Chrome/Opera 和 Microsoft Edge 支持。在不支持的浏览器中，该控件会优雅降级至与 [<input type="text">](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/text) 相同的功能。

在 Chrome/Opera 中，week 控件提供了用于填写星期和年的插槽，弹出式日历界面（可以更直观地选择它们）以及“X”按钮以清除控件的值。



Edge 的 week 控制更加精细，使用滚动的滚轮打开星期和年的选择器。

|  |  |
| --- | --- |
| [**值**](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/week#%E5%80%BC) | 代表周数和年份的字符串，或为空字符串 |
| **事件** | [change](https://developer.mozilla.org/zh-CN/docs/Web/API/HTMLElement/change_event) 和 [input](https://developer.mozilla.org/zh-CN/docs/Web/API/HTMLElement/input_event) |
| **支持的常用属性** | [autocomplete](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#autocomplete)、[list](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#list)、[readonly](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#readonly) 和 [step](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#step) |
| **IDL 属性** | value、valueAsDate、valueAsNumber 和 list |
| **DOM 接口** | [HTMLInputElement](https://developer.mozilla.org/zh-CN/docs/Web/API/HTMLInputElement) |
| **方法** | [select()](https://developer.mozilla.org/zh-CN/docs/Web/API/HTMLInputElement/select)、[stepDown() (en-US)](https://developer.mozilla.org/en-US/docs/Web/API/HTMLInputElement/stepDown) 和 [stepUp() (en-US)](https://developer.mozilla.org/en-US/docs/Web/API/HTMLInputElement/stepUp) |

## [值](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/week#%E5%80%BC)

一个字符串代表所输入的星期/年的值。[星期字符串](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Date_and_time_formats#%E6%98%9F%E6%9C%9F%E5%AD%97%E7%AC%A6%E4%B8%B2)中描述了此输入类型使用的日期和时间值的格式。

你可以通过在 [value](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#value) 属性中包含一个值来为输入设置默认值，如下所示：

<label for="week">What week would you like to start?</label>

<input id="week" type="week" name="week" value="2017-W01">

Copy to Clipboard

要注意的一件事是，显示的格式可能与实际的 value 不同，后者始终采用 yyyy-Www 格式。当将上述例子中的值提交给服务器时，浏览器可能会将其显示为 Week 01, 2017，但是提交的值始终为 week=2017-W01。

你还可以使用输入元素的 value 属性获取并设置 JavaScript 中的值，例如：

const weekControl = document.querySelector('input[type="week"]');

weekControl.value = '2017-W45';

Copy to Clipboard

## [其他属性](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/week#%E5%85%B6%E4%BB%96%E5%B1%9E%E6%80%A7)

除了 [<input>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input) 元素共有的属性外，星期输入还提供以下属性：

### [max](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/week#max)

接受以上[值](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/week#%E5%80%BC)部分中讨论的字符串格式的按时间最新年份和星期数。如果输入到该元素中的 [value](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#value) 超过此值，则元素将无法通过[约束验证](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Constraint_validation)。如果 max 属性的值不是有效的星期字符串，则该元素没有最大值。

此值必须大于或等于 min 属性指定的年和星期。

### [min](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/week#min)

最早接受的年和星期数。如果输入到该元素中的 [value](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#value) 小于此值，则该元素将无法通过[约束验证](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Constraint_validation)。如果 min 属性的值不是有效的星期字符串，则输入没有最小值。

该值必须小于或等于 max 属性的值。

### [readonly](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/week#readonly)

布尔属性值，如果存在，说明此字段不能由用户编辑。然而，其 value 值仍然可以通过 JavaScript 代码直接设置 [HTMLInputElement](https://developer.mozilla.org/zh-CN/docs/Web/API/HTMLInputElement) 的 value 属性来改变。

