

HTML 表格高级功能和辅助功能

在本模块的第二篇文章中，我们将了解 HTML 表格的一些更高级的功能——例如标题/摘要和将行分组到表头、正文和页脚部分——以及为视障用户查看表格的可访问性。

先决条件：	HTML 的基础知识（请参阅 HTML 简介 ）。
客观的：	了解更多高级 HTML 表格功能以及表格的可访问性。

使用 <caption> 为表格添加标题

您可以通过将标题放在 `<caption>` 元素内并将其嵌套在 `<table>` 元素内来为表格添加标题。你应该把它放在开始 `<table>` 标签的正下方。

```
<table>
  <caption>
    Dinosaurs in the Jurassic period
  </caption>

  ...
</table>
```

正如您可以从上面的简短示例中推断的那样，标题旨在包含对表格内容的描述。这对于希望在浏览页面时快速了解表格是否对他们有用的所有读者很有用，尤其是对于盲人用户。与其让屏幕阅读器读出许多单元格的内容只是为了找出表格的内容，用户可以依靠标题然后决定是否更详细地阅读表格。

标题直接放在 `<table>` 标签下方。

注意：该 `summary` 属性也可以用在 `<table>` 元素上以提供描述——这也可以由屏幕阅读器读出。<caption> 但是，我们建议改

用该元素，因为 `summary` 它已被弃用并且视力正常的用户无法阅读（它不会出现在页面上）。

主动学习：添加标题

让我们尝试一下，重新审视我们在上一篇文章中首次遇到的示例。

1. 从[HTML Table Basics](#)末尾打开您的语言老师的学校时间表，或制作我们的[timetable-fixed.html](#) 文件的本地副本。
2. 为表格添加合适的标题。
3. 保存您的代码并在浏览器中打开它以查看它的外观。

注意：您可以在 GitHub 上找到我们的版本 — 请参阅[timetable-caption.html](#)（[也可以实时查看](#)）。

使用 `<thead>`、`<tfoot>` 和 `<tbody>` 添加结构

当您的表在结构上变得有点复杂时，为它们提供更多结构定义会很有用。一种明确的方法是使用 `<thead>`、`<tfoot>` 和 `<tbody>`，这允许您标记表格的页眉、页脚和正文部分。

这些元素不会使屏幕阅读器用户更容易访问表格，并且不会自行增强视觉效果。然而，它们对于样式和布局非常有用——充当将 CSS 添加到表格的有用挂钩。给你一些有趣的例子，在长表格的情况下，你可以让表格的页眉和页脚在每个打印页面上重复，你可以让表格主体显示在一个页面上，并通过上下滚动来显示内容。

使用它们：

- 该 `<thead>` 元素必须包裹表格的标题部分——这通常是包含列标题的第一行，但情况并非总是如此。如果您使用的是 `<col>` / `<colgroup>` 元素，则表头应位于它们的正下方。
- 该 `<tfoot>` 元素需要包裹表格的页脚部分——例如，这可能是最后一行，其中前几行中的项目相加。您可以像预期的那样在表格底部添加表

格页脚，或者在表格标题下方（浏览器仍会将其呈现在表格底部）。

- 该 `<tbody>` 元素需要包装不在表头或表尾中的表内容的其他部分。它会出现现在表头下方，有时会出现现在表尾下方，具体取决于您决定如何构建它。

注意： `<tbody>` 始终包含在每个表中，如果您未在代码中指定它，则隐含地包含它。要检查这一点，请打开您之前未包含的示例之一，`<tbody>` 然后查看[浏览器开发人员工具](#)中的 HTML 代码——您会看到浏览器已为您添加了此标记。您可能想知道为什么你应该费心包括它——你应该，因为它可以让你更好地控制你的表格结构和样式。

