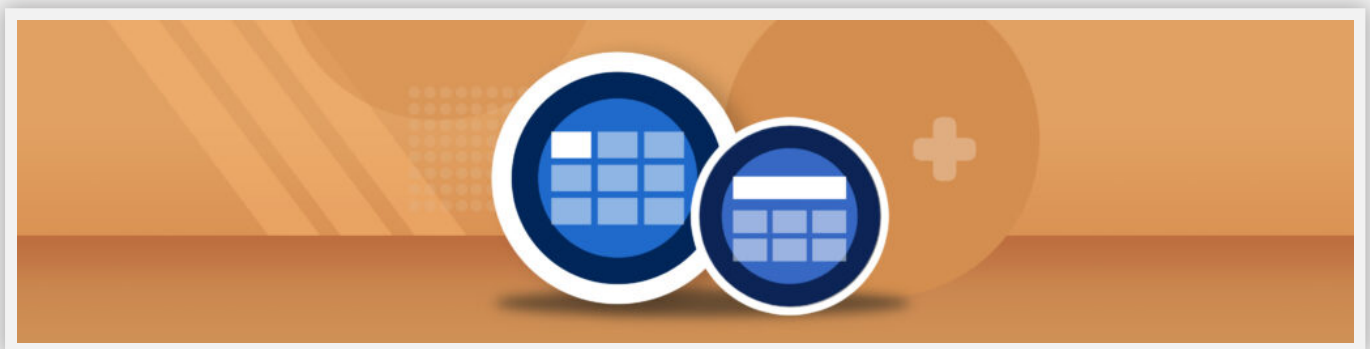




Updated: January 16, 2025

Beginner's guide to accessible tables

by Whitney Lewis



Quick summary

There are two types of tables: data and layout tables. Data tables organize information using headers. Layout tables add structure to a webpage, like columns, and should be avoided.

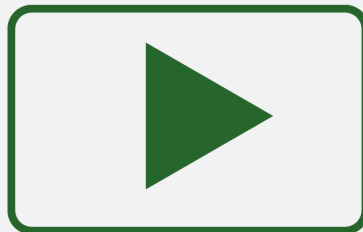
Accessible data tables use proper HTML like table headers, captions, and the scope attribute. This HTML helps screen reader users by announcing how many rows and columns there are and telling them the table headings before reading the information in the cell. That way the user knows what header the data relates to.

In this article, you'll learn:

- [What are data and layout tables?](#)

- [Why accessible data tables matter](#)
- [What makes tables accessible](#)
- [Complex tables](#)
- [Fix inaccessible data tables](#)
- [Check your tables for accessibility activity](#)

After learning about accessible tables, the next step is to check tables on your own website. You can start finding table-related issues today with our [Free Plan for the Pope Tech Web Accessibility Platform](#) – up to 15 pages, free for life, and no credit card required.



Rather watch the video? Watch [Introduction to accessible tables and a screen reader demo \(video\)](#).

What are data and layout tables?

The two types of tables are data and layout tables.

Data tables

Data tables use a table to organize information or data. They're used when it would be too complicated to describe with just text. They can also be used to [make charts and graphs more accessible](#).

There should be a row or column header to help organize the information. Some data

tables also have a caption describing what the table is about. Here's a simple data table with a caption and column and row headers:

Delivery slots:

	Monday	Wednesday	Friday
9:00 AM – 11:00 AM	Not available	Not available	Not available
11:00 AM – 1:00 PM	Available	Available	Not available

Layout tables

Layout tables use a table to add structure to a web page. For example, putting web content – not data – in a table to organize it by columns.

Layout tables should be avoided. Instead, use CSS to determine a page's layout.

[Back to top](#)

Why accessible data tables matter

We use data tables to make certain information easier to read. For visual users, tables make it easier to understand data.

For assistive technology users, **accessible** data tables can make understanding and navigating the information easier by:

- Letting them skip the table instead of going through each cell.
- Telling them how many columns or rows are in the table before entering the table.
- Telling them the table headings before reading the information in the cell, so they know what header(s) the data relates to.

[Back to top](#)

What makes tables accessible

Data tables and layout tables need specific HTML code to be accessible. The screen reader uses this code to announce the information in the table correctly.

