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

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

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

</body>
</html>
```

Example Explained

- 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>

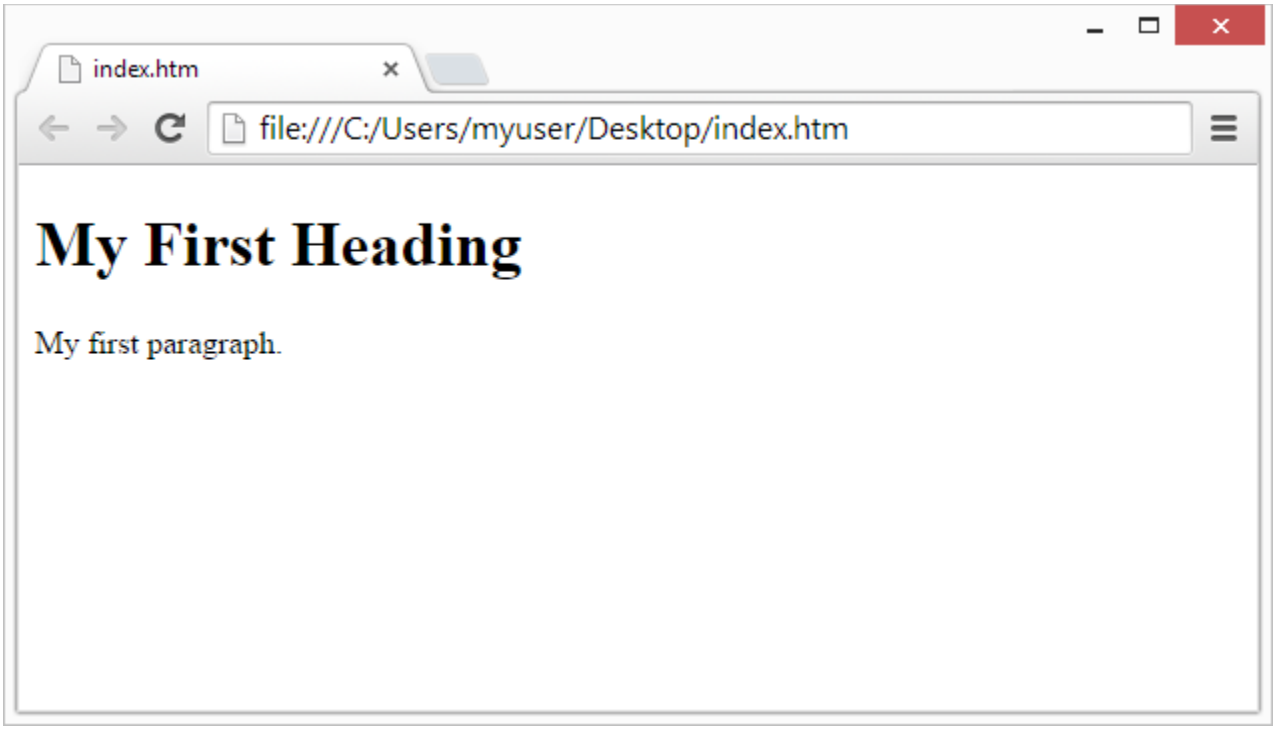
- HTML tags normally come **in pairs** like `<p>` and `</p>`
- The first tag in a pair is the **start tag,** the second tag is the **end tag**
- The end tag is written like the start tag, but with a **forward slash** inserted before the tag name

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:

```
<html>

<head>

<title>Page title</title>

</head>

<body>
```

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

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

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

```
</body>

</html>
```

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 Editors

Write HTML Using Notepad or TextEdit

Web pages can be created and modified by using professional HTML editors.

However, for learning HTML we recommend a simple text editor like Notepad (PC) or TextEdit (Mac).

We believe using a simple text editor is a good way to learn HTML.

Follow the steps below to create your first web page with Notepad or TextEdit.

Step 1: Open Notepad (PC)

Windows 8 or later:

Open the **Start Screen** (the window symbol at the bottom left on your screen). Type **Notepad**.

Windows 7 or earlier:

Open **Start** > **Programs > Accessories > Notepad**

Step 1: Open TextEdit (Mac)

Open **Finder > Applications > TextEdit**

Also change some preferences to get the application to save files correctly. In **Preferences > Format >** choose**"Plain Text"**

Then under "Open and Save", check the box that says "Display HTML files as HTML code instead of formatted text".

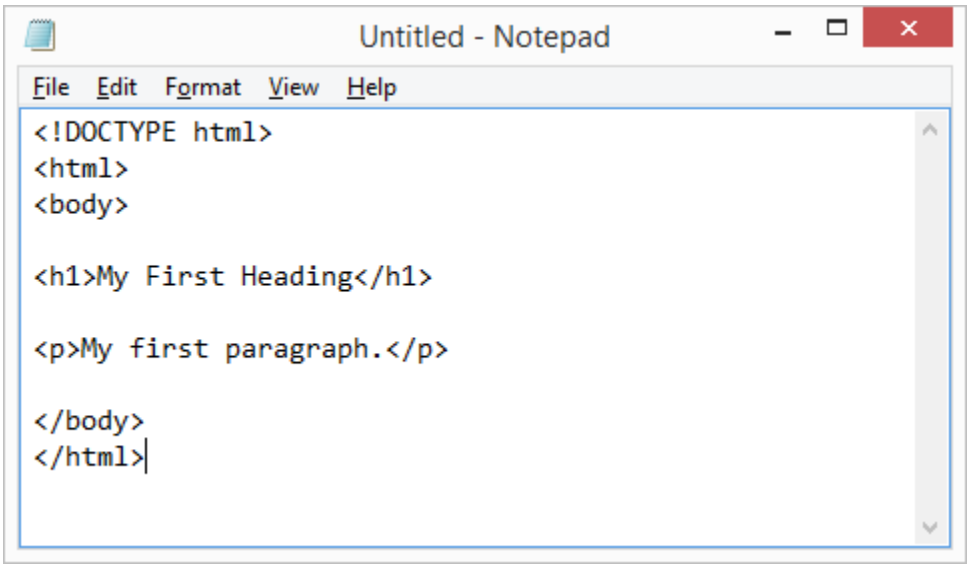
Then open a new document to place the code.

Step 2: Write Some HTML

Write or copy some HTML into Notepad.

```
<!DOCTYPE html>
<html>
```

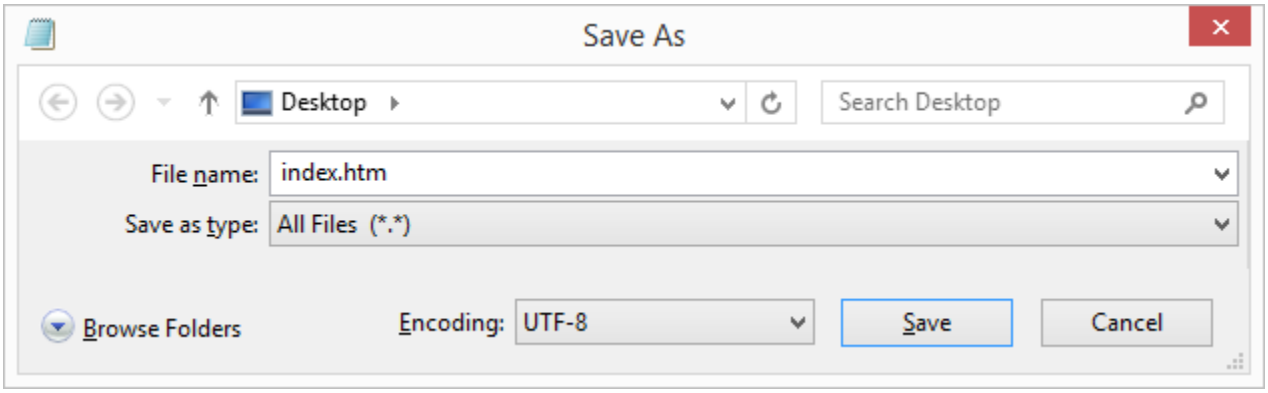
```
<body>