主动学习：添加表结构

让我们将这些新元素付诸实践。

1. 首先，在新文件夹中制作[spending-record.html](#) 和[minimal-table.css](#)的本地副本。
2. 尝试在浏览器中打开它 — 您会发现它看起来还不错，但还有待改进。包含花费金额总和的“SUM”行似乎放错了地方，并且代码中缺少一些细节。
3. 将明显的标题行放在一个 `<thead>` 元素中，将“SUM”行放在一个 `<tfoot>` 元素中，将其余内容放在一个 `<tbody>` 元素中。
4. 保存并刷新，您会看到添加元素 `<tfoot>` 导致“SUM”行下降到表格底部。
5. 接下来，添加一个 `colspan` 属性，使“SUM”单元格跨越前四列，因此实际数字出现在“成本”列的底部。
6. 让我们向表格添加一些简单的额外样式，让您了解这些元素对于应用 CSS 有多么有用。在 HTML 文档的头部，您会看到一个空 `<style>` 元素。在此元素内，添加以下 CSS 代码行：

```
tbody {  
  font-size: 95%;  
  font-style: italic;  
}
```

```
tfoot {  
  font-weight: bold;  
}
```

7. 保存并刷新，看看结果。如果 `<tbody>` 和 `<tfoot>` 元素没有到位，您将不得不编写更复杂的选择器/规则来应用相同的样式。

注意： 我们不希望您现在就完全理解 CSS。当您浏览我们的 CSS 模块时，您将了解更多相关信息（[CSS 简介](#)是一个很好的起点；我们还有一篇专门介绍[样式表](#)的文章）。

您完成的表格应如下所示：

How I chose to spend my money			
Purchase	Location	Date	Evaluation
Haircut	Hairdresser	12/09	Great idea
Lasagna	Restaurant	12/09	Regrets
Shoes	Shoeshop	13/09	Big regrets
Toothpaste	Supermarket	13/09	Good
SUM			

注意： 您也可以在 [GitHub](#) 上以 `spending-record-finished.html` 的形式找到它（[也可以实时查看](#)）。

嵌套表

可以将一个表嵌套在另一个表中，只要包含完整的结构，包括元素 `<table>`。通常不建议这样做，因为它会使标记更加混乱并且屏幕阅读器用户更难访问，并且在许多情况下，您还不如将额外的单元格/行/列插入现有表格中。然而，有时这是必要的，例如，如果您想轻松地从一个来源导入内容。

以下标记显示了一个简单的嵌套表：

```
<table id="table1">
  <tr>
    <th>title1</th>
    <th>title2</th>
    <th>title3</th>
  </tr>
  <tr>
    <td id="nested">
      <table id="table2">
        <tr>
          <td>cell1</td>
          <td>cell2</td>
          <td>cell3</td>
        </tr>
      </table>
    </td>
    <td>cell2</td>
    <td>cell3</td>
  </tr>
  <tr>
    <td>cell4</td>
    <td>cell5</td>
    <td>cell6</td>
  </tr>
</table>
```

它的输出看起来像这样：

title1			title2	title3
cell1	cell2	cell3	cell2	cell3
cell4			cell5	cell6

视障人士专用桌

让我们简要回顾一下我们如何使用数据表。表格可以是一个方便的工具，让我们快速访问数据并允许我们查找不同的值。例如，只需快速浏览下表，即可了解 2016 年 8 月期间在根特售出的戒指数量。为了解其信息，我们在该表中的数据与其列和/或行标题之间建立视觉关联。

2016 年 8 月售出的商品

		衣服			配件	
		裤子	裙子	连衣裙	手链	戒指
比利时	安特卫普	56	22	43	72	23
	根特	46	18	50	61	15
	布鲁塞尔	51	27	38	69	28
荷兰人	阿姆斯特丹	89	34	69	85	38
	乌得勒支	80	12	43	36	19

但是，如果您无法建立这些视觉联想怎么办？那你怎么能读懂上面这样的表格呢？视障人士经常使用屏幕阅读器为他们读出网页上的信息。当您阅读纯文本时这没有问题，但解释表格对于盲人来说可能是一个很大的挑战。然而，通过适当的标记，我们可以用程序化的关联代替视觉关联。

注：根据世界卫生组织 2017 年的数据，大约有 2.53 亿人患有视力障碍。

本文的这一部分提供了使表尽可能易于访问的更多技术。

使用列标题和行标题

屏幕阅读器将识别所有标题并使用它们在这些标题和它们相关的单元格之间建立编程关联。列标题和行标题的组合将识别和解释每个单元格中的数据，以便屏幕阅读器用户可以像视力正常的用户一样解释表格。

