

Example 1

Example Explained

The DOCTYPE declaration defines the document type

The text between <html> and </html> describes the web document

The text between <body> and </body> describes the visible page content

The text between <h1> and </h1> describes a heading

The text between <p> and </p> describes paragraph

Example 2

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

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

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

```
</body>
```

```
</html>
```

Example 3.

HTML Links

HTML links are defined with the <a> tag:

Example

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

Example 4

HTML Images

HTML images are defined with the tag.

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

Example

```

```

Example 5

HTML Attributes

HTML elements can have attributes

Attributes provide additional information about an element

Attributes are always specified in the start tag

Attributes come in name/value pairs like: name="value"

The lang Attribute

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

The language is declared in the lang attribute.

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

Example

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

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

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

```
</body>
```

```
</html>
```

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

Size Attributes

HTML images are defined with the `` tag.

The filename of the source (src), and the size of the image (width and height) are all provided as attributes:

Example

```

```

The alt Attribute

The alt attribute specifies an alternative text to be used, when an HTML element cannot be displayed.

The value of the attribute can be read by "screen readers". This way, someone "listening" to the webpage, i.e. a blind person, can "hear" the element.

Example

```

```

HTML Attributes

Below is an alphabetical list of some attributes often used in HTML:

Attribute	Description
alt	Specifies an alternative text for an image
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)
value	Specifies the value (text content) for an input element.

The HTML `<head>` Element

The HTML `<head>` element has nothing to do with HTML headings.

The HTML `<head>` element contains meta data. Meta data are not displayed.

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

The HTML <title> Element

The HTML <title> element is meta data. It defines the HTML document's title.

The title will not be displayed in the document, but might be displayed in the browser tab.

The HTML <meta> Element

The HTML <meta> element is also meta data.

It can be used to define the character set, and other information about the HTML document.

More Meta Elements

In the chapter about HTML styles you discover more meta elements:

The HTML <style> element is used to define internal CSS style sheets.

The HTML <link> element is used to define external CSS style sheets.

HTML Styles

Every HTML element has a default style (background color is white, text color is black, text-size is 12px ...)

Changing the default style of an HTML element, can be done with the style attribute.

This example changes the default background color from white to lightgrey:

Example

```
<body style="background-color:lightgrey">
```

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

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

```
</body>
```

The HTML Style Attribute

The HTML style attribute has the following syntax:

```
style="property:value"
```

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

HTML Text Color

The color property defines the text color to be used for an HTML element:

Example

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

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

</html>
```

HTML Text Fonts

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

Example

```
<!DOCTYPE html>
<html>

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

</html>
```

Note :The tag, supported in older versions of HTML, is not valid in HTML5.

HTML Text Size

The font-size property defines the text size to be used for an HTML element:

Example

```
<!DOCTYPE html>
<html>

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

</html>
```

HTML Text Alignment

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

Example

```
<!DOCTYPE html>
<html>

<body>
  <h1 style="text-align:center">Centered Heading</h1>
  <p>This is a paragraph.</p>
</body>

</html>
```

HTML Text Formatting Elements

HTML Formatting Elements

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

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

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

Formatting elements were designed to display special types of text:

- Bold text
- Important text
- Italic text
- Emphasized text
- Marked text
- Small text
- Deleted text
- Inserted text
- Subscripts
- Superscripts

HTML Bold and Strong Formatting

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

Example

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

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

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

Example

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