<h1>My First Heading</h1>

<p>My first paragraph.</p>

</body>
</html>
```



Step 3: Save the HTML Page

Save the file on your computer. Select **File > Save as** in the Notepad menu.

Name the file **"index.htm"** and set the encoding to **UTF-8** (which is the preferred encoding for HTML files).

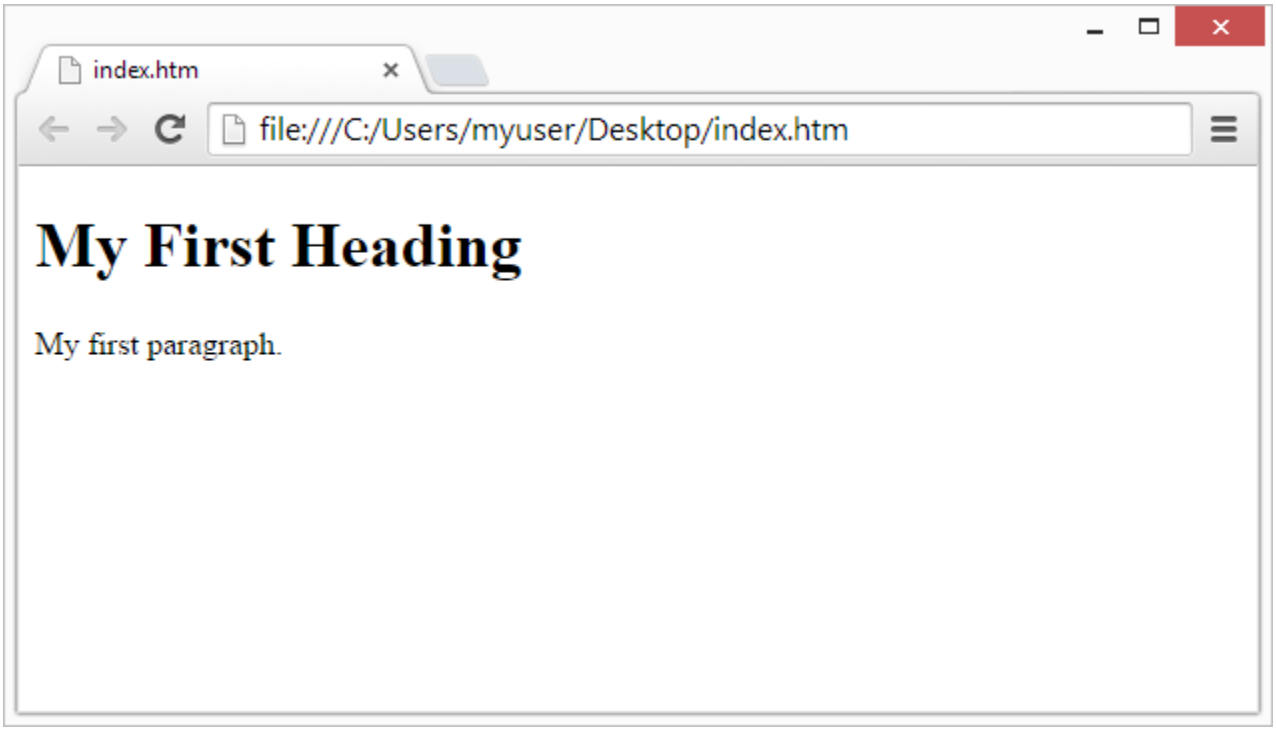


You can use either .htm or .html as file extension. There is no difference, it is up to you.

Step 4: View the HTML Page in Your Browser

Open the saved HTML file in your favorite browser (double click on the file, or right-click - and choose "Open with").

The result will look much like this:



W3Schools Online Editor

With our free online editor, you can edit the HTML code and view the result in your browser.

It is the perfect tool when you want to **test** code fast. It also has color coding and the ability to save and share code with others:

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

HTML Basic Examples

HTML Documents

All HTML documents must start with a document type declaration: `<!DOCTYPE html>`.

The HTML document itself begins with `<html>` and ends with `</html>`.

The visible part of the HTML document is between `<body>` and `</body>`.

Example

```
<!DOCTYPE html>
<html>
<body>

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

</body>
</html>
```

HTML Headings

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

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

Example

```
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
```

HTML Paragraphs

HTML paragraphs are defined with the `<p>` tag:

Example

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

Try it Yourself »

HTML Links

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

Example

```
<a href="https://www.w3schools.com">This is a link</a>
```

Try it Yourself »

The link's destination is specified in the `href` attribute.

Attributes are used to provide additional information about HTML elements.

You will learn more about attributes in a later chapter.

HTML Images

HTML images are defined with the `<img>` tag.

The source file (`src`), alternative text (`alt`), `width`, and `height` are provided as attributes:

Example

```
<img src="w3schools.jpg" alt="W3Schools.com" width="104" height="142">
```

Try it Yourself »

HTML Buttons

HTML buttons are defined with the `<button>` tag:

Example

```
<button>Click me</button>
```

Try it Yourself »

HTML Lists

HTML lists are defined with the `<ul>` (unordered/bullet list) or the `<ol>` (ordered/numbered list) tag, followed by `<li>` tags (list items):

Example

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

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

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:

My first paragraph.

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

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>
```

Try it Yourself »

Example Explained

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 and an **end** tag .

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>
```

Try it Yourself »

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.

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

Example

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

Try it Yourself »

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.

HTML Is Not Case Sensitive

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

The HTML5 standard does not require lowercase tags, but W3C **recommends** lowercase in HTML, and **demands**lowercase for stricter document types like XHTML.

Attributes provide additional information about HTML elements.

HTML Attributes

- All HTML elements can have **attributes**
- Attributes provide **additional information** about an element
- Attributes are always specified in **the start tag**
- Attributes usually come in name/value pairs like: **name="value"**

The href Attribute

HTML links are defined with the `<a>` tag. The link address is specified in the `href` attribute:

Example

```
<a href="https://www.w3schools.com">This is a link</a>
```

Try it Yourself »

You will learn more about links and the `<a>` tag later in this tutorial.

The src Attribute

HTML images are defined with the `<img>` tag.

The filename of the image source is specified in the `src` attribute:

Example

```
<img src="img_girl.jpg">
```

Try it Yourself »

The width and height Attributes

HTML images also have `width` and `height` attributes, which specifies the width and height of the image:

Example

```
<img src="img_girl.jpg" width="500" height="600">
```

Try it Yourself »

The width and height are is specified in pixels by default; so width="500" means 500 pixels wide.

You will learn more about images in our HTML Images chapter.

The alt Attribute

The `alt` attribute specifies an alternative text to be used, if an image cannot be displayed.

The value of the `alt` attribute can be read by screen readers. This way, someone "listening" to the webpage, e.g. a vision impaired person, can "hear" the element.

Example

```
<img src="img_girl.jpg" alt="Girl with a jacket">
```

Try it Yourself »

The `alt` attribute is also useful if the image cannot be displayed (e.g. if it does not exist):

Example

See what happens if we try to display an image that does not exist:

```
<img src="img_typo.jpg" alt="Girl with a jacket">
```

Try it Yourself »

The style Attribute

The `style` attribute is used to specify the styling of an element, like color, font, size etc.

Example

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

Try it Yourself »

You will learn more about styling later in this tutorial, and in our CSS Tutorial.

The lang Attribute

The language of the document can be declared in the `<html>` tag.

The language is declared with the `lang` attribute.

Declaring a language is important for accessibility applications (screen readers) and search engines:

```
<!DOCTYPE html>
<html lang="en-US">
```

```
<body>

...

</body>
</html>
```

The first two letters specify the language (en). If there is a dialect, add two more letters (US).

The title Attribute

Here, a `title` attribute is added to the `<p>` element. The value of the title attribute will be displayed as a tooltip when you mouse over the paragraph:

Example

```
<p title="I'm a tooltip">
This is a paragraph.
</p>
```

Try it Yourself »

We Suggest: Use Lowercase Attributes

The HTML5 standard does not require lowercase attribute names.

The title attribute can be written with uppercase or lowercase like **title** or **TITLE**.

W3C **recommends** lowercase in HTML, and **demands** lowercase for stricter document types like XHTML.

At W3Schools we always use lowercase attribute names.

We Suggest: Quote Attribute Values

The HTML5 standard does not require quotes around attribute values.

The `href` attribute, demonstrated above, *can* be written without quotes:

Bad

```
<a href=https://www.w3schools.com>
```

Try it Yourself »

Good

```
<a href="https://www.w3schools.com">
```

Try it Yourself »

W3C **recommends** quotes in HTML, and **demands** quotes for stricter document types like XHTML.

Sometimes it is **necessary** to use quotes. This example will not display the title attribute correctly, because it contains a space:

Example

```
<p title=About W3Schools>
```

Try it Yourself »

Using quotes are the most common. Omitting quotes can produce errors.
At W3Schools we **always** use quotes around attribute values.

Single or Double Quotes?

Double quotes around attribute values are the most common in HTML, but single quotes can also be used.

In some situations, when the attribute value itself contains double quotes, it is necessary to use single quotes:

```
<p title='John "ShotGun" Nelson'>
```

Or vice versa:

```
<p title="John 'ShotGun' Nelson">
```

Try it Yourself »

Chapter Summary

- All HTML elements can have **attributes**
- The `title` attribute provides additional "tool-tip" information
- The `href` attribute provides address information for links
- The `width` and `height` attributes provide size information for images
- The `alt` attribute provides text for screen readers
- At W3Schools we always use **lowercase** attribute names
- At W3Schools we always **quote** attribute values

HTML Exercises

Exercise:

Add a "tooltip" to the paragraph below with the text "About W3Schools".

```
<p ☒="About W3Schools">W3Schools is a web developer's
site.</p>
```

Submit Answer »

Start the Exercise

HTML Attributes

Below is an alphabetical list of some attributes often used in HTML, which you will learn more about in this tutorial:

Attribute	Description
alt	Specifies an alternative text for an image, when the image cannot be displayed
disabled	Specifies that an input element should be disabled
href	Specifies the URL (web address) for a link
id	Specifies a unique id for an element
src	Specifies the URL (web address) for an image
style	Specifies an inline CSS style for an element
title	Specifies extra information about an element (displayed as a tool tip)

A complete list of all attributes for each HTML element, is listed in our: HTML Attribute Reference.

Headings

Heading 1

Heading 2

Heading 3

Heading 4

Heading 5

Heading 6

Try it Yourself »

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>
```

Try it Yourself »

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>
```

Try it Yourself »

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>
```

Try it Yourself »

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>
.
.
.
```

Try it Yourself »

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

How to View HTML Source?

Have you ever seen a Web page and wondered "Hey! How did they do that?"

View HTML Source Code:

Right-click in an HTML page and select "View Page Source" (in Chrome) or "View Source" (in Edge), or similar in other browsers. This will open a window containing the HTML source code of the page.

Inspect an HTML Element:

Right-click on an element (or a blank area), and choose "Inspect" or "Inspect Element" to see what elements are made up of (you will see both the HTML and the CSS). You can also edit the HTML or CSS on-the-fly in the Elements or Styles panel that opens.

HTML Exercises

Exercise:

Use the correct HTML tag to add a heading with the text "London".

```
<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>
```

Submit Answer »

Start the Exercise

HTML Tag Reference

W3Schools' tag reference contains additional information about these tags and their attributes.

You will learn more about HTML tags and attributes in the next chapters of this tutorial.

Tag	Description
<html>	Defines the root of an HTML document
<body>	Defines the document's body
<head>	A container for all the head elements (title, scripts, styles, meta informatic
<h1> to <h6>	Defines HTML headings
<hr>	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>
```

Try it Yourself »

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>
```

Try it Yourself »

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.
```

Try it Yourself »

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 `<br>` element defines a **line break**.

