**Notes**

**HTML**

The bold and strong tags format the enclosed text to be displayed as bold text. The strong tag also marks the text as important

See the bold tag is action below:

<b>Text goes here</b>

See the strong tag is action below:

<strong>Text goes here</strong>

The italic and emphasis tags format the enclosed text to be displayed as italicized text. The emphasis tag also marks the text as important.

See the italic tag in action below:

<i>Text goes here</i>

See the emphasis tag in action below:

<em>Text goes here</em>

The superscript and subscript tags format the enclosed text to be displayed as superscript or subscript, respectively. Neither tag marks the text as important.

See the superscript and subscript tags in action below:

Superscript: <sup>Text goes here</sup>

Subscript: <sub>Text goes here</sub>

The inserted tag formats the enclosed text as underlined. The deleted tag formats the enclosed text as crossed out. Neither tag marks the text as important.

See the inserted and deleted tags in action below:

Inserted: <ins>Text goes here</ins>

Deleted: <del>Text goes here</del>

The marked tag formats the enclosed text as highlighted.

See the marked tag in action below:

<mark>Text goes here</mark>

The blockquote tag marks the enclosed text as quoted text. Browsers usually display quoted text as being indented with line breaks above and below the indented block.

See the blockquote tag in action below:

Here is an example of some blockquoted text:

<blockquote>

Lorem ipsum dolor sit amet, magna sed quia, elit non

ut corrupti, aenean commodo elit, tincidunt platea

purus quis at dolor a, felis nunc massa pede cras

quis. Lacinia fusce, praesent at pede phasellus, dui

etiam in luctus.

</blockquote>

That was a blockquote!

The code tag displays the enclosed text in a fixed-width font, like you see in many text editors used for programming. It is often used to set apart and display programming code you wish to show your users. This is commonly used on tutorial websites to demonstrate coding concepts.

See the code tag in action below:

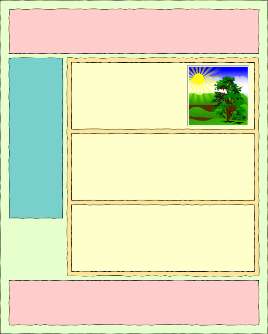
<code>Text goes here</code>

See an HTML comment in action below:

This text will appear on the page.

<!-- This is a comment and will not display -->

This will also appear on the page.



<!DOCTYPE html>

<**html**>

<**head**>

<**title**>Page with Semantic Elements</**title**>

</**head**>

<**body**>

<**div**>

<**div**>

...

</**div**>

<**div**>

...

</**div**>

<**div**>

<**div**>

<**div**>

</**div**>

</**div**>

<**div**>

</**div**>

<**div**>

</**div**>

</**div**>

<**div**>

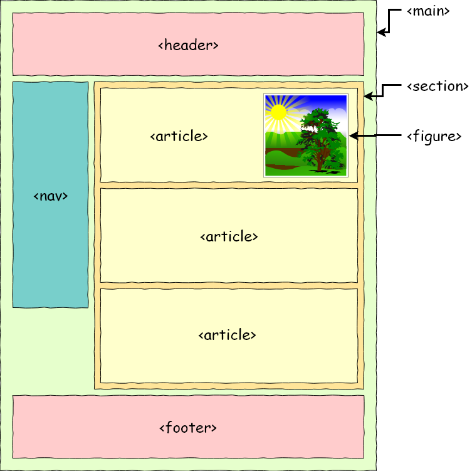
</**div**>

</**div**>

</**body**>

</**html**>

* main: a block that represents the primary section of a page
* header: a block at the top of a page
* footer: a block at the bottom of a page
* nav: a block that includes navigation elements, such as buttons or menus
* article: a block that contains nested blocks related to a single topic, like a newspaper article
* section: a block that identifies a section of content with a specific purpose, like the section of a newspaper
* form: a block that contains the elements of a form
* figure: A block that contains an image and information about that image



<!DOCTYPE html>

<**html**>

<**head**>

<**title**>Page with Semantic Elements</**title**>

</**head**>

<**body**>

<main>

<**header**>

...

</**header**>

<**nav**>

...

</**nav**>

<**section**>

<**article**>

<**figure**>

</**figure**>

</**article**>

<**article**>

</**article**>

<**article**>

</**article**>

</**section**>

<**footer**>

</**footer**>

</main>

</**body**>

</**html**>

<a>Clickable content</a>

<**a** href="https://www.google.com">Google</**a**>.

<**a** href="https://www.google.com">

<**img** src="images/portrait.jpg" />

</**a**>

<**a** href="https://www.google.com" target="\_blank">Google</**a**>

\_blank and \_self. \_blank will open the link in a new browser tab while \_self will open in the same tab, replacing the current page.

