Introduction

HTML5 is the next major revision of the HTML standard superseding HTML 4.01, XHTML 1.0, and XHTML 1.1. HTML5 is a standard for structuring and presenting content on the World Wide Web.

HTML5 is a cooperation between the World Wide Web Consortium (W3C) and the Web Hypertext Application Technology Working Group (WHATWG).

The new standard incorporates features like video playback and drag-and-drop that have been previously dependent on third-party browser plug-ins such as Adobe Flash, Microsoft Silverlight, and Google Gears.

What is HTML?

HTML is the standard markup language for creating Web pages.

- HTML stands for Hyper Text Markup Language
- **Hypertext** refers to the way in which Web pages (HTML documents) are linked together. Thus, the link available on a webpage is called Hypertext.
- As its name suggests, HTML is a **Markup Language** which means you use HTML to simply "mark-up" a text document with tags that tell a Web browser how to structure it to display.
- HTML describes the structure of Web pages using markup
- HTML elements are the building blocks of HTML pages
- 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

Browser Support

The latest versions of Apple Safari, Google Chrome, Mozilla Firefox, and Opera all support many HTML5 features and Internet Explorer 9.0 will also have support for some HTML5 functionality.

The mobile web browsers that come pre-installed on iPhones, iPads, and Android phones all have excellent support for HTML5.

New Features

HTML5 introduces a number of new elements and attributes that helps in building a modern website. Following are great features introduced in HTML5.

- **New Semantic Elements** − These are like <header>, <footer>, and <section>.
- **Forms 2.0** − Improvements to HTML web forms where new attributes have been introduced for <input> tag.
- **Persistent Local Storage** − To achieve without resorting to third-party plugins.
- **WebSocket** − A a next-generation bidirectional communication technology for web applications.

- **Server-Sent Events** − HTML5 introduces events which flow from web server to the web browsers and they are called Server-Sent Events (SSE).
- **Canvas** − This supports a two-dimensional drawing surface that you can program with JavaScript.
- **Audio & Video** − You can embed audio or video on your web pages without resorting to third-party plugins.
- **Geolocation** − Now visitors can choose to share their physical location with your web application.
- **Microdata** − This lets you create your own vocabularies beyond HTML5 and extend your web pages with custom semantics.
- **Drag and drop** − Drag and drop the items from one location to another location on a the same webpage.

Backward Compatibility

HTML5 is designed, as much as possible, to be backward compatible with existing web browsers. New features build on existing features and allow you to provide fallback content for older browsers.

It is suggested to detect support for individual HTML5 features using a few lines of JavaScript.

If you are not familiar with any previous version of HTML, I would recommend to go through our HTML Tutorial before you explore further concepts of HTM5.

HTML Document Structure

Example

```
<!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 Comment Tags

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

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

Notice that there is an exclamation point (!) in the opening tag, but not in the closing tag.
Note: Comments are not displayed by the browser, but they can help document your HTML source code.
With comments you can place notifications and reminders in your HTML:

Example

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

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

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

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

Example

```
<!-- Do not display this at the moment
<img border="0" src="pic_mountain.jpg" alt="Mountain">
-->
```

Conditional Comments

You might stumble upon conditional comments in HTML:

```
<!--[if IE 9]>
   .... some HTML here ....
<![endif]-->
```

Conditional comments defines some HTML tags to be executed by Internet Explorer only.

HTML Tags

HTML tags are element names surrounded by angle brackets:

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

- HTML tags normally come **in pairs** like and
- 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**.

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>
```