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

Example

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

Try it Yourself »

The `<br>` 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>
```

Try it Yourself »

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>
```

Try it Yourself »

HTML Exercises

Exercise:

Use the correct HTML tag to add a paragraph with the text "Hello World!".

```
<html>
<body>

</body>
</html>
```

Submit Answer »

Start the Exercise

HTML Tag Reference

W3Schools' tag reference contains additional information about HTML elements and their attributes.

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

HTML Styles

☐ PreviousNext ☐

Example

I am Red

I am Blue

I am Big

Try it Yourself »

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.

You will learn more about CSS later in this tutorial.

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>
```

Try it Yourself »

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>
```

Try it Yourself »

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>
```

Try it Yourself »

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>
```

Try it Yourself »

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>
```

Try it Yourself »

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

Text Formatting

This text is bold

This text is italic

This is subscript and superscript

Try it Yourself »

HTML Formatting Elements

In the previous chapter, you learned about the HTML **style attribute**.

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

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

Formatting elements were designed to display special types of text:

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

HTML <b> and Elements

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

Example

```
<b>This text is bold</b>
```

Try it Yourself »

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

Example

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

Try it Yourself »

HTML <i> and Elements

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

Example

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

Try it Yourself »

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

Example

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

Try it Yourself »

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

HTML <small> Element

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

Example

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

Try it Yourself »

HTML <mark> Element

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

Example

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

Try it Yourself »

HTML <del> Element

The HTML `<del>` element defines deleted/removed text.

Example

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

Try it Yourself »

HTML <ins> Element

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

Example

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

Try it Yourself »

HTML <sub> Element

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

Example

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

Try it Yourself »

HTML <sup> Element

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

Example

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

Try it Yourself »

HTML Exercises

Exercise:

Add extra importance to the word "degradation" in the paragraph below.

```
<p>
WWF's mission is to stop the degradation of our planet's
```

```
natural environment.
</p>
```

Submit Answer »

Start the Exercise

HTML Text Formatting Elements

Tag	Description
<b>	Defines bold text
	Defines emphasized text
<i>	Defines italic text
<small>	Defines smaller text
	Defines important text
<sub>	Defines subscripted text

<sup>	Defines superscripted text
<ins>	Defines inserted text
<del>	Defines deleted text
<mark>	Defines marked/highlighted text

HTML Quotation and Citation Elements

☐ PreviousNext ☐

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.

Try it Yourself »

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>
```

Try it Yourself »

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>
```

Try it Yourself »

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>
```

Try it Yourself »

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>
```

Try it Yourself »

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>
```

Try it Yourself »

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>
```

Try it Yourself »

HTML Exercises

Exercise:

Use an HTML element to add quotation marks around the letters "cool".

```
<p>
I am so cool.
</p>
```

Submit Answer »

Start the Exercise

HTML Quotation and Citation Elements

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

HTML Comments

☐ PreviousNext ☐

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 -->
```

Try it Yourself »

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
<img border="0" src="pic_trulli.jpg" alt="Trulli">
-->
```

Try it Yourself »

HTML Exercises

Exercise:

Use the HTML comment tag to make a comment out of the "This is a comment" text.

```
<h1>This is a heading</h1>
 This is a comment 
<p>This is a paragraph.</p>
```

HTML Colors

☐ PreviousNext ☐

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:

Tomato

Orange

DodgerBlue

MediumSeaGreen

Gray

SlateBlue

Violet

LightGray

Try it Yourself »

HTML supports 140 standard color names.

Background Color

You can set the background color for HTML elements:

Hello World

Lorem ipsum dolor sit amet, consectetuer 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>
```

Try it Yourself »

Text Color

You can set the color of text:

Hello World

Lorem ipsum dolor sit amet, consectetuer 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>
```

Try it Yourself »

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>
```

Try it Yourself »

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>
```

Try it Yourself »

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)

Try it Yourself »

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)
```

Try it Yourself »

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

Try it Yourself »

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

Example

#000000

#3c3c3c

#787878

#b4b4b4

#f0f0f0

#ffffff

Try it Yourself »

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%)

Try it Yourself »

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%)

Try it Yourself »

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%)
```

Try it Yourself »

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%)
```

Try it Yourself »

HTML Styles - CSS

☐ PreviousNext ☐

CSS = Styles and Colors

Manipulate Text

Colors, Boxes

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.

Tip: You can learn much more about CSS in our CSS Tutorial.

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>
```

Try it Yourself »

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>
```

Try it Yourself »

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>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Try it Yourself »

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>
```

Try it Yourself »

CSS Border

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

Example

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

Try it Yourself »

CSS Padding

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

Example

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

Try it Yourself »

CSS Margin

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

Example

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

Try it Yourself »

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;
}
```

Try it Yourself »

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;
}
```

Try it Yourself »

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="https://www.w3schools.com/html/styles.css">
```

Try it Yourself »

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">
```

Try it Yourself »

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

Example

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

Try it Yourself »

You can read more about file paths in the chapter HTML File Paths.

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 Exercises

Exercise:

Use CSS to set the background color of the document (body) to yellow.

```
<!DOCTYPE html>
<html>
<head>
<style>
□ □
  □:yellow;
□
</style>
</head>
<body>

<h1>My Home Page</h1>
```

```
</body>
</html>
```

Submit Answer »

Start the Exercise

HTML Style Tags

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

HTML Links

☐ PreviousNext ☐

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.w3schools.com/html/">Visit our HTML tutorial</a>
```

Try it Yourself »

The `href` attribute specifies the destination address (https://www.w3schools.com/html/) of the link.

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

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>
```

Try it Yourself »

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>
```

Try it Yourself »

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>
```

Try it Yourself »

To learn more about CSS, go to our CSS Tutorial.

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.w3schools.com/" target="_blank">Visit W3Schools!</a>
```

Try it Yourself »

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.w3schools.com/html/" target="_top">HTML5 tutorial!</a>
```

Try it Yourself »

HTML Links - Image as Link

It is common to use images as links:

Example

```
<a href="default.asp">
  <img src="smiley.gif" alt="HTML
tutorial" style="width:42px;height:42px;border:0;">
</a>
```

Try it Yourself »

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.w3schools.com/html/" title="Go to W3Schools HTML section">Visit our HTML Tutorial</a>
```

Try it Yourself »

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>
```

Try it Yourself »

External Paths

External pages 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 web page:

Example

```
<a href="https://www.w3schools.com/html/default.asp">HTML tutorial</a>
```

Try it Yourself »

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

Example

```
<a href="/html/default.asp">HTML tutorial</a>
```

Try it Yourself »

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

Example

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

HTML Images

☐ PreviousNext ☐

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

Example

```
<img src="pic_trulli.jpg" alt="Italian Trulli">
```

Try it Yourself »

Example

```
<img src="img_girl.jpg" alt="Girl in a jacket">
```

Try it Yourself »

Example

```
<img src="img_chania.jpg" alt="Flowers in Chania">
```

Try it Yourself »

HTML Images Syntax

In HTML, images are defined with the `<img>` tag.

The `<img>` 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:

```
<img src="url">
```

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

```
<img src="img_chania.jpg" alt="Flowers in Chania">
```

Try it Yourself »

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

Example

```
<img src="wrongname.gif" alt="Flowers in Chania">
```

Try it Yourself »

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

```
<img src="img_girl.jpg" alt="Girl in a
jacket" style="width:500px;height:600px;">
```

Try it Yourself »

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

Example

```
<img src="img_girl.jpg" alt="Girl in a jacket" width="500" height="600">
```

Try it Yourself »

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>

<img src="html5.gif" alt="HTML5 Icon" width="128" height="128">
<img src="html5.gif" alt="HTML5 Icon" style="width:128px;height:128px;">

</body>
</html>
```

Try it Yourself »

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

```
<img src="/images/html5.gif" alt="HTML5
Icon" style="width:128px;height:128px;">
```

Try it Yourself »

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

```
<img src="https://www.w3schools.com/images/w3schools_green.jpg" alt="W3School
s.com">
```

Try it Yourself »

You can read more about file paths in the chapter HTML File Paths.

Animated Images

HTML allows animated GIFs:

Example

```
<img src="programming.gif" alt="Computer
Man" style="width:48px;height:48px;">
```

Try it Yourself »

Image as a Link

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

Example

```
<a href="default.asp">
  <img src="smiley.gif" alt="HTML
tutorial" style="width:42px;height:42px;border:0;">
</a>
```

Try it Yourself »

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><img src="smiley.gif" alt="Smiley
face" style="float:right;width:42px;height:42px;">
The image will float to the right of the text.</p>

<p><img src="smiley.gif" alt="Smiley
face" style="float:left;width:42px;height:42px;">
The image will float to the left of the text.</p>
```

Try it Yourself »

Tip: To learn more about CSS Float, read our CSS Float Tutorial.

Image Maps

The `<map>` tag defines an image-map. An image-map is an image with clickable areas.

In the image below, click on the computer, the phone, or the cup of coffee:

Example

```
<img src="workplace.jpg" alt="Workplace" usemap="#workmap">

<map name="workmap">
  <area shape="rect" coords="34,44,270,350" alt="Computer" href="computer.htm
">
  <area shape="rect" coords="290,172,333,250" alt="Phone" href="phone.htm">
  <area shape="circle" coords="337,300,44" alt="Coffee" href="coffee.htm">
</map>
```

Try it Yourself »

The `name` attribute of the `<map>` tag is associated with the `<img>`'s usemap attribute and creates a relationship between the image and the map.

The `<map>` element contains a number of `<area>` tags, that define the clickable areas in the image-map.

Background Image

To add a background image on an HTML element, use the CSS property `background-image`:

Example

To add a background image on a web page, specify the background-image property on the BODY element:

```
<body style="background-image:url('clouds.jpg');">

<h2>Background Image</h2>

</body>
```

Try it Yourself »

Example

To add a background image on a paragraph, specify the background-image property on the P element:

```
<body>

<p style="background-image:url('clouds.jpg');">
...
</p>

</body>
```

Try it Yourself »

To learn more about background images, study our CSS Background Tutorial.

The HTML <picture> Element

HTML5 introduced the `<picture>` element to add more flexibility when specifying image resources.

The `<picture>` element contains a number of <source> elements, each referring to different image sources. This way the browser can choose the image that best fits the current view and/or device.

Each `<source>` element have attributes describing when their image is the most suitable.

The browser will use the first `<source>` element with matching attribute values, and ignore any following `<source>`elements.

Example

Show one picture if the browser window (viewport) is a minimum of 650 pixels, and another image if not, but larger than 465 pixels.