Internal Links

<**a** href="about.html">About Us</**a**>

<**a** href="products/desktop-computer.html">Featured Product</**a**>

### Bookmarks

<**body**>

<**h1**>Bookmark Links in HTML</**h1**>

<**p**><**a** href="#target">This is the question.</**a**></**p**>

<**figure**><**img** src="https://cdn.pixabay.com/photo/2018/06/24/20/32/ivy-3495403\_1280.jpg" alt="" /></**figure**>

<**p** id="target">This is the target.</**p**>

<**figure**><**img** src="https://cdn.pixabay.com/photo/2018/06/18/06/43/nature-3481966\_1280.jpg" alt="" /></**figure**>

</**body**>

## **Common Image Attributes**

### SRC Attribute

<**img** src="https://cdn.pixabay.com/photo/2017/06/21/00/22/dog-2425528\_960\_720.jpg" />

### ALT Attribute

<**img** src="https://cdn.pixabay.com/photo/2017/06/21/00/22/dog-2425528\_960\_720.jpg" alt="Photo of a puppy and a kitten." />

### STYLE Attribute

<**img** src="https://cdn.pixabay.com/photo/2017/06/21/00/22/dog-2425528\_960\_720.jpg"

alt="Photo of a puppy and a kitten." style="**border**: thick red solid;" />

<**img** src="https://cdn.pixabay.com/photo/2017/06/21/00/22/dog-2425528\_960\_720.jpg"

alt="Photo of a puppy and a kitten."

style="**border**: thick red solid; **width**: 300px;" />

<**img** src="https://cdn.pixabay.com/photo/2017/06/21/00/22/dog-2425528\_960\_720.jpg"

alt="Photo of a puppy and a kitten."

style="**border**: thick red solid; **width**: 300px; **height**:465px" />

One way to add an image to a website is by specifying the URL of that image in the src attribute. For example:

<**img** src="https://cdn.pixabay.com/photo/2017/06/21/00/22/dog-2425528\_960\_720.jpg" alt="Photo of a puppy and a kitten." />

Using a relative path to a file inside our website then, the img element would look like:

<**img** src="images/portrait.jpg" alt="Portrait of Marci Kitzmiller" />

## **Using Lists**

<**h2**>Ordered Lists</**h2**>

<**ol** type="A">

<**li**>First Item</**li**>

<**li**>Second Item</**li**>

<**li**>Third Item</**li**>

</**ol**>

| **Type** | **Purpose** |
| --- | --- |
| type=”1” | Numbers used to sequence the list (default) |
| type=”A” | Uppercase letters used to sequence the list |
| type=”a” | Lowercase letters used to sequence the list |
| type=”I” | Uppercase Roman Numerals used to sequence the list |
| type=”i” | Lowercase Roman Numerals used to sequence the list |

<**h2**>Unordered Lists</**h2**>

<**ul** style="**list-style-type**: square;">

<**li**>Milk</**li**>

<**li**>Bread</**li**>

<**li**>Cereal</**li**>

</**ul**

| **Type** | **Purpose** |
| --- | --- |
| list-style-type=”disc” | Bullet or Filled Circle indicator (default) |
| list-style-type=”square” | Square indicator |
| list-style-type=”circle” | Hollow circle indicator (similar to disc, but without the inner fill) |
| list-style-type=”none” | No indicator |

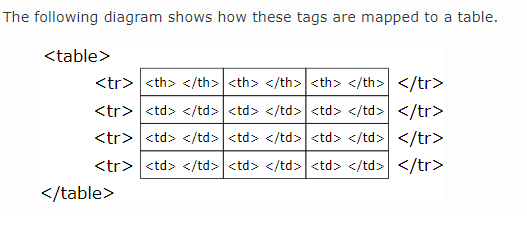
## **Definition Lists**

Although not as common as the other two list types, the definition list provides a useful way to list terms and their associated definition.

