

ISTA 230



Tables

CSS

Tables

HTML tables are used to display related data in a concise manner that is easily understood.

While there are other HTML structures available for grouping related data, HTML tables are particularly well suited for making data scannable, readable, and comparable.

CSS

Tables

Below is the HTML for creating a table with three rows and three columns.

HTML

```
<table>
  <tr>
    <td></td>
    <td><strong>Apples</strong></td>
    <td><strong>Oranges</strong></td>
  </tr>
  <tr>
    <td><strong>Wholesale</strong></td>
    <td>$1.00</td>
    <td>$2.00</td>
  </tr>
  <tr>
    <td><strong>Retail</strong></td>
    <td>$1.50</td>
    <td>$3.00</td>
  </tr>
</table>
```

The parent element is the <table> tag. All other table-related elements are descendants of the table tag. The next child is the table row element (<tr>). Within the table row element, we specify table cells (<td>, for 'table data').

CSS

Tables

Browser

	Apples	Oranges
Wholesale	\$1.00	\$2.00
Retail	\$1.50	\$3.00

CSS

Tables - Accessibility

Browser

	Apples	Oranges
Wholesale	\$1.00	\$2.00
<u>Retail</u>	<u>\$1.50</u>	<u>\$3.00</u>

Imagine that you were asked to read the above table and report the retail price of apples. Given the formatting of the table, sighted readers would be able to tell us that apples retail for \$1.50.

However, for visually-impaired users the question is a bit more challenging. Without being able to see the visual cues we've included in our formatting, they must rely on a screen reader to provide the data for them.

Were a screen reader to try and read the content from the last row of the above HTML table, it would read the following:

[Retail] [dollar one point five zero] [dollar three point zero zero]

CSS

Table Headings

The first step in making our tables more accessible is to replace our combo `<td>` tags with a table heading element (`<th>`);

HTML

```
<table>
  <tr>
    <td></td>
    <th>Apples</th>
    <th>Oranges</th>
  </tr>
  <tr>
    <th>Wholesale</th>
    <td>$1.00</td>
    <td>$2.00</td>
  </tr>
  <tr>
    <th>Retail</th>
    <td>$1.50</td>
    <td>$3.00</td>
  </tr>
</table>
```

CSS

Table Headings

- Useful for delineating headings from data
- Displayed as bold text in most modern browsers
- Text is center-aligned on most modern browsers
- Provides more structure and meaning

CSS

Tables - Accessibility

Browser

	Apples	Oranges
Wholesale	\$1.00	\$2.00
Retail	<u>\$1.50</u>	\$3.00

Using table headings, our table is still visually formatted just as we first intended. However, a screen reader is now able to associate our table headings with individual table cells. When a screen reader reads the data from the last row, second column, it will sound like the following:

[column two, row three, apples retail, dollar three point zero zero]

While using table headings does improve the user's experience when using a screen reader, it is still particularly tedious. This is one of the reasons that it is recommended to only use tables when absolutely necessary.

CSS

Table Captions

Instead of using another HTML element to provide the title of our table, we can use the `<caption>` element. This provides us with a better semantic relationship as the caption is clearly associated with a given table (whereas a `<div>` or `<h1>` might not be clearly related).

HTML

```
<table>
  <caption>Current Fruit Prices</caption>
  <tr>
    <td></td>
    <th>Apples</th>
    <th>Oranges</th>
  </tr>
  <tr>
    <th>Wholesale</th>
    <td>$1.00</td>
    <td>$2.00</td>
  </tr>
  <tr>
    <th>Retail</th>
    <td>$1.50</td>
    <td>$3.00</td>
  </tr>
</table>
```

CSS

Table Captions

- Allows you to title your tabular data
- Can only include one caption per table
- Always comes immediately after the opening `<table>` tag
- Text is centered on most modern browsers

CSS

Table Summary

In addition to providing a caption, we can also provide a 'summary' attribute for our table element.