```
<picture>
  <source media="(min-width: 650px)" srcset="img_pink_flowers.jpg">
  <source media="(min-width: 465px)" srcset="img_white_flower.jpg">
  <img src="img_orange_flowers.jpg" alt="Flowers" style="width:auto;">
</picture>
```

Try it Yourself »

Note: Always specify an `<img>` element as the last child element of the `<picture>` element. The `<img>`element is used by browsers that do not support the `<picture>` element, or if none of the `<source>` tags matched.

HTML Screen Readers

A screen reader is a software program that reads the HTML code, converts the text, and allows the user to "listen" to the content. Screen readers are useful for people who are visually impaired or learning disabled.

Chapter Summary

- Use the HTML `<img>` 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 `<img>`'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 Exercises

Exercise:

Use the HTML image attributes to set the size of the image to 250 pixels wide and 400 pixels tall.

```
<img src="scream.png" ☒="250" ☒="400">
```

Submit Answer »

Start the Exercise

HTML Image Tags

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

HTML Tables

☐ PreviousNext ☐

HTML Table Example

Company	Contact	Cou
Alfreds Futterkiste	Maria Anders	Gerr
Centro comercial Moctezuma	Francisco Chang	Mexi
Ernst Handel	Roland Mendel	Aust
Island Trading	Helen Bennett	UK

Laughing Bacchus Winecellars	Yoshi Tannamuri	Cana
Magazzini Alimentari Riuniti	Giovanni Rovelli	Italy

Try it Yourself »

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

Try it Yourself »

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;
}
```

Try it Yourself »

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;
}
```

Try it Yourself »

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;
}
```

Try it Yourself »

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;
}
```

Try it Yourself »

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;
}
```

Try it Yourself »

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>
```

Try it Yourself »

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>
```

Try it Yourself »

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>
```

Try it Yourself »

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;
}
```

Try it Yourself »

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;
}
```

Try it Yourself »

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 Exercises

Exercise:

Add a table row with two table headers.

The two table headers should have the value "Name" and "Age".

```
<table>
  ☒
```

```
    
    
  
  <tr>
    <td>Jill Smith</td>
    <td>50</td>
  </tr>
</table>
```

Submit Answer »

Start the Exercise

HTML Table Tags

Tag	Description
<table>	Defines a table
<th>	Defines a header cell in a table
<tr>	Defines a row in a table
<td>	Defines a cell in a table

<caption>	Defines a table caption
<colgroup>	Specifies a group of one or more columns in a table for formatting
<col>	Specifies column properties for each column within a <colgroup> element
<thead>	Groups the header content in a table
<tbody>	Groups the body content in a table
<tfoot>	Groups the footer content in a table

HTML Lists

☐ PreviousNext ☐

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

Try it Yourself »

Unordered HTML List

An unordered list starts with the <ul> tag. Each list item starts with the <li> 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>
```

Try it Yourself »

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>
```

Try it Yourself »

Example - Circle

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

Try it Yourself »

Example - Square

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

Try it Yourself »

Example - None

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

Try it Yourself »

Ordered HTML List

An ordered list starts with the <ol> tag. Each list item starts with the <li> tag.

The list items will be marked with numbers by default:

Example

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

Try it Yourself »

Ordered HTML List - The Type Attribute

The `type` attribute of the <ol> 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>
```

Try it Yourself »

Uppercase Letters:

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

Try it Yourself »

Lowercase Letters:

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

Try it Yourself »

Uppercase Roman Numbers:

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

Try it Yourself »

Lowercase Roman Numbers:

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

Try it Yourself »

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>
```

Try it Yourself »

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>
```

Try it Yourself »

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>
```

Try it Yourself »

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>
```

Try it Yourself »

Tip: You can learn much more about CSS in our CSS Tutorial.

Chapter Summary

- Use the HTML `<ul>` element to define an unordered list
- Use the CSS `list-style-type` property to define the list item marker
- Use the HTML `<ol>` element to define an ordered list
- Use the HTML `type` attribute to define the numbering type
- Use the HTML `<li>` 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 Exercises

Exercise:

Add a list item with the text "Coffee" inside the `<ul>` element.

```
<ul>Coffee</ul>
```

Submit Answer »

Start the Exercise

HTML List Tags

Tag	Description
	Defines an unordered list
	Defines an ordered list
	Defines a list item
<dl>	Defines a description list

<dt>	Defines a term in a description list
<dd>	Describes the term in a description list

HTML Block and Inline Elements

☐ PreviousNext ☐

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 element is a block-level element.

Example

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

Try it Yourself »

Block level elements in HTML:

<address>

<article>

<aside>

<blockquote>

<canvas>

<dd>

<dl>

<dt>

<fieldset>

<figcaption>

<figure>

<footer>

<form>

<h1>-<h6>

<header>

<hr>

<main>

<nav>

<noscript>

<p>

<pre>

```
<section>

<table>

<tfoot>

<ul>

<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

```
<span>Hello World</span>
```

Try it Yourself »

Inline elements in HTML:

```
<a>

<abbr>

<acronym>

<b>

<bdo>

<big>

<br>

<button>
```

<cite>

<code>

<dfn>

<i>

<input>

<kbd>

<label>

<map>

<object>

<output>

<q>

<samp>

<script>

<select>

<small>

<sub>

<sup>

<textarea>

`<time>`

`<tt>`

`<var>`

The 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>
```

Try it Yourself »

The Element

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

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

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

Example

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

Try it Yourself »

HTML Grouping Tags

Tag	Description
	Defines a section in a document (block-level)
	Defines a section in a document (inline)

HTML The class Attribute

☐ PreviousNext ☐

Using The class Attribute

The HTML `class` attribute is used to define equal styles for elements with the same class name.

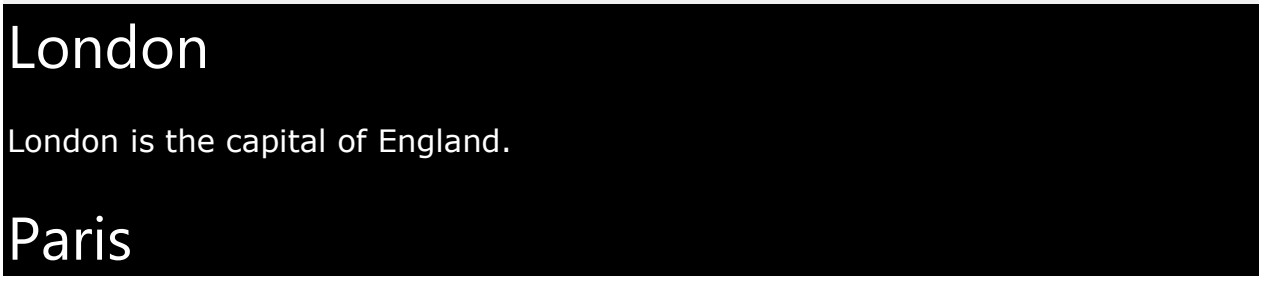
So, all HTML elements with the same `class` attribute will get the same style.

Here we have three `<div>` elements that point to the same class name:

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
.cities {
  background-color: black;
  color: white;
  margin: 20px;
  padding: 20px;
}
</style>
</head>
<body>

<div class="cities">
  <h2>London</h2>
  <p>London is the capital of England.</p>
</div>

<div class="cities">
  <h2>Paris</h2>
  <p>Paris is the capital of France.</p>
</div>

<div class="cities">
  <h2>Tokyo</h2>
  <p>Tokyo is the capital of Japan.</p>
</div>

</body>
</html>
```

Result:



Paris is the capital of France.

Tokyo

Tokyo is the capital of Japan.

Try it Yourself »

Using The class Attribute on Inline Elements

The HTML `class` attribute can also be used on inline elements:

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
span.note {
  font-size: 120%;
  color: red;
}
</style>
</head>
<body>

<h1>My <span class="note">Important</span> Heading</h1>
<p>This is some <span class="note">important</span> text.</p>

</body>
</html>
```

Try it Yourself »

Tip: The `class` attribute can be used on **any** HTML element.

Note: The class name is case sensitive!

Tip: You can learn much more about CSS in our CSS Tutorial.

Select Elements With a Specific Class

In CSS, to select elements with a specific class, write a period (.) character, followed by the name of the class:

Example

Use CSS to style all elements with the class name "city":

```
<style>
.city {
  background-color: tomato;
  color: white;
  padding: 10px;
}
</style>

<h2 class="city">London</h2>
<p>London is the capital of England.</p>

<h2 class="city">Paris</h2>
<p>Paris is the capital of France.</p>

<h2 class="city">Tokyo</h2>
<p>Tokyo is the capital of Japan.</p>
```

Result:

London

London is the capital of England.

Paris

Paris is the capital of France.

Tokyo

Tokyo is the capital of Japan.

Try it Yourself »

Multiple Classes

HTML elements can have more than one class name, each class name must be separated by a space.

