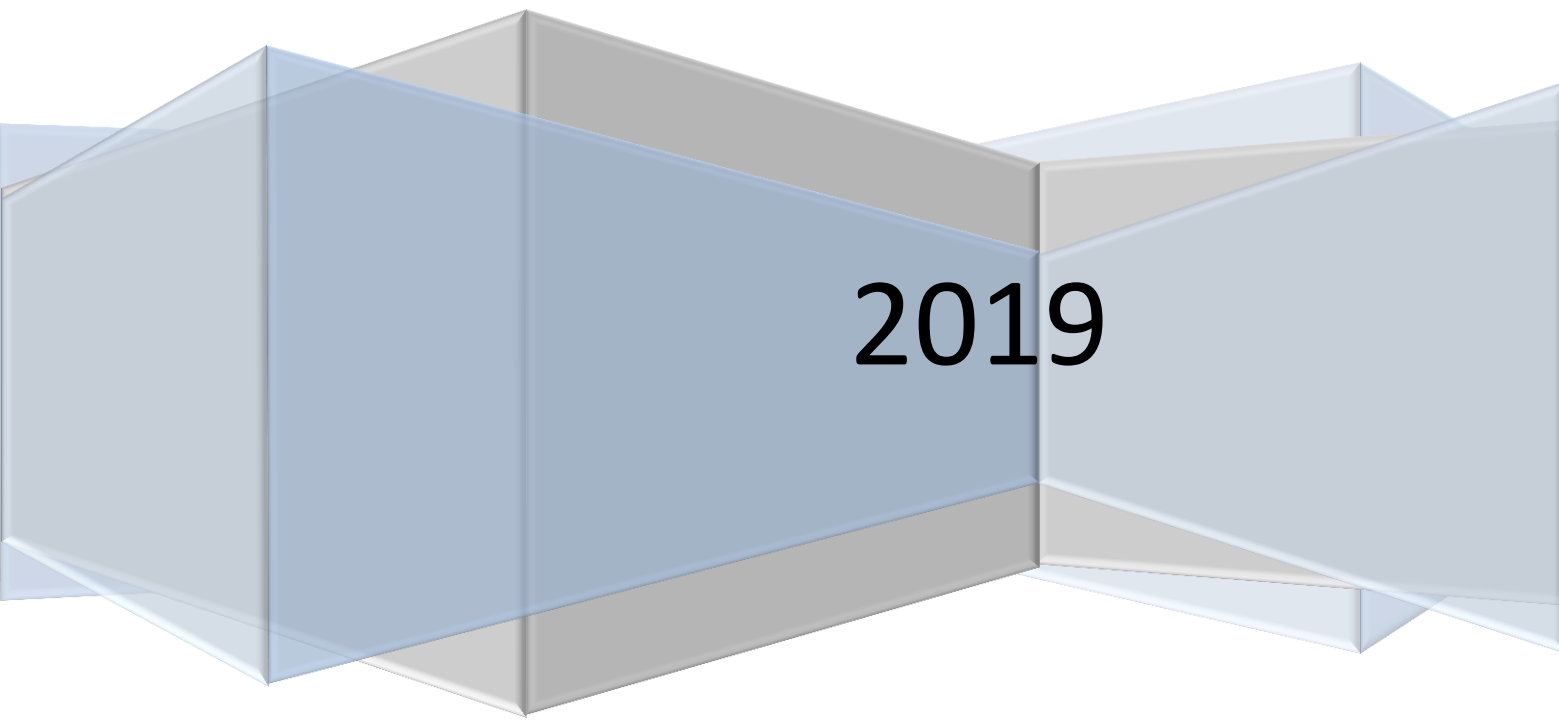


University of Kirkuk
College of Science
Computer Science Department
4th class

Web Design & Programming HTML & CSS

Lectures

M.Sc. Banaz A. Qader



2019

Introduction to HTML

What is HTML?

HTML is the standard markup language for creating Web pages.

- HTML stands for Hyper Text Markup Language
- HTML describes the structure of a Web page
- HTML consists of a series of elements
- HTML elements tell the browser how to display the content
- HTML elements are represented by tags
- HTML tags label pieces of content such as "heading", "paragraph", "table", and so on
- Browsers do not display the HTML tags, but use them to render the content of the page

