

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。

创建超链接

超链接非常重要——它们使互联网成为一个[互联的网络](#)。本文介绍了创建链接所需的语法，并且讨论了链接的最佳实践。

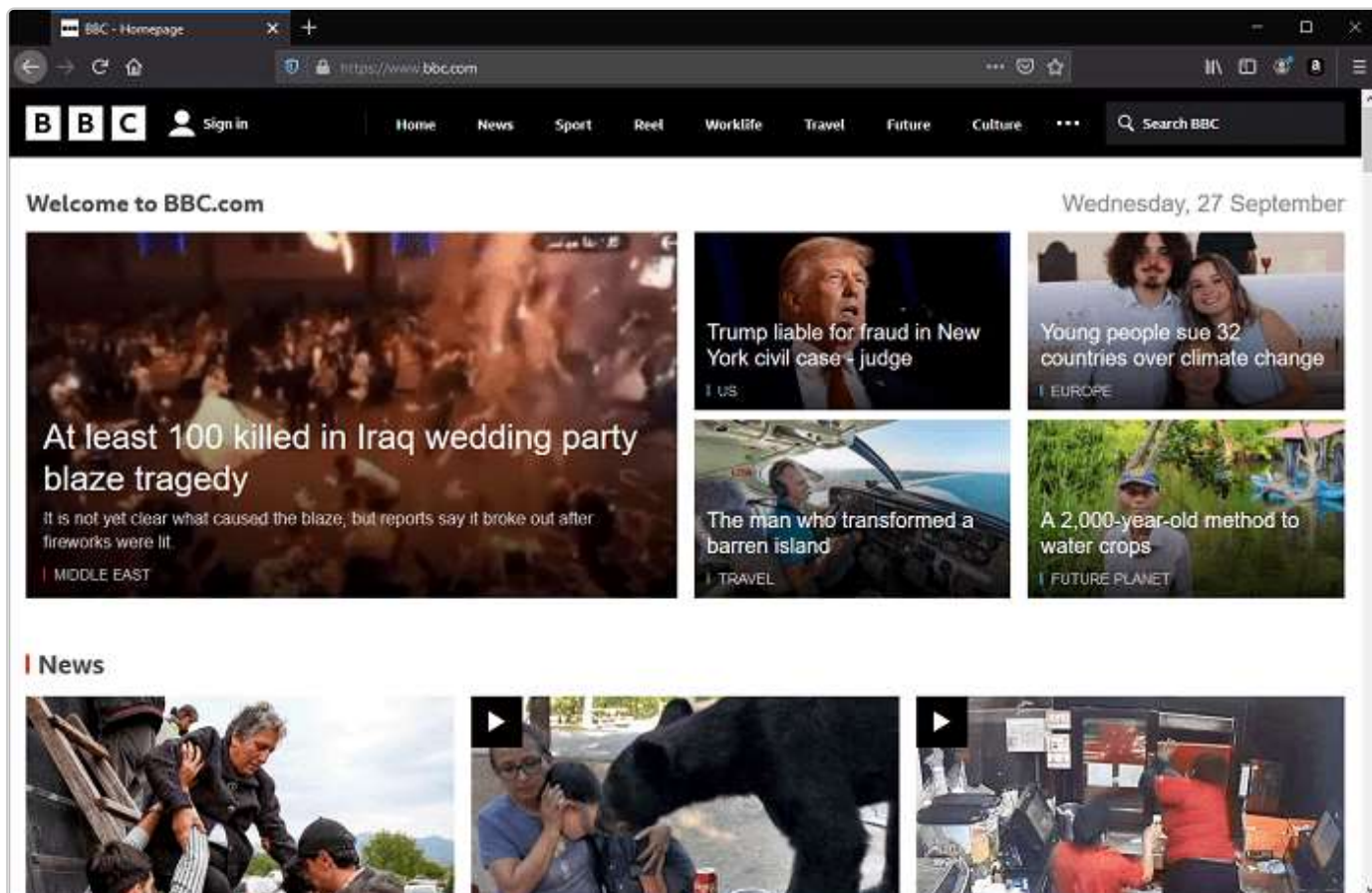
前提:	熟悉基本 HTML（包含在 开始学习 HTML 中），HTML 文本格式（包含在 HTML 文字处理基础 中）。
目标:	学习如何有效地实现一个把多个文件链接在一起的超文本链接。