There are three elements that make up a definition list:

1. The <dl></dl> tags designate the type of list and serve as a grouping container for all items in the list.
2. The <dt></dt> tags designate the term to be defined
3. The <dd></dd> tags provide the definition/description for the term it is associated with.
4. <**h2**>Definition Lists</**h2**>
5. <**dl**>
6. <**dt**>One</**dt**>
7. <**dd**>The first positive number in our counting system.</**dd**>
8. <**dt**>Two</**dt**>
9. <**dd**>The first whole number greater in value than one.</**dd**>
10. <**dt**>Three</**dt**> <**dd**>The next whole number in increasing value after two.
11. </**dd**>
12. </**dl**>

|  |  |
| --- | --- |
| **Element Name** | **Element Purpose** |
| <table></table> | marks the beginning and end of a table |
| <tr></tr> | table row: marks the beginning and end of a row within a table |
| <th></th> | table header: marks the beginning and end of a cell in a header row |
| <td></td> | table data: marks the beginning and end of a cell in a table row |



### Table with Headings

<!DOCTYPE html>

<**html** lang="en">

<**head**>

<**title**>Tables in HTML</**title**>

</**head**>

<**body**>

<**table** border="1">

<**tr**>

<**th**>First Name</**th**>

<**th**>Last Name</**th**>

<**th**>Phone</**th**>

</**tr**>

<**tr**>

<**td**>John</**td**>

<**td**>Doe</**td**>

<**td**>555-1212</**td**>

</**tr**>

<**tr**>

<**td**>Sally</**td**>

<**td**>Smith</**td**>

<**td**>555-1123</**td**>

</**tr**>

<**tr**>

<**td**>Sam</**td**>

<**td**>Jones</**td**>

<**td**>555-4321</**td**>

</**tr**>

</**table**>

</**body**>

</**html**>

### Table with Caption

<**table** border="1">

<**caption**>Contact Information</**caption**>

<**tr**>

<**th**>First Name</**th**>

<**th**>Last Name</**th**>

<**th**>Phone</**th**>

</**tr**>

<**tr**>

<**td**>John</**td**>

<**td**>Doe</**td**>

<**td**>555-1212</**td**>

</**tr**>

<**tr**>

<**td**>Sally</**td**>

<**td**>Smith</**td**>

<**td**>555-1123</**td**>

</**tr**>

<**tr**>

<**td**>Sam</**td**>

<**td**>Jones</**td**>

<**td**>555-4321</**td**>

</**tr**>

</**table**>

### Table Width

<**table** border="1" width="90%">

<**caption**>Contact Information</**caption**>

<**tr**>

<**th**>First Name</**th**>

### Changing Background Colors

<**table** border="1" width="90%">

<**caption**>Contact Information</**caption**>

<**tr** bgcolor="palegreen">

<**th**>First Name</**th**>

### Cells that Span Multiple Columns

<**table** border="1" width="90%">

<**caption**>Contact Information</**caption**>

<**tr**>

<**th** colspan="2">Name</**th**>

<**th**>Phone</**th**>

</**tr**>

<**tr**>

<**td**>John</**td**>

<**td**>Doe</**td**>

<**td**>555-1212</**td**>

</**tr**>

<**tr**>

<**td**>Sally</**td**>

<**td**>Smith</**td**>

<**td**>555-1123</**td**>

</**tr**>

### Cells That Span Multiple Rows

<**th**>Category</**th**>

<**th** colspan="2">Name</**th**>

<**th**>Phone</**th**>

</**tr**>

<**tr**>

<**td** rowspan="3">Friends:</**td**>

<**td**>John</**td**>

<**td**>Doe</**td**>

<**td**>555-1212</**td**>

</**tr**>

<**tr**>

<**td**>Sally</**td**>

<**td**>Smith</**td**>

<**td**>555-1123</**td**>

</**tr**>

<**tr**>

<**td**>Sam</**td**>

<**td**>Jones</**td**>

<**td**>555-4321</**td**>

| **Form Tag Name** | **Purpose** |
| --- | --- |
| <form> </form> | Marks the beginning and end of the form. |
| <input /> | Represents some type of input element based on its type attribute. Valid type attribute values include: text, radio, submit, password, checkbox, button, and more. |
| <select> </select> | Marks the beginning and end of a drop-down list. |
| <option> </option> | Marks the beginning and end of an option in the drop-down list. |
| <textarea> </textarea> | Marks the beginning and end of a multi-line text input box. |

If the form is to properly submit its data to a server, the <form> tag must have the following two attributes:

1. action – specifies the URL where the form data will be submitted.
2. method – specifies the HTTP method (POST or GET) that should be used to submit the data.
3. <**body**>
4. <**h1**>Form Example</**h1**>
5. <**form** action="fakePage.html" method="POST">
7. </**form**>

## **Input Tag**

<input type="type of input" name="name of input" >

text

<**form** action="fakePage.html" method="POST">

Text Input: <**input** type="text" name="myTextInput" />

</**form**>



radio

<**form** action="fakePage.html" method="POST">

Text Input: <**input** type="text" name="myTextInput" /><**br** />

Radio Buttons:<**br** />

<**input** type="radio" name="answer" value="YES" checked />Yes<**br** />

<**input** type="radio" name="answer" value="NO" />No<**br** />

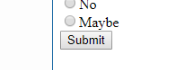
<**input** type="radio" name="answer" value="MAYBE" />Maybe<**br** />

