

从对象到 iframe——其他嵌入技术

到现在为止，您应该真正掌握了将内容嵌入网页的窍门，包括图像、视频和音频。在这一点上，我们想采取一些侧面的步骤，看看一些允许您将各种内容类型嵌入到您的网页中的元素：和 [<iframe>](#) 元素。s 用于嵌入其他网页，另外两个允许您嵌入外部资源，例如 PDF 文件。 [<embed>](#) [<object>](#) [<iframe>](#)

先决条件：	基本的计算机知识、 安装的基本软件、使用文件 的基本知识、熟悉 HTML 基础知识（如 HTML 入门 中所述）以及本模块中的先前文章。
客观的：	了解如何使用 <object> 、 <embed> 和 将项目嵌入到网页中 <iframe> ，例如 PDF 文档和其他网页。

嵌入的简史

很久以前在 Web 上，使用**框架**来创建网站很流行——网站的一小部分存储在单独的 HTML 页面中。这些嵌入在称为**框架集**的主文档中，它允许您指定屏幕上每个框架填充的区域，就像调整表格的列和行的大小一样。这些被认为是 90 年代中后期最酷的事情，并且有证据表明，将网页分成这样的小块可以提高下载速度——尤其是在当时网络连接速度如此之慢的情况下。然而，它们确实有很多问题，随着网络速度变得更快，这些问题远远超过了任何积极因素，所以你再也看不到它们被使用了。

不久之后（90 年代末、2000 年代初），插件技术变得非常流行，例如 [Java Applets](#)和[Flash](#)——它们允许 Web 开发人员将丰富的内容嵌入到网页中，例如视频和动画，而这些仅通过 HTML 是无法实现的. 嵌入这些技术是通过 [<object>](#) 和较少使用的元素实现的 [<embed>](#)，它们在当时非常有用。由于许多问题，包括可访问性、安全性、文件大小等，它们已经过时了。现在主流浏览器已经停止支持Flash之类的插件了。

最后，该 `<iframe>` 元素出现了（连同其他嵌入内容的方式，例如 `<canvas>`、`<video>` 等）这提供了一种将整个 Web 文档嵌入另一个文档的方法，就好像它是一个 `` 或其他这样的元素一样，并且今天经常使用。

历史课结束后，让我们继续看看如何使用其中的一些。

主动学习：经典嵌入使用

在本文中，我们将直接进入主动学习部分，让您立即真正了解嵌入技术的用途。网络世界对 [YouTube](#) 非常熟悉，但许多人不知道它提供的一些共享设施。让我们看看 YouTube 如何允许我们使用 `<iframe>`。

1. 首先，去 YouTube 找一个你喜欢的视频。
2. 在视频下方，您会找到一个共享按钮 — 选择此按钮可显示共享选项。
3. 选择“嵌入”按钮，您将获得一些 `<iframe>` 代码——复制它。
4. 将其插入下面的 *Input* 框中，并查看 *Output* 中的结果。

对于奖励积分，您还可以尝试在示例中嵌入 [Google 地图](#)：

1. 转到 Google 地图并找到您喜欢的地图。
2. 单击 UI 左上角的“汉堡菜单”（三个水平线）。
3. 选择共享或嵌入地图选项。
4. 选择嵌入地图选项，这将为您提供一些 `<iframe>` 代码——复制它。
5. 将其插入下面的 *Input* 框中，并查看 *Output* 中的结果。

如果你犯了一个错误，你总是可以使用重置按钮重置它。如果您真的遇到困难，请按“显示解决方案”按钮查看答案。

Live output

Nyan Cat [original]



详细的 iframe

所以，这既简单又有趣，对吧？[<iframe>](#) 元素旨在允许您将其他 Web 文档嵌入到当前文档中。这非常适合将第三方内容整合到您的网站中，您可能无法直接控制并且不想实施您自己的版本——例如来自在线视频提供商的视频、Disqus 等评论系统、[在线](#) 地图地图提供商、广告横幅等。甚至您在本课程中使用的实时可编辑示例也是使用 `<iframe>`s 实现的。