什么是超链接？

超链接是互联网提供的最令人兴奋的创新之一，它们从一开始就一直是互联网的一个特性，使互联网成为[互联的网络](#)。超链接使我们能够将我们的文档链接到任何其他文档（或其他资源），也可以链接到文档的指定部分，我们可以在一个简单的网址上提供应用程序。几乎任何网络内容都可以转换为链接，点击（或激活）超链接将使网络浏览器转到另一个网址（[URL](#)）。

备注： URL 可以指向 HTML 文件、文本文件、图像、文本文档、视频和音频文件以及可以在网络上保存的任何其他内容。如果浏览器不知道如何显示或处理文件，它会询问你是否要打开文件（需要选择合适的本地应用来打开或处理文件）或下载文件（以后处理它）。

以 BBC 的主页为例，里面就包含了非常多的链接，各自连到不同新闻、网站的其他地方（导航功能），或者登入/注册页面（用户工具）等等。



链接的解析

通过将文本或其他内容包裹在 `<a>` 元素内，并给它一个包含网址的 `href` 属性（也称为**超文本引用或目标**，它将包含一个网址）来创建一个基本链接。

HTML

```
<p>
```

我创建了一个指向

```
<a href="https://www.mozilla.org/zh-CN/">Mozilla 主页</a>的连接。
```

```
</p>
```

结果如下所示：

我创建了一个指向 [Mozilla 主页](https://www.mozilla.org/zh-CN/) 的连接。

块级链接

就像之前提到的那样，任何内容，甚至[块级内容](#)都可以作为链接出现。如果想让标题元素变为链接，就把它包裹在锚点元素（`<a>`）内，像这个代码段一样：

```
<a href="https://developer.mozilla.org/zh-CN/">
  <h1>MDN Web 文档</h1>
</a>
<p>自从 2005 年起，就开始记载包括 CSS、HTML、JavaScript 等网络技术。</p>
```

它将标题转化为了链接：

Play

MDN Web 文档

自从 2005 年起，就开始记载包括 CSS、HTML、JavaScript 等网络技术。

图片链接

如果有需要作为链接的图片，使用 `<a>` 元素来包裹要引用图片的 `` 元素。

```
<a href="https://developer.mozilla.org/zh-CN/">
  
</a>
```

它将 MDN 徽标转化为了链接：

Play



备注： 你会在以后的文章中了解到更多如何在 Web 中使用图片的信息。

使用 title 属性添加支持信息

你可能要添加到你的链接的另一个属性是 `title`。这旨在包含关于链接的补充信息，例如页面包含什么样的信息或需要注意的事情。

HTML

Play

```
<p>
  我创建了一个指向<a
    href="https://www.mozilla.org/zh-CN/"
    title="了解 Mozilla 使命以及如何参与贡献的最佳站点。"
  >Mozilla 主页</a
  >的超链接。
</p>
```

结果如下（当鼠标指针悬停在链接上时，标题将作为提示信息出现）：

Play

备注： 链接的标题仅当鼠标悬停在其上时才会显示，这意味着使用键盘来导航网页的人很难获取到标题信息。如果标题信息对于页面非常重要，你应该使用所有用户都能方便获取的方式来呈现，例如放在常规文本中。