HTML

```
<table summary="This table describes...">
  <caption>Current Fruit Prices</caption>
  <tr>
    <td></td>
    <th>Apples</th>
    <th>Oranges</th>
  </tr>
  <tr>
    <th>Wholesale</th>
    <td>$1.00</td>
    <td>$2.00</td>
  </tr>
  <tr>
    <th>Retail</th>
    <td>$1.50</td>
    <td>$3.00</td>
  </tr>
</table>
```

CSS

Table Summary

- Used by screen readers to provide a more detailed description of your table data.
- Not displayed on the screen!
- Especially important for more complex tables

CSS

Colspan

Consider the following schedule, created using an HTML table. You'll notice that the user is free from 8a-12p on both Monday and Tuesday. Additionally, the user is also free from 12p-2p on Tuesday.

Browser

Day	8am	10am	12pm
Monday	Free	Free	Busy
Tuesday	Free	Free	Free

CSS

Colspan

To create the schedule, the designer has used multiple table cells. However, since multiple cells contain the same data, we can combine these to create a less cluttered table.

HTML

```
<table>
  <tr>
    <th>Day</th>
    <th>8am</th>
    <th>10am</th>
    <th>12pm</th>
  </tr>
  <tr>
    <th>Monday</th>
    <td>Free</td>
    <td>Free</td>
    <td>Busy</td>
  </tr>
  <tr>
    <th>Tuesday</th>
    <td>Free</td>
    <td>Free</td>
    <td>Free</td>
  </tr>
</table>
```

CSS

Colspan

Using the 'colspan' attribute, we can specify that a given table cell should span more than one column.

HTML

```
<table>
  <tr>
    <th>Day</th>
    <th>8am</th>
    <th>10am</th>
    <th>12pm</th>
  </tr>
  <tr>
    <th>Monday</th>
    <td colspan="2">Free</td>
    <td>Busy</td>
  </tr>
  <tr>
    <th>Tuesday</th>
    <td colspan="3">Free</td>
  </tr>
</table>
```

CSS

Colspan

Browser

Day	8am	10am	12pm
Monday	Free	Busy	
Tuesday	Free		

The table cells above have been outlined to show the effect of the 'colspan' attribute.

CSS

Rowspan

Alternatively, we can use the 'rowspan' attribute to specify that a given table cell should span more than one row.

HTML

```
<table>
  <tr>
    <th>Day</th>
    <th>8am</th>
    <th>10am</th>
    <th>12pm</th>
  </tr>
  <tr>
    <th>Monday</th>
    <td rowspan="2">Free</td>
    <td rowspan="2">Free</td>
    <td>Busy</td>
  </tr>
  <tr>
    <th>Tuesday</th>
    <td>Free</td>
  </tr>
</table>
```

CSS

Rowspan

Browser

Day	8am	10am	12pm
Monday	Free	Free	Busy
Tuesday	Free	Free	Free

The table cells above have been outlined to show the effect of the 'rowspan' attribute.

CSS

Colspan AND Rowspan

We could also use both 'colspan' and 'rowspan' attribute to specify that a given table cell should span more than one column AND more than one row.

HTML

```
<table>
  <tr>
    <th>Day</th>
    <th>8am</th>
    <th>10am</th>
    <th>12pm</th>
  </tr>
  <tr>
    <th>Monday</th>
    <td rowspan="2" colspan="2">Free</td>
    <td>Busy</td>
  </tr>
  <tr>
    <th>Tuesday</th>
    <td>Free</td>
  </tr>
</table>
```

CSS

Colspan AND Rowspan

Browser

Day	8am - 10am	10am - 12pm
Monday	Free	Busy
Tuesday		Free

The table cells above have been outlined to show the effect of using both colspan and rowspan attributes.

CSS

Colspan AND Rowspan

Browser

Day	8am - 10am	10am - 12pm
Monday	Free	Busy
Tuesday	Free	Free

When using colspan and rowspan, it is important to consider how accessibility is impacted. Because cells span multiple columns/rows, multiple headers may apply. However, screen readers will only read one vertical and one horizontal header for each cell.