我们已经在上一篇文章中介绍了标题 — 请参阅[使用 <th> 元素添加标题](#)。

范围属性

本文的一个新主题是属性 `scope`，它可以添加到 `<th>` 元素中，以准确地告诉屏幕阅读器标题是哪些单元格的标题——例如，它是所在行的标题，还是列的标题？回顾我们之前的支出记录示例，您可以明确地将列标题定义为列标题，如下所示：

```
<thead>
  <tr>
    <th scope="col">Purchase</th>
    <th scope="col">Location</th>
    <th scope="col">Date</th>
    <th scope="col">Evaluation</th>
    <th scope="col">Cost (€)</th>
  </tr>
</thead>
```

每行都可以有一个这样定义的标题（如果我们添加行标题和列标题）：

```
<tr>
  <th scope="row">Haircut</th>
  <td>Hairdresser</td>
  <td>12/09</td>
  <td>Great idea</td>
  <td>30</td>
</tr>
```

例如，屏幕阅读器将识别这样结构的标记，并允许其用户一次读出整列或整行。

`scope` 还有两个可能的值——`colgroup` 和 `rowgroup`。这些用于位于多列或多行顶部的标题。如果您回顾一下本文这一部分开头的“2016 年 8 月售出的商品”表，您会看到“衣服”单元格位于“裤子”、“裙子”和“连衣裙”单元格上方。所有这些单元格都应标记为标题 (`<th>`)，但“衣服”是位于顶部并定义其他三个子标题的标题。因此，“衣服”的属性应为 `scope="colgroup"`，而其他衣服的属性应为 `scope="col"`。

id 和 headers 属性

使用 scope attribute 的另一种方法是使用 [id](#) 和 [headers](#) attributes 在标题和单元格之间创建关联。它们的使用方法如下：

1. id 您为每个元素添加一个唯一的 <th> 。
2. headers 您向每个元素添加一个属性 <td> 。每个属性都必须包含充当该单元格标题的所有元素的 headers 列表，以空格分隔。id <th>

这为您的 HTML 表格提供了表格中每个单元格位置的明确定义，由其所属的每一列和行的标题定义，有点像电子表格。为了使其正常工作，该表确实需要列标题和行标题。

回到我们的支出成本示例，前两个片段可以这样重写：



使用基于 FIDO 的密钥身份验证为 Web 和本机应用程序转换您的登录方式。免费开始。

Mozilla 广告

不想看广告？

```
<th id="location">Location</th>
<th id="date">Date</th>
<th id="evaluation">Evaluation</th>
<th id="cost">Cost (€)</th>
</tr>
</thead>
<tbody>
  <tr>
    <th id="haircut">Haircut</th>
    <td headers="location haircut">Hairdresser</td>
    <td headers="date haircut">12/09</td>
    <td headers="evaluation haircut">Great idea</td>
    <td headers="cost haircut">30</td>
  </tr>

  ...
</tbody>
```

注意：此方法在标题和数据单元格之间创建了非常精确的关联，但它使用了更多标记并且不会留下任何错误空间。对于大多数

表，该 `scope` 方法通常就足够了。

主动学习：使用范围和标题

1. 对于最后的练习，我们希望您首先在新目录中制作[items-sold.html](#)和[minimal-table.css](#)的本地副本。
2. 现在尝试添加适当的 `scope` 属性以使该表更合适。
3. 最后，尝试制作另一个起始文件副本，这次使用 `id` 和 `headers` 属性使表更易于访问。

注意：您可以根据我们完成的示例检查您的工作——请参阅[items-sold-scope.html](#)（也可查看此实时）和[items-sold-headers.html](#)（也可查看此实时）。

概括

关于 HTML 中的表格，您还可以了解一些其他内容，但目前您只需要知道这些。[接下来，您可以使用我们的HTML 表格评估](#)来测试自己。玩得开心！

如果您已经在学习 CSS 并且在评估中取得了不错的成绩，您可以继续学习[样式化 HTML 表格](#)——请参阅[样式化表格](#)。

如果您想开始学习 CSS，请查看[CSS 学习区](#)！

此页面最后修改于 2023 年 3 月 12 日由[MDN 贡献者](#)提供。