在深入使用 `<iframe>` 元素之前，需要注意一些安全问题。假设您想使用该元素将 MDN 词汇表包含在您的某个网页上 [<iframe>](#)，您可以尝试下一个代码示例。如果您将下面的代码添加到您的某个页面中，您可能会惊讶地看到一条错误消息而不是词汇表页面：

```
<head>
  <style>
    iframe {
      border: none;
    }
  </style>
</head>
<body>
  <iframe
    src="https://developer.mozilla.org/en-US/docs/Glossary"
    width="100%"
    height="500"
    allowfullscreen
    sandbox>
    <p>
      <a href="/en-US/docs/Glossary">
        Fallback link for browsers that don't support iframes
      </a>
    </p>
  </iframe>
</body>
```

如果您查看浏览器的控制台，您会看到如下错误消息：

```
Refused to display 'https://developer.mozilla.org/' in a frame
because it set 'X-Frame-Options' to 'deny'.
```

下面的安全部分详细介绍了为什么会出现此错误，但首先，让我们看看我们的代码在做什么。[👉](#)

该示例包括使用 `an` 所需的基本要素 `<iframe>`：

[`border: none`](#)

如果使用，则 `<iframe>` 显示时没有周围边框。否则，默认情况下，浏览器会显示 `<iframe>` 带有周围边框的（这通常是不可取的）。

[`allowfullscreen`](#)

如果设置，则 `<iframe>` 可以使用[全屏 API](#)将其置于全屏模式（有点超出本文的范围。）

[src](#)

此属性与 [<video>](#) / 一样 [](#)，包含指向要嵌入的文档的 URL 的路径。

[width](#) 和 [height](#)

这些属性指定您希望 iframe 的宽度和高度。

[sandbox](#)

此属性比其他功能 `<iframe>`（例如 IE 10 及更高版本）适用于更现代的浏览器，需要更高的安全设置；我们将在下一节中对此进行更多讨论。

注意：为了提高速度，最好 `src` 在主要内容加载完成后使用 JavaScript 设置 iframe 的属性。这使您的页面更快可用并减少您的官方页面加载时间（一个重要的[SEO](#)指标。）

安全问题

上面我们提到了安全问题——现在让我们更详细地讨论一下。我们不希望您第一次就完全理解所有这些内容；我们只是想让你意识到这个问题，并提供一个参考，当你变得更有经验并开始考虑 `<iframe>` 在你的实验和工作中使用 `s` 时，可以回来参考。此外，没有必要害怕和不使用 `<iframe>` `s`——你只需要小心。继续阅读.....

浏览器制造商和 Web 开发人员已经了解到 iframe 是 Web 上的坏人（通常称为**黑客**，或更准确地说，**骇客**）的常见目标（官方术语：**攻击向量**），如果他们试图恶意修改您的网页，或诱骗人们做他们不想做的事情，例如泄露用户名和密码等敏感信息。正因为如此，规范工程师和浏览器开发人员已经开发了各种安全机制来提高 `s` 的安全性，并且还需要考虑最佳实践——我们将在下面介绍其中的一些。`<iframe>`

