

此页面由社区从英文翻译而来。了解更多并加入 MDN Web Docs 社区。

## HTML5 的输入 (input) 类型

在[前一篇文章](#)中我们学习了 `<input>` 元素，涵盖了自 HTML 早期以来可用的 `type` 属性的原始值。现在我们来具体看一看较新的表单控件，包含一些新添加至 HTML 5 的 `input` 类型，以允许收集更多特定类型的数据。

前提:	计算机基础知识和对于 <a href="#">HTML 的基本理解</a> 。
目标:	了解可用于创建本地表单控件的较新的 <code>input</code> 类型值，以及如何使用 HTML 实现它们。

**备注：** 本篇文章中讨论的大多数特性都受到了广泛支持，如果有任何例外将会在文章中说明，如果你需要更多浏览器支持的细节，你应该查看我们的 [HTML 表单元素参考](#)，特别是深入的 [<input> 类型](#) 参考。

由于 HTML 表单控件的外观可能与设计者的规格有很大的不同，web 开发者有时会建立自己的自定义表单控件。我们在一个高级教程中介绍了这一点：[如何构建自定义表单控件](#)。

## E-mail 地址字段

将 `type` 属性设置为 `email` 就可以使用这种控件：

HTML

```
<input type="email" id="email" name="email" />
```

当使用了这种 [type](#) 类型时，用户需要输入一个合法的电子邮件地址，任何其他输入都会使得浏览器在表单提交时显示错误信息。你可以在下面的截图中看到这个行为：



Your comment:  
I am a comment

Your email:  
k

Please enter an email address.