A Simple HTML Document

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>
<h1>My First Heading</h1>
<p>My first paragraph.</p>
</body>
</html>
```

Example Explanation

- The <!DOCTYPE html> declaration defines this document to be HTML5
- The <html> element is the root element of an HTML page
- The <head> element contains meta information about the document
- The <title> element specifies a title for the document

- The <body> element contains the visible page content
- The <h1> element defines a large heading
- The <p> element defines a paragraph

HTML Tags

HTML tags are element names surrounded by angle brackets:

<tagname>content goes here...</tagname>

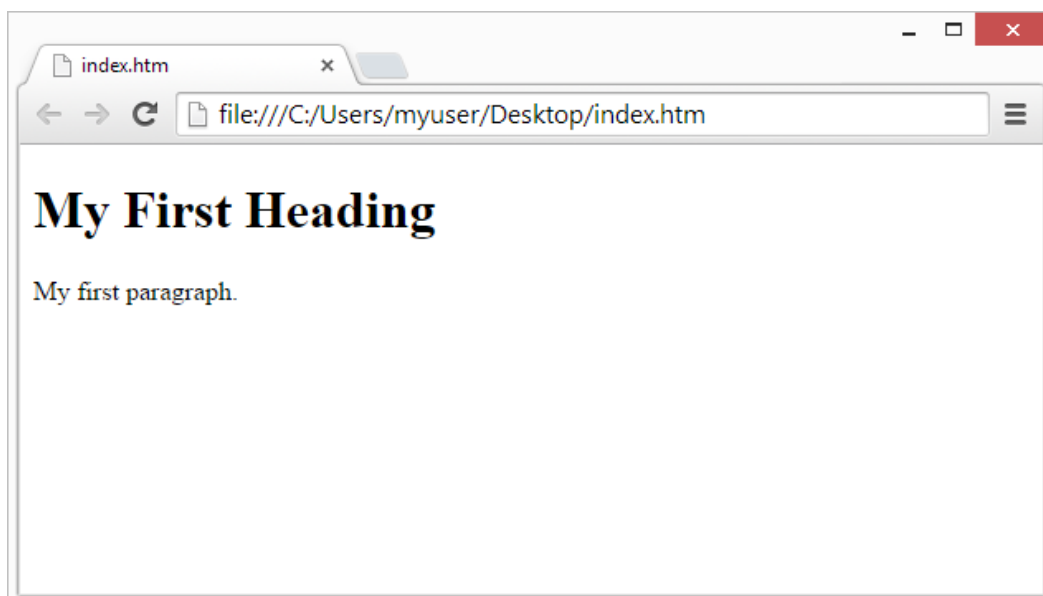
- HTML tags normally come in pairs like <p> and </p>
- The first tag in a pair is the start tag, the second tag is the end tag
- The end tag is written like the start tag, but with a **forward slash** inserted before the tag name

Tip: The start tag is also called the opening tag, and the end tag the closing tag.

Web Browsers

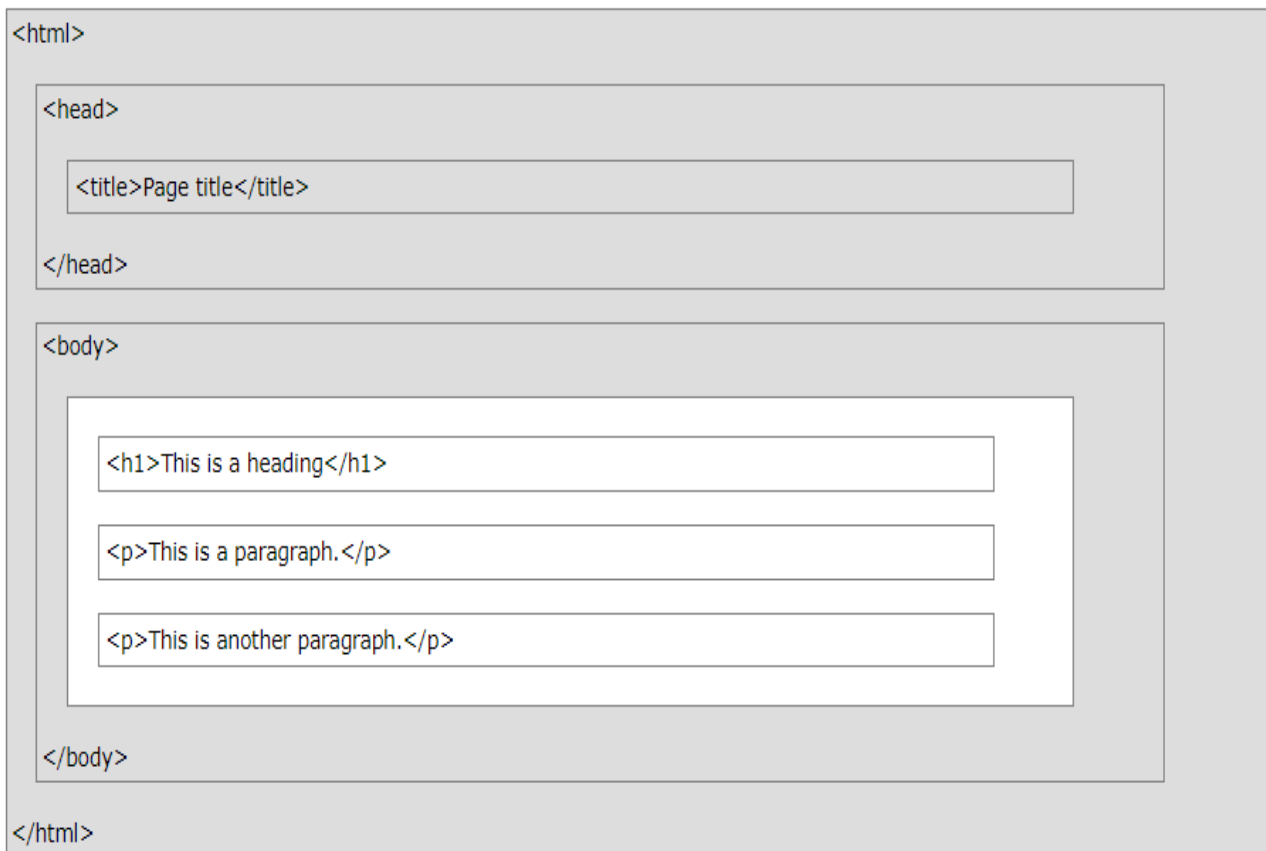
The purpose of a web browser (Chrome, Edge, Firefox, Safari) is to read HTML documents and display them.

The browser does not display the HTML tags, but uses them to determine how to display the document:



HTML Page Structure

Below is a visualization of an HTML page structure:



Note: Only the content inside the `<body>` section (the white area above) is displayed in a browser.

The `<!DOCTYPE>` Declaration

The `<!DOCTYPE>` declaration represents the document type, and helps browsers to display web pages correctly.

It must only appear once, at the top of the page (before any HTML tags).

The `<!DOCTYPE>` declaration is not case sensitive.

The `<!DOCTYPE>` declaration for HTML5 is: `<!DOCTYPE html>`

HTML Versions

Since the early days of the web, there have been many versions of HTML:

Version	Year
HTML	1991
HTML 2.0	1995
HTML 3.2	1997
HTML 4.01	1999
XHTML	2000
HTML5	2014

HTML Elements

An HTML element usually consists of a **start** tag and an **end** tag, with the content inserted in between:

`<tagname>`Content goes here...`</tagname>`

The HTML **element** is everything from the start tag to the end tag:

`<p>`My first paragraph.`</p>`

Start tag	Element content	End tag
<code><h1></code>	My First Heading	<code></h1></code>
<code><p></code>	My first paragraph.	<code></p></code>
<code>
</code>	-	-

HTML elements with no content are called empty elements. Empty elements do not have an end tag, such as the `
` element (which indicates a line break).

Nested HTML Elements

HTML elements can be nested (elements can contain elements).

All HTML documents consist of nested HTML elements.

This example contains four HTML elements:

Example

```
<!DOCTYPE html>
<html>
<body>
<h1>My First Heading</h1>
<p>My first paragraph.</p>
</body>
</html>
```

Example Explanation

The `<html>` element defines the **whole document**.

It has a **start** tag `<html>` and an **end** tag `</html>`.

Inside the `<html>` element is the `<body>` element.

```
<html>
<body>
<h1>My First Heading</h1>
<p>My first paragraph.</p>
</body>
</html>
```

The `<body>` element defines the **document body**.

It has a **start** tag `<body>` and an **end** tag `</body>`.

Inside the `<body>` element is two other HTML elements: `<h1>` and `<p>`.

```
<body>
<h1>My First Heading</h1>
<p>My first paragraph.</p>
</body>
```

The `<h1>` element defines a **heading**.

It has a **start** tag `<h1>` and an **end** tag `</h1>`.

The element **content** is: My First Heading.

```
<h1>My First Heading</h1>
```

The `<p>` element defines a **paragraph**.

It has a **start** tag `<p>` and an **end** tag `</p>`.

The element **content** is: My first paragraph.