注意：[点击劫持](#)是一种常见的 iframe 攻击，黑客将不可见的 iframe 嵌入到您的文档中（或将您的文档嵌入到他们自己的恶意

网站中) 并使用它来捕获用户的交互。这是误导用户或窃取敏感数据的常见方式。

不过, 首先是一个简单的示例——尝试将我们上面显示的前面的示例加载到您的浏览器中——您可以在 [在 GitHub 上找到它](#) (也可以[查看源代码](#)。) 而不是您期望的页面, 您可能会看到某种消息“我无法打开此页面”的效果, 如果您查看[浏览器开发人员工具](#)中的控制台, 您会看到一条消息告诉您原因。在 Firefox 中, 你会得到类似“X-Frame-Options”指令设置为“DENY”的框架中“<https://developer.mozilla.org/en-US/docs/Glossary>”加载被拒绝的信息”。这是因为构建 MDN 的开发人员在为网站页面提供服务的服务器上包含了一项设置, 以禁止将它们嵌入到 s 中 (请参阅下面的 <iframe> 配置[CSP 指令](#)。) 这是有道理的——整个 MDN 页面并不能真正使感觉嵌入到其他页面中, 除非你想做一些事情, 比如将它们嵌入你的网站并声称它们是你自己的——或者试图通过点击劫持来窃取数据, 这两者都是非常糟糕的事情。另外, 如果每个人都开始这样做, 所有额外的带宽将开始让 Mozilla 花费大量资金。

仅在必要时嵌入

有时嵌入第三方内容是有意义的——比如 youtube 视频和地图——但是如果你只在完全必要的时候嵌入第三方内容, 你可以避免很多麻烦。网络安全的一个好的规则是“你永远不能太谨慎。如果你做到了, 无论如何都要仔细检查。如果其他人做到了, 假设它是危险的, 直到证明不是这样。”

除了安全之外, 您还应该注意知识产权问题。大多数内容都受版权保护, 无论是离线还是在线, 甚至是您可能意想不到的内容 (例如, [维基共享资源](#)上的大多数图像)。除非您拥有该内容或所有者给予您明确的书面许可, 否则切勿在您的网页上显示内容。侵犯版权的处罚是严厉的。再一次, 你永远不能太谨慎。

如果内容已获得许可, 您必须遵守许可条款。例如, MDN 上的内容是[根据 CC-BY-SA 许可的](#)。这意味着, 您在引用我们的内容时必须[适当地注明出处](#), 即使您进行了重大更改。

使用 HTTPS

[HTTPS](#)是[HTTP](#)的加密版本。您应该尽可能使用 HTTPS 为您的网站提供服务：

1. HTTPS 减少了远程内容在传输过程中被篡改的可能性，
2. HTTPS 阻止嵌入内容访问父文档中的内容，反之亦然。

启用 HTTPS 的站点需要安装特殊的安全证书。许多托管服务提供商都提供支持 HTTPS 的托管服务，您无需自行进行任何设置来放置证书。但是，如果您确实需要为自己的站点设置 HTTPS 支持，[Let's Encrypt](#) 提供了可用于自动创建和安装必要证书的工具和说明——内置了对最广泛使用的 Web 服务器的支持，包括 Apache Web 服务器、Nginx 等。Let's Encrypt 工具旨在使该过程尽可能简单，因此没有充分的理由避免使用它或其他可用方式来使您的站点启用 HTTPS。

注意： [GitHub 页面](#)默认允许通过 HTTPS 提供内容，因此它对于托管内容很有用。如果您使用的是不同的托管服务提供商并且不确定，请咨询他们。

始终使用 sandbox 属性

您希望尽可能少地让攻击者在您的网站上做坏事，因此您应该只为嵌入内容提供完成其工作所需的权限。当然，这也适用于您自己的内容。可以适当使用（或用于测试）但不会对代码库的其余部分造成任何伤害（意外或恶意）的代码容器称为沙箱。

未沙盒化的内容可能能够执行 JavaScript、提交表单、触发弹出窗口等。默认情况下，您应该使用 sandbox 不带参数的属性来施加所有可用的限制，如我们前面的示例所示。

如果绝对需要，您可以一个一个地添加权限（在 sandbox="" 属性值内）——查看 [sandbox](#) 所有可用选项的参考条目。一个重要的注意事项是，您永远 allow-scripts 不应将和 都添加 allow-same-origin 到您的 sandbox 属性中——在这种情况下，嵌入的内容可能会绕过阻止站点执行脚本的[同源策略](#)，并使用 JavaScript 完全关闭沙盒。