</**form**>



submit

<**input** type="submit"/><**br** />



reset

<**input** type="submit" /><**input** type="reset" /><**br** />



Password

Password Input: <**input** type="password" name="userPassword" /><**br** />



Checkbox

<**input** type="checkbox" name="myChoices" value="Salad" checked/>Salad<**br** />



### Select Tag

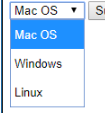
<**select** name="phones">

<**option** value="iPhone6">Mac OS</**option**>

<**option** value="GalaxyS6">Windows</**option**>

<**option** value="Nexus5">Linux</**option**>

</**select**>



### Textarea Tag

<**textarea** name="briefDescription"></**textarea**><**br** />



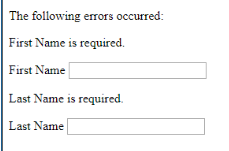
## **Making fields required**

Text Input: <input type="text" name="myTextInput" **required** />

**CSS**

.warning class="warning"

#about id="about"



By default, we do not need these warnings to show, so we can hide them with CSS. When we want to target a class, the selector starts with a dot (.), like .warning. Because the class applies to both div and p elements, we can use a single rule to target all of them.

<!DOCTYPE html>

<**html**>

<**head**>

<**title**>CSS Examples</**title**>

<**style**>

.warning {

**display**: none;

}

</**style**>

</**head**>

<**body**>

<**div** class="warning">

<**p**>The following errors occurred:</**p**>

</**div**>

<**p** class="warning">First Name is required.</**p**>

<**label** for="firstName">First Name</**label**>

<**input** type="textbox" name="firstName"/>

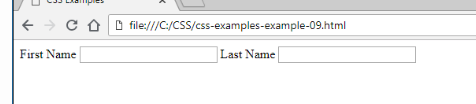
<**p** class="warning">Last Name is required.</**p**>

<**label** for="lastName">Last Name</**label**>

<**input** type="textbox" name="lastName"/>

</**body**>

</**html**>



## **Apply a Rule to More Than One Selector**

<**style**>

#product1, #product3 {

**color**: blue;

}

#product2 {

<**style**>

.warning {

**font-weight**: bold;

**color**: red;

}

**div**.warning {

**font-style**: italic;

}

</**style**>

</**head**>

<**body**>

<**div** class="warning">

<**p**>The following errors occurred:</**p**>

</**div**>

<**p** class="warning">First Name is required.</**p**>

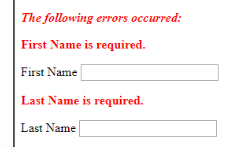
<**label** for="firstName">First Name</**label**>

<**input** type="textbox" name="firstName"/>

<**p** class="warning">Last Name is required.</**p**>

<**label** for="lastName">Last Name</**label**>

<**input** type="textbox" name="lastName"/>



## **Internal Style Sheets**

## **External Style Sheets**

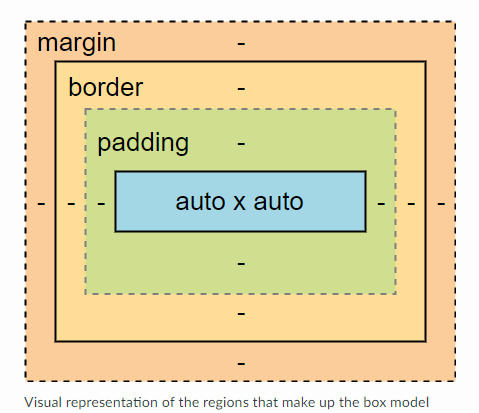
<link rel="stylesheet"

type="text/css"  
 href="sampleStyles.css" />

You can also link to an external style sheet that is already published to the Web, using @import.

@import url("http://mysite.com/styles/sharedStyles.css");

## **The CSS Box Model**



* padding: whitespace around the content, between the content and the border
* border: a line that creates a visual separation between the box's content and the content around it
* margin: whitespace between the border and the content around the box

#### **Tip**

The term whitespace comes from printing layouts and it refers to planned empty space in a page layout. However, it is actually transparent space, so "whitespace" will be whatever color the background uses.

Whitespace is an important part of a page layout because it can help make a page look less cluttered, as well as helping the user identify distinct blocks of content on the page.