Example

Style elements with the class name "city", also style elements with the class name "main":

```
<h2 class="city main">London</h2>
<h2 class="city">Paris</h2>
<h2 class="city">Tokyo</h2>
```

Try it Yourself »

In the example above, the first `<h2>` element belongs to both the "city" class and the "main" class.

Different Tags Can Share Same Class

Different tags, like `<h2>` and `<p>`, can have the same class name and thereby share the same style:

Example

```
<h2 class="city">Paris</h2>
<p class="city">Paris is the capital of France</p>
```

Try it Yourself »

Using The class Attribute in JavaScript

The class name can also be used by JavaScript to perform certain tasks for elements with the specified class name.

JavaScript can access elements with a specified class name by using the `getElementsByClassName()` method:

Example

When a user clicks on a button, hide all elements with the class name "city":

```
<script>
function myFunction() {
  var x = document.getElementsByClassName("city");
  for (var i = 0; i < x.length; i++) {
    x[i].style.display = "none";
  }
}
</script>
```

Try it Yourself »

Don't worry if you don't understand the code in the example above.

You will learn more about JavaScript in our HTML JavaScript chapter, or you can study our JavaScript Tutorial.

HTML Exercises

Exercise:

Create a class selector named "special".

Add a color property with the value "blue" inside the "special" class.

```
<!DOCTYPE html>
<html>
<head>
<style>
 
   ;

</style>
</head>
<body>

<p class="special">My paragraph</p>

</body>
</html>
```

HTML The id Attribute

☐ PreviousNext ☐

Using The id Attribute

The `id` attribute specifies a unique id for an HTML element (the value must be unique within the HTML document).

The id value can be used by CSS and JavaScript to perform certain tasks for the element with the specific id value.

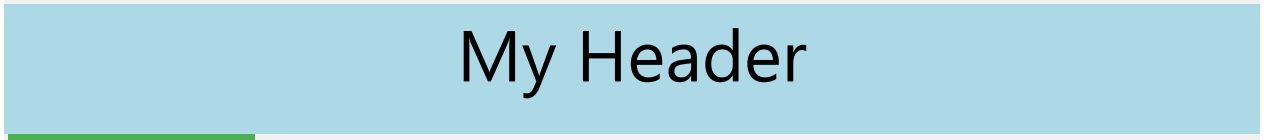
In CSS, to select an element with a specific id, write a hash (#) character, followed by the id of the element:

Example

Use CSS to style an element with the id "myHeader":

```
<style>
#myHeader {
  background-color: lightblue;
  color: black;
  padding: 40px;
  text-align: center;
}
</style>

<h1 id="myHeader">My Header</h1>
```

Result:



Try it Yourself »

Tip: The id attribute can be used on **any** HTML element.

Note: The id value is case-sensitive.

Note: The id value must contain at least **one** character, and must **not** contain whitespace (spaces, tabs, etc.).

Difference Between Class and ID

An HTML element can only have one unique id that belongs to that single element, while a class name can be used by multiple elements:

Example

```
<style>
/* Style the element with the id "myHeader" */
#myHeader {
  background-color: lightblue;
```

```
  color: black;
  padding: 40px;
  text-align: center;
}

/* Style all elements with the class name "city" */
.city {
  background-color: tomato;
  color: white;
  padding: 10px;
}
</style>

<!-- A unique element -->
<h1 id="myHeader">My Cities</h1>

<!-- Multiple similar elements -->
<h2 class="city">London</h2>
<p>London is the capital of England.</p>

<h2 class="city">Paris</h2>
<p>Paris is the capital of France.</p>

<h2 class="city">Tokyo</h2>
<p>Tokyo is the capital of Japan.</p>
```

Try it Yourself »

Tip: You can learn much more about CSS in our CSS Tutorial.

Bookmarks with ID and Links

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>
```

Try it Yourself »

Using The id Attribute in JavaScript

JavaScript can access an element with a specified id by using the `getElementById()` method:

Example

Use the id attribute to manipulate text with JavaScript:

```
<script>
function displayResult() {
  document.getElementById("myHeader").innerHTML = "Have a nice day!";
}
</script>
```

Try it Yourself »

Tip: Study JavaScript in the HTML JavaScript chapter, or in our JavaScript Tutorial.

HTML Exercises

Exercise:

Add the correct HTML attribute to make the H1 element red.

```
<!DOCTYPE html>
<html>
<head>
<style>
#myheader {color:red;}
</style>
</head>
<body>

<h1 >My Home Page</h1>

</body>
</html>
```

HTML Forms

☐ PreviousNext ☐

HTML Form Example

First name:

Last name:

Try it Yourself »

The <form> Element

The HTML `<form>` element defines a form that is used to collect user input:

```
<form>
.
form elements
.
</form>
```

An HTML form contains **form elements**.

Form elements are different types of input elements, like text fields, checkboxes, radio buttons, submit buttons, and more.

The <input> Element

The `<input>` element is the most important form element.

The `<input>` element can be displayed in several ways, depending on the **type** attribute.

Here are some examples:

Type	Description
<input type="text">	Defines a one-line text input field
<input type="radio">	Defines a radio button (for selecting one of many choices)
<input type="submit">	Defines a submit button (for submitting the form)

You will learn a lot more about input types later in this tutorial.

Text Input

`<input type="text">` defines a one-line input field for **text input**:

Example

```
<form>
  First name:<br>
  <input type="text" name="firstname"><br>
  Last name:<br>
  <input type="text" name="lastname">
</form>
```

Try it Yourself »

This is how it will look like in a browser:

First name:

Last name:


Note: The form itself is not visible. Also note that the default width of a text field is 20 characters.

Radio Button Input

`<input type="radio">` defines a **radio button**.

Radio buttons let a user select ONE of a limited number of choices:

Example

```
<form>
  <input type="radio" name="gender" value="male" checked> Male<br>
  <input type="radio" name="gender" value="female"> Female<br>
  <input type="radio" name="gender" value="other"> Other
</form>
```

Try it Yourself »

This is how the HTML code above will be displayed in a browser:

 Male
 Female
 Other

The Submit Button

`<input type="submit">` defines a button for **submitting** the form data to a **form-handler**.

The form-handler is typically a server page with a script for processing input data.

The form-handler is specified in the form's **action** attribute:

Example

```
<form action="/action_page.php">
  First name:<br>
  <input type="text" name="firstname" value="Mickey"><br>
  Last name:<br>
  <input type="text" name="lastname" value="Mouse"><br><br>
  <input type="submit" value="Submit">
</form>
```

Try it Yourself »

This is how the HTML code above will be displayed in a browser:

First name:

Last name:

The Action Attribute

The `action` attribute defines the action to be performed when the form is submitted.

Normally, the form data is sent to a web page on the server when the user clicks on the submit button.

In the example above, the form data is sent to a page on the server called "/action_page.php". This page contains a server-side script that handles the form data:

```
<form action="/action_page.php">
```

If the `action` attribute is omitted, the action is set to the current page.

The Target Attribute

The `target` attribute specifies if the submitted result will open in a new browser tab, a frame, or in the current window.

The default value is "`_self`" which means the form will be submitted in the current window.

To make the form result open in a new browser tab, use the value "`_blank`":

Example

```
<form action="/action_page.php" target="_blank">
```

Try it Yourself »

Other legal values are "`_parent`", "`_top`", or a name representing the name of an iframe.

The Method Attribute

The `method` attribute specifies the HTTP method (**GET** or **POST**) to be used when submitting the form data:

Example

```
<form action="/action_page.php" method="get">
```

Try it Yourself »

or:

Example

```
<form action="/action_page.php" method="post">
```

Try it Yourself »

When to Use GET?

The default method when submitting form data is GET.

However, when GET is used, the submitted form data will be **visible in the page address field**:

```
/action_page.php?firstname=Mickey&lastname=Mouse
```

Notes on GET:

- Appends form-data into the URL in name/value pairs
- The length of a URL is limited (about 3000 characters)
- Never use GET to send sensitive data! (will be visible in the URL)
- Useful for form submissions where a user wants to bookmark the result
- GET is better for non-secure data, like query strings in Google

When to Use POST?

Always use POST if the form data contains sensitive or personal information. The POST method does not display the submitted form data in the page address field.

Notes on POST:

- POST has no size limitations, and can be used to send large amounts of data.
- Form submissions with POST cannot be bookmarked

The Name Attribute

Each input field must have a `name` attribute to be submitted.

If the `name` attribute is omitted, the data of that input field will not be sent at all.

This example will only submit the "Last name" input field:

Example

```
<form action="/action_page.php">
  First name:<br>
  <input type="text" value="Mickey"><br>
  Last name:<br>
  <input type="text" name="lastname" value="Mouse"><br><br>
  <input type="submit" value="Submit">
</form>
```

Try it Yourself »

Grouping Form Data with <fieldset>

The `<fieldset>` element is used to group related data in a form.

The `<legend>` element defines a caption for the `<fieldset>` element.

Example

```
<form action="/action_page.php">
  <fieldset>
    <legend>Personal information:</legend>
    First name:<br>
    <input type="text" name="firstname" value="Mickey"><br>
    Last name:<br>
    <input type="text" name="lastname" value="Mouse"><br><br>
    <input type="submit" value="Submit">
  </fieldset>
</form>
```

Try it Yourself »

This is how the HTML code above will be displayed in a browser:

Personal information:First name:

Last name:

HTML Exercises

Exercise:

In the form below, add an input field with the type "button" and the value "OK".