```
<p>My first paragraph.</p>
```

Do Not Forget the End Tag

Some HTML elements will display correctly, even if you forget the end tag:

Example

```
<html>
<body>
<p>This is a paragraph
<p>This is a paragraph
</body>
</html>
```

The example above works in all browsers, because the closing tag is considered optional.

Never rely on this. It might produce unexpected results and/or errors if you forget the end tag.

Empty HTML Elements

HTML elements with no content are called empty elements.

`
` is an empty element without a closing tag (the `
` tag defines a line break):

Example

```
<p>This is a <br> paragraph with a line break.</p>
```

Empty elements can be "closed" in the opening tag like this: `
`.

HTML5 does not require empty elements to be closed. But if you want stricter validation, or if you need to make your document readable by XML parsers, you must close all HTML elements properly.

Tip: HTML tags are not case sensitive: `<P>` means the same as `<p>`.

HTML Headings

Headings

Heading 1

Heading 2

Heading 3

Heading 4

Heading 5

Heading 6

HTML Headings

Headings are defined with the `<h1>` to `<h6>` tags.

`<h1>` defines the most important heading. `<h6>` defines the least important heading.

Example

```
<h1>Heading 1</h1>
```

```
<h2>Heading 2</h2>
```

```
<h3>Heading 3</h3>
```

```
<h4>Heading 4</h4>
```

```
<h5>Heading 5</h5>
```

```
<h6>Heading 6</h6>
```

Note: Browsers automatically add some white space (a margin) before and after a heading.

Headings are Important

Search engines use the headings to index the structure and content of your web pages.

Users often skim a page by its headings. It is important to use headings to show the document structure.

`<h1>` headings should be used for main headings, followed by `<h2>` headings, then the less important `<h3>`, and so on.

Note: Use HTML headings for headings only. Don't use headings to make text **BIG** or **bold**.

Bigger Headings

Each HTML heading has a default size. However, you can specify the size for any heading with the `style` attribute, using the CSS `font-size` property:

Example

```
<h1 style="font-size:60px;">Heading 1</h1>
```

HTML Horizontal Rules

The `<hr>` tag defines a thematic break in an HTML page, and is most often displayed as a horizontal rule.

The `<hr>` element is used to separate content (or define a change) in an HTML page:

Example

```
<h1>This is heading 1</h1>
```

```
<p>This is some text.</p>
```

```
<hr>
```

```
<h2>This is heading 2</h2>
```

```
<p>This is some other text.</p>
```

```
<hr>
```

The HTML <head> Element

The HTML `<head>` element is a container for metadata. HTML metadata is data about the HTML document. Metadata is not displayed.

The `<head>` element is placed between the `<html>` tag and the `<body>` tag:

Example

```
<!DOCTYPE html>
<html>
<head>
  <title>My First HTML</title>
  <meta charset="UTF-8">
</head>
<body>
```

Note: Metadata typically define the document title, character set, styles, scripts, and other meta information.

HTML Tag Reference

The following table contains additional information about these tags and their attributes.

Tag	Description
<code><html></code>	Defines the root of an HTML document
<code><body></code>	Defines the document's body
<code><head></code>	A container for all the head elements (title, scripts, styles, meta information, and more)
<code><h1></code> to <code><h6></code>	Defines HTML headings
<code><hr></code>	Defines a thematic change in the content

HTML Paragraphs

The HTML `<p>` element defines a **paragraph**:

Example

```
<p>This is a paragraph.</p>  
<p>This is another paragraph.</p>
```

Note: Browsers automatically add some white space (a margin) before and after a paragraph.

HTML Display

You cannot be sure how HTML will be displayed.

Large or small screens, and resized windows will create different results.

With HTML, you cannot change the output by adding extra spaces or extra lines in your HTML code.

The browser will remove any extra spaces and extra lines when the page is displayed:

Example

```
<p>
```

This paragraph
contains a lot of lines
in the source code,
but the browser
ignores it.

```
</p>
```

```
<p>
```

This paragraph
contains a lot of spaces
in the source code,
but the browser
ignores it.

```
</p>
```

Don't Forget the End Tag

Most browsers will display HTML correctly even if you forget the end tag:

Example

`<p>`This is a paragraph.

`<p>`This is another paragraph.

The example above will work in most browsers, but do not rely on it.

Note: Dropping the end tag can produce unexpected results or errors.

HTML Line Breaks

The HTML `
` element defines a **line break**.

Use `
` if you want a line break (a new line) without starting a new paragraph:

Example

`<p>`This is`
`a paragraph`
`with line breaks.`</p>`

The `
` tag is an empty tag, which means that it has no end tag.

The Poem Problem

This poem will display on a single line:

Example

`<p>`
My Bonnie lies over the ocean.

My Bonnie lies over the sea.

My Bonnie lies over the ocean.

Oh, bring back my Bonnie to me.
`</p>`

The HTML `<pre>` Element

The HTML `<pre>` element defines preformatted text.

The text inside a `<pre>` element is displayed in a fixed-width font (usually Courier), and it preserves both spaces and line breaks:

Example

```
<pre>
My Bonnie lies over the ocean.

My Bonnie lies over the sea.

My Bonnie lies over the ocean.

Oh, bring back my Bonnie to me.
</pre>
```

HTML Tag Reference

This table contains additional information about HTML elements and their attributes.

Tag	Description
<code><p></code>	Defines a paragraph
<code>
</code>	Inserts a single line break
<code><pre></code>	Defines pre-formatted text

HTML Block and Inline Elements

Every HTML element has a default display value depending on what type of element it is.

The two display values are: block and inline.

- Block-level Elements

A block-level element always starts on a new line and takes up the full width available (stretches out to the left and right as far as it can).

The <div> element is a block-level element.

Example

```
<div>Hello World</div>
```

Block level elements in HTML:

```
<address>  
<article>  
<aside>  
<blockquote>  
<canvas>  
<dd>  
<div>  
<dl>  
<dt>  
<fieldset>  
<figcaption>  
<figure>  
<footer>  
<form>  
<h1>-<h6>  
<header>  
<hr>
```


<main>
<nav>
<noscript>

<p>
<pre>
<section>
<table>
<tfoot>

<video>

- Inline Elements

An inline element does not start on a new line and only takes up as much width as necessary.

This is an inline element inside a paragraph.

Example

Hello World

Inline elements in HTML:

<a>
<abbr>
<acronym>

<bdo>
<big>

<button>
<cite>
<code>

<dfn>

<i>

<input>
<kbd>
<label>
<map>
<object>
<output>
<q>
<samp>
<script>
<select>
<small>

<sub>
<sup>
<textarea>
<time>
<tt>
<var>

The <div> Element

The `<div>` element is often used as a container for other HTML elements.

The `<div>` element has no required attributes, but `style`, `class` and `id` are common.

When used together with CSS, the `<div>` element can be used to style blocks of content:

Example

```
<div style="background-color:black; color:white; padding:20px;">
  <h2>London</h2>
  <p>London is the capital city of England. It is the most populous city in
the United Kingdom, with a metropolitan area of over 13 million
inhabitants.</p>
</div>
```

The Element

The `` element is often used as a container for some text.

The `` element has no required attributes, but `style`, `class` and `id` are common.

When used together with CSS, the `` element can be used to style parts of the text:

Example