[column two, row two, 8am Monday, Free]

In the above example, the cell is identified as "8am Monday", even though the cell actually expresses data for both 8am and 10am, and for both Monday and Tuesday.

CSS

Table Headers and IDs

To help address accessibility concerns for cells that span multiple columns/rows, we can use the following approach:

First, we'll provide id attributes for all of our table headings.

Next, we'll use the 'headers' attribute on our table cell that spans multiple columns/rows. Inside of the 'headers' attribute, we'll specify the id attributes of every table heading we want associated with the table cell.

Note that the id values used for the 'headers' attribute should be included in the order in which you want screen readers to read them.

HTML

```
<table>
  <tr>
    <th id="day">Day</th>
    <th id="hr8">8am</th>
    <th id="hr10">10am</th>
    <th id="hr12">12pm</th>
  </tr>
  <tr>
    <th id="mon">Monday</th>
    <td headers="mon tues hr8 hr10" rowspan="2" colspan="2">Free</td>
    <td>Busy</td>
  </tr>
  <tr>
    <th id="tues">Tuesday</th>
    <td>Free</td>
  </tr>
</table>
```

CSS

Table Headers and IDs

Browser

Day	8am	10am	12pm
Monday			Busy
Tuesday	<u>Free</u>		Free

Using the 'headers' attribute, screen readers would now read the large table cell as:

[column two, row two, Monday Tuesday 8am 10am, Free]

CSS

When To Use Tables

Because of the accessibility issues they pose as well as their rigidity when displayed on smaller screens, tables should only be used for the following cases:

- If it's tabular data
- If the data needs to be comparable
- If the data is sortable by multiple facets

Even if the above criteria are met, that doesn't necessarily mean that a table is the only HTML element that can be used.

CSS

Is it a table?

The data below is a good case for using an HTML table. The data is tabular in nature and is likely to be compared side by side.

Rank	Area	Census Population		Change, 1990 to 2000	
		April 1, 2000	April 1, 1990	Number	Percent
1	Nevada	1,998,257	1,201,833	796,424	66.3
2	Arizona	5,130,632	3,665,228	1,465,404	40.0
3	Colorado	4,301,261	3,294,394	1,006,867	30.6
4	Utah	2,233,169	1,722,850	510,319	29.6
5	Idaho	1,293,953	1,006,749	287,204	28.5
6	Georgia	8,186,453	6,478,216	1,708,237	26.4
7	Florida	15,982,378	12,937,926	3,044,452	23.5
8	Texas	20,851,820	16,986,510	3,865,310	22.8
9	North Carolina	8,049,313	6,628,637	1,420,676	21.4
10	Washington	5,894,121	4,866,692	1,027,429	21.1

CSS

Is it a table?

The data below is a screenshot of a Netflix queue. Netflix uses an HTML table for this data. However, the same layout could possibly be achieved using a <div> element for each movie, with child elements for each section (title, rating, genre, etc.).

List Order	Title	Star Rating	Genre	Expected Availability	Format	Remove
1	Now You See Me	★★★★☆	Thrillers		DVD	X
2 <small>TOP ▲</small>	Stories We Tell	★★★★☆	Documentary		DVD	X
3 <small>TOP ▲</small>	Extremely Loud and Incredibly Close	★★★★☆	Drama		DVD	X
4 <small>TOP ▲</small>	Sherlock Holmes: A Game of Shadows	★★★★☆	Action & Adventure		DVD	X
5 <small>TOP ▲</small>	The Man with the Iron Fists	★★★☆☆	Action & Adventure		DVD	X
6 <small>TOP ▲</small>	Lawless	★★★★☆	Drama		DVD	X

CSS

Is it a table?