**备注：** 由于只读字段不可以拥有值，required 不会对指定了 readonly 属性的字段起作用。

### [step](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/week#step)

step 属性指定了值必须满足的粒度，或者是下文描述的特殊值 any。值必须满足基础的步进值，才有效。如果指定了 [min](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/week#min) 属性，则由 min 属性决定，否则，使用 [value](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#value) 属性的值，如果上述两个值都不存在，则提供适当的默认值。

字符串值 any 意味着不使用步进值，任意值都可以接受（除其他制约因素如 [min](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/week#min) 或 [max](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/week#max) 之外）。

**备注：** 当用户输入的数据不符合步进配置时，[用户代理](https://developer.mozilla.org/zh-CN/docs/Glossary/User_agent)可能会四舍五入到最近的有效值，当有两个同样接近的选项时，更倾向于正方向的数字。

对于 week 输入，step 的值以周为单位，比例因子为 604,800,000（因为基础数值以毫秒为单位）。step 的默认值为 1，表示 1 周。默认的步进基数是 -259,200,000，这是 1970 年第一周的开始（"1970-W01"）。

目前，尚不清楚当与 *week* 输入一起使用时，*"any"* 的值对 *step* 意味着什么。确定该信息后，将对其进行更新。

## [使用星期输入](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/week#%E4%BD%BF%E7%94%A8%E6%98%9F%E6%9C%9F%E8%BE%93%E5%85%A5)

乍看之下，星期输入听起来很方便，因为它们提供了用于选择星期的简单 UI，并且标准化了发送到服务器的数据格式，而与用户的浏览器或区域设置无关。但是，<input type="week"> 存在问题，因为不能保证所有浏览器都支持该特性。

我们将研究 <input type="week"> 的基本和更复杂的用法，然后在以后提供有关缓解浏览器支持问题的建议（请参阅[处理浏览器支持](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/week#%E5%A4%84%E7%90%86%E6%B5%8F%E8%A7%88%E5%99%A8%E6%94%AF%E6%8C%81)）。

### [星期的基本用途](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/week#%E6%98%9F%E6%9C%9F%E7%9A%84%E5%9F%BA%E6%9C%AC%E7%94%A8%E9%80%94)

<input type="week"> 的最简单用法涉及基本的 <input> 和 [<label>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/label) 元素组合，如下所示：

<form>

<label for="week">What week would you like to start?</label>

<input id="week" type="week" name="week" />

</form>

Copy to Clipboard

### [控制输入框大小](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/week#%E6%8E%A7%E5%88%B6%E8%BE%93%E5%85%A5%E6%A1%86%E5%A4%A7%E5%B0%8F)

<input type="week"> 不支持诸如 [size](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#size) 的表单大小属性。您必须依靠 [CSS](https://developer.mozilla.org/zh-CN/docs/Web/CSS) 来确定大小。

### [使用 step 属性](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/week#%E4%BD%BF%E7%94%A8_step_%E5%B1%9E%E6%80%A7)

你应该能够使用 [step](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#step) 属性来更改每次递增或递减的跳转周数，但是这似乎对支持浏览器没有任何影响。

## [验证方式](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/week#%E9%AA%8C%E8%AF%81%E6%96%B9%E5%BC%8F)

默认情况下，<input type="week"> 不会对输入的值应用任何验证。UI 实现通常不允许指定不是有效的星期或年的任何内容，尽管这很有用，但是仍然可以在字段为空的情况下提交，并且你可能希望限制可选择的周数范围。

### [设置最大和最小星期](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/week#%E8%AE%BE%E7%BD%AE%E6%9C%80%E5%A4%A7%E5%92%8C%E6%9C%80%E5%B0%8F%E6%98%9F%E6%9C%9F)

您可以使用 [min](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#min) 和 [max](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#max) 属性来限制用户可以选择的有效周数。在以下示例中，我们设置了可供选择的最小值 Week 01, 2017 和最大值 Week 52, 2017：