```
<p><strong>This text is bold</strong>.</p>
```

HTML Italic and Emphasized Formatting

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

Example

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

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

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

Example

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

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

HTML Small Formatting

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

Example

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

HTML Marked Formatting

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

Example

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

HTML Deleted Formatting

The HTML `` element defines deleted (removed) of text.

Example

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

HTML Inserted Formatting

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

Example

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

HTML Subscript Formatting

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

Example

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

HTML Superscript Formatting

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

Example

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

HTML Tutorial

HTML HOME HTML Introduction HTML Editors HTML Basic HTML Elements HTML Attributes HTML Headings HTML Paragraphs HTML Styles HTML Formatting HTML Quotations HTML Computercode HTML Comments HTML CSS HTML Links HTML Images HTML Tables HTML Lists HTML Blocks HTML Classes HTML Layout HTML Responsive HTML Forms HTML Iframes HTML Color Names HTML Color Values HTML Color Shades HTML JavaScript HTML Head HTML Entities HTML Symbols HTML Charset HTML URL Encode HTML XHTML HTML5

HTML5 Intro HTML5 Support HTML5 Elements HTML5 Semantics HTML5 Migration HTML5 Syntax

HTML Forms

HTML Form Elements HTML Input Types HTML Input Attributes

HTML Graphics

HTML Canvas HTML SVG

HTML Media

HTML Media HTML Video HTML Audio HTML Plug-ins HTML YouTube

HTML APIs

HTML Geolocation HTML Drag/Drop HTML Local Storage HTML App Cache HTML Web Workers HTML SSE

HTML Examples

HTML Examples HTML Quiz HTML5 Quiz HTML Certificate HTML5 Certificate HTML Summary

HTML References

HTML Tag List HTML Attributes HTML Events HTML Canvas HTML Audio/Video HTML

Doctypes HTML Colornames HTML Colorpicker HTML Colormixer HTML Character Sets HTML URL Encode HTML Lang Codes HTTP Messages HTTP Methods Keyboard Shortcuts HTML Text Formatting Elements

« Previous

Next Chapter »

Text Formatting

This text is bold

This text is italic

This is superscript

HTML Formatting Elements

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

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

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

Formatting elements were designed to display special types of text:

Bold text

Important text

Italic text

Emphasized text

Marked text

Small text

Deleted text

Inserted text

Subscripts

Superscripts

HTML Bold and Strong Formatting

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

Example

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

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

Try it Yourself »

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

Example

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

```
<p><strong>This text is bold</strong>.</p>
```

Try it Yourself »

HTML Italic and Emphasized Formatting

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

Example

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

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

Try it Yourself »

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

Example

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

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

Try it Yourself »

Note Browsers display `` as ``, and `` as `<i>`.

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

HTML Small Formatting

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

Example

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

Try it Yourself »

HTML Marked Formatting

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

Example

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

Try it Yourself »

HTML Deleted Formatting

The HTML `` element defines deleted (removed) of text.

Example

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

Try it Yourself »

HTML Inserted Formatting

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

Example

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


Try it Yourself »

HTML Subscript Formatting

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

Example

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

Try it Yourself »

HTML Superscript Formatting

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

Example

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

Try it Yourself »

Test Yourself with Exercises!

Exercise 1 » Exercise 2 » Exercise 3 »

HTML Text Formatting Elements

Tag Description

`` 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

`` Defines deleted text

`<mark>` Defines marked/highlighted text

HTML `<q>` for Short Quotations

The HTML `<q>` element defines a short quotation.

Browsers usually insert quotation marks around the `<q>` element.

Example

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

HTML `<blockquote>` for Long Quotations

The HTML `<blockquote>` element defines a quoted section.

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.

HTML `<abbr>` for Abbreviations

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

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

Example

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

HTML `<address>` for Contact Information

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

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

Example

```
<address>
```

```
Written by Jon Doe.<br>
```

```
Visit us at:<br>
```

```
Example.com<br>
```

```
Box 564, Disneyland<br>
```

```
USA
```

```
</address>
```

HTML `<cite>` for Work Title

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

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

Example

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

HTML `<bdo>` for Bi-Directional Override

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

If your browser supports `bdo`, this text will be written from right to left:

Example

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

HTML Quotations, Citations, and Definition 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

`<q>` Defines an inline (short) quotation

`<cite>` Defines the title of a work

`<dfn>` Defines a definition term

HTML Computer Code Formatting

Normally, HTML uses variable letter size, and variable letter spacing.

This is not wanted when displaying examples of computer code.

The `<kbd>`, `<samp>`, and `<code>` elements all support fixed letter size and spacing.

HTML Keyboard Formatting

The HTML `<kbd>` element defines keyboard input:

Example