Accessible data tables

Some HTML elements needed when making accessible data tables are:

- **The table header `<th>` tag.** The table headers can be assigned to the first row, column, or both. And, when the table has headers, the screen reader announces the header before the data in the cell, so the screen reader user knows what heading the data is associated with.
- **The scope attribute `<scope>` added to the table header `<th>` tag.** Scope tells the screen reader if the header is a column or row header.
- **We suggest adding a caption with a `<caption>` tag.** A caption is like a heading for the table, so it should describe what the table is about. The screen reader announces the caption before reading the data in the cells. A heading could be used instead of a caption. For more information, read: [Should I use a heading instead of a table caption?](#)
- **Complex data tables should use a [summary](#).** A summary describes the layout of a table and is only needed for complex tables with unusual structures. It can help users understand how to navigate the table.

Table headers, captions, possible layout tables, and problematic empty table headers are all issues Pope Tech's Web Accessibility Platform detects. You can check your own tables today using our [Free plan – up to 15 pages, free for life, no credit card required](#).

Accessible layout tables

Sometimes, a layout table is unavoidable. For example, an older website that's using layout tables to structure the content. If you're not able to remove the layout table and replace it with CSS, there is a way to make the layout table more accessible.

In these cases, we don't want the screen reader to announce that it's a table. Instead, we'd want the screen reader to just announce the content. To get this functionality with a layout table, there are two things you need to do in your code:

- **Assign the table the [ARIA role of presentation](#).** This removes all the table and its

children’s HTML semantics from the accessibility API. However, the content is still available to assistive technologies. The code would look like this: `<table role="presentation">`

- **Make sure the reading order of the content is correct in the code.** A screen reader will read the content based on the order in the HTML. So, if the order looks correct visually, but it’s mixed up in the code, it wouldn’t make sense when announced by a screen reader.

[Back to top](#)

Complex tables and accessibility

Complex tables have cells that span multiple rows or columns. Or, they have multiple column or row headers in one table.

Here’s an example from W3C’s table tutorials of a table with spanning cells and multiple headings.

Poster availability

Each poster has color options. Each color option has different available sizes. The first column is the poster name. The second column are the color options. The third, fourth, and fifth column are the available sizes.

Poster name	Color	Sizes available		
Zodiac	Full color	A2	A3	A4
	Black and white	A1	A2	A3
	Sepia	A3	A4	A5
Angels	Black and white	A1	A3	A4
	Sepia	A2	A3	A5

A table like this uses more HTML table elements to make sure a screen reader announces

it correctly. Since the table is so complex, it also has a summary describing the organization of the table, which would help a screen reader navigate it.

Instead of having more complex tables, our suggestion is to restructure them into multiple tables. It'll be a better experience for screen reader users and could also be easier to read for visual users.

The table above could be restructured to be two tables: one table for the zodiac poster and another for the angel poster. This removes the poster name column, which means there aren't multiple spanning headers.

From there, I made the caption "Sizes available for [poster name] by color" so it's clear which poster it is, there are different colors, and there are different sizes for each color. Now, each of my tables only needs one set of headers.

Sizes available for Zodiac poster by color

Full color	Black and white	Sephia
A2	A1	A3
A3	A2	A4
A4	A3	A5

Sizes available for Angel poster by color

Black and white	Sephia
A1	A2
A3	A3
A4	A5

To review the code for complex tables or more complex table examples, check out [W3C's table tutorials](#).

[Back to top](#)

Fix inaccessible data tables

To fix a data table that doesn't have table headers, the scope attribute, or a caption, add them in the HTML or use your web content editor.

Some web content editors don't have the option to add a caption or make the first column and row headers.

Most editors will have the option to switch to an HTML view though.

Once you switch to HTML, you can add a caption, headers, and the scope attribute to your table in the HTML. Here are the steps:

1. Switch to your content editor's HTML view.
2. Use **command or control + F** to pull up your browser's find tool. You can use this to easily find the table in the HTML. Search for text in the table or the word "table".
3. Change the `<td>` cells that should be headers to `<th>`.
4. Add the scope attribute to your headers. The column headers with the scope attribute will look like this: `<th scope="col"> Header text </th>`. The row headers with the scope attribute will look like this: `<th scope="row"> Header text </th>`
5. Right after the `<table>` tag add your table caption with this HTML: `<caption>Put brief description here</caption>`

Check out the examples below for code to reference.

Table with header cells in the top row only

Your HTML will look something like this if you only have **header cells in the top row**:

```
<table>
```

```

<caption>Fruit and the month they are ripe.</caption>
<tr>
  <th scope="col">Fruit type</th>
  <th scope="col">Month it's ripe</th>
</tr>
<tr>
  <td>Apple</td>
  <td>September</td>
</tr>
<tr>
  <td>Strawberry</td>
  <td>May-July</td>
</tr>
</table>

```

Fruit and the month they are ripe.