```
<h1>My <span style="color:red">Important</span> Heading</h1>
```

HTML Grouping Tags

Tag	Description
<code><div></code>	Defines a section in a document (block-level)
<code></code>	Defines a section in a document (inline)

HTML Text Formatting

Text Formatting

This text is bold

This text is italic

This is _{subscript} and ^{superscript}

HTML Formatting Elements

HTML defines special **elements** for defining text with a special **meaning**.

HTML uses elements like `` and `<i>` for formatting output, like **bold** or *italic* text.

Formatting elements were designed to display special types of text:

- `` - Bold text
- `` - Important text
- `<i>` - Italic text
- `` - Emphasized text
- `<mark>` - Marked text
- `<small>` - Small text
- `` - Deleted text
- `<ins>` - Inserted text
- `<sub>` - Subscript text
- `<sup>` - Superscript text

HTML `` and `` Elements

The HTML `` element defines **bold** text, without any extra importance.

Example

``This text is bold``

The HTML `` element defines **strong** text, with added semantic "strong" importance.

Example

```
<strong>This text is strong</strong>
```

HTML `<i>` and `` Elements

The HTML `<i>` element defines *italic* text, without any extra importance.

Example

```
<i>This text is italic</i>
```

The HTML `` element defines *emphasized* text, with added semantic importance.

Example

```
<em>This text is emphasized</em>
```

Note: Browsers display `` as ``, and `` as `<i>`. However, there is a difference in the meaning of these tags: `` and `<i>` defines bold and italic text, but `` and `` means that the text is "important".

HTML `<small>` Element

The HTML `<small>` element defines smaller text:

Example

```
<h2>HTML <small>Small</small> Formatting</h2>
```

HTML `<mark>` Element

The HTML `<mark>` element defines marked/highlighted text:

Example

```
<h2>HTML <mark>Marked</mark> Formatting</h2>
```

HTML Element

The HTML `` element defines deleted/removed text.

Example

```
<p>My favorite color is <del>blue</del> red.</p>
```

HTML <ins> Element

The HTML `<ins>` element defines inserted/added text.

Example

```
<p>My favorite <ins>color</ins> is red.</p>
```

HTML <sub> Element

The HTML `<sub>` element defines subscripted text.

Example

```
<p>This is <sub>subscripted</sub> text.</p>
```

HTML <sup> Element

The HTML `<sup>` element defines superscripted text.

Example

```
<p>This is <sup>superscripted</sup> text.</p>
```

HTML Text Formatting Elements

Tag	Description
<u></u>	Defines bold text
<u></u>	Defines emphasized text
<u><i></u>	Defines italic text
<u><small></u>	Defines smaller text
<u></u>	Defines important text
<u><sub></u>	Defines subscripted text
<u><sup></u>	Defines superscripted text
<u><ins></u>	Defines inserted text
<u></u>	Defines deleted text
<u><mark></u>	Defines marked/highlighted text

HTML Quotation and Citation Elements

Quotation

Here is a quote from WWF's website:

For 50 years, WWF has been protecting the future of nature. The world's leading conservation organization, WWF works in 100 countries and is supported by 1.2 million members in the United States and close to 5 million globally.

HTML <q> for Short Quotations

The HTML `<q>` element defines a short quotation. Browsers usually insert quotation marks around the `<q>` element.

Example

```
<p>WWF's goal is to: <q>Build a future where people live in harmony with nature.</q></p>
```

HTML <blockquote> for Quotations

The HTML `<blockquote>` element defines a section that is quoted from another source.

Browsers usually indent `<blockquote>` elements.

Example

```
<p>Here is a quote from WWF's website:</p>
<blockquote cite="http://www.worldwildlife.org/who/index.html">
For 50 years, WWF has been protecting the future of nature.
The world's leading conservation organization,
WWF works in 100 countries and is supported by
1.2 million members in the United States and
close to 5 million globally.
</blockquote>
```

HTML <abbr> for Abbreviations

The HTML `<abbr>` element defines an abbreviation or an acronym.

Marking abbreviations can give useful information to browsers, translation systems and search-engines.

Example

```
<p>The <abbr title="World Health Organization">WHO</abbr> was founded in 1948.</p>
```

HTML <address> for Contact Information

The HTML `<address>` element defines contact information (author/owner) of a document or an article.

The `<address>` element is usually displayed in italic. Most browsers will add a line break before and after the element.

Example

```
<address>  
Written by John Doe.<br>  
Visit us at:<br>  
Example.com<br>  
Box 564, Disneyland<br>  
USA  
</address>
```

HTML <cite> for Work Title

The HTML `<cite>` element defines the title of a work.

Browsers usually display `<cite>` elements in italic.

Example

```
<p><cite>The Scream</cite> by Edvard Munch. Painted in 1893.</p>
```

HTML <bdo> for Bi-Directional Override

The HTML `<bdo>` element defines bi-directional override.

The `<bdo>` element is used to override the current text direction:

Example

`<bdo dir="rtl">`This text will be written from right to left`</bdo>`

HTML Quotation and Citation Elements

Tag	Description
<code><abbr></code>	Defines an abbreviation or acronym
<code><address></code>	Defines contact information for the author/owner of a document
<code><bdo></code>	Defines the text direction
<code><blockquote></code>	Defines a section that is quoted from another source
<code><cite></code>	Defines the title of a work
<code><q></code>	Defines a short inline quotation

HTML Comments

Comment tags are used to insert comments in the HTML source code.

HTML Comment Tags

You can add comments to your HTML source by using the following syntax:

```
<!-- Write your comments here -->
```

Notice that there is an exclamation point (!) in the opening tag, but not in the closing tag.

Note: Comments are not displayed by the browser, but they can help document your HTML source code.

With comments you can place notifications and reminders in your HTML:

Example

```
<!-- This is a comment -->
```

```
<p>This is a paragraph.</p>
```

```
<!-- Remember to add more information here -->
```

Comments are also great for debugging HTML, because you can comment out HTML lines of code, one at a time, to search for errors:

Example

```
<!-- Do not display this image at the moment
```

```

```

```
-->
```

HTML Styles

Example

I am Red

I am Blue

I am Big

The HTML Style Attribute

Setting the style of an HTML element, can be done with the **style** attribute.

The HTML **style** attribute has the following **syntax**:

```
<tagname style="property:value;">
```

The **property** is a CSS property. The **value** is a CSS value.

Background Color

The CSS **background-color** property defines the background color for an HTML element.

This example sets the background color for a page to powderblue:

Example

```
<body style="background-color:powderblue;">  
<h1>This is a heading</h1>  
<p>This is a paragraph.</p>  
</body>
```

Text Color

The CSS `color` property defines the text color for an HTML element:

Example

```
<h1 style="color:blue;">This is a heading</h1>  
<p style="color:red;">This is a paragraph.</p>
```

Fonts

The CSS `font-family` property defines the font to be used for an HTML element:

Example

```
<h1 style="font-family:verdana;">This is a heading</h1>  
<p style="font-family:courier;">This is a paragraph.</p>
```

Text Size

The CSS `font-size` property defines the text size for an HTML element:

Example