The data below is a screenshot of a calendar displayed on a website. While the presentation would certainly lead someone to believe that the underlying HTML included a table, the calendar was actually created using an ordered list. While it's debatable whether a table would have been a better approach, this does prove that HTML elements are very flexible and can be made to look however you, as the designer, want them to look.

April

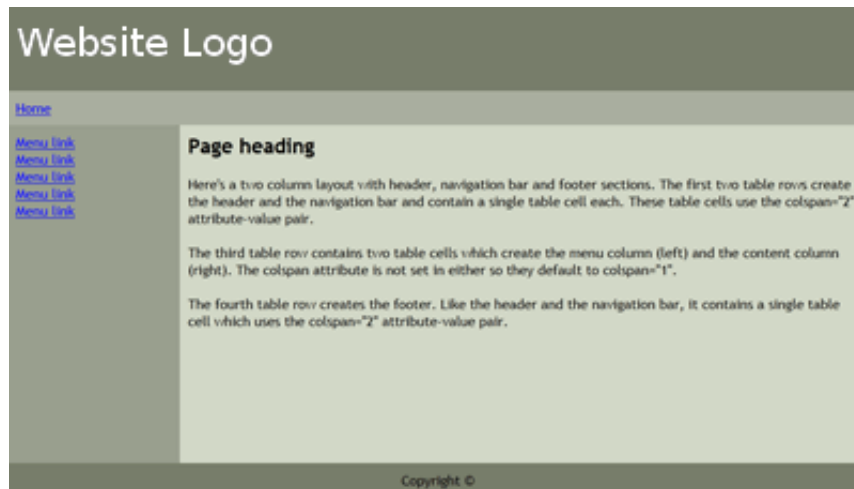
Sun	Mon	Tue	Wed	Thu	Fri	Sat
28	29	30	31	1 April Fools	2 Good Friday	3
4 Easter Sunday	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22 Earth Day	23	24
25	26	27	28	29	30	1

CSS

Table-Based Layouts

Years ago, it was common practice for website designers to use tables to format their entire web page. Using 'colspan' and 'rowspan', designers could essentially create a grid for each of their website elements, knowing that the table would always hold each piece in place.

For example, if a designer wanted a website that looked like this...



...they would build their HTML like this:

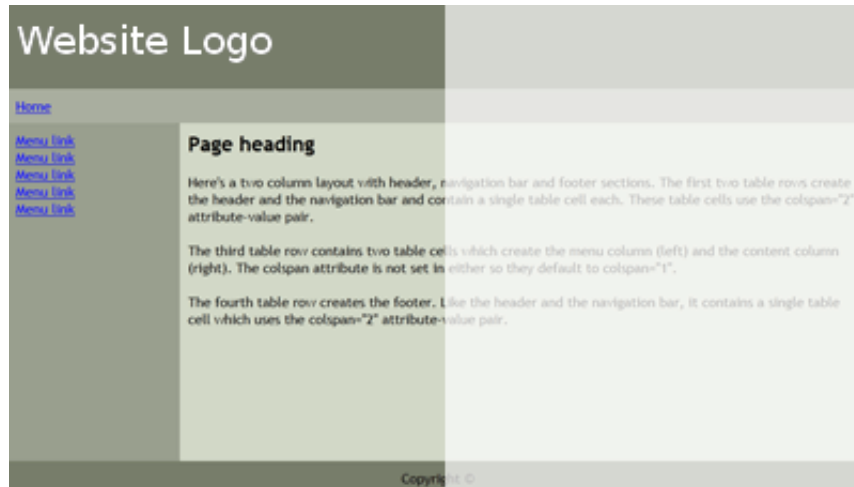
HTML

```
<table>
  <tr>
    <td colspan="2">
      <h2>Website Logo</h2>
    </td>
  </tr>
  <tr>
    <td colspan="2">
      <h2>Home</h2>
    </td>
  </tr>
  <tr>
    <td>
      <ul> ... navigation ...
    </td>
    <td>
      <h2>Page Heading</h2>
    </td>
  </tr>
  <tr>
    <td colspan="2">Copyright &copy;</td>
  </tr>
</table>
```

CSS

Table-Based Layouts

While the table-based layout would display properly on the designers screen, it would inevitably not fit on a smaller screen, resulting in horizontal scrolling. This became particularly problematic as users started visiting websites on their cell phones and were unable to see 90% of the webpage without scrolling.