```
<form>
<>
</form>
```

Submit Answer »

Start the Exercise

Here is the list of all `<form>` attributes:

Attribute	Description
accept-charset	Specifies the charset used in the submitted form (default: the page charse
action	Specifies an address (url) where to submit the form (default: the submitti

autocomplete	Specifies if the browser should autocomplete the form (default: on).
enctype	Specifies the encoding of the submitted data (default: is url-encoded).
method	Specifies the HTTP method used when submitting the form (default: GET).
name	Specifies a name used to identify the form (for DOM usage: document.form
novalidate	Specifies that the browser should not validate the form.
target	Specifies the target of the address in the action attribute (default: _self).

HTML Form Elements

☐ PreviousNext ☐

This chapter describes all HTML form elements.

The <input> Element

The most important form element is the `<input>` element.

The `<input>` element can be displayed in several ways, depending on the `type` attribute.

Example

```
<input name="firstname" type="text">
```

Try it Yourself »

If the `type` attribute is omitted, the input field gets the default type: "text".

All the different input types are covered in the next chapter.

The <select> Element

The `<select>` element defines a **drop-down list**:

Example

```
<select name="cars">
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="fiat">Fiat</option>
  <option value="audi">Audi</option>
</select>
```

Try it Yourself »

The `<option>` elements defines an option that can be selected.

By default, the first item in the drop-down list is selected.

To define a pre-selected option, add the `selected` attribute to the option:

Example

```
<option value="fiat" selected>Fiat</option>
```

Try it Yourself »

Visible Values:

Use the `size` attribute to specify the number of visible values:

Example

```
<select name="cars" size="3">
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="fiat">Fiat</option>
  <option value="audi">Audi</option>
</select>
```

Try it Yourself »

Allow Multiple Selections:

Use the `multiple` attribute to allow the user to select more than one value:

Example

```
<select name="cars" size="4" multiple>
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="fiat">Fiat</option>
  <option value="audi">Audi</option>
</select>
```

Try it Yourself »

The <textarea> Element

The `<textarea>` element defines a multi-line input field (**a text area**):

Example

```
<textarea name="message" rows="10" cols="30">
The cat was playing in the garden.
</textarea>
```

Try it Yourself »

The `rows` attribute specifies the visible number of lines in a text area.

The `cols` attribute specifies the visible width of a text area.

This is how the HTML code above will be displayed in a browser:

You can also define the size of the text area by using CSS:

Example

```
<textarea name="message" style="width:200px; height:600px;">
The cat was playing in the garden.
</textarea>
```

Try it Yourself »

The <button> Element

The `<button>` element defines a clickable **button**:

Example

```
<button type="button" onclick="alert('Hello World!')">Click Me!</button>
```

Try it Yourself »

This is how the HTML code above will be displayed in a browser:

Click Me!

Note: Always specify the **type** attribute for the button element. Different browsers may use different default types for the button element.

HTML5 Form Elements

HTML5 added the following form elements:

- `<datalist>`
- `<output>`

Note: Browsers do not display unknown elements. New elements that are not supported in older browsers will not "destroy" your web page.

HTML5 <datalist> Element

The `<datalist>` element specifies a list of pre-defined options for an `<input>` element.

Users will see a drop-down list of the pre-defined options as they input data.

The `list` attribute of the `<input>` element, must refer to the `id` attribute of the `<datalist>` element.

Example

```
<form action="/action_page.php">
  <input list="browsers">
  <datalist id="browsers">
    <option value="Internet Explorer">
    <option value="Firefox">
    <option value="Chrome">
    <option value="Opera">
    <option value="Safari">
  </datalist>
</form>
```

Try it Yourself »

HTML5 <output> Element

The `<output>` element represents the result of a calculation (like one performed by a script).

Example

Perform a calculation and show the result in an `<output>` element:

```
<form action="/action_page.php"
  oninput="x.value=parseInt(a.value)+parseInt(b.value)">
  0
  <input type="range"  id="a" name="a" value="50">
  100 +
  <input type="number" id="b" name="b" value="50">
  =
  <output name="x" for="a b"></output>
  <br><br>
  <input type="submit">
</form>
```

Try it Yourself »

HTML Exercises

Exercise:

In the form below, add an empty drop down list with the name "cars".

```
<form action="/action_page.php">
<>
</>
</form>
```

Submit Answer »

Start the Exercise

HTML Form Elements

Tag	Description
<form>	Defines an HTML form for user input
<input>	Defines an input control
<textarea>	Defines a multiline input control (text area)
<label>	Defines a label for an <input> element
<fieldset>	Groups related elements in a form
<legend>	Defines a caption for a <fieldset> element
<select>	Defines a drop-down list
<optgroup>	Defines a group of related options in a drop-down list

<option>	Defines an option in a drop-down list
<button>	Defines a clickable button
<datalist>	Specifies a list of pre-defined options for input controls
<output>	Defines the result of a calculation

For a complete list of all available HTML tags, visit our HTML Tag

HTML Input Types

Previous Next

This chapter describes the different input types for the <input> element.

HTML Input Types

Here are the different input types you can use in HTML:

- `<input type="button">`
- `<input type="checkbox">`
- `<input type="color">`
- `<input type="date">`
- `<input type="datetime-local">`
- `<input type="email">`
- `<input type="file">`

- `<input type="hidden">`
- `<input type="image">`
- `<input type="month">`
- `<input type="number">`
- `<input type="password">`
- `<input type="radio">`
- `<input type="range">`
- `<input type="reset">`
- `<input type="search">`
- `<input type="submit">`
- `<input type="tel">`
- `<input type="text">`
- `<input type="time">`
- `<input type="url">`
- `<input type="week">`

Input Type Text

`<input type="text">` defines a **one-line text input field**:

Example

```
<form>
  First name:<br>
  <input type="text" name="firstname"><br>
  Last name:<br>
  <input type="text" name="lastname">
</form>
```

Try it Yourself »

This is how the HTML code above will be displayed in a browser:

First name:

Last name:

Input Type Password

`<input type="password">` defines a **password field**:

Example

```
<form>
  User name:<br>
  <input type="text" name="username"><br>
  User password:<br>
  <input type="password" name="psw">
</form>
```

Try it Yourself »

This is how the HTML code above will be displayed in a browser:

User name:

User password:

The characters in a password field are masked (shown as asterisks or circles).

Input Type Submit

`<input type="submit">` defines a button for **submitting** form data to a **form-handler**.

The form-handler is typically a server page with a script for processing input data.

The form-handler is specified in the form's `action` attribute:

Example

```
<form action="/action_page.php">
  First name:<br>
  <input type="text" name="firstname" value="Mickey"><br>
```

```
  Last name:<br>
  <input type="text" name="lastname" value="Mouse"><br><br>
  <input type="submit" value="Submit">
</form>
```

Try it Yourself »

This is how the HTML code above will be displayed in a browser:

First name:

Last name:

If you omit the submit button's value attribute, the button will get a default text:

Example

```
<form action="/action_page.php">
  First name:<br>
  <input type="text" name="firstname" value="Mickey"><br>
  Last name:<br>
  <input type="text" name="lastname" value="Mouse"><br><br>
  <input type="submit">
</form>
```

Try it Yourself »

Input Type Reset

`<input type="reset">` defines a **reset button** that will reset all form values to their default values:

Example

```
<form action="/action_page.php">
  First name:<br>
```

```
  <input type="text" name="firstname" value="Mickey"><br>
  Last name:<br>
  <input type="text" name="lastname" value="Mouse"><br><br>
  <input type="submit" value="Submit">
  <input type="reset">
</form>
```

Try it Yourself »

This is how the HTML code above will be displayed in a browser:

First name:

Last name:

If you change the input values and then click the "Reset" button, the form-data will be reset to the default values.

Input Type Radio

`<input type="radio">` defines a **radio button**.

Radio buttons let a user select ONLY ONE of a limited number of choices:

Example

```
<form>
  <input type="radio" name="gender" value="male" checked> Male<br>
  <input type="radio" name="gender" value="female"> Female<br>
  <input type="radio" name="gender" value="other"> Other
</form>
```

Try it Yourself »

This is how the HTML code above will be displayed in a browser:

☒ Male
☒ Female
☒ Other

Input Type Checkbox

`<input type="checkbox">` defines a **checkbox**.

Checkboxes let a user select ZERO or MORE options of a limited number of choices.

Example

```
<form>
  <input type="checkbox" name="vehicle1" value="Bike"> I have a bike<br>
  <input type="checkbox" name="vehicle2" value="Car"> I have a car
</form>
```

Try it Yourself »

This is how the HTML code above will be displayed in a browser:

☒ I have a bike
☒ I have a car

Input Type Button

`<input type="button">` defines a **button**:

Example

```
<input type="button" onclick="alert('Hello World!')" value="Click Me!">
```

Try it Yourself »

This is how the HTML code above will be displayed in a browser:

HTML5 Input Types

HTML5 added several new input types:

- color
- date
- datetime-local
- email
- month
- number
- range
- search
- tel
- time
- url
- week

New input types that are not supported by older web browsers, will behave as `<input type="text">`.

Input Type Color

The `<input type="color">` is used for input fields that should contain a color.

Depending on browser support, a color picker can show up in the input field.

Example

```
<form>
  Select your favorite color:
  <input type="color" name="favcolor">
</form>
```

Try it Yourself »

Input Type Date

The `<input type="date">` is used for input fields that should contain a date.

Depending on browser support, a date picker can show up in the input field.

Example

```
<form>
  Birthday:
  <input type="date" name="bday">
</form>
```

Try it Yourself »

You can also use the `min` and `max` attributes to add restrictions to dates:

Example

```
<form>
  Enter a date before 1980-01-01:
  <input type="date" name="bday" max="1979-12-31"><br>
  Enter a date after 2000-01-01:
  <input type="date" name="bday" min="2000-01-02"><br>
</form>
```

Try it Yourself »

Input Type Datetime-local

The `<input type="datetime-local">` specifies a date and time input field, with no time zone.

Depending on browser support, a date picker can show up in the input field.

Example