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">
```

The width and height Attributes

Images in HTML have a set of size attributes, which specifies the width and height of the image:

Example

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

The image size is specified in pixels: width="500" means 500 pixels wide.

The alt Attribute

The **alt** attribute specifies an alternative text to be used, when an image cannot be displayed.
The value of the attribute can be read by screen readers. This way, someone "listening" to the webpage, e.g. a blind person, can "hear" the element.

Example

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

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 four 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 "Ignore rich text commands in HTML files".
Then open a new document to place the code.

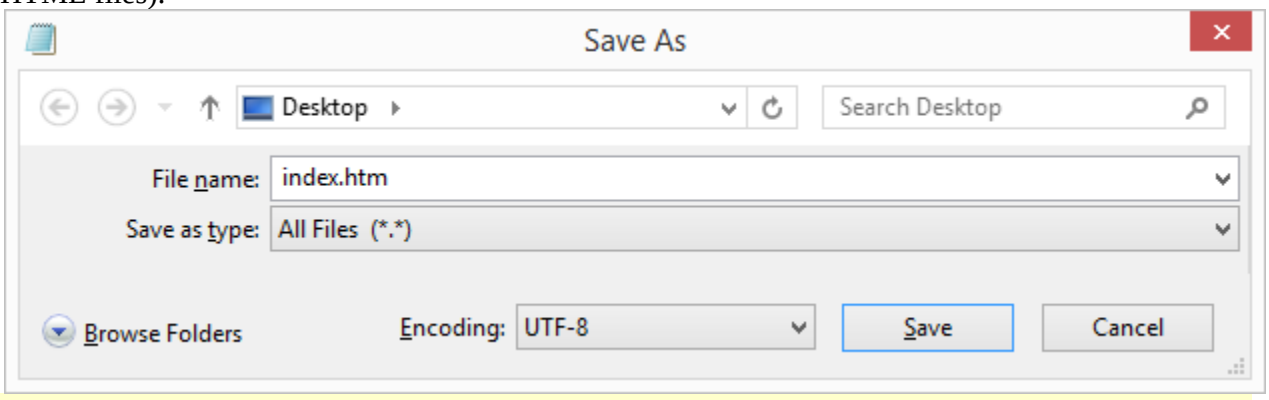
Step 2: Write Some HTML

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

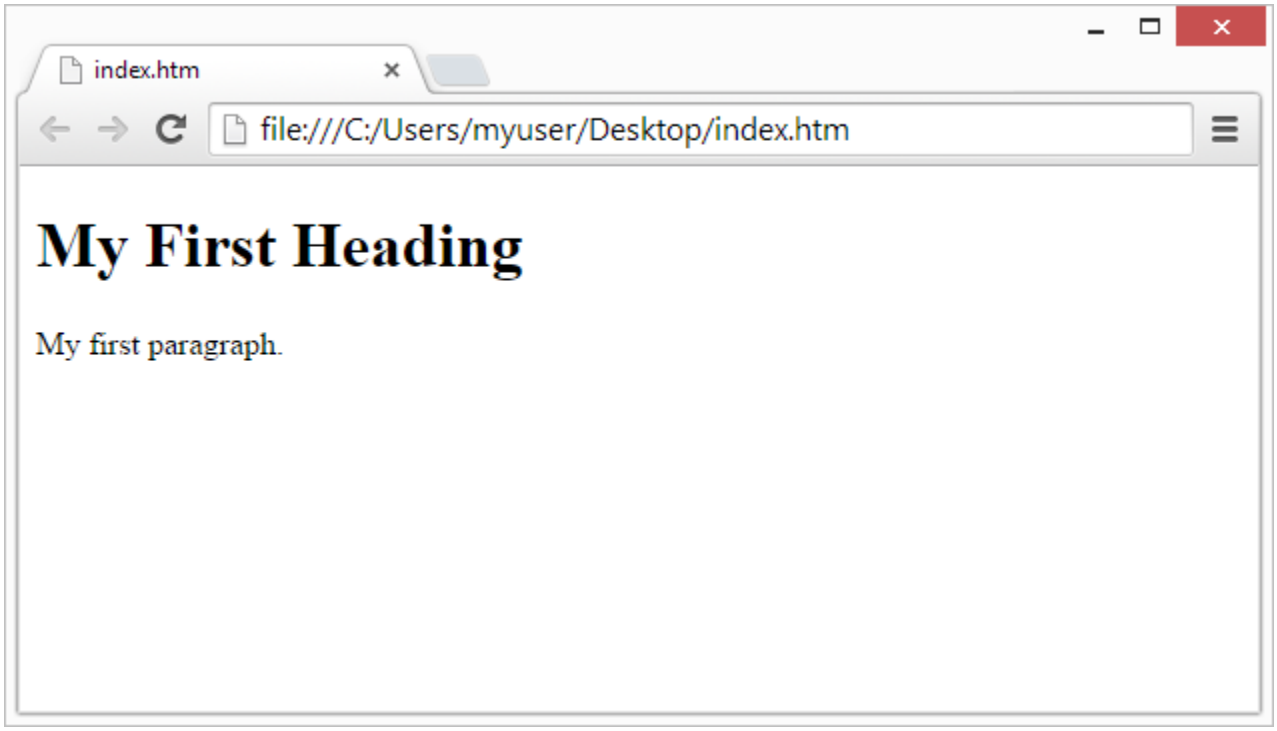
Save the file on your computer. Select **File** > **Save as** in the Notepad menu.
Name the file **"index.htm"** and set the encoding to **UTF-8** (which is the preferred encoding for HTML files).



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:



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>**.

Text Formatting Tags

HTML Formatting is a process of formatting text for better look and feel. There are many formatting tags in HTML. These tags are used to make text bold, italicized, or underlined. There are almost 12 options available that how text appears in HTML and XHTML.

Heading Tags

Any document starts with a heading. You can use different sizes for your headings. HTML also has six levels of headings, which use the elements **<h1>, <h2>, <h3>, <h4>, <h5>,** and **<h6>**. While displaying any heading, browser adds one line before and one line after that heading.

Example

Live Demo

```
<!DOCTYPE html>
<html>

  <head>
    <title>Heading Example</title>
  </head>

  <body>
    <h1>This is heading 1</h1>
    <h2>This is heading 2</h2>
    <h3>This is heading 3</h3>
    <h4>This is heading 4</h4>
    <h5>This is heading 5</h5>
    <h6>This is heading 6</h6>
  </body>

</html>
```

This will produce the following result −

Paragraph Tag

The ** ** tag offers a way to structure your text into different paragraphs. Each paragraph of text should go in between an opening and a closing tag as shown below in the example −

Example

Live Demo

```
<!DOCTYPE html>
<html>

  <head>
    <title>Paragraph Example</title>
  </head>

  <body>
    <p>Here is a first paragraph of text.</p>
    <p>Here is a second paragraph of text.</p>
    <p>Here is a third paragraph of text.</p>
  </body>

</html>
```

This will produce the following result −

Line Break Tag

Whenever you use the **
** element, anything following it starts from the next line. This tag is an example of an **empty** element, where you do not need opening and closing tags, as there is nothing to go in between them.

The
 tag has a space between the characters **br** and the forward slash. If you omit this space, older browsers will have trouble rendering the line break, while if you miss the forward slash character and just use
 it is not valid in XHTML.

Example

Live Demo

```
<!DOCTYPE html>
<html>

  <head>
    <title>Line Break  Example</title>
  </head>

  <body>
    <p>Hello<br />
      You delivered your assignment ontime.<br />
      Thanks<br />
      Mahnaz</p>
  </body>

</html>
```

This will produce the following result −

Centering Content

You can use ** ** tag to put any content in the center of the page or any table cell.

Example

Live Demo

```
<!DOCTYPE html>
<html>

  <head>
    <title>Centring Content Example</title>
  </head>

  <body>
    <p>This text is not in the center.</p>
```

```
    <center>
      <p>This text is in the center.</p>
    </center>
  </body>

</html>
```

This will produce following result −

Horizontal Lines

Horizontal lines are used to visually break-up sections of a document. The **<hr>** tag creates a line from the current position in the document to the right margin and breaks the line accordingly.

For example, you may want to give a line between two paragraphs as in the given example below −

Example

Live Demo