Content is what is at the core of the element, which could include images or text. In this particular example, the core is the text "Your Journey Begins Here." The dimensions of the content can be dependent on the content itself, or it can be controlled via the height and width CSS properties. These dimensions can take fixed and relative values. An example of a fixed value is using a specific number of pixels (e.g., 216px). An example of a relative value is to specify a percentage of the parent container – the declaration width=90%; means that the content's width will be displayed at a size equivalent to 90% of its parent container, for example, but the exact width depends on the width of the parent container.

top padding of 8px, a right padding of 16px, a bottom padding of 4px, and a left padding of 4px.

padding: 8px 16px 4px 4px;

If all the sides shared the same padding value

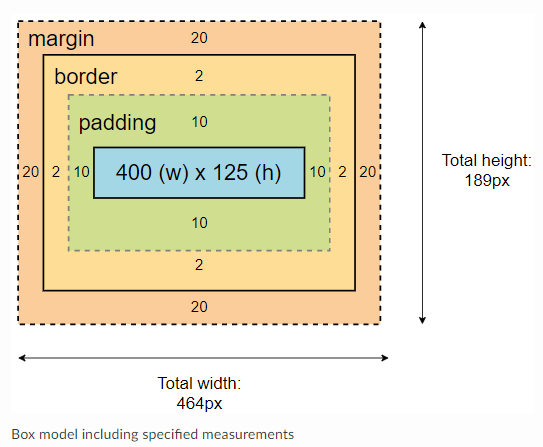
padding: 5px;

The border encloses the padding and the content. The CSS border, border-top, border-left, border-right, and border-bottom shorthand properties give us more flexibility, in that they include:

* Thickness (border-width) – typically in medium, thin, thick, or a fixed length
* Color (border-color) – any CSS-accepted color format, including color names, rgb(a), or hexadecimal.
* Style (border-style) – solid, double, dashed, dotted, groove, ridge, inset, outset, hidden, and none

Use the border shorthand when the style applies to all borders. For example, the following rule would apply a thin, gray, solid border on all sides:

border: thin #888 solid;



### background-image

Specifies the URL of the image to use, using the format background-image:url("image-address.png");.

### background-repeat

* no-repeat: does not repeat
* repeat-x: repeats horizontally
* repeat-y: repeats vertically
* repeat: repeats both horizontally and vertically

### background-attachment

* scroll: scrolls
* fix: anchored/fixed in place

### background-position

Determines the initial position of an image with two values that apply to X (horizontal) and Y (vertical) positioning; values can be percentages, lengths, or keywords.

* X position keywords include:
  + right
  + left
  + center
* Y position keywords include:
  + top
  + bottom
  + center

### Background Shorthand

All of these background properties can be shortened to the format:

background: color image repeat attachment position;

If any of those properties are not specified, they are assumed to be their default values. For example, this:

background-image: url("../images/confetti.gif");

backround-repeat: repeat;

can be simplified to:

background: url("../images/confetti.gif") repeat;

**body** {

**background-image**:url("paper-1914901\_100.png");

**background-color**: #cccccc;

}

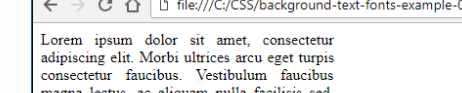
</**style**>

### Alignment

Much like how you can align text in a word processing document, you can do the same in HTML. The CSS attribute text-align allows us to align text to the left of the page, to the right, centered, or justifed. By default, text is aligned to its native position – left if the language's direction is ltr (left-to-right, like English) and right if the direction is rtl (right-to-left, like Arabic or Hebrew).

<**div** style="**text-align**: justify;**width**: 300px;">

Lorem ipsum dolor sit amet, consectetur



### Indentation

<**div** style="**text-indent**: 15%;**width**: 300px;">

Lorem ipsum dolor sit amet, consectetur

### Capitalization

* capitalize – capitalize the first character of each word
* uppercase – convert all characters TO UPPERCASE
* lowercase – CONVERT ALL CHARACTERS to lowercase
* none – no capitalization effects
* <**p** style="**text-transform**: uppercase;">
* Now this is crazy!
* <**span** style="**text-transform**: none; **color**: dodgerblue;">
* Yay!
* </**span**>
* <**span** style="**text-transform**: capitalize;">
* we love CSS!
* </**span**>



<**span** style="**text-decoration**:underline;">

<**span** style="**text-decoration**:overline;">

<**span** style="**text-decoration**:line-through;">



## **Font**

### Style

* italic – italicized font
* oblique – slanted font
* normal – default font style, no slanting

### Size

* Absolute – xx-small, x-small, small, medium, large, x-large, xx-large
* Length – em, ex, px, cm, mm, in, pt, or pc
* Percentage – %
* Relative – smaller, larger

### Weight

The weight of a font is the thickness of the font. The font-weight CSS property controls the strength of the font being styled. Weight values include:

* Numeric values - in increments of 100 between 100 and 900
* normal – default weight, in the middle
* bold – thick characters
* Relative – bolder, lighter

### Family

#### **Generic Families**

* serif – normal fonts with serifs (small lines often added at the end of the lines that make up the letter) such as Garamond, Georgia, New York, Times, and Times New Roman
* sans-serif – normal fonts without serifs such as Arial, Geneva, Helvetica, Lucida Sans, Trebuchet, and Verdana
* monospace – fixed-width fonts such as Courier, Consolas, Lucida Console, and Monaco
* cursive – fonts that emulate handwriting such as Comic Sans MS, Lucida Handwriting, and Zapf Chancery
* fantasy – decorative fonts such as Copperplate, Desdemona, Impact, and Kino
* If you want the browser to use a specific font, you can specify that as well, such as:
* font-family: 'Comic Sans MS';

## **Pseudo-Classes**

So far, we have examined styling HTML elements with CSS rules in their default states. However, sometimes you may need to style some elements in a particular state. Pseudo-classes are used for styling special states of HTML elements. They appear in the CSS selector section of a CSS rule, like this:

selector:pseudo-class {

/\* CSS declarations \*/

}

## **Links**

* link – unvisited link
* visited – visited link (meaning that you have recently opened the target page in your browser)
* hover – mouse over the link
* active – selected link

In this example, our requirements are:

* Unvisited links are green.
* Visited links are red.
* All links are underlined, except when hovering over a link.
* All links are in a monospace font, except when hovering over a link. When hovering over the link, the text is in cursive.
* We want to make sure it is clear which link we are selecting. Active (selected) links are bold.
* a {
* font-family: monospace;
* text-decoration: underline;
* font-weight: none;
* }
* a:link {
* color: green;
* }
* a:visited {
* color: red;
* }
* a:hover {
* text-decoration: none;
* color: #00c;
* font-family: cursive;
* }
* a:active {
* font-weight: bold;
* }

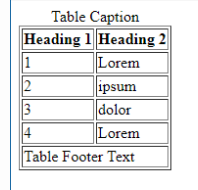
## **Tables**

Tables can be tricky HTML elements to style, in part because there are so many elements that make up a table. Before we get into the CSS side, though, let's look at three more tags that are helpful in table HTML.

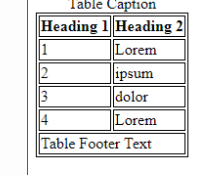
* <thead> </thead> – used for grouping table header content
* <tbody> </tbody> – used for grouping table body content
* <tfoot> </tfoot> – used for grouping table footer content
* <table>
* <caption>Table Caption</caption>
* <thead>
* <tr>
* <th>Heading 1</th>
* <th>Heading 2</th>
* </tr>
* </thead>
* <tbody>
* <tr>
* <td>1</td>
* <td>Lorem</td>
* </tr>
* <tr>
* <td>2</td>
* <td>ipsum</td>
* </tr>
* <tr>
* <td>3</td>
* <td>dolor</td>
* </tr>
* <tr>
* <td>4</td>
* <td>Lorem</td>
* </tr>
* </tbody>
* <tfoot>
* <tr>
* <td colspan="2">Table Footer Text</td>
* </tr>
* </tfoot>
* </table>

### Borders

By default, a table does not display borders. If we add the HTML attribute border="1" to our table tag (<table border="1">),



* Thickness (border-width) – typically in medium/thin/thick or a fixed width
* Color (border-color) – in any CSS-accepted color format
* Style (border-style) – solid, double, dashed, dotted, groove, ridge, inset, outset, hidden, and none
* table, th, td {
* border: 1px solid black;
* }

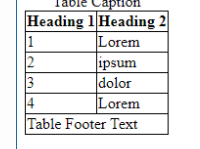


table, th, td {

border: 1px solid black;

border-collapse: collapse;

}



table, th, td {

border: 1px solid black;

border-collapse: collapse;

padding-top: 5px;

padding-right: 0px;

padding-bottom: 5px;

padding-left: 10px;

}

This can also be written with the shorthand syntax:

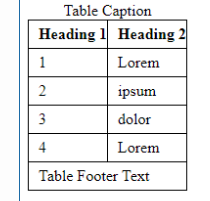
table, th, td {

border: 1px solid black;

border-collapse: collapse;

padding: 5px 0px 5px 10px;

}



### Widths and Heights

table {

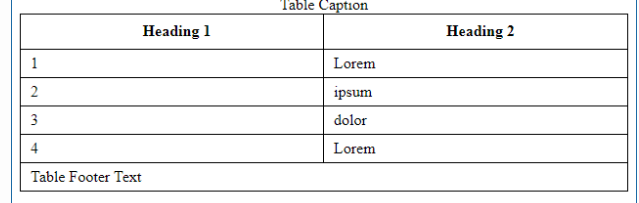
width: 100%;