CSS

Table-Based Layouts

If you use table-based layouts on your assignments, your assignment will **NOT** be graded and you will receive 0 for the assignment.

There is no excuse in this day and age to use tables for your webpage layouts. With the vast number of CSS properties we have discussed, you have more than enough tools to create a well-crafted design without relying on HTML tables to provide you with a grid-like layout. By not relying on HTML tables, you'll also have a more flexible design, which will become important as we discussing website design for mobile devices.

It is worth noting that while typically a frowned-upon practice, using table-based layouts is still a common approach when sending HTML-based email messages. Companies, such as Target or Best Buy, who want to present you with a fully-designed ad within your email will use a combination of in-line CSS, fixed-width layouts, absolute URLs, and HTML tables to craft their email messages. While these are practices typically avoided when building a traditional email, they are necessary to make sure that enhanced email messages are rendered consistently across multiple email clients.

CSS

Table Styling

Given the number of HTML elements that go into building a table, we have a number of options to consider when it comes to styling.

HTML

```
<table summary="This table describes...">
  <caption>Current Fruit Prices</caption>
  <tr>
    <td></td>
    <th>Apples</th>
    <th>Oranges</th>
  </tr>
  <tr>
    <th>Wholesale</th>
    <td>$1.00</td>
    <td>$2.00</td>
  </tr>
  <tr>
    <th>Retail</th>
    <td>$1.50</td>
    <td>$3.00</td>
  </tr>
</table>
```

CSS

Table Styling

By applying a border to the <table> element we can create a border around all of the table cells **BUT** not around the table caption.

CSS

```
table
{
  border: 2px solid #000;
}
```


CSS

Table Styling

Browser

Current Fruit Prices		
	Apples	Oranges
Wholesale	\$1.00	\$2.00
Retail	\$1.50	\$3.00

CSS

Table Styling

If we want to include a border around the <caption> element, we'll need to include a CSS selector for it as well.

CSS

```
table,  
caption  
{  
  border: 2px solid #000;  
}
```

CSS

Table Styling

It is worth noting that the border between the top of the table and the bottom of the caption is 4px wide (2px for the top table border and 2px for the bottom caption border).

Browser

Current Fruit Prices		
	Apples	Oranges
Wholesale	\$1.00	\$2.00
Retail	\$1.50	\$3.00

CSS

Table Styling

We can also apply a border to our table cells. It is worth noting that you'll need to provide CSS selectors for both the `<td>` and the `<th>` elements.

CSS

```
table,  
caption,  
td,  
th  
{  
  border: 2px solid #000;  
}
```

CSS

Table Styling

Browser

Current Fruit Prices		
	Apples	Oranges
Wholesale	\$1.00	\$2.00
Retail	\$1.50	\$3.00

CSS

Table Styling

By default, the width of table cells in a given column is based on the cell with the 'widest' content. For example, the middle column below is based on the size of the text, 'Granny Smith Apples'.

Browser

Current Fruit Prices		
	Granny Smith Apples	Oranges
Wholesale	\$1.00	\$2.00
Retail	\$1.50	\$3.00

CSS

Table Styling

To provide a more consistent look to our table, we can specify the width of our table headings. It is worth noting that by doing so, we implicitly set the width of the `<td>` cells as well.

CSS

```
th
{
  width: 33%;
}
```

CSS

Table Styling

Browser

Current Fruit Prices		
	Granny Smith Apples	Oranges
Wholesale	\$1.00	\$2.00
Retail	\$1.50	\$3.00

CSS

Table Styling - Border Spacing

The border-spacing property allows us to specify how much space should be displayed between table cells.