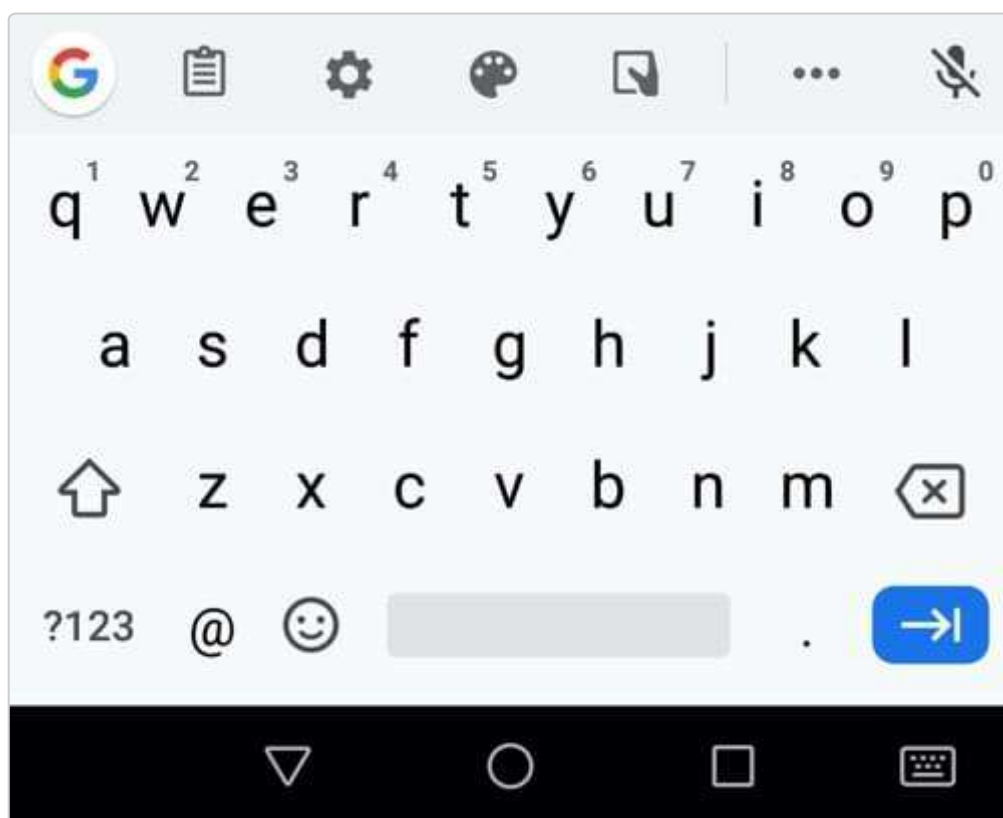
Send now

你也可以搭配使用 [multiple\\_\(en-US\)](#) 属性，以允许在同一个 email 输入框中输入多个电子邮件地址，以英文逗号分隔：

HTML

```
<input type="email" id="email" name="email" multiple />
```

在某些设备上——特别是像智能手机这样具有动态键盘的触摸设备——可能会出现一个包括 @ 键的更适合于输入电子邮件地址的虚拟键盘。请看下面的 Firefox Android 版键盘截图，以示说明：



**备注：** 你可以在[基本 input 例子](#) 中找到基本文本 input 类型的例子（也请看看[源代码](#)）。

这是使用这些较新的 input 类型的另一个很好的理由——可以为这些设备的用户改善用户体验。

## 客户端验证

正如你在上面看到的，email 与其他较新的 input 类型一样，提供了内置的客户端错误验证，在数据被发送到服务器之前由浏览器执行。它是引导用户准确填写表格的一个有用的辅助工具，可以节省时间：可以立即知道你的数据是否正确，而不需要等待服务器返回结果。

但它不应该被认为是一种详尽的安全措施！你的应用程序始终应该在服务器端和客户端对任何表单提交的数据进行安全检查，因为客户端验证太容易被关闭了，所以恶意用户仍然可以很容易地将坏数据发送到你的服务器。请参阅[网站安全](#)来了解可能会发生什么。实现服务端认证超出了本章的范围，但你应该熟稔于心。

注意，在默认限制条件下，a@b 也是一个合法的电子邮件地址，因为 email input 类型默认也允许内部网络的电子邮件地址。为了实现不同的验证行为，你可以使用 [pattern](#) 属性，而且可以自定义错误信息；我们将在[客户端表单认证](#)文章中进一步说明如何使用这些功能。

**备注：** 如果输入的数据不是一个电子邮件地址，会匹配 `:invalid` 伪类，且 `validityState.typeMismatch_(en-US)` 属性总会返回 `true`。

## 查询字段

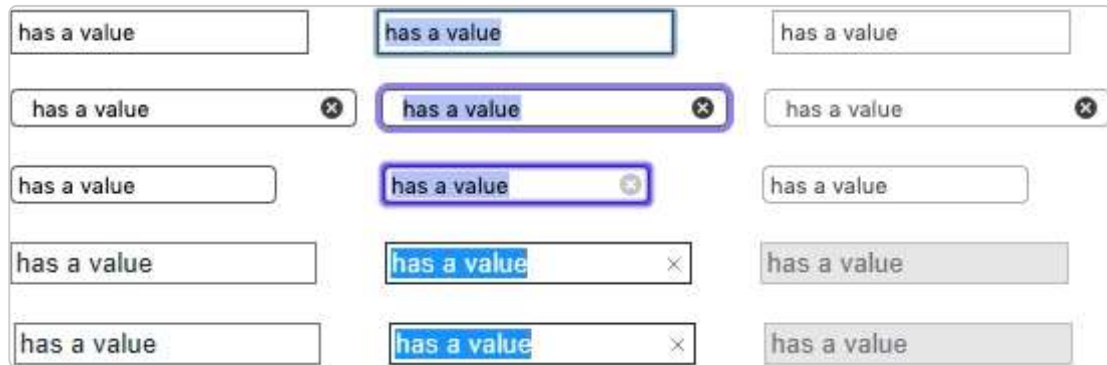
查询字段（Search fields）旨在用于在页面和应用程序上创建搜索框。将 [type](#) 属性设置为 `search` 就可以使用这种控件：

HTML

```
<input type="search" id="search" name="search" />
```

text 字段和 search 字段的主要区别是浏览器赋予它们的外观显示。通常情况下，search 字段拥有圆角边框，有时会显示“ⓧ”标志，当点击它时会清除输入框。另外，在动态键盘设备上，键盘的回车键会显示“search”，或显示为放大镜图标。