```
<!DOCTYPE html>
<html>

  <head>
    <title>Horizontal Line Example</title>
  </head>

  <body>
    <p>This is paragraph one and should be on top</p>
    <hr />
    <p>This is paragraph two and should be at bottom</p>
  </body>

</html>
```

This will produce the following result −

Again **<hr />** tag is an example of the **empty** element, where you do not need opening and closing tags, as there is nothing to go in between them.

The **<hr />** element has a space between the characters **hr** and the forward slash. If you omit this space, older browsers will have trouble rendering the horizontal line, while if you miss the forward slash character and just use **<hr>** it is not valid in XHTML

Preserve Formatting

Sometimes, you want your text to follow the exact format of how it is written in the HTML document. In these cases, you can use the preformatted tag **<pre>**.

Any text between the opening **<pre>** tag and the closing **</pre>** tag will preserve the formatting of the source document.

Example

Live Demo

```
<!DOCTYPE html>
<html>

  <head>
    <title>Preserve Formatting Example</title>
  </head>

  <body>
    <pre>
      function testFunction( strText ){
        alert (strText)
      }
    </pre>
  </body>

</html>
```

This will produce the following result −

Try using the same code without keeping it inside **<pre>...</pre>** tags

HTML <font> Tag. Not Supported in HTML5.

The <font> tag is not supported in HTML5. Use CSS instead.
The <font> tag specifies the font face, font size, and color of text.

Example

Specify the font size, font face and color of text:

```
<font size="3" color="red">This is some text!</font>
<font size="2" color="blue">This is some text!</font>
<font face="verdana" color="green">This is some text!</font>
```

Optional Attributes

Attribute	Value	Description
color	*rgb(x,x,x)*	Not supported in HTML5.

	#xxxxxx colorname	Specifies the color of text
face	*font_family*	Not supported in HTML5. Specifies the font of text
size	*number*	Not supported in HTML5. Specifies the size of text

Nonbreaking Spaces

Suppose you want to use the phrase "12 Angry Men." Here, you would not want a browser to split the "12, Angry" and "Men" across two lines −

```
An example of this technique appears in the movie "12 Angry Men."
```

In cases, where you do not want the client browser to break text, you should use a nonbreaking space entity ** ** instead of a normal space. For example, when coding the "12 Angry Men" in a paragraph, you should use something similar to the following code −

Example

```
<!DOCTYPE html>
<html>

  <head>
    <title>Nonbreaking Spaces Example</title>
  </head>

  <body>
    <p>An example of this technique appears in the movie "12 Angry Men."</p>
  </body>

</html>
```

Bold Text

If you write anything within ..... element, is shown in bold letters.
See this example:

1. ** **Write Your First Paragraph in bold text.** **

Output:
Write Your First Paragraph in bold text.

Italic Text

If you write anything within ..... element, is shown in italic letters.
See this example:

```
<p> <i>Write Your First Paragraph in italic text.</i></p>
```

Output:
Write Your First Paragraph in italic text.

HTML Marked formatting

If you want to mark or highlight a text, you should write the content within <mark>.....</mark>.
See this example:

```
<h2>  I want to put a <mark> Mark</mark> on your face</h2>
```

Output:

I want to put a Mark on your face

Underlined Text

If you write anything within <u>.....</u> element, is shown in underlined text.
See this example:

```
<p> <u>Write Your First Paragraph in underlined text.</u></p>
```

Output:
Write Your First Paragraph in underlined text.

Strike Text

Anything written within <strike>.....</strike> element is displayed with strikethrough. It is a thin line which cross the statement.
See this example:

```
<p> <strike>Write Your First Paragraph with strikethrough</strike>.</p>
```

Output:
~~Write Your First Paragraph with strikethrough.~~

Monospaced Font

If you want that each letter has the same width then you should write the content within <tt>.....</tt> element.
Note: We know that most of the fonts are known as variable-width fonts because different letters have different width. (for example: 'w' is wider than 'i'). Monospaced Font provides similar space among every letter.
See this example:

```
<p>Hello <tt>Write Your First Paragraph in monospaced font.</tt></p>
```

Output:
Hello Write Your First Paragraph in monospaced font.

Superscript Text

If you put the content within ^{.....} element, is shown in superscript ; means it is displayed half a character's height above the other characters.
See this example:

```
<p>Hello <sup>Write Your First Paragraph in superscript.</sup></p>
```

Output:
Hello Write Your First Paragraph in superscript.

Subscript Text

If you put the content within _{.....} element, is shown in subscript ; means it is displayed half a character's height below the other characters.
See this example:

```
<p>Hello <sub>Write Your First Paragraph in subscript.</sub></p>
```

Output:
Hello Write Your First Paragraph in subscript.

Deleted Text

Anything that puts within ~~.....~~ is displayed as deleted text.
See this example:

```
<p>Hello <del>Delete your first paragraph.</del></p>
```

Output:
Hello ~~Delete your first paragraph.~~

Inserted Text

Anything that puts within <ins>.....</ins> is displayed as inserted text.
See this example:

```
<p> <del>Delete your first paragraph.</del><ins>Write another paragraph.</ins></p>
```

Output:
Delete your first paragraph.Write another paragraph.

Larger Text

If you want to put your font size larger than the rest of the text then put the content within <big>.....</big>. It increase one font size larger than the previous one.
See this example:

```
<p>Hello <big>Write the paragraph in larger font.</big></p>
```

Output:
Hello Write the paragraph in larger font.

Smaller Text

If you want to put your font size smaller than the rest of the text then put the content within <small>.....</small>tag. It reduces one font size than the previous one.
See this example:

```
<p>Hello <small>Write the paragraph in smaller font.</small></p>
```

Output:

Hello Write the paragraph in smaller font.

HTML Image

HTML img tag is used to display image on the web page. HTML img tag is an empty tag that contains attributes only, closing tags are not used in HTML image element.
Let's see an example of HTML image.