```
<form>
  Birthday (date and time):
  <input type="datetime-local" name="bdaytime">
</form>
```

Try it Yourself »

Input Type Email

The `<input type="email">` is used for input fields that should contain an e-mail address.

Depending on browser support, the e-mail address can be automatically validated when submitted.

Some smartphones recognize the email type, and add ".com" to the keyboard to match email input.

Example

```
<form>
  E-mail:
  <input type="email" name="email">
</form>
```

Try it Yourself »

Input Type File

The `<input type="file">` defines a file-select field and a "Browse" button for file uploads.

Example

```
<form>
  Select a file: <input type="file" name="myFile">
</form>
```

Try it Yourself »

Input Type Month

The `<input type="month">` allows the user to select a month and year.

Depending on browser support, a date picker can show up in the input field.

Example

```
<form>
  Birthday (month and year):
  <input type="month" name="bdaymonth">
</form>
```

Try it Yourself »

Input Type Number

The `<input type="number">` defines a **numeric** input field.

You can also set restrictions on what numbers are accepted.

The following example displays a numeric input field, where you can enter a value from 1 to 5:

Example

```
<form>
  Quantity (between 1 and 5):
  <input type="number" name="quantity" min="1" max="5">
</form>
```

Try it Yourself »

Input Restrictions

Here is a list of some common input restrictions:

Attribute	Description
disabled	Specifies that an input field should be disabled
max	Specifies the maximum value for an input field
maxlength	Specifies the maximum number of character for an input field
min	Specifies the minimum value for an input field
pattern	Specifies a regular expression to check the input value against
readonly	Specifies that an input field is read only (cannot be changed)

required	Specifies that an input field is required (must be filled out)
size	Specifies the width (in characters) of an input field
step	Specifies the legal number intervals for an input field
value	Specifies the default value for an input field

You will learn more about input restrictions in the next chapter.

The following example displays a numeric input field, where you can enter a value from 0 to 100, in steps of 10. The default value is 30:

Example

```
<form>
  Quantity:
  <input type="number" name="points" min="0" max="100" step="10" value="30">
</form>
```

Try it Yourself »

Input Type Range

The `<input type="range">` defines a control for entering a number whose exact value is not important (like a slider control). Default range is 0 to 100. However, you can set restrictions on what numbers are accepted with the `min`, `max`, and `step` attributes:

Example

```
<form>
  <input type="range" name="points" min="0" max="10">
</form>
```

Try it Yourself »

Input Type Search

The `<input type="search">` is used for search fields (a search field behaves like a regular text field).

Example

```
<form>
  Search Google:
  <input type="search" name="googlesearch">
</form>
```

Try it Yourself »

Input Type Tel

The `<input type="tel">` is used for input fields that should contain a telephone number.

Example

```
<form>
  Telephone:
  <input type="tel" name="phone" pattern="[0-9]{3}-[0-9]{2}-[0-9]{3}">
</form>
```

Try it Yourself »

Input Type Time

The `<input type="time">` allows the user to select a time (no time zone).

Depending on browser support, a time picker can show up in the input field.

Example

```
<form>
  Select a time:
  <input type="time" name="usr_time">
</form>
```

Try it Yourself »

Input Type Url

The `<input type="url">` is used for input fields that should contain a URL address.

Depending on browser support, the url field can be automatically validated when submitted.

Some smartphones recognize the url type, and adds ".com" to the keyboard to match url input.

Example

```
<form>
  Add your homepage:
  <input type="url" name="homepage">
</form>
```

Try it Yourself »

Input Type Week

The `<input type="week">` allows the user to select a week and year.

Depending on browser support, a date picker can show up in the input field.

Example

```
<form>
  Select a week:
  <input type="week" name="week_year">
</form>
```

HTML Input Attributes

☐ PreviousNext ☐

The value Attribute

The `value` attribute specifies the initial value for an input field:

Example

```
<form action="">
  First name:<br>
  <input type="text" name="firstname" value="John">
</form>
```

Try it Yourself »

The readonly Attribute

The `readonly` attribute specifies that the input field is read only (cannot be changed):

Example

```
<form action="">
  First name:<br>
  <input type="text" name="firstname" value="John" readonly>
</form>
```

Try it Yourself »

The disabled Attribute

The `disabled` attribute specifies that the input field is disabled.

A disabled input field is unusable and un-clickable, and its value will not be sent when submitting the form:

Example

```
<form action="">
  First name:<br>
  <input type="text" name="firstname" value="John" disabled>
</form>
```

Try it Yourself »

The size Attribute

The `size` attribute specifies the size (in characters) for the input field:

Example

```
<form action="">
  First name:<br>
  <input type="text" name="firstname" value="John" size="40">
</form>
```

Try it Yourself »

The maxlength Attribute

The `maxlength` attribute specifies the maximum allowed length for the input field:

Example

```
<form action="">
  First name:<br>
  <input type="text" name="firstname" maxlength="10">
</form>
```

Try it Yourself »

With a `maxlength` attribute, the input field will not accept more than the allowed number of characters.

The `maxlength` attribute does not provide any feedback. If you want to alert the user, you must write JavaScript code.

Note: Input restrictions are not foolproof, and JavaScript provides many ways to add illegal input. To safely restrict input, it must be checked by the receiver (the server) as well!

HTML5 Attributes

HTML5 added the following attributes for `<input>`:

- autocomplete

- autofocus
- form
- formaction
- formenctype
- formmethod
- formnovalidate
- formtarget
- height and width
- list
- min and max
- multiple
- pattern (regexp)
- placeholder
- required
- step

and the following attributes for `<form>`:

- autocomplete
- novalidate

The autocomplete Attribute

The `autocomplete` attribute specifies whether a form or input field should have autocomplete on or off.

When autocomplete is on, the browser automatically completes the input values based on values that the user has entered before.

Tip: It is possible to have autocomplete "on" for the form, and "off" for specific input fields, or vice versa.

The `autocomplete` attribute works with `<form>` and the following `<input>` types: text, search, url, tel, email, password, datepickers, range, and color.

Example

An HTML form with autocomplete on (and off for one input field):

```
<form action="/action_page.php" autocomplete="on">
  First name:<input type="text" name="fname"><br>
  Last name: <input type="text" name="lname"><br>
  E-mail: <input type="email" name="email" autocomplete="off"><br>
  <input type="submit">
</form>
```

Try it Yourself »

Tip: In some browsers you may need to activate the autocomplete function for this to work.

The novalidate Attribute

The `novalidate` attribute is a `<form>` attribute.

When present, novalidate specifies that the form data should not be validated when submitted.

Example

Indicates that the form is not to be validated on submit:

```
<form action="/action_page.php" novalidate>
  E-mail: <input type="email" name="user_email">
  <input type="submit">
</form>
```

Try it Yourself »

The autofocus Attribute

The `autofocus` attribute specifies that the input field should automatically get focus when the page loads.

Example

Let the "First name" input field automatically get focus when the page loads:

```
First name:<input type="text" name="fname" autofocus>
```

Try it Yourself »

The form Attribute

The `form` attribute specifies one or more forms an `<input>` element belongs to.

Tip: To refer to more than one form, use a space-separated list of form ids.

Example

An input field located outside the HTML form (but still a part of the form):

```
<form action="/action_page.php" id="form1">
  First name: <input type="text" name="fname"><br>
  <input type="submit" value="Submit">
</form>

Last name: <input type="text" name="lname" form="form1">
```

Try it Yourself »

The formaction Attribute

The `formaction` attribute specifies the URL of a file that will process the input control when the form is submitted.

The formaction attribute overrides the action attribute of the `<form>` element.

The formaction attribute is used with `type="submit"` and `type="image"`.

Example

An HTML form with two submit buttons, with different actions:

```
<form action="/action_page.php">
  First name: <input type="text" name="fname"><br>
  Last name: <input type="text" name="lname"><br>
  <input type="submit" value="Submit"><br>
  <input type="submit" formaction="/action_page2.php"
  value="Submit as admin">
</form>
```

Try it Yourself »

The formenctype Attribute

The `formenctype` attribute specifies how the form data should be encoded when submitted (only for forms with method="post").

The `formenctype` attribute overrides the enctype attribute of the `<form>` element.

The `formenctype` attribute is used with `type="submit"` and `type="image"`.

Example

Send form-data that is default encoded (the first submit button), and encoded as "multipart/form-data" (the second submit button):

```
<form action="/action_page_binary.asp" method="post">
  First name: <input type="text" name="fname"><br>
  <input type="submit" value="Submit">
  <input type="submit" formenctype="multipart/form-data"
  value="Submit as Multipart/form-data">
</form>
```

Try it Yourself »

The formmethod Attribute

The `formmethod` attribute defines the HTTP method for sending form-data to the action URL.

The `formmethod` attribute overrides the method attribute of the `<form>` element.

The `formmethod` attribute can be used with `type="submit"` and `type="image"`.

Example

The second submit button overrides the HTTP method of the form:

```
<form action="/action_page.php" method="get">
  First name: <input type="text" name="fname"><br>
  Last name: <input type="text" name="lname"><br>
  <input type="submit" value="Submit">
  <input type="submit" formmethod="post" value="Submit using POST">
</form>
```

Try it Yourself »

The formnovalidate Attribute

The `formnovalidate` attribute overrides the novalidate attribute of the `<form>` element.

The `formnovalidate` attribute can be used with `type="submit"`.

Example

A form with two submit buttons (with and without validation):

```
<form action="/action_page.php">
  E-mail: <input type="email" name="userid"><br>
  <input type="submit" value="Submit"><br>
  <input type="submit" formnovalidate value="Submit without validation">
</form>
```

Try it Yourself »

The formtarget Attribute

The `formtarget` attribute specifies a name or a keyword that indicates where to display the response that is received after submitting the form.

The `formtarget` attribute overrides the target attribute of the `<form>` element.

The `formtarget` attribute can be used with `type="submit"` and `type="image"`.

Example

A form with two submit buttons, with different target windows:

```
<form action="/action_page.php">
  First name: <input type="text" name="fname"><br>
  Last name: <input type="text" name="lname"><br>
  <input type="submit" value="Submit as normal">
  <input type="submit" formtarget="_blank"
  value="Submit to a new window">
</form>
```

Try it Yourself »

The height and width Attributes

The `height` and `width` attributes specify the height and width of an `<input type="image">` element.

Always specify the size of images. If the browser does not know the size, the page will flicker while images load.

Example

Define an image as the submit button, with height and width attributes:

```
<input type="image" src="img_submit.gif" alt="Submit" width="48" height="48">
```

Try it Yourself »

The list Attribute

The `list` attribute refers to a `<datalist>` element that contains pre-defined options for an <input> element.

Example

An <input> element with pre-defined values in a <datalist>:

```
<input list="browsers">

<datalist id="browsers">
  <option value="Internet Explorer">
  <option value="Firefox">
  <option value="Chrome">
  <option value="Opera">
  <option value="Safari">
</datalist>
```

Try it Yourself »

The min and max Attributes

The `min` and `max` attributes specify the minimum and maximum values for an `<input>` element.

The `min` and `max` attributes work with the following input types: number, range, date, datetime-local, month, time and week.

Example

<input> elements with min and max values:

```
Enter a date before 1980-01-01:
<input type="date" name="bday" max="1979-12-31">

Enter a date after 2000-01-01:
<input type="date" name="bday" min="2000-01-02">

Quantity (between 1 and 5):
<input type="number" name="quantity" min="1" max="5">
```

Try it Yourself »

The multiple Attribute

The `multiple` attribute specifies that the user is allowed to enter more than one value in the `<input>` element.

The `multiple` attribute works with the following input types: email, and file.

Example

A file upload field that accepts multiple values:

```
Select images: <input type="file" name="img" multiple>
```

Try it Yourself »

The pattern Attribute

The `pattern` attribute specifies a regular expression that the `<input>` element's value is checked against.

The `pattern` attribute works with the following input types: text, search, url, tel, email, and password.

Tip: Use the global title attribute to describe the pattern to help the user.

Tip: Learn more about regular expressions in our JavaScript tutorial.

Example

An input field that can contain only three letters (no numbers or special characters):

```
Country code: <input type="text" name="country_code" pattern="[A-Za-
z]{3}" title="Three letter country code">
```

Try it Yourself »

The placeholder Attribute

The `placeholder` attribute specifies a hint that describes the expected value of an input field (a sample value or a short description of the format).

The hint is displayed in the input field before the user enters a value.

The `placeholder` attribute works with the following input types: text, search, url, tel, email, and password.

Example

An input field with a placeholder text:

```
<input type="text" name="fname" placeholder="First name">
```

Try it Yourself »

The required Attribute

The `required` attribute specifies that an input field must be filled out before submitting the form.

The `required` attribute works with the following input types: text, search, url, tel, email, password, date pickers, number, checkbox, radio, and file.

Example

A required input field:

```
Username: <input type="text" name="usrname" required>
```

Try it Yourself »

The step Attribute

The `step` attribute specifies the legal number intervals for an `<input>` element.

Example: if step="3", legal numbers could be -3, 0, 3, 6, etc.

Tip: The step attribute can be used together with the max and min attributes to create a range of legal values.

The `step` attribute works with the following input types: number, range, date, datetime-local, month, time and week.

Example

An input field with a specified legal number intervals:

```
<input type="number" name="points" step="3">
```

Try it Yourself »

HTML Exercises

Exercise:

In the input field below, add placeholder that says "Your name here".

```
<form action="/action_page.php">
<input type="text" >
</form>
```

Submit Answer »

Start the Exercise

HTML Form and Input Elements

Tag	Description
<form>	Defines an HTML form for user input
<input>	Defines an input control

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 element is a block-level element.

Example

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

Try it Yourself »

Block level elements in HTML:

```
<address>
```

<article>

<aside>

<blockquote>

<canvas>

<dd>

<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

```
<span>Hello World</span>
```

Try it Yourself »

Inline elements in HTML:

<a>

<abbr>

<acronym>

<b>

<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 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>
```

Try it Yourself »

The Element

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

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

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

Example

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

Try it Yourself »

HTML Grouping Tags

Tag	Description
	Defines a section in a document (block-level)
	Defines a section in a document (inline)

HTML The class Attribute

Using The class Attribute

The HTML `class` attribute is used to define equal styles for elements with the same class name.

So, all HTML elements with the same `class` attribute will get the same style.

Here we have three `<div>` elements that point to the same class name:

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
.cities {
```

```
  background-color: black;
  color: white;
  margin: 20px;
  padding: 20px;
}
</style>
</head>
<body>

<div class="cities">
  <h2>London</h2>
  <p>London is the capital of England.</p>
</div>

<div class="cities">
  <h2>Paris</h2>
  <p>Paris is the capital of France.</p>
</div>

<div class="cities">
  <h2>Tokyo</h2>
  <p>Tokyo is the capital of Japan.</p>
</div>

</body>
</html>
```

Using The class Attribute on Inline Elements

The HTML `class` attribute can also be used on inline elements:

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
span.note {
  font-size: 120%;
  color: red;
}
</style>
```

```
</head>
<body>

<h1>My <span class="note">Important</span> Heading</h1>
<p>This is some <span class="note">important</span> text.</p>

</body>
</html>
```

Try it Yourself »

Tip: The `class` attribute can be used on **any** HTML element.

Note: The class name is case sensitive!

Tip: You can learn much more about CSS in our CSS Tutorial.

Select Elements With a Specific Class

In CSS, to select elements with a specific class, write a period (.) character, followed by the name of the class:

Example

Use CSS to style all elements with the class name "city":

```
<style>
.city {
  background-color: tomato;
  color: white;
  padding: 10px;
}
</style>

<h2 class="city">London</h2>
<p>London is the capital of England.</p>

<h2 class="city">Paris</h2>
<p>Paris is the capital of France.</p>
```

```
<h2 class="city">Tokyo</h2>
<p>Tokyo is the capital of Japan.</p>
```

Result:

London

London is the capital of England.

Paris

Paris is the capital of France.

Tokyo

Tokyo is the capital of Japan.

Try it Yourself »

Multiple Classes

HTML elements can have more than one class name, each class name must be separated by a space.

Example

Style elements with the class name "city", also style elements with the class name "main":

```
<h2 class="city main">London</h2>
<h2 class="city">Paris</h2>
<h2 class="city">Tokyo</h2>
```

Try it Yourself »

In the example above, the first `<h2>` element belongs to both the "city" class and the "main" class.

Different Tags Can Share Same Class

Different tags, like `<h2>` and `<p>`, can have the same class name and thereby share the same style:

Example

```
<h2 class="city">Paris</h2>
<p class="city">Paris is the capital of France</p>
```

Try it Yourself »

Using The class Attribute in JavaScript

The class name can also be used by JavaScript to perform certain tasks for elements with the specified class name.

JavaScript can access elements with a specified class name by using the `getElementsByClassName()` method:

Example

When a user clicks on a button, hide all elements with the class name "city":

```
<script>
function myFunction() {
  var x = document.getElementsByClassName("city");
  for (var i = 0; i < x.length; i++) {
    x[i].style.display = "none";
  }
}
</script>
```

HTML The id Attribute

☐ PreviousNext ☐

Using The id Attribute

The `id` attribute specifies a unique id for an HTML element (the value must be unique within the HTML document).

The id value can be used by CSS and JavaScript to perform certain tasks for the element with the specific id value.

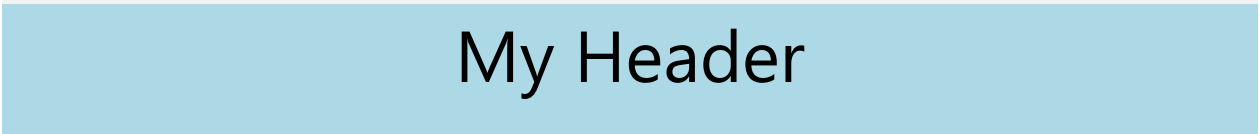
In CSS, to select an element with a specific id, write a hash (#) character, followed by the id of the element:

Example

Use CSS to style an element with the id "myHeader":

```
<style>
#myHeader {
  background-color: lightblue;
  color: black;
  padding: 40px;
  text-align: center;
}
</style>

<h1 id="myHeader">My Header</h1>
```

Result:



Try it Yourself »

Tip: The id attribute can be used on **any** HTML element.

Note: The id value is case-sensitive.

Note: The id value must contain at least **one** character, and must **not** contain whitespace (spaces, tabs, etc.).

Difference Between Class and ID

An HTML element can only have one unique id that belongs to that single element, while a class name can be used by multiple elements:

Example

```
<style>
/* Style the element with the id "myHeader" */
#myHeader {
  background-color: lightblue;
  color: black;
  padding: 40px;
  text-align: center;
}

/* Style all elements with the class name "city" */
.city {
  background-color: tomato;
  color: white;
  padding: 10px;
}
</style>

<!-- A unique element -->
<h1 id="myHeader">My Cities</h1>

<!-- Multiple similar elements -->
<h2 class="city">London</h2>
<p>London is the capital of England.</p>

<h2 class="city">Paris</h2>
<p>Paris is the capital of France.</p>

<h2 class="city">Tokyo</h2>
<p>Tokyo is the capital of Japan.</p>
```

Try it Yourself »

Tip: You can learn much more about CSS in our CSS Tutorial.

Bookmarks with ID and Links

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>
```

Try it Yourself »

Using The id Attribute in JavaScript

JavaScript can access an element with a specified id by using the `getElementById()` method:

Example

Use the id attribute to manipulate text with JavaScript:

```
<script>
function displayResult() {
  document.getElementById("myHeader").innerHTML = "Have a nice day!";
}
</script>
```

Difference Between Class and ID