下面的截图显示了 macOS 上的 Firefox 71、Safari 13 和 Chrome 79，以及 Windows 10 上的 Edge 18 和 Chrome 79 的非空查询字段，请注意，清除图标仅当在输入框中存在值得时候才会显示，且与 Safari 不同的是，仅当聚焦状态时才会显示此图标。



另外一个值得提示的特性是，`search` 字段的值可以自动地保存下来，在同一网站的自动完成框中复用输入，这样的特性倾向于在大多数现代浏览器中自动进行。

## 电话号码字段

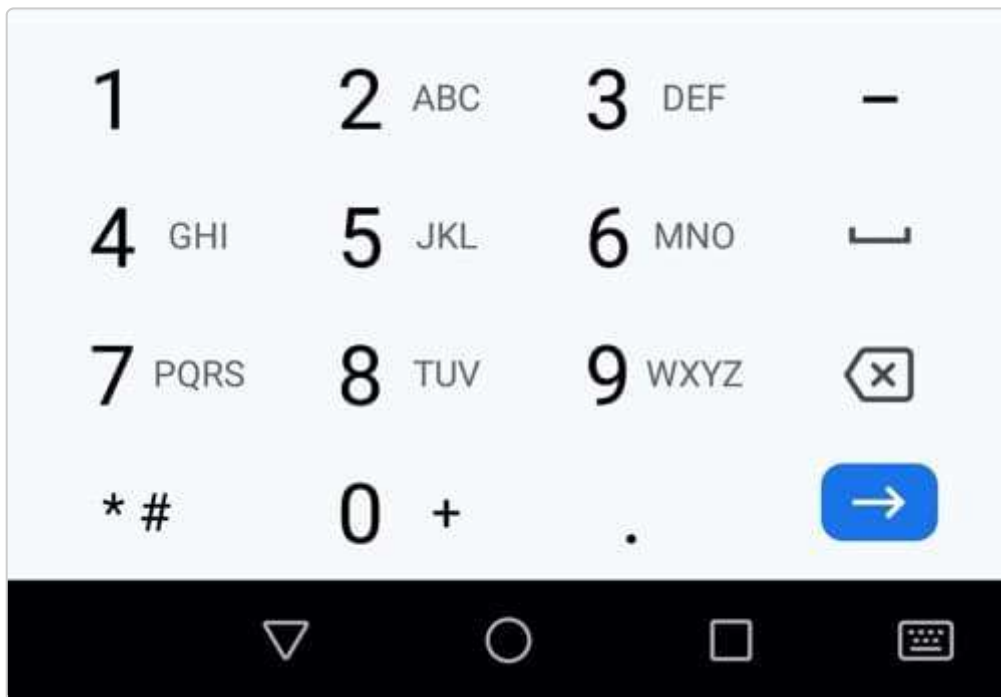
在 `type` 属性中使用 `tel` 值，即可创建一个专门用于输入电话号码的文本域：

HTML

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

当使用带有动态键盘的移动设备访问带有 `type="tel"` 的表单时，大多数设备会显示数字键盘。这意味着只要数字键盘有用，这种类型就很有用，而且不只是用于电话号码。