CSS

```
table
{
  border-spacing:0;
}
```

CSS

Table Styling

When border spacing is set to '0', we notice that the border between table cells press up against one another, giving the impression that the borders are twice the specified width.

Browser

Current Fruit Prices		
	Granny Smith Apples	Oranges
Wholesale	\$1.00	\$2.00
Retail	\$1.50	\$3.00

By default, borders within table elements are displayed side by side. However, we can adjust this using the border-collapse property.

CSS

Table Styling - Border Spacing

By setting the border-collapse property to 'collapse', we can specify that only borders between table elements should collapse on one another, resulting in only one border being displayed.

CSS

```
table
{
  border-spacing:0;
  border-collapse: collapse;
}
```

CSS

Table Styling

Browser

Current Fruit Prices		
	Granny Smith Apples	Oranges
Wholesale	\$1.00	\$2.00
Retail	\$1.50	\$3.00

CSS

Table Styling - Border Spacing

Because border-collapse has no effect on the caption/table border, we need to remove the bottom border of the caption (or the top border of the table) in order to get a consistent set of borders.

CSS

```
caption
{
  border-bottom: 0;
}
```

CSS

Table Styling

Browser

Current Fruit Prices		
	Granny Smith Apples	Oranges
Wholesale	\$1.00	\$2.00
Retail	\$1.50	\$3.00

CSS

border-spacing Property

CSS

```
table { border-spacing: 0; }
```

```
/*  
 * Sets the distance between adjacent cells of a table  
 * (when using non-collapsed borders).  
 *  
 * Default Value: none  
 * Inherited: Yes  
 */
```

CSS

border-collapse Property

CSS

```
table { border-collapse: 0; }
```

```
/*  
 * Specifies whether borders should collapse into a  
 * single border between table cells.  
 *  
 * Default Value: none  
 * Inherited: Yes  
 */
```


CSS

Table Styling - Border Collapse

HTML

```
<th id='userHeading'>Username</th>  
<th id='passHeading'>Password</th>
```

CSS

```
table { border-spacing: 0; border-collapse: collapse; }  
th#userHeading { border: 2px dotted #f00; }  
th#passHeading { border: 2px solid #0f0; }
```

Given the HTML and CSS above, the question must be asked: which border will show between the two table headings?

How should these borders collapse on one another?

CSS

Table Border Conflict Resolution

- Border styles take precedence in the following order:
 - border-style:hidden
 - 'hidden' has the same effect as 'none' when it comes to border style. However, it has higher precedence in context of HTML tables.
 - border-width (wider borders take precedence)
 - border-style (in order of precedence: double, solid, dashed, dotted, ridge, outset, groove, inset, none)

CSS

Table Border Conflict Resolution

- If border width and style are the same, then Border styles take precedence in the following order:
 - A style set on a cell take precedence over a row which take precedence over a table
 - Whichever is farthest to the left

CSS

Table Styling - Border Collapse

HTML

```
<th id='userHeading'>Username</th>  
<th id='passHeading'>Password</th>
```

CSS

```
table { border-spacing: 0; border-collapse: collapse; }  
th#userHeading { border: 2px dotted #f00; }  
th#passHeading { border: 2px solid #0f0; }
```

Because border style 'solid' takes precedence over border style 'dotted', the border in between the two table headings will be a solid green border.

Browser

Username	Password
----------	----------

CSS

Table Styling

When centering text within table cells, it is worth noting that the default vertical alignment for table cells is to vertically align text to the middle of the cell. However, we can change this by using the 'vertical-align' property.