}

th {

height: 25px;

}

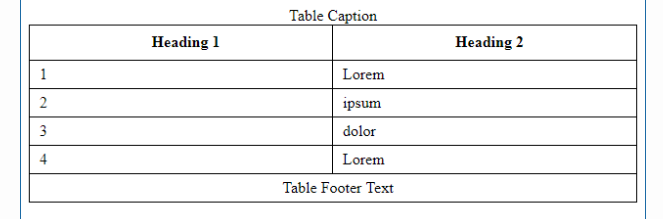


## **Text Alignment**

tfoot td {

text-align: center;

}



### Styling Table Row Backgrounds

tbody tr:nth-child(even) {

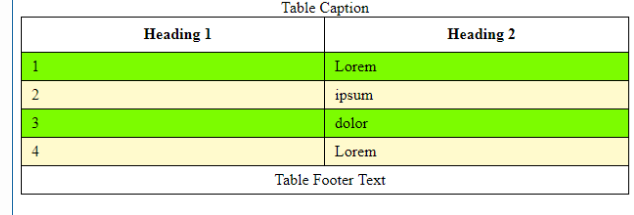
background-color: lemonchiffon;

}

tbody tr:nth-child(odd) {

background-color: lawngreen;

}



## **Final Check**

The complete source code for this table is:

<!DOCTYPE html>

<**html** lang="en">

<**head**>

<**title**>Links and Tables</**title**>

<**style**>

**table**, **th**, **td** {

**border**: 1px solid black;

**border-collapse**: collapse;

**padding**: 5px 0px 5px 10px;

}

**table** {

**width**: 100%;

}

**th** {

**height**: 25px;

}

**tfoot** **td** {

**text-align**: center;

}

**tbody** **tr**:**nth-child**(even) {

**background-color**: lemonchiffon;

}

**tbody** **tr**:**nth-child**(odd) {

**background-color**: lawngreen;

}

</**style**>

</**head**>

<**body**>

<**table**>

<**caption**>Table Caption</**caption**>

<**thead**>

<**tr**>

<**th**>Heading 1</**th**>

<**th**>Heading 2</**th**>

</**tr**>

</**thead**>

<**tbody**>

<**tr**>

<**td**>1</**td**>

<**td**>Lorem</**td**>

</**tr**>

<**tr**>

<**td**>2</**td**>

<**td**>ipsum</**td**>

</**tr**>

<**tr**>

<**td**>3</**td**>

<**td**>dolor</**td**>

</**tr**>

<**tr**>

<**td**>4</**td**>

<**td**>Lorem</**td**>

</**tr**>

</**tbody**>

<**tfoot**>

<**tr**>

<**td** colspan="2">Table Footer Text</**td**>

</**tr**>

</**tfoot**>

</**table**>

</**body**>

</**html**>

## **Bootstrap**

download the CSS and JavaScript files

Linking to a CDN

linking out to the same minified CSS and JavaScript files stored on their website

The abbreviation CDN stands for Content Delivery Network

In order for Bootstrap JavaScript to work, you must add jQuery and popper.js to your website, BEFORE adding Bootstrap.

## **Meta Elements**

<**head**>

<**meta** charset="utf-8">

<**meta** name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

## **Add Bootstrap CSS**

<**title**>Bootstrap Sample Page</**title**>

<!-- Bootstrap CDN copied from https://getbootstrap.com/docs/4.1/getting-started/introduction/ -->

<**link** rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css" integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO" crossorigin="anonymous">

</**head**>

## **Add the JavaScript files**

Now we need to include the files required for the JavaScript content to work. Because Bootstrap is built using the popper.js and jQuery frameworks, the browser must load those frameworks first, before loading the Bootstrap framework, or Bootstrap will not work correctly.

Using the links provided on Bootstrap's Introduction page, add all three frameworks at the bottom of the body element, just above the </body> tag.