下方截图为在 Android 系统的 Firefox 浏览器上访问该类型表单的键盘：



由于世界各地的电话号码格式多种多样，这种类型的字段对用户输入的值没有任何限制（这意味着它可能包括字母等非数字值）。

像之前提及的那样，[pattern](#) 属性也可以应用在这里来附加一些限制，你会在[客户端表单验证](#)文章中学到更多内容。

## URL 字段

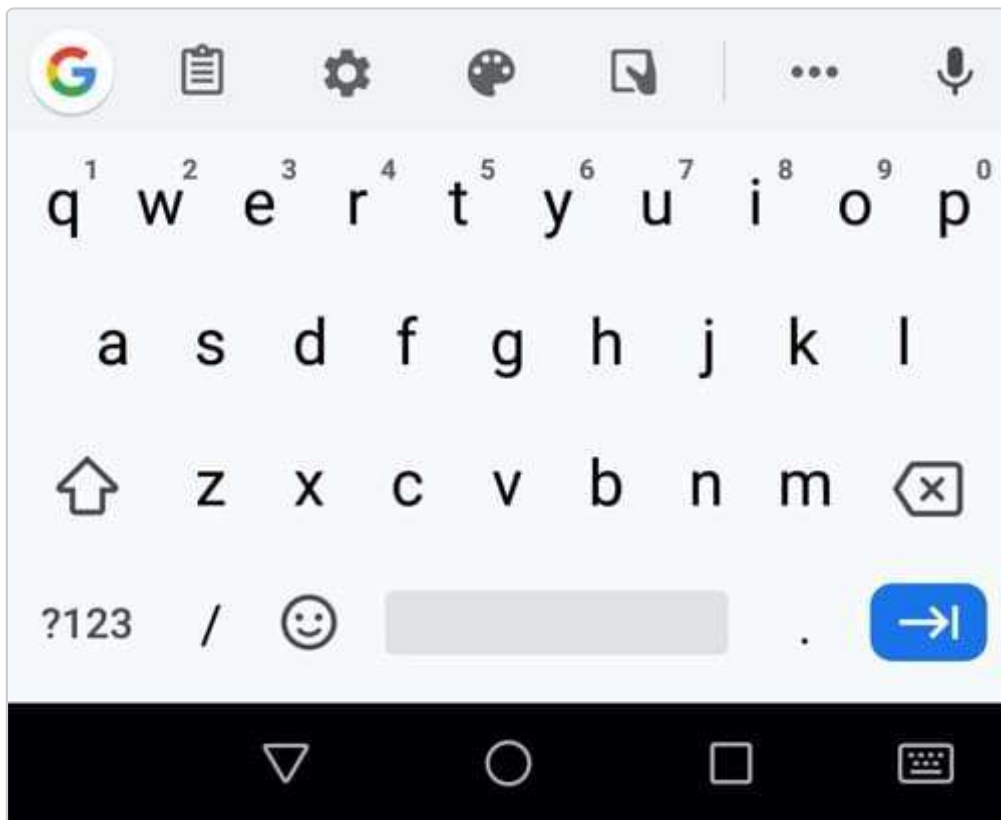
在 [type](#) 属性值中使用 `url`，即可创建一个用于输入网址的文本字段：

HTML

```
<input type="url" id="url" name="url" />
```

它为字段添加了特殊的验证约束。浏览器会在没有协议（例如 `http:`）输入或网址格式不对的情况下报告错误。在具有动态键盘的设备上，键盘通常会显示部分或全部冒号、句号和正斜杠作为默认键。