主动学习：创建你自己的示例链接

使用本地代码编辑器和我们的[入门模板](#) 来创建一个 HTML 文档。

- 在 HTML 内，尝试添加一个或者多个段落或其他你知道类型的内容。

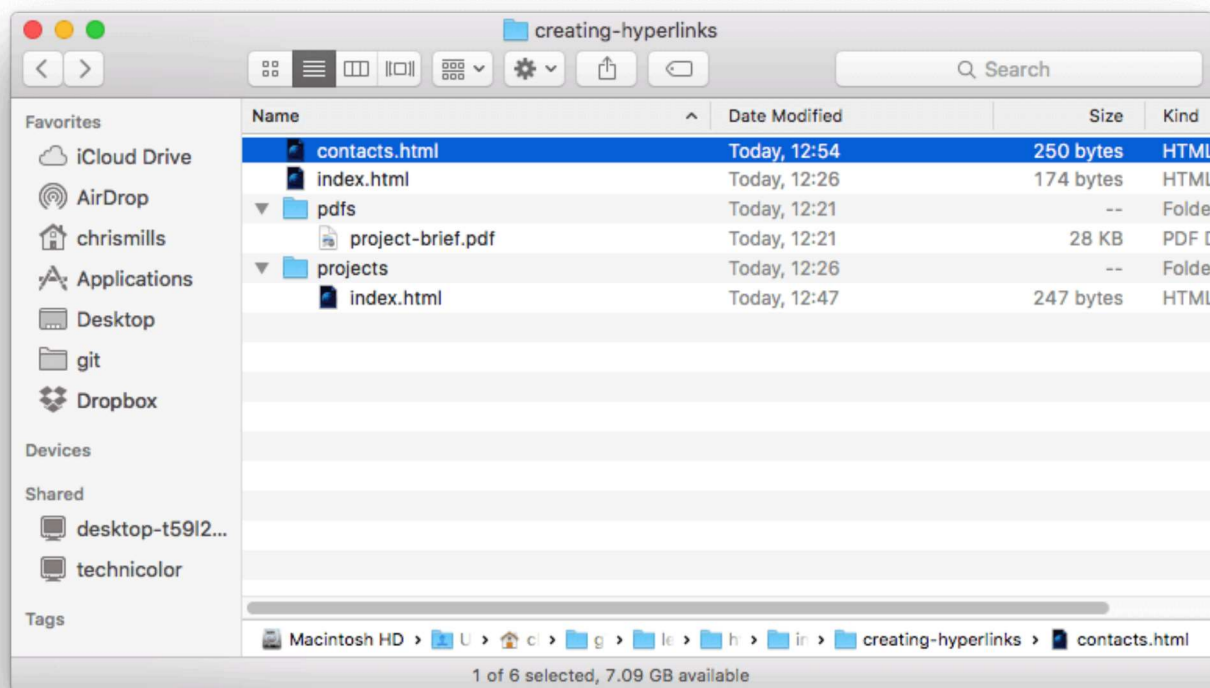
- 将某些内容转换为链接。
- 包含 title 属性。

统一资源定位符（URL）与路径（path）快速入门

要全面地了解链接目标，你需要了解统一资源定位符和文件路径。本小节将介绍相关的信息。

统一资源定位符（Uniform Resource Locator，URL）是一个定义了在网络上的位置的一个文本字符串。例如，Mozilla 的简体中文主页位于 <https://www.mozilla.org/zh-CN/>。

URL 使用路径查找文件。路径指定文件系统中你感兴趣的文件所在的位置。看一下一个简单的目录结构的例子（参见[创建超链接](#) 目录）。



此目录结构的**根目录**称为 `creating-hyperlinks`。当在本地处理网站时，你会有一个包含整个网站的目录。在**根目录**，我们有 `index.html` 和 `contacts.html` 文件。在真实的网站上，`index.html` 将会成为我们的主页或着陆页（作为网站或网站特定部分的入口的网页）。