<form>

<label for="week">What week would you like to start?</label>

<input id="week" type="week" name="week" min="2017-W01" max="2017-W52" />

<span class="validity"></span>

</form>

Copy to Clipboard

这是上面示例中使用的 CSS。在这里，我们利用 [:valid](https://developer.mozilla.org/zh-CN/docs/Web/CSS/:valid) 和 [:invalid](https://developer.mozilla.org/zh-CN/docs/Web/CSS/:invalid) CSS 伪类属性根据当前值是否有效来设置输入的样式。我们必须将图标放在输入旁边的 [<span>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/span) 而不是输入本身上，因为在 Chrome 中，生成的内容位于表单控件内，无法设置样式或显示有效性。

div {

margin-bottom: 10px;

position: relative;

}

input[type="number"] {

width: 100px;

}

input + span {

padding-right: 30px;

}

input:invalid + span::after {

position: absolute;

content: "✖";

padding-left: 5px;

}

input:valid + span::after {

position: absolute;

content: "✓";

padding-left: 5px;

}

Copy to Clipboard

结果是，在支持的浏览器中，2017 年 W01 到 W52 之间只有几周才被视为有效，并且可以选择。

### [使星期值成为必需值](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/week#%E4%BD%BF%E6%98%9F%E6%9C%9F%E5%80%BC%E6%88%90%E4%B8%BA%E5%BF%85%E9%9C%80%E5%80%BC)

另外，你可以使用 [required](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input#required) 属性来强制填写星期。因此，如果您尝试提交空白的星期字段，则支持的浏览器将显示错误。

让我们看一个例子；在这里，我们设置了最小和最大星期，并令该字段必填：

<form>

<div>

<label for="week">What week would you like to start?</label>

<input

id="week"

type="week"

name="week"

min="2017-W01"

max="2017-W52"

required />

<span class="validity"></span>

</div>

<div>

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

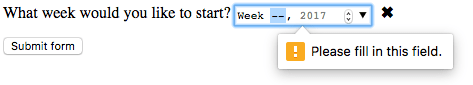
</div>

</form>

Copy to Clipboard

如果您尝试提交不带任何值的表单，浏览器将显示错误。现在尝试使用示例：

这是使用不支持该控件的浏览器的屏幕截图：

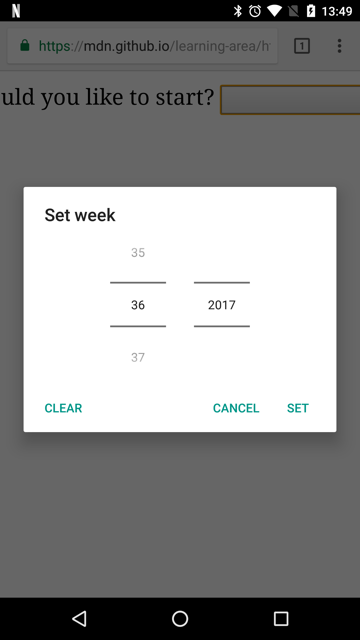


**警告：** HTML 表单验证不能代替脚本来确保输入的数据采用正确的格式。对于某人来说，对 HTML 进行调整以使其绕过验证或完全删除验证太容易了，甚至也可以完全绕开 HTML 并将数据直接提交到服务器。如果服务器端代码无法验证其接收到的数据，则在提交格式不正确的（或太大，类型错误……）的数据时，灾难可能会发生。

## [处理浏览器支持](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/week#%E5%A4%84%E7%90%86%E6%B5%8F%E8%A7%88%E5%99%A8%E6%94%AF%E6%8C%81)

如上所述，现在使用星期输入的主要问题是浏览器支持：Safari 和 Firefox 在桌面上不支持它，而旧版本的 IE 不支持它。

诸如 Android 和 iOS 之类的移动平台确实很好地利用了这种输入类型，提供了专门的 UI 控件，使在触摸屏环境中选择值变得非常容易。例如，Android 版 Chrome 上的 week 选择器如下所示：