```
<h1 style="font-size:300%;">This is a heading</h1>  
<p style="font-size:160%;">This is a paragraph.</p>
```

Text Alignment

The CSS `text-align` property defines the horizontal text alignment for an HTML element:

Example

```
<h1 style="text-align:center;">Centered Heading</h1>  
<p style="text-align:center;">Centered paragraph.</p>
```

Chapter Summary

- Use the `style` attribute for styling HTML elements
- Use `background-color` for background color
- Use `color` for text colors
- Use `font-family` for text fonts
- Use `font-size` for text sizes
- Use `text-align` for text alignment

HTML Colors

HTML colors are specified using predefined color names, or RGB, HEX, HSL, RGBA, HSLA values.

- Color Names

In HTML, a color can be specified by using a color name:



HTML supports [140 standard color names](#).

Background Color

You can set the background color for HTML elements:

Hello World

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

Example

```
<h1 style="background-color:DodgerBlue;">Hello World</h1>  
<p style="background-color:Tomato;">Lorem ipsum...</p>
```

Text Color

You can set the color of text:

Hello World

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

Example

```
<h1 style="color:Tomato;">Hello World</h1>  
<p style="color:DodgerBlue;">Lorem ipsum...</p>  
<p style="color:MediumSeaGreen;">Ut wisi enim...</p>
```

Border Color

You can set the color of borders:

Hello World

Hello World

Hello World

Example

```
<h1 style="border:2px solid Tomato;">Hello World</h1>
```

```
<h1 style="border:2px solid DodgerBlue;">Hello World</h1>
```

```
<h1 style="border:2px solid Violet;">Hello World</h1>
```

- Color Values

In HTML, colors can also be specified using RGB values, HEX values, HSL values, RGBA values, and HSLA values:

Same as color name "Tomato":

rgb(255, 99, 71)

#ff6347

hsl(9, 100%, 64%)

Same as color name "Tomato", but 50% transparent:

Example

```
<h1 style="background-color:rgb(255, 99, 71);">...</h1>
```

```
<h1 style="background-color:#ff6347;">...</h1>
```

```
<h1 style="background-color:hsl(9, 100%, 64%;">...</h1>
```

```
<h1 style="background-color:rgba(255, 99, 71, 0.5);">...</h1>
```

```
<h1 style="background-color:hsla(9, 100%, 64%, 0.5);">...</h1>
```

1. RGB Value

In HTML, a color can be specified as an RGB value, using this formula:

rgb(*red, green, blue*)

Each parameter (red, green, and blue) defines the intensity of the color between 0 and 255.

For example, rgb(255, 0, 0) is displayed as red, because red is set to its highest value (255) and the others are set to 0.

To display black, set all color parameters to 0, like this: rgb(0, 0, 0).

To display white, set all color parameters to 255, like this: rgb(255, 255, 255).

Experiment by mixing the RGB values below:

rgb(255, 99, 71)

RED= 255 , GREEN= 99 , BLUE= 71

Example

rgb(255, 0, 0)

rgb(0, 0, 255)

rgb(60, 179, 113)

rgb(238, 130, 238)

rgb(255, 165, 0)

rgb(106, 90, 205)

Shades of gray are often defined using equal values for all the 3 light sources:

Example

rgb(0, 0, 0)

rgb(60, 60, 60)

rgb(120, 120, 120)

rgb(180, 180, 180)

rgb(240, 240, 240)

rgb(255, 255, 255)

2. HEX Value

In HTML, a color can be specified using a hexadecimal value in the form:

#rrggbb

Where rr (red), gg (green) and bb (blue) are hexadecimal values between 00 and ff (same as decimal 0-255).

For example, #ff0000 is displayed as red, because red is set to its highest value (ff) and the others are set to the lowest value (00).

Example

#ff0000

#0000ff

#3cb371

#ee82ee

#ffa500

#6a5acd

Shades of gray are often defined using equal values for all the 3 light sources:

Example

#000000

#3c3c3c

#787878

#b4b4b4

#f0f0f0

#ffffff

3. HSL Value

In HTML, a color can be specified using hue, saturation, and lightness (HSL) in the form:

hsl(hue, saturation, lightness)

Hue is a degree on the color wheel from 0 to 360. 0 is red, 120 is green, and 240 is blue.

Saturation is a percentage value, 0% means a shade of gray, and 100% is the full color.

Lightness is also a percentage, 0% is black, 50% is neither light or dark, 100% is white

Example

hsl(0, 100%, 50%)

hsl(240, 100%, 50%)

hsl(147, 50%, 47%)

hsl(300, 76%, 72%)

hsl(39, 100%, 50%)

hsl(248, 53%, 58%)

Saturation

Saturation can be described as the intensity of a color.

100% is pure color, no shades of gray

50% is 50% gray, but you can still see the color.

0% is completely gray, you can no longer see the color.

Example

hsl(0, 100%, 50%)

hsl(0, 80%, 50%)

hsl(0, 60%, 50%)

hsl(0, 40%, 50%)

hsl(0, 20%, 50%)

hsl(0, 0%, 50%)

Lightness

The lightness of a color can be described as how much light you want to give the color, where 0% means no light (black), 50% means 50% light (neither dark nor light) 100% means full lightness (white).

Example

hsl(0, 100%, 0%)

hsl(0, 100%, 25%)

hsl(0, 100%, 50%)

hsl(0, 100%, 75%)

hsl(0, 100%, 90%)

hsl(0, 100%, 100%)

Shades of gray are often defined by setting the hue and saturation to 0, and adjust the lightness from 0% to 100% to get darker/lighter shades:

Example

hsl(0, 0%, 0%)

hsl(0, 0%, 24%)

hsl(0, 0%, 47%)

hsl(0, 0%, 71%)

hsl(0, 0%, 94%)

hsl(0, 0%, 100%)

4. RGBA Value

RGBA color values are an extension of RGB color values with an alpha channel - which specifies the opacity for a color.

An RGBA color value is specified with:

`rgba(red, green, blue, alpha)`

The alpha parameter is a number between 0.0 (fully transparent) and 1.0 (not transparent at all):

Example

`rgba(255, 99, 71, 0)`

`rgba(255, 99, 71, 0.2)`

`rgba(255, 99, 71, 1)`

5. HSLA Value

HSLA color values are an extension of HSL color values with an alpha channel - which specifies the opacity for a color.

An HSLA color value is specified with:

`hsla(hue, saturation, lightness, alpha)`

The alpha parameter is a number between 0.0 (fully transparent) and 1.0 (not transparent at all):

Example

`hsla(9, 100%, 64%, 0)`

`hsla(9, 100%, 64%, 0.2)`

`hsla(9, 100%, 64%, 1)`

HTML Styles - CSS

CSS = Styles and Colors

M a n i p u l a t e T e x t

C o l o r s , **B o x e s**

Styling HTML with CSS

CSS stands for **C**ascading **S**tyle **S**heets.

CSS describes **how HTML elements are to be displayed on screen, paper, or in other media.**

CSS **saves a lot of work.** It can control the layout of multiple web pages all at once.

CSS can be added to HTML elements in 3 ways:

- **Inline** - by using the style attribute in HTML elements
- **Internal** - by using a `<style>` element in the `<head>` section
- **External** - by using an external CSS file

The most common way to add CSS, is to keep the styles in separate CSS files. However, here we will use inline and internal styling, because this is easier to demonstrate, and easier for you to try it yourself.

- Inline CSS

An inline CSS is used to apply a unique style to a single HTML element.

An inline CSS uses the style attribute of an HTML element.

This example sets the text color of the `<h1>` element to blue:

Example