```
<h2>HTML Image Example</h2>
<img src="good_morning.jpg" alt="Good Morning Friends"/>
```

Output:

Attributes of HTML img tag

The src and alt are important attributes of HTML img tag. All attributes of HTML image tag are given below.

1) src

It is a necessary attribute that describes the source or path of the image. It instructs the browser where to look for the image on the server.
The location of image may be on the same directory or another server.

2) alt

The alt attribute defines an alternate text for the image, if it can't be displayed. The value of the alt attribute describe the image in words. The alt attribute is considered good for SEO prospective.

3) width

It is an optional attribute which is used to specify the width to display the image. It is not recommended now. You should apply CSS in place of width attribute.

4) height

It specifies the height of the image. The HTML height attribute also supports iframe, image and object elements. It is not recommended now. You should apply CSS in place of height attribute.

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

In HTML, links are defined with the **<a>** tag:

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

Example

```
<a href="https://www.gitam.edu/">Visit our GITAM</a>
```

The **href** attribute specifies the destination address (https://www.gitam.edu/html/) of the link.
The **link text** is the visible part (Visit our GITAM).
Clicking on the link text will send you to the specified address.
Note: Without a forward slash on subfolder addresses, you might generate two requests to the server. Many servers will automatically add a forward slash to 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 http://www....).

Example

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

HTML Link Colors

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

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

You can change the default colors, by using styles:

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 »

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>
```

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).

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>
```

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>
```

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>
```

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

Example

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

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

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

HTML Table

HTML table tag is used to display data in tabular form (row * column). There can be many columns in a row.

HTML tables are used to manage the layout of the page e.g. header section, navigation bar, body content, footer section etc. But it is recommended to use div tag over table to manage the layout of the page .

HTML Table Tags

Tag	Description
<table>	It defines a table.
<tr>	It defines a row in a table.
<th>	It defines a header cell in a table.
<td>	It defines a cell in a table.
<caption>	It defines the table caption.
<colgroup>	It specifies a group of one or more columns in a table for formatting.
<col>	It is used with <colgroup> element to specify column properties for each column.
<tbody>	It is used to group the body content in a table.
<thead>	It is used to group the header content in a table.
<tfoot>	It is used to group the footer content in a table.

HTML Table Example

Let's see the example of HTML table tag. It output is shown above.

```
<table>
<tr><th>First_Name</th><th>Last_Name</th><th>Marks</th></tr>
<tr><td>Sonoo</td><td>Jaiswal</td><td>60</td></tr>
<tr><td>James</td><td>William</td><td>80</td></tr>
<tr><td>Swati</td><td>Sironi</td><td>82</td></tr>
<tr><td>Chetna</td><td>Singh</td><td>72</td></tr>
</table>
```

Output:

First_Name	Last_Name	Marks
Sonoo	Jaiswal	60
James	William	80
Swati	Sironi	82
Chetna	Singh	72

In the above html table, there are 5 rows and 3 columns = 5 * 3 = 15 values.

HTML Table with Border

There are two ways to specify border for HTML tables.

1. By border attribute of table in HTML
2. By border property in CSS

---

1) HTML Border attribute

You can use border attribute of table tag in HTML to specify border. But it is not recommended now.

```
<table border="1">
<tr><th>First_Name</th><th>Last_Name</th><th>Marks</th></tr>
<tr><td>Sonoo</td><td>Jaiswal</td><td>60</td></tr>
<tr><td>James</td><td>William</td><td>80</td></tr>
<tr><td>Swati</td><td>Sironi</td><td>82</td></tr>
<tr><td>Chetna</td><td>Singh</td><td>72</td></tr>
</table>
```

Output:

First_Name	Last_Name	Marks
Sonoo	Jaiswal	60
James	William	80
Swati	Sironi	82

Chetna	Singh	72

---

2) CSS Border property

It is now recommended to use border property of CSS to specify border in table.

```
<style>
table, th, td {
   border: 1px solid black;
}
</style>
```

You can collapse all the borders in one border by border-collapse property.

```
<style>
table, th, td {
   border: 2px solid black;
   border-collapse: collapse;
}
</style>
```

Output:

Name	Last Name	Marks
Sonoo	Jaiswal	60
James	William	80
Swati	Sironi	82
Chetna	Singh	72

HTML Table with cell padding

You can specify padding for table header and table data by two ways:

1. By cellpadding attribute of table in HTML
2. By padding property in CSS

The cellpadding attribute of HTML table tag is obselete now. It is recommended to use CSS. So let's see the code of CSS.

```
<style>
table, th, td {
  border: 1px solid pink;
  border-collapse: collapse;
}
th, td {
  padding: 10px;
}
</style>
```

Output:

Name	Last Name	Marks
Sonoo	Jaiswal	60
James	William	80
Swati	Sironi	82
Chetna	Singh	72

HTML Table with colspan

If you want to make a cell span more than one column, you can use the colspan attribute.
Let's see the example that span two columns.
CSS code:

```
<style>
table, th, td {
  border: 1px solid black;
  border-collapse: collapse;
}
th, td {
  padding: 5px;
```

```
}
</style>
```

HTML code:

```
<table style="width:100%">
  <tr>
    <th>Name</th>
    <th colspan="2">Mobile No.</th>
  </tr>
  <tr>
    <td>Ajeet Maurya</td>
    <td>7503520801</td>
    <td>9555879135</td>
  </tr>
</table>
```

Output:

Name	**Mobile No.**	
Ajeet Maurya	7503520801	9555879135

HTML Table with rowspan

If you want to make a cell span more than one row, you can use the rowspan attribute.
Let's see the example that span two rows.
CSS code:

```
<style>
table, th, td {
  border: 1px solid black;
  border-collapse: collapse;
}
th, td {
  padding: 10px;
}
</style>
```

HTML code:

```
<table>
<tr><th>Name</th><td>Ajeet Maurya</td></tr>
<tr><th rowspan="2">Mobile No.</th><td>7503520801</td></tr>
<tr><td>9555879135</td></tr>
</table>
```

Output:

Name	Ajeet Maurya
Mobile No.	7503520801
	9555879135

HTML table with caption

HTML caption is diplayed above the table. It must be used after table tag only.