看看下面这个例子（取自于 Android 上的 Firefox 浏览器）：

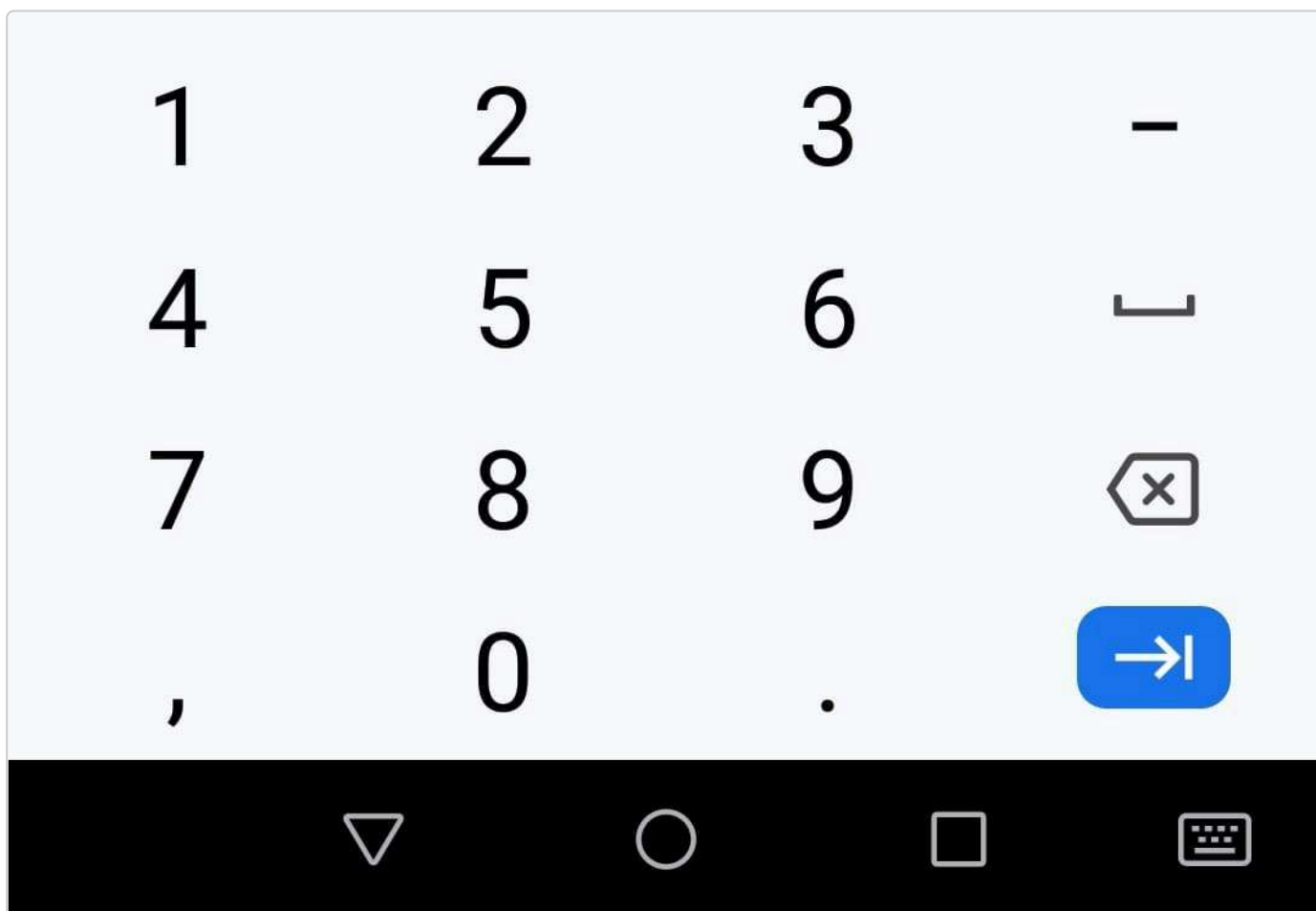


**备注：** URL 格式正确可不意味着所指向的地址一定实际存在！

## 数字字段

用于输入数字的控件可以由 `type` 为 `number` 的 `<input>` 元素创建。这个控件外观与文本域类似，但只允许浮点数输入，并通常以旋转器（spinner）的形式提供按钮来增加和减少控件的值。在有动态键盘的设备上，一般会显示数字键盘。

看看下面这个例子（取自于 Android 上的 Firefox 浏览器）：



使用 `number input` 类型，你可以使用 [min](#) 和 [max](#) 属性控制允许输入的最小值和最大值。

你也可以使用 `step` 属性来设定每次按下 `spinner` 按钮增加或减少的值。默认情况下，`number input` 类型只允许整数值输入，为了允许浮点数输入，要指定 [step="any" \(en-US\)](#)。如果省略了此值，`step` 会默认为 `1`，意味着只有自然数是有效的输入。

我们来看一些例子。第一个例子创建了可从 `1` 到 `10` 之间选择值的数字选择器，且单击一次按钮所增长或减少的值为 `2`。

HTML

```
<input type="number" name="age" id="age" min="1" max="10" step="2" />
```