注意：如果攻击者可以诱骗人们直接访问恶意内容（在 `<iframe>` 之外），则沙盒不会提供任何保护。如果某些内容有可能是恶意的（例如，用户生成的内容），请将其从不同的域提供给您的主站点。

配置 CSP 指令

[CSP](#)代表[内容安全策略](#)并提供[一组 HTTP 标头](#)（当网页从 Web 服务器提供服务时随网页一起发送的元数据）旨在提高 HTML 文档的安全性。在保护 `<iframe>` 时，您可以[将服务器配置为发送适当的 `X-Frame-Options` 标头](#)。这可以防止其他网站将您的内容嵌入到他们的网页中（这将启用[点击劫持](#)和许多其他攻击），正如我们之前看到的，这正是 MDN 开发人员所做的。

注意：您可以阅读 Frederik Braun 的帖子[On the X-Frame-Options Security Header](#)，了解有关此主题的更多背景信息。显然，这超出了本文的完整解释范围。

<embed> 和 <object> 元素



使用 One DevOps 平台简化软件开发。立即开始您的 30 天免费试用！

Mozilla 广告

不想看广告？

但是，您不太可能经常使用这些元素。如果您需要显示 PDF，通常最好链接到它们，而不是将它们嵌入到页面中。

从历史上看，这些元素也曾用于嵌入由 [Adobe Flash](#) 等浏览器[插件](#)处理的内容，但这项技术现在已经过时，并且不受现代浏览器的支持。

如果您发现自己需要嵌入插件内容，那么您至少需要以下信息：

	<embed>	<object>
嵌入内容的 URL	<code>embed</code>	<code>object</code>

	<code><embed></code>	<code><object></code>
嵌入内容的 准确媒体类型	<code>embed</code>	<code>object</code>
由插件控制的框的高度和宽度（以 CSS 像素为单位）	height width	height width
名称和值，将插件作为参数提供	具有这些名称和值的临时属性	单标签 <code><param></code> 元素，包含在 <code><object></code>
独立的 HTML 内容作为不可用资源的后备	不支持（ <code><noembed></code> 已过时）	包含在元素 <code><object></code> 之后 <code><param></code>

让我们看一个 `<object>` 将 PDF 嵌入到页面中的示例（参见[实时示例](#) 和[源代码](#)）：

```
<object data="mypdf.pdf" type="application/pdf" width="800"
height="1200">
  <p>
    You don't have a PDF plugin, but you can
    <a href="mypdf.pdf">download the PDF file. </a>
  </p>
</object>
```

PDF 是纸张和数字之间的必要垫脚石，但它们带来了许多[可访问性挑战](#)，并且在小屏幕上可能难以阅读。它们在某些圈子中仍然很受欢迎，但最好链接到它们，以便可以在单独的页面上下载或阅读它们，而不是将它们嵌入网页中。

测试你的技能！

您已读完本文，但您还记得最重要的信息吗？在继续之前，您可以找到一些进一步的测试来验证您是否保留了此信息 — 请参阅[测试您的技能：多媒体和嵌入](#)。

概括

在 web 文档中嵌入其他内容的主题很快就会变得非常复杂，因此在本文中，我们尝试以一种简单、熟悉的方式介绍它，这种方式会立即看起来相关，同时仍然暗示一些更高级的功能所涉及的技术。首先，除了在页面上包含地图和视频等第三方内容外，您不太可能使用嵌入。然而，随着您变得更有经验，您可能会开始发现它们的更多用途。

除了我们在此讨论的技术之外，还有许多其他技术涉及嵌入外部内容。我们在之前的文章中看到了一些，例如 [<video>](#)、[<audio>](#) 和 [](#)，但还有其他一些有待发现，例如 [<canvas>](#) JavaScript 生成的 2D 和 3D 图形，以及 [<svg>](#) 嵌入矢量图形。我们将在本模块的下一篇文章中介绍[SVG](#)。

此页面最后修改于 2023 年 4 月 21 日由[MDN 贡献者](#)提供。