Fruit type	Month it's ripe
Apple	September
Strawberry	May-July

Table with header cells in the first column only

Your HTML will look something like this if you only have **header cells in the first column**:

```

<table>
<caption>Fruit and the month they are ripe.</caption>
<tr>
  <th scope="row">Fruit type</th>
  <td>Apple</td>
  <td>Strawberry</td>
</tr>

```



```

<tr>
  <th scope="row">Month it's ripe</th>
  <td>September</td>
  <td>May-July</td>
</tr>
</table>

```

Fruit and the month they are ripe.

Fruit type	Apple	Strawberry
Month it's ripe	Sept.	May-July

Both column and row headers

Your HTML will look something like this if you have both **row** and **column** headers:

```

<table>
  <caption>Delivery slots:</caption>
  <tr>
    <td></td>
    <th scope="col">Monday</th>
    <th scope="col">Wednesday</th>
    <th scope="col">Friday</th>
  </tr>
  <tr>
    <th scope="row">9:00 AM - 11:00 AM</th>
    <td>Closed</td>
    <td>Closed</td>
    <td>Closed</td>
  </tr>
  <tr>
    <th scope="row">11:00 AM - 1:00 PM</th>
    <td>Open</td>

```

```
<td>Open</td>
<td>Closed</td>
</tr>
</table>
```

Delivery slots:

	Mon.	Wed.	Fri.
9:00 AM – 11:00 AM	Closed	Closed	Closed
11:00 AM – 1:00 PM	Open	Open	Closed

[Back to top](#)

Key takeaways

- Data tables organize information that's too complex for text. Layout tables structure a page's content.
- Avoid using layout tables to make columns in your content. Instead, use CSS to adjust your page's layout.
- Use table headers, the scope attribute, and a table caption to make your data table accessible.
- Use the ARIA role of presentation and make sure your content is in the correct reading order in the code to make layout tables more accessible.
- Tables made with the proper HTML code tell screen reader users how many columns and rows there are before entering the table and announce the table headings before reading the information in the cell.
- Instead of having complex tables, restructure them into multiple tables.

Visit [September Accessibility Focus: Tables and Lists](#) for more articles and video resources.

Make your tables for accessibility activity

Now that you're familiar with accessible tables, you're ready to find and fix table issues on your own website. These are the results you'll focus on:

- [Empty table header](#)
- [Layout table](#)
- [Possible table caption](#)

Follow these steps to find and fix tables issues on your own website or Canvas course:

1. Use the Pope Tech Platform, Canvas Accessibility Guide, or the free WAVE extension tool to find the results listed above.
 - [Canvas Accessibility Guide](#) – Test the syllabus, home page, and other pages for one course.
 - Pope Tech Platform – From your Dashboard, [drill down to the table results](#).
 - [WAVE extension tool \(free\)](#) – Test four pages on a website or course you contribute to.
2. Review ten results.
 - If it's an empty table header, add headers to the table.
 - If it's a possible caption, review the content and add it as a caption to the table using the correct HTML if needed.
 - If it's a layout table, determine if it could be removed and replaced with CSS. If it can't be, add `role=presentation` to the table and make sure the content is in the right order in the HTML.
3. BONUS: If you don't have one already, schedule a monthly accessibility check-in (even if it's just you) to celebrate progress and remove blockers.

Creating accessible tables going forward

To avoid having to come back and fix tables in the future, answer these questions to come up with ways to help content creators create accessible tables:

- Do content creators know how to use the [WAVE extension tool](#), or is training needed?
- Does your content editor make it easy to add headers and captions to tables? Does it add the scope attribute automatically? If not, tables might need to be made or adjusted in HTML.
- Would a table template be helpful? If so, consider making a template of the table in your content editor or in HTML code. Then, only the content needs to be filled in, and the HTML is already accessible.

[Back to top](#)

Search Articles

Search

Monthly Accessibility Insights

Sign up to get monthly accessibility content. Unsubscribe at any time.

First Name

Email Address

Subscribe

Free Web Accessibility Scanning

Start improving your web accessibility today with our free plan.

No Credit Card required.

Free Plan



[Sitemap](#)

[Accessibility Statement](#)

[About Us](#)

[WAVE Suite](#)

[Contact Us](#)