第二个例子创建了可从 `0` 到 `1` 之间选择值得数字选择器，且单击一次按钮所增长或减少的值为 `0.01`。

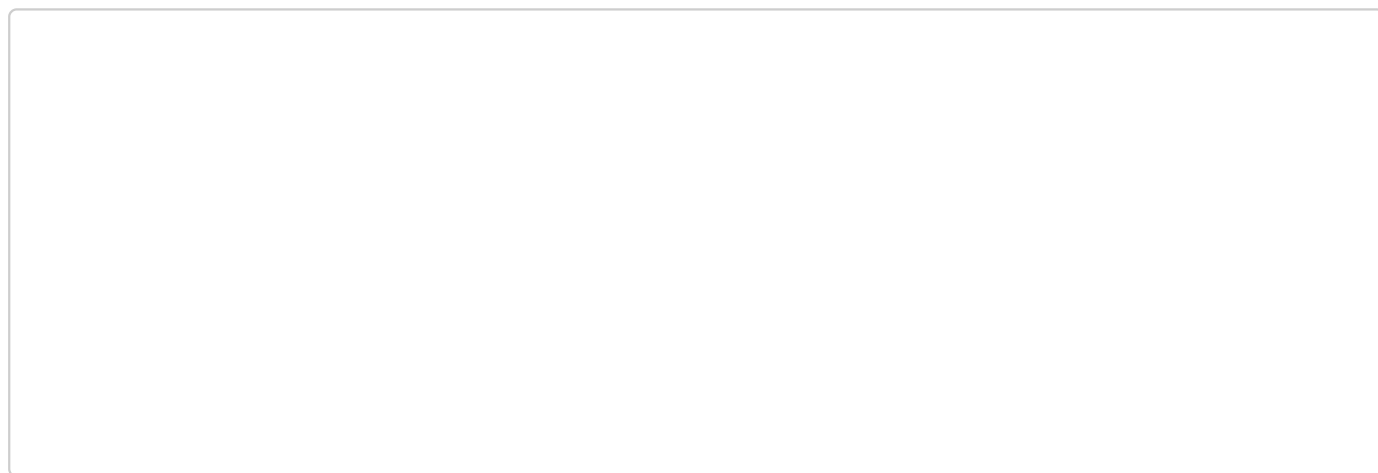
HTML

```
<input type="number" name="change" id="pennies" min="0" max="1" step="0.01" />
```

当有效值的范围有限时，`number` 输入类型才有意义，例如一个人的年龄或身高。如果范围太大，渐进式增加没有意义（如范围为 00001 到 99999 的美国 ZIP 码）的话，使用 `tel` 类型可能会更好；它提供了数字键盘，而放弃了数字输入器的 spinner UI 功能。

## 滑块控件

另外一种选择数字的方式是使用**滑块（slider）**。你在买房网站等网站上经常看到这种情况，你想设置一个最高的房产价格来进行过滤。让我们看一个现场示例来说明这一点：



从使用上来说，滑块的准确性不如文本字段。因此，它们被用来挑选**精确值**不一定那么重要的数字。

在 `<input>` 元素中使用 `range` 作为属性 `type` 的值，就可以创建一个滑块，滑块可以通过鼠标、触摸，或用键盘的方向键移动。

正确配置滑块组件非常重要。推荐分别配置 [min\\_\(en-US\)](#)、[max\\_\(en-US\)](#) 和 [step\\_\(en-US\)](#) 属性来设置滑块的最小值、最大值和增量值。

我们来看看示例背后的代码，这样你就可以看到它是如何完成的。首先是基本的 HTML 代码：

### HTML

```
<label for="price">Choose a maximum house price: </label>
<input
  type="range"
  name="price"
  id="price"
  min="50000"
  max="500000"
```



```
step="100"
value="250000" />
<output class="price-output" for="price"></output>
```

本示例创建了一个其取值为 50000 到 500000 之间的滑块，每次的增量值是 100。我们使用 `value` 属性设定了此滑块的默认值为 250000。

使用滑块的一个问题是，它们不提供任何种类的视觉反馈来说明当前的值是什么。这是我们附加了一个包含当前值输出的 `<output>` 元素的原因。你可以在任何元素内显示一个输入值或一个计算的输出值，但是 `<output>` 是特殊的，就像 `<label>` 那样，它可以指定 `for` 属性，允许你将其与输出值来自的一个或多个元素联系起来。

要真正显示当前值，并在其变化时更新，你必须使用 JavaScript，这相对容易做到：

JS

```
const price = document.querySelector("#price");
const output = document.querySelector(".price-output");

output.textContent = price.value;

price.addEventListener("input", () => {
  output.textContent = price.value;
});
```

这里我们将 `range` 输入元素和 `output` 元素存为了两个变量。然后我们马上将 `output` 的 `textContent` 属性设置为 `input` 的 `value`。最终，我们设置了一个事件监听器，确保每次范围滑块移动时，`output` 的 `textContent` 总是可以及时更新为新值。