我们的根目录还有两个目录——`pdfs` 和 `projects`，它们分别包含一个 PDF（`project-brief.pdf`）文件和一个 `index.html` 文件。请注意你可以在同一项目中有两个 `index.html` 文件，前提是他们在不同的文件系统目录下。第二个 `index.html` 或许是项目相关信息的着陆页。

- **指向当前目录：**如果你想在 `index.html`（顶层 `index.html`）中包含一个指向 `contacts.html` 的超链接，你只需要指定想要链接到的文件名。因为它与当前文件是在同一个目录的，所以你应该使用的 URL 是 `contacts.html`：

HTML

```
<p>
    要联系某位工作人员，请访问我们的<a href="contacts.html">联系人页面</a>。
</p>
```

- **指向子目录：**如果你想在 `index.html`（顶层 `index.html`）中包含一个指向 `projects/index.html` 的超链接，你需要先进入 `projects` 项目目录，然后通过指定目录的名称，然后是正斜杠，然后是文件的名称指明要链接到的文件 `index.html`。因此你要使用的 URL 是 `projects/index.html`：

HTML

```
<p>请访问我的<a href="projects/index.html">项目主页</a>。</p>
```

- **指向上级目录：**如果你想在 `projects/index.html` 中包含一个指向 `pdfs/project-brief.pdf` 的超链接，你必须先返回上级目录，然后再回到 `pdfs` 目录。“返回上一个目录级”使用两个英文点号表示（`..`），因此你应该使用的 URL 是 `../pdfs/project-brief.pdf`：

HTML

```
<p>点击打开<a href="../pdfs/project-brief.pdf">项目简介</a>。</p>
```

备注：如果需要，你可以将这些功能的多个实例组合成复杂的 URL。例如 `../../../../complex/path/to/my/file.html`。

文档片段

超链接除了可以链接到文档外，也可以链接到 HTML 文档的特定部分（被称为**文档片段**）。要做到这一点，你必须首先给要链接到的元素分配一个 `id` 属性。通常情况下，链接到一个特定的标题是有意义的，这看起来就像下面这样：

HTML

```
<h2 id="Mailing_address">邮寄地址</h2>
```

为了链接到那个特定的 `id`，要把它放在 URL 的末尾，并在前面包含井号（`#`），例如：

HTML

```
<p>
    要提供意见和建议，请将信件邮寄至<a href="contacts.html#Mailing_address"
    >我们的地址</a
    >。
</p>
```

你甚至可以在同一份文档下，通过链接文档片段，来链接到*当前文档的另一部分*：

HTML

```
<p>本页面底部可以找到<a href="#Mailing_address">公司邮寄地址</a>。</p>
```

绝对 URL 和相对 URL

你可能会在网络上遇到两个术语，**绝对 URL** 和**相对 URL**（或者称为，**绝对链接**和**相对链接**）：

绝对 URL：指向由其在 Web 上的绝对位置定义的位置，包括[协议](#)和[域名](#)。像下面的例子，如果 index.html 页面上传到了 projects 这一个目录。并且 projects 目录位于 web 服务站的根目录，web 站点的域名为 https://www.example.com，那么这个页面就可以通过 https://www.example.com/projects/index.html 访问（或者通过 https://www.example.com/projects/ 来访问，因为在没有指定特定的 URL 的情况下，大多数 web 服务器会默认访问加载 index.html 这类页面）

不管绝对 URL 在哪里使用，它总是指向确定的相同位置。

相对 URL：指向与你链接的文件相关的位置，更像我们在前面一节中所看到的位置。例如，如果我们想从示例文件链接 https://www.example.com/projects/index.html 转到相同目录下的一个 PDF 文件，URL 就是文件名（例如 project-brief.pdf），没有其他的信息要求。如果 PDF 文件能够在 projects 的子目录 pdfs 中访问到，相对路径就是 pdfs/project-brief.pdf（对应的绝对 URL 是 https://www.example.com/projects/pdfs/project-brief.pdf）