```
<h1 style="color:blue;">This is a Blue Heading</h1>
```

- Internal CSS

An internal CSS is used to define a style for a single HTML page.

An internal CSS is defined in the `<head>` section of an HTML page, within a `<style>` element:

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
body {background-color: powderblue;}
h1 {color: blue;}
p {color: red;}
</style>
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

- External CSS

An external style sheet is used to define the style for many HTML pages.

With an external style sheet, you can change the look of an entire web site, by changing one file!

To use an external style sheet, add a link to it in the `<head>` section of the HTML page:

Example

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="styles.css">
</head>
</html>
```

```
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

An external style sheet can be written in any text editor. The file must not contain any HTML code, and must be saved with a .css extension.

Here is how the "styles.css" looks:

```
body {
  background-color: powderblue;
}
h1 {
  color: blue;
}
p {
  color: red;
}
```

CSS Fonts

The CSS `color` property defines the text color to be used.

The CSS `font-family` property defines the font to be used.

The CSS `font-size` property defines the text size to be used.

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
  color: blue;
  font-family: verdana;
  font-size: 300%;
}
p {
  color: red;
  font-family: courier;
  font-size: 160%;
```

```
}  
</style>  
</head>  
<body>  
<h1>This is a heading</h1>  
<p>This is a paragraph.</p>  
</body>  
</html>
```

CSS Border

The CSS **border** property defines a border around an HTML element:

Example

```
p {  
  border: 1px solid powderblue;  
}
```

CSS Padding

The CSS **padding** property defines a padding (space) between the text and the border:

Example

```
p {  
  border: 1px solid powderblue;  
  padding: 30px;  
}
```

CSS Margin

The CSS **margin** property defines a margin (space) outside the border:

Example

```
p {  
  border: 1px solid powderblue;  
  margin: 50px;  
}
```

The id Attribute

To define a specific style for one special element, add an **id** attribute to the element:

```
<p id="p01">I am different</p>
```

then define a style for the element with the specific id:

Example

```
#p01 {  
  color: blue;  
}
```

Note: The id of an element should be unique within a page, so the id selector is used to select one unique element!

The class Attribute

To define a style for special types of elements, add a **class** attribute to the element:

```
<p class="error">I am different</p>
```

then define a style for the elements with the specific class:

Example

```
p.error {  
  color: red;  
}
```

External References

External style sheets can be referenced with a full URL or with a path relative to the current web page.

This example uses a full URL to link to a style sheet:

Example

```
<link rel="stylesheet" href="www.mywebsite/styles.css">
```

This example links to a style sheet located in the html folder on the current web site:

Example

```
<link rel="stylesheet" href="/html/styles.css">
```

This example links to a style sheet located in the same folder as the current page:

Example

```
<link rel="stylesheet" href="styles.css">
```

Chapter Summary

- Use the HTML `style` attribute for inline styling
- Use the HTML `<style>` element to define internal CSS
- Use the HTML `<link>` element to refer to an external CSS file
- Use the HTML `<head>` element to store `<style>` and `<link>` elements
- Use the CSS `color` property for text colors
- Use the CSS `font-family` property for text fonts
- Use the CSS `font-size` property for text sizes
- Use the CSS `border` property for borders
- Use the CSS `padding` property for space inside the border
- Use the CSS `margin` property for space outside the border

HTML Style Tags

Tag	Description
<code><style></code>	Defines style information for an HTML document
<code><link></code>	Defines a link between a document and an external resource

HTML Links

Links are found in nearly all web pages. Links allow users to click their way from page to page.

HTML Links - Hyperlinks

HTML links are hyperlinks.

You can click on a link and jump to another document.

When you move the mouse over a link, the mouse arrow will turn into a little hand.

Note: A link does not have to be text. It can be an image or any other HTML element.

HTML Links - Syntax

Hyperlinks are defined with the HTML `<a>` tag:

```
<a href="url">link text</a>
```

Example

```
<a href="https://www.google.com/html/">Visit our page</a>
```

The `href` attribute specifies the destination address (`https://www.google.com/html/`) of the link.

The **link text** is the visible part (Visit our page).

Clicking on the link text will send you to the specified address.

Note: Without a forward slash at the end of subfolder addresses, you might generate two requests to the server. Many servers will automatically add a forward slash to the end of the address, and then create a new request.

Local Links

The example above used an absolute URL (a full web address).

A local link (link to the same web site) is specified with a relative URL (without https://www....).

Example

```
<a href="html_images.asp">HTML Images</a>
```

HTML Link Colors

By default, a link will appear like this (in all browsers):

- An unvisited link is underlined and blue
- A visited link is underlined and purple
- An active link is underlined and red

You can change the default colors, by using CSS:

Example

```
<style>
a:link {
    color: green;
    background-color: transparent;
    text-decoration: none;
}

a:visited {
    color: pink;
    background-color: transparent;
    text-decoration: none;
}

a:hover {
    color: red;
    background-color: transparent;
    text-decoration: underline;
```

```
}
```

```
a:active {  
  color: yellow;  
  background-color: transparent;  
  text-decoration: underline;  
}  
</style>
```

Links are often styled as buttons, by using CSS:

Example

```
<style>  
a:link, a:visited {  
  background-color: #f44336;  
  color: white;  
  padding: 15px 25px;  
  text-align: center;  
  text-decoration: none;  
  display: inline-block;  
}  
a:hover, a:active {  
  background-color: red;  
}  
</style>
```

HTML Links - The target Attribute

The **target** attribute specifies where to open the linked document.

The **target** attribute can have one of the following values:

- **_blank** - Opens the linked document in a new window or tab
- **_self** - Opens the linked document in the same window/tab as it was clicked (this is default)
- **_parent** - Opens the linked document in the parent frame
- **_top** - Opens the linked document in the full body of the window
- *framename* - Opens the linked document in a named frame

This example will open the linked document in a new browser window/tab:

Example

```
<a href="https://www.google.com/" target="_blank">Visit google </a>
```

Tip: If your webpage is locked in a frame, you can use **target="_top"** to break out of the frame:

Example

```
<a href="https://www.google.com/html/" target="_top">HTML5  
tutorial!</a>
```

HTML Links - Image as Link

It is common to use images as links:

Example

```
<a href="default.asp">  
    
</a>
```

Note: **border:0;** is added to prevent IE9 (and earlier) from displaying a border around the image (when the image is a link).