```
<table>
<caption>Student Records</caption>
<tr><th>First_Name</th><th>Last_Name</th><th>Marks</th></tr>
<tr><td>Vimal</td><td>Jaiswal</td><td>70</td></tr>
<tr><td>Mike</td><td>Warn</td><td>60</td></tr>
<tr><td>Shane</td><td>Warn</td><td>42</td></tr>
<tr><td>Jai</td><td>Malhotra</td><td>62</td></tr>
</table>
```

Styling HTML table even and odd cells

CSS code:

```
<style>
table, th, td {
  border: 1px solid black;
  border-collapse: collapse;
```

```
}
th, td {
   padding: 10px;
}
table#alter tr:nth-child(even) {
   background-color: #eee;
}
table#alter tr:nth-child(odd) {
   background-color: #fff;
}
table#alter th {
   color: white;
   background-color: gray;
}
</style>
```

Output:

First_Name	Last_Name	Marks
Sonoo	Jaiswal	60
James	William	80
Swati	Sironi	82
Chetna	Singh	72

- 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 Lists

HTML Lists are used to specify lists of information. All lists may contain one or more list elements. There are three different types of HTML lists:

1. Ordered List or Numbered List (ol)
2. Unordered List or Bulleted List (ul)
3. Description List or Definition List (dl)

HTML Ordered List or Numbered List

In the ordered HTML lists, all the list items are marked with numbers. It is known as numbered list also. The ordered list starts with <ol> tag and the list items start with <li> tag.

```
<ol>
 <li>Aries</li>
 <li>Bingo</li>
 <li>Leo</li>
 <li>Oracle</li>
</ol>
```

Output:

1. Aries
2. Bingo
3. Leo
4. Oracle

Click here for full details of HTML ordered list. HTML Ordered List

---

HTML Unordered List or Bulleted List

In HTML Unordered list, all the list items are marked with bullets. It is also known as bulleted list also. The Unordered list starts with <ul> tag and list items start with the <li> tag.

```
<ul>
 <li>Aries</li>
 <li>Bingo</li>
```

```
 <li>Leo</li>
 <li>Oracle</li>
</ul>
```

Output:

- Aries
- Bingo
- Leo
- Oracle

Click here for full details of HTML unordered list. HTML Unordered List

---

HTML Description List or Definition List

HTML Description list is also a list style which is supported by HTML and XHTML. It is also known as definition list where entries are listed like a dictionary or encyclopedia.
The definition list is very appropriate when you want to present glossary, list of terms or other name-value list.
The HTML definition list contains following three tags:

1. **<dl> tag** defines the start of the list.
2. **<dt> tag** defines a term.
3. **<dd> tag** defines the term definition (description).

```
<dl>
 <dt>Aries</dt>
 <dd>-One of the 12 horoscope sign.</dd>
 <dt>Bingo</dt>
 <dd>-One of my evening snacks</dd>
 <dt>Leo</dt>
 <dd>-It is also an one of the 12 horoscope sign.</dd>
 <dt>Oracle</dt>
 <dd>-It is a multinational technology corporation.</dd>
</dl>
```

Test it Now

Output:

Aries
-One of the 12 horoscope sign.
Bingo
-One of my evening snacks
Leo
-It is also an one of the 12 horoscope sign.
Oracle
-It is a multinational technology corporation.

HTML Ordered List | HTML Numbered List

HTML Ordered List or Numbered List displays elements in numbered format. The HTML ol tag is used for ordered list. There can be different types of numbered list:

- Numeric Number (1, 2, 3)
- Capital Roman Number (I II III)
- Small Romal Number (i ii iii)
- Capital Alphabet (A B C)
- Small Alphabet (a b c)

To represent different ordered lists, there are 5 types of attributes in <ol> tag.

Type	Description
Type "1"	This is the default type. In this type, the list items are numbered with numbers.
Type "I"	In this type, the list items are numbered with upper case roman numbers.
Type "i"	In this type, the list items are numbered with lower case roman numbers.
Type "A"	In this type, the list items are numbered with upper case letters.
Type "a"	In this type, the list items are numbered with lower case letters.

HTML Ordered List Example

Let's see the example of HTML ordered list that displays 4 topics in numbered list. Here we are not defining type="1" because it is the default type.

```
<ol>
 <li>HTML</li>
 <li>Java</li>
```

```
 <li>JavaScript</li>
 <li>SQL</li>
</ol>
```

Output:

1. HTML
2. Java
3. JavaScript
4. SQL

ol type="I"

Let's see the example to display list in roman number uppercase.

```
<ol type="I">
 <li>HTML</li>
 <li>Java</li>
 <li>JavaScript</li>
 <li>SQL</li>
</ol>
```

Output:

I. HTML
II. Java
III. JavaScript
IV. SQL

ol type="i"

Let's see the example to display list in roman number lowercase.

```
<ol type="i">
 <li>HTML</li>
 <li>Java</li>
 <li>JavaScript</li>
 <li>SQL</li>
</ol>
```

Output:

i. HTML
ii. Java
iii. JavaScript
iv. SQL

ol type="A"

Let's see the example to display list in alphabet uppercase.

```
<ol type="A">
 <li>HTML</li>
 <li>Java</li>
 <li>JavaScript</li>
 <li>SQL</li>
</ol>
```

Output:

A. HTML
B. Java
C. JavaScript
D. SQL

---

ol type="a"

Let's see the example to display list in alphabet lowercase.