**备注：** 在 CSS Tricks 网站上有一篇关于该主题的很好的教程：[The Output Element](#)。

## 日期和时间选择器

收集日期和时间值历来是 web 开发者的噩梦。为了获得良好的用户体验，提供一个日历选择用户界面是很重要的，使用户能够选择日期，而不需要切换到本地日历应用程序的上下文，或者可能以难以解析的不同格式输入。上个千年（1000~1999 年）的最后一分钟可以用以下不同的方式表示，例如：1999/12/31, 23:59 或 12/31/99T11:59PM。

HTML 日期控件可用于处理这种特定的数据，提供日历控件并使数据统一。

日期和时间控件可由 [<input>](#) 元素和一个合适的 [type](#) 属性值来创建，该值取决于要收集的类型（日期、时间、还是以上全部）。这里有一个示例，在浏览器不支持的情况下会自动回退为 [<select>](#) 元素：

让我们简单地看看不同的可用类型。请注意，这些类型的用法相当复杂，特别是考虑到浏览器的支持（见下文）；要想了解全部细节，请跟随下面的链接进入到每种类型的参考页面，包括详细的例子。

## datetime-local

[<input type="datetime-local">](#) 创建了显示和选择一个没有特定时区信息的日期和时间的控件。

HTML

```
<input type="datetime-local" name="datetime" id="datetime" />
```

## month

[<input type="month">](#) 创建了显示和选择带有年份信息的某个月的控件。

HTML

```
<input type="month" name="month" id="month" />
```

## time

[<input type="time">](#) 创建了显示和选择时间的控件。时间可能会以 12 小时制显示，但一定会以 24 小时制形式返回。

```
<input type="time" name="time" id="time" />
```

## week

[<input type="week">](#) 创建了显示和选择一年中特定编号周的控件。

一周以周一开始，一直运行到周日结束。另外，每年的第一周总会包含那一年首个星期四，其中可能不包括当年的第一天，也可能包括前一年的最后几天。

```
<input type="week" name="week" id="week" />
```

## 限制日期/时间值

所有的日期和时间控件总可以由 [min\\_\(en-US\)](#) 和 [max\\_\(en-US\)](#) 属性控制，可由 [step\\_\(en-US\)](#) 属性进一步做控制，具体值随着 input 类型的不同而产生变化。

```
<label for="myDate">When are you available this summer?</label>
<input
  type="date"
  name="myDate"
  min="2013-06-01"
  max="2013-08-31"
  step="7"
  id="myDate" />
```


## 颜色选择器控件

颜色总是有点难处理。有许多方法来表达它们，如 RGB 值（十进制或十六进制）、HSL 值、关键词等。

用于输入颜色的控件可以由 [type](#) 为 color 的 [<input>](#) 元素创建：

```
<input type="color" name="color" id="color" />
```

在支持的情况下，点击一个颜色控件将倾向于显示操作系统的默认颜色选择功能，以便你真正做出选择。以下是在 macOS 上 Firefox 浏览器的屏幕截图，提供了一个例子：

 Firefox for Android email keyboard, with ampersand displayed by default.

这里有一个在线示例供你尝试：



返回值总是颜色的小写的 6 位十六进制数表示。

## 技能测试！

你已经看完了这篇文章的主要内容，但你还记得所有重要的信息吗？在继续学习其他内容之前，你可以在这里找到进一步的测试来确定你是否掌握了这些知识点——请参见[技能测试：HTML5 控件\(en-US\)](#)。

## 总结

至此，我们结束了对 HTML5 表单输入类型的考察。还有一些其他的控件类型，由于其行为非常特殊，所以不能轻易地归为一类，但仍有必要了解一下。我们将在下一篇文章中介绍这些类型。

## 进阶内容

- [如何构建表单控件](#)
- [使用 JavaScript 发送表单](#)
- [表单控件兼容性列表](#)

## Help improve MDN

Was this page helpful to you?

[Learn how to contribute.](#)

This page was last modified on 2023年11月30日 by [MDN contributors](#).