Link Titles

The **title** attribute specifies extra information about an element. The information is most often shown as a tooltip text when the mouse moves over the element.

Example

```
<a href="https://www.google.com/html/" title="Go to HTML section">
```

```
Visit our HTML Tutorial</a>
```

HTML Links - Create a Bookmark

HTML bookmarks are used to allow readers to jump to specific parts of a Web page.

Bookmarks can be useful if your webpage is very long.

To make a bookmark, you must first create the bookmark, and then add a link to it.

When the link is clicked, the page will scroll to the location with the bookmark.

Example

First, create a bookmark with the **id** attribute:

```
<h2 id="C4">Chapter 4</h2>
```

Then, add a link to the bookmark ("Jump to Chapter 4"), from within the same page:

```
<a href="#C4">Jump to Chapter 4</a>
```

Or, add a link to the bookmark ("Jump to Chapter 4"), from another page:

Example

```
<a href="html_demo.html#C4">Jump to Chapter 4</a>
```

Chapter Summary

- Use the `<a>` element to define a link
- Use the `href` attribute to define the link address
- Use the `target` attribute to define where to open the linked document
- Use the `` element (inside `<a>`) to use an image as a link
- Use the `id` attribute (`id="value"`) to define bookmarks in a page
- Use the `href` attribute (`href="#value"`) to link to the bookmark

HTML Link Tags

Tag	Description
<code><a></code>	Defines a hyperlink

HTML Images

Images can improve the design and the appearance of a web page

Example

```

```

Example

```

```

Example

```

```

HTML Images Syntax

In HTML, images are defined with the `` tag.

The `` tag is empty, it contains attributes only, and does not have a closing tag.

The `src` attribute specifies the URL (web address) of the image:

```

```

The alt Attribute

The `alt` attribute provides an alternate text for an image, if the user for some reason cannot view it (because of slow connection, an error in the `src` attribute, or if the user uses a screen reader).

The value of the `alt` attribute should describe the image:

Example

```

```

If a browser cannot find an image, it will display the value of the `alt` attribute:

Example

```

```

Note: The `alt` attribute is required. A web page will not validate correctly without it.

Image Size - Width and Height

You can use the `style` attribute to specify the width and height of an image.

Example

```

```

Alternatively, you can use the `width` and `height` attributes:

Example

```

```

The `width` and `height` attributes always defines the width and height of the image in pixels.

Note: Always specify the width and height of an image. If width and height are not specified, the page might flicker while the image loads.

Width and Height, or Style?

The `width`, `height`, and `style` attributes are valid in HTML.

However, we suggest using the `style` attribute. It prevents styles sheets from changing the size of images:

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
img {
    width: 100%;
}
</style>
</head>
<body>


</body>
</html>
```

Images in Another Folder

If not specified, the browser expects to find the image in the same folder as the web page.

However, it is common to store images in a sub-folder. You must then include the folder name in the `src` attribute:

Example

```

```

Images on Another Server

Some web sites store their images on image servers.

Actually, you can access images from any web address in the world:

Example

```

```

Animated Images

HTML allows animated GIFs:

Example

```

```

Image as a Link

To use an image as a link, put the `` tag inside the `<a>` tag:

Example

```
<a href="default.asp">  
    
</a>
```

Note: `border:0;` is added to prevent IE9 (and earlier) from displaying a border around the image (when the image is a link).

Image Floating

Use the CSS `float` property to let the image float to the right or to the left of a text:

Example

```
<p>
```

The image will float to the right of the text.</p>

```
<p>
```

The image will float to the left of the text.</p>

Chapter Summary

- Use the HTML `` element to define an image
- Use the HTML `src` attribute to define the URL of the image
- Use the HTML `alt` attribute to define an alternate text for an image, if it cannot be displayed
- Use the HTML `width` and `height` attributes to define the size of the image
- Use the CSS `width` and `height` properties to define the size of the image (alternatively)
- Use the CSS `float` property to let the image float
- Use the HTML `<map>` element to define an image-map
- Use the HTML `<area>` element to define the clickable areas in the image-map
- Use the HTML ``'s element `usemap` attribute to point to an image-map
- Use the HTML `<picture>` element to show different images for different devices

Note: Loading images takes time. Large images can slow down your page. Use images carefully.

HTML Image Tags

Tag	Description
<code></code>	Defines an image
<code><map></code>	Defines an image-map
<code><area></code>	Defines a clickable area inside an image-map
<code><picture></code>	Defines a container for multiple image resources

HTML Table

HTML Table Example

Company	Contact	Country
Alfreds Futterkiste	Maria Anders	Germany
Centro comercial Moctezuma	Francisco Chang	Mexico
Ernst Handel	Roland Mendel	Austria
Island Trading	Helen Bennett	UK
Laughing Bacchus Winecellars	Yoshi Tannamuri	Canada
Magazzini Alimentari Riuniti	Giovanni Rovelli	Italy

Defining an HTML Table

An HTML table is defined with the `<table>` tag.

Each table row is defined with the `<tr>` tag. A table header is defined with the `<th>` tag. By default, table headings are bold and centered. A table data/cell is defined with the `<td>` tag.

Example

```
<table style="width:100%">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
</table>
```

```
<td>Eve</td>
<td>Jackson</td>
<td>94</td>
</tr>
</table>
```

Note: The `<td>` elements are the data containers of the table. They can contain all sorts of HTML elements; text, images, lists, other tables, etc.

HTML Table - Adding a Border

If you do not specify a border for the table, it will be displayed without borders.

A border is set using the CSS `border` property:

Example

```
table, th, td {
    border: 1px solid black;
}
```

Remember to define borders for both the table and the table cells.

HTML Table - Collapsed Borders

If you want the borders to collapse into one border, add the CSS `border-collapse` property:

Example

```
table, th, td {
    border: 1px solid black;
    border-collapse: collapse;
}
```

HTML Table - Adding Cell Padding

Cell padding specifies the space between the cell content and its borders.

If you do not specify a padding, the table cells will be displayed without padding.

To set the padding, use the CSS `padding` property:

Example

```
th, td {  
    padding: 15px;  
}
```

HTML Table - Left-align Headings

By default, table headings are bold and centered.

To left-align the table headings, use the CSS `text-align` property:

Example

```
th {  
    text-align: left;  
}
```

HTML Table - Adding Border Spacing

Border spacing specifies the space between the cells.

To set the border spacing for a table, use the CSS `border-spacing` property:

Example

```
table {  
    border-spacing: 5px;  
}
```

Note: If the table has collapsed borders, `border-spacing` has no effect.

HTML Table - Cells that Span Many Columns

To make a cell span more than one column, use the `colspan` attribute:

Example