```
<ol type="a">
 <li>HTML</li>
 <li>Java</li>
 <li>JavaScript</li>
 <li>SQL</li>
</ol>
```

Output:

a. HTML

b. Java
c. JavaScript
d. SQL

start attribute

The start attribute is used with ol tag to specify from where to start the list items.
<ol type="1" start="5"> : It will show numeric values starting with "5".
<ol type="A" start="5"> : It will show capital alphabets starting with "E".
<ol type="a" start="5"> : It will show lower case alphabets starting with "e".
<ol type="I" start="5"> : It will show Roman upper case value starting with "V".
<ol type="i" start="5"> : It will show Roman lower case value starting with "v".

```
<ol type="i" start="5">
 <li>HTML</li>
 <li>Java</li>
 <li>JavaScript</li>
 <li>SQL</li>
</ol>
```

Output:

v. HTML
vi. Java
vii. JavaScript
viii. SQL

HTML Unordered List | HTML Bulleted List

HTML Unordered List or Bulleted List displays elements in bulleted format. The HTML ul tag is used for the unordered list. There can be 4 types of bulleted list:

- disc
- circle
- square
- none

To represent different ordered lists, there are 4 types of attributes in <ul> tag.

Type	Description

Type "disc"	This is the default style. In this style, the list items are marked with bullets.
Type "circle"	In this style, the list items are marked with circles.
Type "square"	In this style, the list items are marked with squares.
Type "none"	In this style, the list items are not marked .

HTML Unordered List Example

```
<ul>
 <li>HTML</li>
 <li>Java</li>
 <li>JavaScript</li>
 <li>SQL</li>
</ul>
```

Output:

- HTML
- Java
- JavaScript
- SQL

ul type="circle"

```
<ul type="circle">
 <li>HTML</li>
 <li>Java</li>
 <li>JavaScript</li>
 <li>SQL</li>
</ul>
```

Output:

- HTML
- Java
- JavaScript

- SQL

ul type="square"

```
<ul type="square">
 <li>HTML</li>
 <li>Java</li>
 <li>JavaScript</li>
 <li>SQL</li>
</ul>
```

Output:

- HTML
- Java
- JavaScript
- SQL

ul type="none"

```
<ul type="none">
 <li>HTML</li>
 <li>Java</li>
 <li>JavaScript</li>
 <li>SQL</li>
</ul>
```

Output:

- HTML
- Java
- JavaScript
- SQL

HTML Description List | HTML Definition List

HTML Description List or Definition List displays elements in definition form like in dictionary. The <dl>, <dt> and <dd> tags are used to define description list.
The 3 HTML description list tags are given below:

1. **<dl> tag** defines the description list.
2. **<dt> tag** defines data term.
3. **<dd> tag** defines data definition (description).

```
<dl>
 <dt>HTML</dt>
 <dd>is a markup language</dd>
 <dt>Java</dt>
 <dd>is a programming language and platform</dd>
<dt>JavaScript</dt>
<dd>is a scripting language</dd>
 <dt>SQL</dt>
 <dd>is a query language</dd>
</dl>
```

Output:

HTML
    is a markup language
Java
    is a programming language and platform
JavaScript
    is a scripting language
SQL
    is a query language

HTML Form

An **HTML form** is *a section of a document* which contains controls such as text fields, password fields, checkboxes, radio buttons, submit button, menus etc.
An HTML form facilitates the user to enter data that is to be sent to the server for processing.

Why use HTML Form

HTML forms are required if you want to collect some data from of the site visitor.
For example: If a user want to purchase some items on internet, he/she must fill the form such as shipping address and credit/debit card details so that item can be sent to the given address.

HTML Form Syntax

```
<form action="server url" method="get|post">
 //input controls e.g. textfield, textarea, radiobutton, button
```

</form>

Form Attributes

Apart from common attributes, following is a list of the most frequently used form attributes –

Sr.No	Attribute & Description
1	**Action** Backend script ready to process your passed data.
2	**Method** Method to be used to upload data. The most frequently used are GET and POST methods.
3	**Target** Specify the target window or frame where the result of the script will be displayed. It takes values like _blank, _self, _parent etc.
4	**Enctype** You can use the enctype attribute to specify how the browser encodes the data before it sends it to the server. Possible values are − **application/x-www-form-urlencoded** − This is the standard method most forms use in simple scenarios. **mutlipart/form-data** − This is used when you want to upload binary data in the form of files like image, word file etc.

HTML Form Tags

Let's see the list of HTML 5 form tags.

Tag	Description
<form>	It defines an HTML form to enter inputs by the used side.
<input>	It defines an input control.
<textarea>	It defines a multi-line input control.
<label>	It defines a label for an input element.

<fieldset>	It groups the related element in a form.
<legend>	It defines a caption for a <fieldset> element.
<select>	It defines a drop-down list.
<optgroup>	It defines a group of related options in a drop-down list.
<option>	It defines an option in a drop-down list.
<button>	It defines a clickable button.

HTML 5 Form Tags

Let's see the list of HTML 5 form tags.

Tag	Description
<datalist>	It specifies a list of pre-defined options for input control.
<keygen>	It defines a key-pair generator field for forms.
<output>	It defines the result of a calculation.

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">
```

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>
```

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>
```

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>
```

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>
```

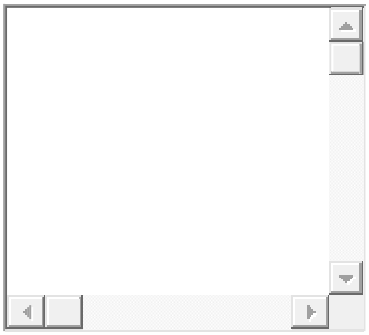
The <textarea> Element

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>
```

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>
```

The <button> Element

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

Example

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

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

Click Me!

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>
```

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>
```

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

HTML Input Types

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

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>
```

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>
```

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>
```

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

First name:
Mickey
Last name:
Mouse

Submit

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>
```

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>
```

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

First name:
Mickey
Last name:
Mouse

Submit Reset

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>
```

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>
```

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!">
```

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.