```
<p>To open a file, select:</p>
```

```
<p><kbd>File | Open...</kbd></p>
```

HTML Sample Formatting

The HTML `<samp>` element defines a computer output sample:

Example

```
<samp>
```

```
demo.example.com login: Apr 12 09:10:17
```

```
Linux 2.6.10-grsec+gg3+e+fhs6b+nfs+gr0501+++p3+c4a+gr2b-reslog-v6.189
```

```
</samp>
```

HTML Code Formatting

The HTML `<code>` element defines programming code sample:

Example

```
<code>
```

```
var person = { firstName:"John", lastName:"Doe", age:50, eyeColor:"blue" }
```

```
</code>
```

The `<code>` element does not preserve extra whitespace and line-breaks:

Example

```
<p>Coding Example:</p>
```

```
<code>
```

```
var person = {  
  firstName:"John",  
  lastName:"Doe",  
  age:50,  
  eyeColor:"blue"  
}
```

```
</code>
```

To fix this, you must wrap the code in a `<pre>` element:

Example

```
<p>Coding Example:</p>
```

```
<code>
```

```
<pre>
```

```
var person = {  
  firstName:"John",  
  lastName:"Doe",
```

```
    age:50,  
    eyeColor:"blue"  
}  
</pre>  
</code>
```

HTML Variable Formatting

The HTML `<var>` element defines a mathematical variable:

Example

```
<p>Einstein wrote:</p>
```

```
<p><var>E = m c<sup>2</sup></var></p>
```

HTML Computer Code Elements

Tag	Description
-----	-------------

<code><code></code>	Defines computer code text
---------------------------	----------------------------

<code><kbd></code>	Defines keyboard text
--------------------------	-----------------------

<code><samp></code>	Defines sample computer code
---------------------------	------------------------------

<code><var></code>	Defines a variable
--------------------------	--------------------

<code><pre></code>	Defines preformatted text
--------------------------	---------------------------

HTML Comment Tags

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

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
```

```

```

```
-->
```

Conditional Comments

You might stumble upon conditional comments in HTML:

```
<!--[if IE 8]>
```

```
.... some HTML here ....
```

```
<![endif]-->
```

HTML Styles – CSS

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
body {background-color:lightgray}
h1 {color:blue}
p {color:green}
</style>
</head>
```

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

```
</html>
```

Styling HTML with CSS

CSS stands for Cascading Style Sheets

Styling can be added to HTML elements in 3 ways:

- Inline - using a style attribute in HTML elements

- Internal - using a <style> element in the HTML <head> section

- External - using one or more external CSS files

CSS Syntax

CSS styling has the following syntax:

```
element { property:value ; property:value }
```

The element is an HTML element name. The property is a CSS property. The value is a CSS value.

Multiple styles are separated with semicolon.

Inline Styling (Inline CSS)

Inline styling is useful for applying a unique style to a single HTML element:

Inline styling uses the style attribute.

This inline styling changes the text color of a single heading:

Example

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

Internal Styling (Internal CSS)

An internal style sheet can be used to define a common style for all HTML elements on a page.

Internal styling is defined in the <head> section of an HTML page, using a <style> element:

Example

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
body {background-color:lightgrey}
h1 {color:blue}
p {color:green}
</style>
</head>
```

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

```
</html>
```

External Styling (External CSS)

External style sheet are ideal when the style is applied to many pages.

With external style sheets, you can change the look of an entire site by changing one file.