一个相对的 URL 将指向不同的位置，这取决于引用的文件的实际位置——例如，如果我们把 index.html 文件从 projects 目录移动到 Web 站点的根目录（最高级别，而不是任何目录中），里面的 pdfs/project-brief.pdf 相对 URL 将会指向 https://www.example.com/pdfs/project-brief.pdf，而不是 https://www.example.com/projects/pdfs/project-brief.pdf

当然，project-brief.pdf 文件和 pdfs 文件夹的位置不会因为你移动了 index.html 文件而突然发生变化——这将使你的链接指向错误的位置，因此如果单击它，它将无法工作。你得小心点！

链接最佳实践

下面是一些在编写链接时可以遵循的最佳实践。

使用清晰的链接措辞

把链接放在你的页面上很容易。这还不够。我们需要让所有的读者都可以使用链接，无论他们当前的环境和用于访问的工具（无障碍）。例如：

- 使用屏幕阅读器的用户喜欢从页面上的一个链接跳到另一个链接，并且脱离上下文来阅读链接。
- 搜索引擎使用链接文本来索引目标文件，所以在链接文本中包含关键词是一个很好的主意，以有效地描述与之相关的信息。
- 读者往往会浏览页面而不是阅读每一个字，他们的眼睛会被页面的特征所吸引，比如链接。他们会找到描述性的链接。

下面是一个具体的例子：

好的链接文本：[下载 Firefox](#)

HTML

```
<p><a href="https://www.mozilla.org/firefox/"> 下载 Firefox </a></p>
```

不好的链接文本：[点击这里](#) 下载 Firefox

HTML

```
<p><a href="https://www.mozilla.org/firefox/"> 点击这里 </a>下载 Firefox</p>
```

其他提示：

- 不要重复 URL 作为链接文本的一部分——URL 看起来很丑，当屏幕阅读器一个字母一个字母的读出来的时候听起来就更丑了。
- 不要在链接文本中说“链接”或“链接到”——它只是无用的内容。屏幕阅读器告诉人们有一个链接。可视化用户也会知道有一个链接，因为链接通常是用不同的颜色设计的，并且存在下划

线（这个惯例一般不应该被打破，因为用户习惯了它）。

- 保持你的链接标签尽可能短——这非常重要，因为屏幕阅读器需要解释整个链接文本。
- 尽量减少相同文本的多个副本链接到不同地方的情况。如果存在标记为“单击此处”、“单击此处”、“单击此处”这样脱离上下文的链接文本，容易对使用屏幕阅读器的用户带来问题。

链接到非 HTML 资源——留下清晰的指示

当链接到一个需要下载的资源（如 PDF 或 Word 文档）或流媒体（如视频或音频）或有另一个潜在的意想不到的效果（打开一个弹出窗口），你应该添加明确的措辞，以减少混乱。

如下的例子会让人反感：

- 你在使用低带宽连接的情况下，点击一个链接，意外地突然开始下载大文件。

让我们看看一些例子，看看在这里可以使用什么样的文本：

HTML

```
<p>
  <a href="https://www.example.com/large-report.pdf">
    下载销售报告（PDF，大小为 10 MB）
  </a>
</p>

<p>
  <a href="https://www.example.com/video-stream/" target="_blank">
    观看视频（将在新标签页中播放，HD 画质）
  </a>
</p>
```

在下载链接时使用 download 属性

当你链接到要下载的资源而不是在浏览器中打开时，你可以使用 `download` 属性来提供一个默认的保存文件名。下面是一个 Firefox 的 Windows 最新版本下载链接的示例：

HTML

```
<a
  href="https://download.mozilla.org/?product=firefox-latest-ssl&os=win64&lang=zh-CN"
  download="firefox-latest-64bit-installer.exe">
```


主动学习：创建一个导航菜单

在这个练习中，我们希望你把一些页面和导航菜单链接起来，创建一个多页面的网站。这是创建网站的一种常见方式——每个页面都使用相同的页面结构，包括相同的导航菜单，所以当点击链接时，给人的印象是你停留在同一个地方，而不同的内容被带了出来。