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

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.

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

You can also add restrictions to dates:

```
<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>
```

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.

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

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 adds ".com" to the keyboard to match email input.

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

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.

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

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:

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

Input Restrictions

Here is a list of some common input restrictions (some are new in HTML5):

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:

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

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:

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

Input Type Search

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

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

Input Type Tel

The **<input type="tel">** is used for input fields that should contain a telephone number.
The tel type is currently supported only in Safari 8.

```
<form>
  Telephone:
  <input type="tel" name="usrtel">
</form>
```

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.

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

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.

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

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.

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

HTML Input Attributes

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>
```

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>
```

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>
```

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>
```

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>
```

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.

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>
```

The novalidate Attribute

The **novalidate** attribute is a <form> attribute.
When present, novalidate specifies that the form data should not be validated when submitted.

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>
```

The autofocus Attribute

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

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

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

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.

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">
```

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".
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>
```

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".

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>
```

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>
```

The formnovalidate Attribute

The **formnovalidate** attribute overrides the novalidate attribute of the <form> element.
The formnovalidate attribute can be used with type="submit".

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>
```

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".

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>
```

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.

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.

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>
```

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.

<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">

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.

A file upload field that accepts multiple values:

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

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.

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">

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.

An input field with a placeholder text:

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

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>

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">
```

HTML Form and Input Elements

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

HTML <frameset> Tag. Not Supported in HTML5.

Example

A simple three-framed page:

```
<frameset cols="25%,*,25%">
  <frame src="frame_a.htm">
  <frame src="frame_b.htm">
  <frame src="frame_c.htm">
</frameset>
```

More "Try it Yourself" examples below.

Definition and Usage

The <frameset> tag is not supported in HTML5.
The <frameset> tag defines a frameset.
The <frameset> element holds one or more <frame> elements. Each <frame> element can hold a separate document.
The <frameset> element specifies HOW MANY columns or rows there will be in the frameset, and HOW MUCH percentage/pixels of space will occupy each of them.
Note: If you want to validate a page containing frames, be sure the <!DOCTYPE> is set to either "HTML Frameset DTD" or "XHTML Frameset DTD".

Differences Between HTML 4.01 and HTML5

The <frameset> tag is not supported in HTML5.

Differences Between HTML and XHTML

NONE.

Optional Attributes

Attribute	Value	Description
cols	*pixels* *%* *	Not supported in HTML5. Specifies the number and size of columns in a frameset
rows	*pixels* *%* *	Not supported in HTML5. Specifies the number and size of rows in a frameset

HTML <frame> Tag. Not Supported in HTML5.

Example

A simple three-framed page:

```
<frameset cols="25%,50%,25%">
  <frame src="frame_a.htm">
  <frame src="frame_b.htm">
  <frame src="frame_c.htm">
</frameset>
```

More "Try it Yourself" examples below.

Definition and Usage

The <frame> tag is not supported in HTML5.
The <frame> tag defines one particular window (frame) within a <frameset>.
Each <frame> in a <frameset> can have different attributes, such as border, scrolling, the ability to resize, etc.
Note: If you want to validate a page containing frames, be sure the <!DOCTYPE> is set to either "HTML Frameset DTD" or "XHTML Frameset DTD".

Differences Between HTML 4.01 and HTML5

The <frame> tag is not supported in HTML5.

Differences Between HTML and XHTML

In HTML, the <frame> tag has no end tag. In XHTML, the <frame> tag must be properly closed.

Optional Attributes

Attribute	Value	Description
frameborder	0 1	Not supported in HTML5. Specifies whether or not to display a border around a frame
longdesc	*URL*	Not supported in HTML5. Specifies a page that contains a long description of the content of a frame
marginheight	*Pixels*	Not supported in HTML5. Specifies the top and bottom margins of a frame
marginwidth	*Pixels*	Not supported in HTML5. Specifies the left and right margins of a frame
name	*Text*	Not supported in HTML5. Specifies the name of a frame
noresize	noresize	Not supported in HTML5. Specifies that a frame is not resizable
scrolling	yes no auto	Not supported in HTML5. Specifies whether or not to display scrollbars in a frame
src	*URL*	Not supported in HTML5. Specifies the URL of the document to show in a frame

HTML5 New Elements

New Elements in HTML5

Below is a list of the new HTML5 elements, and a description of what they are used for.

New Semantic/Structural Elements

HTML5 offers new elements for better document structure:

Tag	Description
<article>	Defines an article in a document
<aside>	Defines content aside from the page content
<bdi>	Isolates a part of text that might be formatted in a different direction from other text outside it
<details>	Defines additional details that the user can view or hide
<dialog>	Defines a dialog box or window
<figcaption>	Defines a caption for a <figure> element
<figure>	Defines self-contained content
<footer>	Defines a footer for a document or section
<header>	Defines a header for a document or section
<main>	Defines the main content of a document
<mark>	Defines marked/highlighted text
<menuitem>	Defines a command/menu item that the user can invoke from a popup menu
<meter>	Defines a scalar measurement within a known range (a gauge)
<nav>	Defines navigation links
<progress>	Represents the progress of a task

<rp>	Defines what to show in browsers that do not support ruby annotations
<rt>	Defines an explanation/pronunciation of characters (for East Asian typography)
<ruby>	Defines a ruby annotation (for East Asian typography)
<section>	Defines a section in a document
<summary>	Defines a visible heading for a <details> element
<time>	Defines a date/time
<wbr>	Defines a possible line-break

HTML5 Semantic Elements

Semantics is the study of the meanings of words and phrases in a language.
Semantic elements = elements with a meaning.

What are Semantic Elements?

A semantic element clearly describes its meaning to both the browser and the developer.
Examples of **non-semantic** elements: and - Tells nothing about its content.
Examples of **semantic** elements: <form>, <table>, and <article> - Clearly defines its content.

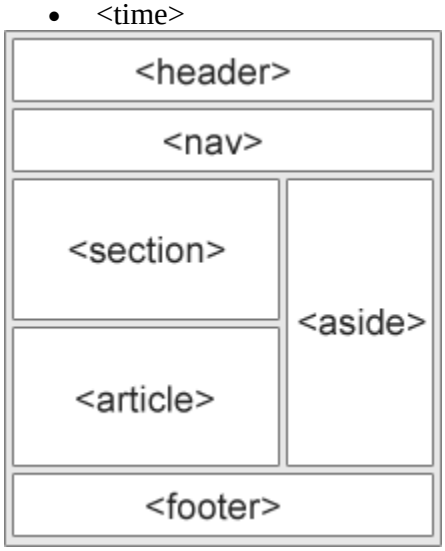
HTML5 semantic elements are supported in all modern browsers.
In addition, you can "teach" older browsers how to handle "unknown elements".
Read about it in HTML5 Browser Support.

New Semantic Elements in HTML5

Many web sites contain HTML code like: to indicate navigation, header, and footer.
HTML5 offers new semantic elements to define different parts of a web page:

- <article>
- <aside>
- <details>
- <figcaption>
- <figure>
- <footer>

- <header>
- <main>
- <mark>
- <nav>
- <section>
- <summary>
- <time>



HTML5 <section> Element

The <section> element defines a section in a document.
According to W3C's HTML5 documentation: "A section is a thematic grouping of content, typically with a heading."
A home page could normally be split into sections for introduction, content, and contact information.

Example