```
<table style="width:100%">
  <tr>
    <th>Name</th>
    <th colspan="2">Telephone</th>
  </tr>
  <tr>
    <td>Bill Gates</td>
    <td>55577854</td>
    <td>55577855</td>
  </tr>
</table>
```

HTML Table - Cells that Span Many Rows

To make a cell span more than one row, use the `rowspan` attribute:

Example

```
<table style="width:100%">
  <tr>
    <th>Name:</th>
    <td>Bill Gates</td>
  </tr>
  <tr>
    <th rowspan="2">Telephone:</th>
    <td>55577854</td>
  </tr>
  <tr>
    <td>55577855</td>
  </tr>
</table>
```

HTML Table - Adding a Caption

To add a caption to a table, use the `<caption>` tag:

Example

```
<table style="width:100%">
  <caption>Monthly savings</caption>
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>$100</td>
  </tr>
  <tr>
    <td>February</td>
    <td>$50</td>
  </tr>
</table>
```

Note: The `<caption>` tag must be inserted immediately after the `<table>` tag.

A Special Style for One Table

To define a special style for a special table, add an `id` attribute to the table:

Example

```
<table id="t01">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
  </tr>
```

```
<tr>
  <td>Eve</td>
  <td>Jackson</td>
  <td>94</td>
</tr>
</table>
```

Now you can define a special style for this table:

```
table#t01 {
  width: 100%;
  background-color: #f1f1c1;
}
```

And add more styles:

```
table#t01 tr:nth-child(even) {
  background-color: #eee;
}
table#t01 tr:nth-child(odd) {
  background-color: #fff;
}
table#t01 th {
  color: white;
  background-color: black;
}
```

Chapter Summary

- Use the HTML `<table>` element to define a table
- Use the HTML `<tr>` element to define a table row
- Use the HTML `<td>` element to define a table data
- Use the HTML `<th>` element to define a table heading
- Use the HTML `<caption>` element to define a table caption
- Use the CSS `border` property to define a border

- Use the CSS `border-collapse` property to collapse cell borders
 - Use the CSS `padding` property to add padding to cells
 - Use the CSS `text-align` property to align cell text
 - Use the CSS `border-spacing` property to set the spacing between cells
 - Use the `colspan` attribute to make a cell span many columns
 - Use the `rowspan` attribute to make a cell span many rows
 - Use the `id` attribute to uniquely define one table
-

HTML Table Tags

Tag	Description
<code><table></code>	Defines a table
<code><th></code>	Defines a header cell in a table
<code><tr></code>	Defines a row in a table
<code><td></code>	Defines a cell in a table
<code><caption></code>	Defines a table caption
<code><colgroup></code>	Specifies a group of one or more columns in a table for formatting
<code><col></code>	Specifies column properties for each column within a <code><colgroup></code> element
<code><thead></code>	Groups the header content in a table
<code><tbody></code>	Groups the body content in a table
<code><tfoot></code>	Groups the footer content in a table

HTML Lists

HTML List Example

An Unordered List:

- Item
- Item
- Item
- Item

An Ordered List:

1. First item
2. Second item
3. Third item
4. Fourth item

Unordered HTML List

An unordered list starts with the [``](#) tag. Each list item starts with the [``](#) tag.

The list items will be marked with bullets (small black circles) by default:

Example

```
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

Unordered HTML List - Choose List Item Marker

The CSS `list-style-type` property is used to define the style of the list item marker:

Value	Description
disc	Sets the list item marker to a bullet (default)
circle	Sets the list item marker to a circle
square	Sets the list item marker to a square
none	The list items will not be marked

Example - Disc

```
<ul style="list-style-type:disc;">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

Example - Circle

```
<ul style="list-style-type:circle;">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

Example - Square

```
<ul style="list-style-type:square;">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

Example - None

```
<ul style="list-style-type:none;">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

Ordered HTML List

An ordered list starts with the [](#) tag. Each list item starts with the [](#) tag.

The list items will be marked with numbers by default:

Example

```
<ol>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

Ordered HTML List - The Type Attribute

The **type** attribute of the [](#) tag, defines the type of the list item marker:

Type	Description
type="1"	The list items will be numbered with numbers (default)
type="A"	The list items will be numbered with uppercase letters
type="a"	The list items will be numbered with lowercase letters
type="I"	The list items will be numbered with uppercase roman numbers
type="i"	The list items will be numbered with lowercase roman numbers

Numbers:

```
<ol type="1">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

Uppercase Letters:

```
<ol type="A">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

Lowercase Letters:

```
<ol type="a">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

Uppercase Roman Numbers:

```
<ol type="I">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

Lowercase Roman Numbers:

```
<ol type="i">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

HTML Description Lists

HTML also supports description lists.

A description list is a list of terms, with a description of each term.

The `<dl>` tag defines the description list, the `<dt>` tag defines the term (name), and the `<dd>` tag describes each term:

Example

```
<dl>
  <dt>Coffee</dt>
  <dd>- black hot drink</dd>
  <dt>Milk</dt>
  <dd>- white cold drink</dd>
</dl>
```

Nested HTML Lists

List can be nested (lists inside lists):

Example

```
<ul>
  <li>Coffee</li>
  <li>Tea
    <ul>
      <li>Black tea</li>
      <li>Green tea</li>
    </ul>
  </li>
  <li>Milk</li>
</ul>
```

Note: List items can contain new list, and other HTML elements, like images and links, etc.

Control List Counting

By default, an ordered list will start counting from 1. If you want to start counting from a specified number, you can use the `start` attribute:

Example

```
<ol start="50">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

Horizontal List with CSS

HTML lists can be styled in many different ways with CSS.

One popular way is to style a list horizontally, to create a navigation menu:

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  overflow: hidden;
  background-color: #333333;
}

li {
  float: left;
}

li a {
  display: block;
  color: white;
  text-align: center;
  padding: 16px;
```

```
    text-decoration: none;
}

li a:hover {
    background-color: #111111;
}
</style>
</head>

<body>
<ul>
    <li><a href="#home">Home</a></li>
    <li><a href="#news">News</a></li>
    <li><a href="#contact">Contact</a></li>
    <li><a href="#about">About</a></li>
</ul>
</body>
</html>
```

Chapter Summary

- Use the HTML `` element to define an unordered list
- Use the CSS `list-style-type` property to define the list item marker
- Use the HTML `` element to define an ordered list
- Use the HTML `type` attribute to define the numbering type
- Use the HTML `` element to define a list item
- Use the HTML `<dl>` element to define a description list
- Use the HTML `<dt>` element to define the description term
- Use the HTML `<dd>` element to describe the term in a description list
- Lists can be nested inside lists
- List items can contain other HTML elements
- Use the CSS property `float:left` or `display:inline` to display a list horizontally

HTML List Tags

Tag	Description
<u></u>	Defines an unordered list
<u></u>	Defines an ordered list
<u></u>	Defines a list item
<u><dl></u>	Defines a description list
<u><dt></u>	Defines a term in a description list
<u><dd></u>	Describes the term in a description list