<!-- CDN links copied from https://getbootstrap.com/docs/4.1/getting-started/introduction/ -->

<**script** src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous"></**script**>

<**script** src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js" integrity="sha384-ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPIPm49" crossorigin="anonymous"></**script**>

<**script** src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js" integrity="sha384-ChfqqxuZUCnJSK3+MXmPNIyE6ZbWh2IMqE241rYiqJxyMiZ6OW/JmZQ5stwEULTy" crossorigin="anonymous"></**script**>

</**body**>

</**html**>

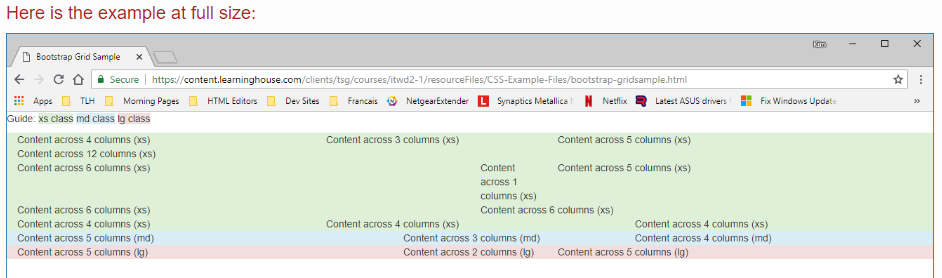
While JavaScript files can be added to the head element with the style sheets, that means the browser will have to read those files before it can display any of the page's content in the body element. To speed up the time required to load the content (and improve user experience), it is common practice to add larger JavaScript files inside the body element, below any content in the page and just above the closing </body> tag. That way the user can see and start to use the HTML content while the browser is loading the JavaScript files in the background

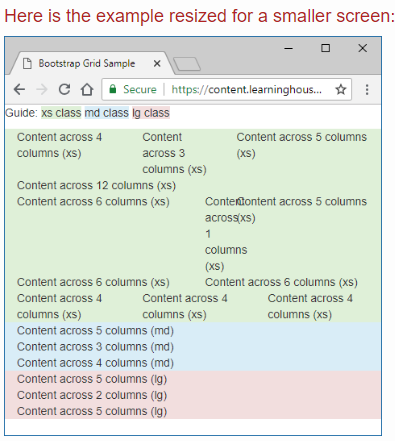
## **Responsive Design with Bootstrap’s Grid System**

Bootstrap’s grid system allows you to lay out your page in a grid that is 12 columns across. It is put together using [flexbox (Links to an external site.)Links to an external site.](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Flexible_Box_Layout/Basic_Concepts_of_Flexbox) which helps automatically size some of our components. Here are some things to note:

* As mentioned above, your content needs to be enclosed in a div with either a ***container*** (fixed-width) or ***container-fluid*** (full-width) class.
* There are CSS classes for the rows and columns of the grid:
  + ***row*** is used to group content into a row.
  + ***col*** classes are there for grouping content across multiple columns. You can specify exactly how many columns the HTML element should span using a notation like **col-4**, which tells Bootstrap that the content should span four grid columns.
  + ***sm***, ***md***, and ***lg*** refer to the devices that they are targeting. If a smaller device style is applied and a larger device style is not present, the smaller device style will apply to the element regardless of the viewing device.
  + In previous versions, **xs** referenced extra small screens. This has been replaced in Bootstrap 4 with just **col** by itself with no size indicator appended.
  + As of Bootstrap 4, If you don't specify a column width for a column (***col*** vs ***col-12***) it will automatically resize the columns in a single row to fit into the 12 grid columns evenly.  For example, three ***col*** columns in a ***row*** will automatically resize to be the equivalent of ***col-4***. Similarly, two ***col*** columns in a ***row*** will automatically resize to be the equivalent of ***col-6*** each.

To better understand the grid, open up [**bootstrap-gridsample.html** (Links to an external site.)Links to an external site.](https://content.learninghouse.com/clients/tsg/courses/itwd2-1/resourceFiles/CSS-Example-Files/bootstrap-gridsample.html). This file displays multiple extra small formatted rows so that you can see how to spread content across 12 columns. There are also medium and large formatted rows.





## **Looking at CSS Media Queries**

Behind the grid system are CSS media queries. These are ways in CSS to apply styles based on certain conditions, including:

* Width and height of the viewport and device
* Orientation – landscape or portrait mode
* Types – including screen, speech (for screen readers), print (for styling output for a printer), and all

Media queries are noted by @media and follow the following format:

@media type and (property:value) {

selector {

property:value;

property:value;

}

selector2{

property:value;

property:value;

}

}

## **Example Media Queries in Bootstrap**

| **Media Query** | **Meaning** |
| --- | --- |
| @media print | For output sent to a printer |
| @media (min-width: 768px) | For a viewport at least 768px wide |
| @media screen and (max-width: 767px) | For viewing with a screen and the max viewing width is 767px |
| @media (max-device-width: 480px) and (orientation: landscape) | For devices that have a max device width of 480px and are displaying the site in landscape mode |
| @media all and (transform-3d), (-webkit-transform-3d) | For all types that support CSS3’s 3D transformations |