```
<section>
  <h1>WWF</h1>
  <p>The World Wide Fund for Nature (WWF) is....</p>
</section>
```

HTML5 <article> Element

The <article> element specifies independent, self-contained content.
An article should make sense on its own, and it should be possible to read it independently from the rest of the web site.
Examples of where an <article> element can be used:

- Forum post
- Blog post
- Newspaper article

Example

```
<article>
  <h1>What Does WWF Do?</h1>
  <p>WWF's mission is to stop the degradation of our planet's natural environment,
  and build a future in which humans live in harmony with nature.</p>
</article>
```

Nesting <article> in <section> or Vice Versa?

The <article> element specifies independent, self-contained content.
The <section> element defines section in a document.
Can we use the definitions to decide how to nest those elements? No, we cannot!
So, on the Internet, you will find HTML pages with <section> elements containing <article> elements, and <article> elements containing <sections> elements.
You will also find pages with <section> elements containing <section> elements, and <article> elements containing <article> elements.
Example for a newspaper: The sport **articles** in the sport **section**, may have a technical **section** in each **article**.

HTML5 <header> Element

The <header> element specifies a header for a document or section.
The <header> element should be used as a container for introductory content.
You can have several <header> elements in one document.
The following example defines a header for an article:

Example

```
<article>
  <header>
    <h1>What Does WWF Do?</h1>
    <p>WWF's mission:</p>
  </header>
  <p>WWF's mission is to stop the degradation of our planet's natural environment,
  and build a future in which humans live in harmony with nature.</p>
</article>
```

HTML5 <footer> Element

The <footer> element specifies a footer for a document or section.

A <footer> element should contain information about its containing element.
A footer typically contains the author of the document, copyright information, links to terms of use, contact information, etc.
You may have several <footer> elements in one document.

Example

```
<footer>
  <p>Posted by: Hege Refsnes</p>
  <p>Contact information: <a href="mailto:someone@example.com">
  someone@example.com</a>.</p>
</footer>
```

HTML5 <nav> Element

The <nav> element defines a set of navigation links.
Notice that NOT all links of a document should be inside a <nav> element. The <nav> element is intended only for major block of navigation links.

Example

```
<nav>
  <a href="/html/">HTML</a> |
  <a href="/css/">CSS</a> |
  <a href="/js/">JavaScript</a> |
  <a href="/jquery/">jQuery</a>
</nav>
```

HTML5 <aside> Element

The <aside> element defines some content aside from the content it is placed in (like a sidebar).
The aside content should be related to the surrounding content.

Example

```
<p>My family and I visited The Epcot center this summer.</p>

<aside>
  <h4>Epcot Center</h4>
  <p>The Epcot Center is a theme park in Disney World, Florida.</p>
</aside>
```

HTML5 <figure> and <figcaption> Elements

The purpose of a figure caption is to add a visual explanation to an image.

In HTML5, an image and a caption can be grouped together in a **<figure>** element:

Example

```
<figure>
  <img src="pic_mountain.jpg" alt="The Pulpit Rock" width="304" height="228">
  <figcaption>Fig1. - The Pulpit Rock, Norway.</figcaption>
</figure>
```

The **<img>** element defines the image, the **<figcaption>** element defines the caption.

Why Semantic Elements?

With HTML4, developers used their own id/class names to style elements: header, top, bottom, footer, menu, navigation, main, container, content, article, sidebar, topnav, etc.
This made it impossible for search engines to identify the correct web page content.
With the new HTML5 elements (<header> <footer> <nav> <section> <article>), this will become easier.
According to the W3C, a Semantic Web: "Allows data to be shared and reused across applications, enterprises, and communities."

Semantic Elements in HTML5

Below is an alphabetical list of the new semantic elements in HTML5.
The links go to our complete HTML5 Reference.

Tag	Description
<article>	Defines an article
<aside>	Defines content aside from the page content
<details>	Defines additional details that the user can view or hide
<figcaption>	Defines a caption for a <figure> element
<figure>	Specifies self-contained content, like illustrations, diagrams, photos, code listings, etc.
<footer>	Defines a footer for a document or section
<header>	Specifies a header for a document or section
<main>	Specifies the main content of a document

<mark>	Defines marked/highlighted text
<nav>	Defines navigation links
<section>	Defines a section in a document
<summary>	Defines a visible heading for a <details> element
<time>	Defines a date/time