你需要将以下四页的本地副本放在同一目录中。有关完整的文件列表，请参见 [navigation-menu-start](#) 目录：

- [index.html](#)
- [projects.html](#)
- [pictures.html](#)
- [social.html](#)

你应该：

1. 在一个页面上的指定位置添加一个无序列表，其中包含要链接到的页面的名称。导航菜单通常只是一个链接列表，因此这在语义上是确定的。
2. 将每个页面名称转换为该页的链接。
3. 将导航菜单复制到每个页面。
4. 在每一页上，只删除同一页的链接——一个页面包含自己的链接是令人困惑和毫无意义的，而缺少链接会对你当前的页面起到很好的视觉提示作用。

最终的例子应该是这样的：

备注： 如果你卡住了，或者不确定是否正确，可以访问 [navigation-menu-marked-up](#) 目录，来查看正确的答案。

电子邮件链接

当点击一个链接或按钮时，可能会开启新的邮件的发送而不是连接到一个资源或页面。这种场景可以使用 `<a>` 元素和 `mailto:` URL 协议实现。

其最基本和最常用的使用形式为一个指明收件人电子邮件地址的 `mailto:` 链接。例如：

HTML

```
<a href="mailto:nowhere@mozilla.org">向 nowhere 发邮件</a>
```

这会创建一个链接，看起来像这样：[向 nowhere 发邮件](#)。

实际上，电子邮件地址是可选的。如果你省略了它（也就是说，你的 `href` 属性仅仅是简单的“mailto:”），发送新的电子邮件的窗口也会被用户的邮件客户端打开，只是没有收件人的地址信息，这通常在“分享”链接是很有用的，用户可以给他们选择的地址发送邮件。

指定详细信息

除了电子邮件地址，你还可以提供其他信息。事实上，任何标准的邮件头字段可以被添加到你提供的 `mailto` URL 中。其中最常用的是主题（subject）、抄送（cc）和主体（body）（这不是一个真正的标头字段，但允许你为新邮件指定一个简短的内容消息）。每个字段及其值被指定为查询项。

下面是一个包含 cc、bcc、主题和主体的示例：

HTML

```
<a
  href="mailto:nowhere@mozilla.org?
cc=name2@rapidtables.com&bcc=name3@rapidtables.com&subject=The%20subject%20of%20the%20email&
body=The%20body%20of%20the%20email">
  发送含有 cc、bcc、主题和主体的邮件
</a>
```

备注： 每个字段的值必须使用 URL 编码，即使用百分号转义的 非打印字符（不可见字符如制表符、换行符、分页符）和空格。同时注意使用问号（`?`）来分隔主 URL 与参数值，以及使用 `&` 符来分隔 `mailto:` URL 中的各个参数。这是标准的 URL 查询标记方法。阅读 [GET 方法](#) 以了解哪种 URL 查询标记方法是更常用的。

这里有一些其他的示例 `mailto` 链接：

- <mailto:>
- <mailto:nowhere@mozilla.org>
- <mailto:nowhere@mozilla.org,nobody@mozilla.org>
- <mailto:nowhere@mozilla.org?cc=nobody@mozilla.org>
- <mailto:nowhere@mozilla.org?cc=nobody@mozilla.org&subject=This%20is%20the%20subject>

技能测试！

你已经到了这篇文章的结尾，但还能记住最重要的信息吗？你可以找到一些进一步的测试，以验证你在继续前进之前已经保留了这些信息——见[技能测试：链接 \(en-US\)](#)。

小结

这就是链接！在课程的后期，当你开始研究链接的样式时，会回顾链接部分。接下来是 HTML，我们将回顾文本语义，并查看一些你会发现有用的更高级/不寻常的功能，[文本格式进阶](#)是你的下一站。

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](#).