CSS

```
td
{
  text-align:center;
  vertical-align: top;
}
```

Browser

Appointment Info	Monday	Tuesday Afternoon	Wednesday Night
---------------------	--------	----------------------	--------------------

CSS

vertical-align Property

CSS

```
td { vertical-align: top; }
```

```
/*  
 * Specifies the vertical alignment  
 * of an element. Useful for specifying  
 * the vertical alignment of table  
 * cells.  
 *  
 * Default Value: none  
 * Inherited: Yes  
 *  
 */
```

CSS

Table Styling - Backgrounds

While background colors and background images can be applied to tables, it is important to keep scanability and readability in mind. While applying a decorative gradient background to a table might be nice from an aesthetic standpoint, adding a more readable background color to your table cells would be a worthwhile effort.

CSS

```
table { background: #fff url('gradient.png') repeat-x 0 0; }  
caption { background-color: #036; color: white; }  
td { background-color:#fff; border: 1px solid #036; }
```

Browser

Tournament Entries				
Team 1	Team 2	Team 3	Team 4	Team 5
Jim	Fred	Max	Bill	Kyle
Jane	Frita	Mary	Bonnie	Kim
John	Frank	Matt	Bud	Ken
Jess	Flo	Martha	Barb	Kayla

CSS

Table Styling - Cell Width and Borders

Adding borders is a good approach to making table cells more distinguishable.

CSS

```
th
{
  background: #fff url('gradient.png') repeat-x 0 0;
  color: #fff;
}

td { border: 1px solid #036; }
```

Browser

Team 1	Team 2	Team 3	Team 4	Team 5
Jim	Fred	Max	Bill	Kyle
Jane	Frita	Mary	Bonnie	Kim
John	Frank	Matt	Bud	Ken
Jess	Flo	Martha	Barb	Kayla

CSS

Table Styling - Cell Width and Borders

As more spacing is provided between table cells, borders become less necessary for making cell data easily scannable.

CSS

```
table { width: 100%; }  
th  
{  
  background: #fff url('gradient.png') repeat-x 0 0;  
  color: #fff;  
  width: 20%;  
}
```

Browser

Tournament Entries				
Team 1	Team 2	Team 3	Team 4	Team 5
Jim	Fred	Max	Bill	Kyle
Jane	Frita	Mary	Bonnie	Kim
John	Frank	Matt	Bud	Ken
Jess	Flo	Martha	Barb	Kayla

CSS

Table Styling - Zebra Striping

Without any sort of division between the table cells, the data found in the table below looks somewhat cluttered and is more difficult to read than we would like.

CSS

```
th { background: #000; color: #fff; }
```

Browser

STUDENTS		
Name	Grade Level	Major
Jim	Freshman	Psychology
Jane	Sophomore	Computer Science
John	Junior	History
Jess	Senior	Art
Justin	Senior+	Music

CSS

Table Styling - Zebra Striping

To help make the data more readable, we'll add a class attribute to every other row in the table.

HTML

```
<table>
  <tr><td>Jim</td>...</tr>
  <tr class="alt"><td>Jane</td>...</tr>
  <tr><td>John</td>...</tr>
  <tr class="alt"><td>Jess</td>...</tr>
  <tr><td>Justin</td>...</tr>
</table>
```

CSS

Table Styling - Zebra Striping

We can then use that class attribute to style every other row within the table. By using a different background color for alternating rows, we provide the user with clear visual markers for each row.

We can also provide additional styling for our table cells when the user hovers their mouse over a given table row.

CSS

```
th { background: #000; color: #fff; }  
tr td { background-color:#fff; }  
tr.alt td { background-color:#eee; }  
tr:hover td {background-color: #ccc; }
```

Browser

STUDENTS		
Name	Grade Level	Major
Jim	Freshman	Psychology
Jane	Sophomore	Computer Science
John	Junior	History
Jess	Senior	Art
Justin	Senior+	Music

CSS

Table Styling - General Guidelines

- Headings should be clearly identifiable
- Rows and columns should be easily distinguishable
- The more white space between elements, the less necessary borders become

CSS

HTML Tables

Use of HTML tables should adhere to the following guidelines:

- Use only when appropriate
- Supply appropriate HTML attributes for accessibility
- Style for easy scanning, reading, and comparing