不支持该控件的浏览器会优雅地降级为文本输入，但这在用户界面的一致性（所提供的控件将有所不同）和数据处理方面都产生了问题。

第二个问题是更严重的。如前所述，在输入一个 week 的情况下，实际值始终会归一化为 yyyy-Www 格式。当浏览器回退至通用文本输入时，没有什么可以指导用户正确格式化输入的格式（这肯定是不直观的）。人们可以通过多种方式来写星期值。例如：

* Week 1 2017
* Jan 2-8 2017
* 2017-W01
* 等等。

目前，以跨浏览器方式处理星期/年的最佳方法是让用户在单独的控件中输入星期数和年份（特别是在 [<select>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/select) 元素中，参见下面的示例），或使用 JavaScript 库（例如 [jQuery 日期选择器](https://jqueryui.com/datepicker/)）。

## [示例](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input/week#%E7%A4%BA%E4%BE%8B)

在此示例中，我们创建了两组用于选择星期的 UI 元素：使用 <input type="week"> 创建的原生选择器，以及为不支持 week 输入类型的旧版浏览器准备的两个用于选择星期/年的 [<select>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/select) 元素。

HTML 看起来像这样：

<form>

<div class="nativeWeekPicker">

<label for="week">What week would you like to start?</label>

<input

id="week"

type="week"

name="week"

min="2017-W01"

max="2018-W52"

required />

<span class="validity"></span>

</div>

<p class="fallbackLabel">What week would you like to start?</p>

<div class="fallbackWeekPicker">

<div>

<span>

<label for="week">Week:</label>

<select id="fallbackWeek" name="week"></select>

</span>

<span>

<label for="year">Year:</label>

<select id="year" name="year">

<option value="2017" selected>2017</option>

<option value="2018">2018</option>

</select>

</span>

</div>

</div>

</form>

Copy to Clipboard

星期值由下面的 JavaScript 代码动态生成。

该代码中可能有趣的另一部分是特性检测代码。要检测浏览器是否支持 <input type="week">，我们创建一个新的 [<input>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input) 元素，尝试将其 type 设置为 week，然后立即检查其 type 值。不支持的浏览器将返回 text，因为 week 类型将回退为 text 类型。如果不支持，<input type="week"> 我们将隐藏原生选择器并显示后备选择器 UI（[<select>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/select) 元素）。

// 获取 UI 元素

const nativePicker = document.querySelector('.nativeWeekPicker');

const fallbackPicker = document.querySelector('.fallbackWeekPicker');

const fallbackLabel = document.querySelector('.fallbackLabel');

const yearSelect = document.querySelector('#year');

const weekSelect = document.querySelector('#fallbackWeek');

// 最初，隐藏回退元素

fallbackPicker.style.display = 'none';

fallbackLabel.style.display = 'none';

// 测试一个新的 date 输入框是否会回退至 text 输入框

const test = document.createElement('input');

try {

test.type = 'week';

} catch (e) {

console.log(e.description);

}

// 如果回退了，运行 if 代码块中的代码

if ( test.type === 'text') {

// 隐藏原生选择器，显示回退元素

nativePicker.style.display = 'none';

fallbackPicker.style.display = 'block';

fallbackLabel.style.display = 'block';

// 动态生成星期数

populateWeeks();

}

function populateWeeks() {

// 生成含有 52 个选择的星期数

for (let i = 1; i <= 52; i++) {

const option = document.createElement('option');

option.textContent = (i < 10) ? `0${i}` : i;

weekSelect.appendChild(option);

}

}

**备注：** 请记住，有些年份有 53 周（请参阅[每年的周数](https://zh.wikipedia.org/wiki/ISO%E9%80%B1%E6%97%A5%E6%9B%86#%E6%AF%8F%E5%B9%B4%E7%9A%84%E6%98%9F%E6%9C%9F))！在开发生产应用程序时，需要考虑到这一点。

# 表单的特殊属性

### 1）readonly ：只读，这个只能作用在文本框

### 2）disabled ：禁用控件，使得控件不可用，可以作用在任何控件，被禁用的控件的值不会提交给服务器