External styles are defined in the <head> section of an HTML page, in the <link> element:

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>
```

CSS Fonts

The CSS property color defines the text color to be used for an HTML element.

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

The CSS property font-size defines the text size to be used for an HTML element.

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>

<html>

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

</html>

```

The CSS Box Model

Every visible HTML element has a box around it, even if you cannot see it.

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

```

<!DOCTYPE html>
<html>

<head>
<style>
p {
  border:1px solid grey;
}
</style>
</head>
<body>

<h1>This is a heading</h1>

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

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

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

</body>
</html>

```

The CSS padding property defines a padding (space) inside the border:

```

<html>

```

```
<head>
<style>
p {
  border:1px solid grey;
  padding:10px;
}
</style>
</head>
<body>

<h1>This is a heading</h1>

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

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

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

</body>
</html>
```

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

```
<html>

<head>
<style>
p {
  border:1px solid grey;
  padding:10px;
  margin:30px;
}
</style>
</head>
<body>

<h1>This is a heading</h1>

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

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

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

</body>
</html>
```

The id Attribute

All the examples above use CSS to style HTML elements in a general way.

The CSS styles define an equal style for all equal elements.

To define a special style for a special element, first add an id attribute to the element:

Example

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

```
<html>
```

```
<head>
```

```
<style>
```

```
p#p01 {
```

```
  color: blue;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

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

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

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

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

```
</body>
```

```
</html>
```

The class Attribute

To define a style for a special type (class) of elements, add a class attribute to the element:

Example

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

```
<html>
```

```
<head>
```

```
<style>
```

```
p.error {
```

```
  color:red;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

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

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

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

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

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

```
</body>
```

```
</html>
```

HTML Style Tags

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

HTML Links - Hyperlinks

HTML links are hyperlinks.

A hyperlink is an element, a text, or an image that you can click on, and jump to another document.

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

Link Syntax:

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

Example:

```
<a href="http://www.w3schools.com/html/">Visit our HTML tutorial</a>
```

The href attribute specifies the destination address (http://www.w3schools.com/html/)

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

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

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 Links - Colors and Icons

When you move the mouse cursor over a link, two things will normally happen:

- The mouse arrow will turn into a little hand
- The color of the link element will change

By default, links will appear as 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 defaults, using styles:

```
!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
a:link {
```

```
  color:#000000;
```

```
  background-color:transparent;
```

```

    text-decoration:none;
}
a:visited {
    color:#000000;
    background-color:transparent;
    text-decoration:none;
}
a:hover {
    color:#ff0000;
    background-color:transparent;
    text-decoration:underline;
}
a:active {
    color:#ff0000;
    background-color:transparent;
    text-decoration:underline;
}
</style>
</head>

<body>

<p>You can change the default colors of links</p>

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

</body>
</html>

```

```

<!DOCTYPE html>
<html>

<body>
<h2>Spectacular Mountains</h2>

</body>

</html>

```

Note : Always specify image size. If the size is unknown, the page will flicker while the image loads.

HTML Images Syntax

In HTML, images are defined with the tag.

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

The src attribute defines the url (web address) of the image:

```

```

The alt Attribute

The alt attribute specifies an alternate text for the image, if it cannot be displayed.

The value of the alt attribute should describe the image in words:

Example

```

```

HTML Screen Readers

Screen readers are software programs that can read what is displayed on a screen.

Used on the web, screen readers can "reproduce" HTML as text-to-speech, sound icons, or braille output.

Screen readers are used by people who are blind, visually impaired, or learning disabled.

Note Screen readers can read the alt attribute.

Image Size - Width and Height

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

The values are specified in pixels (use px after the value):

Example

```

```

Alternatively, you can use width and height attributes.

The values are specified in pixels (without px after the value):

Example

```

```

Width and Height or Style?

Both the width, the height, and the style attributes, are valid in the latest HTML5 standard.

We suggest you use the style attribute. It prevents styles sheets from changing the default size of images:

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

<body>
```

```


```

```
</body>
```

```
</html>
```

Images in Another Folder

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

However, it is common on the web, to store images in a sub-folder, and refer to the folder in the image name:

Example

```

```

If a browser cannot find an image, it will display a broken link icon:

```
<html>
```

```
<body>
```

```
<p>
```

If a browser cannot find an image, it will display a broken link icon.

```
</p>
```

```

```

```
</body>
```

```
</html>
```

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:

```

```

Animated Images

The GIF standard allows animated images:

```
<html>
```

```
<body>
```

```
<p>
```

The GIF standard allows moving images.

```
</p>
```

```

```

```
</body>
```

```
</html>
```

Note that the syntax of inserting animated images is no different from non-animated images.

Using an Image as a Link

It is common to use images as links:

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

```
<p>The image is a link. You can click on it.</p>
```

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

</a>
```

```
<p>
```

We have added "border:0" to prevent IE9 (and earlier) from displaying a border around the image.

```
</p>
```

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

Image Maps

For an image, you can create an image map, with clickable areas:

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

```
<p>Click on the sun or on one of the planets to watch it closer:</p>
```

```

```

```
<map name="planetmap">
  <area shape="rect" coords="0,0,82,126" alt="Sun" href="sun.htm">
  <area shape="circle" coords="90,58,3" alt="Mercury" href="mercur.htm">
  <area shape="circle" coords="124,58,8" alt="Venus" href="venus.htm">
</map>
```

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

Image Floating

You can let an image float to the left or right of a paragraph:

```
html>
<body>
```

```
<p>
```

```

A paragraph with an image. A paragraph with an image.
A paragraph with an image. A paragraph with an image.
A paragraph with an image. A paragraph with an image.
</p>
```

```
<p>The image floats to the left of the text.</p>
```

<p>Please use the CSS float property. The align attribute is deprecated in HTML 4, and not supported in HTML5.</p>

</body>
</html>

Chapter Summary

- Use the HTML element to define images
- Use the HTML src attribute to define the image file name
- Use the HTML alt attribute to define an alternative text
- Use the HTML width and height attributes to define the image size
- Use the CSS width and height properties to define the image size (alternatively)
- Use the CSS float property to define image floating
- Use the CSS usemap attribute to point to an image map
- Use the HTML <map> element to define an image map
- Use the HTML <area> element to define image map areas

HTML Image Tags

Tag	Description
	Defines an image
<map>	Defines an image-map
<area>	Defines a clickable area inside an image-map

Defining HTML Tables

```
html>  
<body>  
  
<table style="width:100%">  
  <tr>  
    <td>Jill</td>  
    <td>Smith</td>  
    <td>50</td>  
  </tr>  
  <tr>  
    <td>Eve</td>  
    <td>Jackson</td>  
    <td>94</td>  
  </tr>  
  <tr>  
    <td>John</td>  
    <td>Doe</td>  
    <td>80</td>  
  </tr>  
</table>  
  
</body>  
</html>
```

Example explained:

Tables are defined with the <table> tag.

Tables are divided into table rows with the <tr> tag.

Table rows are divided into table data with the <td> tag.

A table row can also be divided into table headings with the <th> tag.

An HTML Table with a Border Attribute

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

A border can be added using the border attribute:

```
<html>
<body>

<table border="1" style="width:100%">
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
  <tr>
    <td>John</td>
    <td>Doe</td>
    <td>80</td>
  </tr>
</table>

</body>
</html>
```

Note: The border attribute is on its way out of the HTML standard! It is better to use CSS.

To add borders, use the CSS border property:

```
<html>

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



```
<body>

<table style="width:100%">
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
  <tr>
    <td>John</td>
    <td>Doe</td>
    <td>80</td>
  </tr>
</table>

</body>
</html>
```

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

An HTML Table with Collapsed Borders

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

```
<html>

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

<body>

<table style="width:100%">
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
```

```
<tr>
  <td>John</td>
  <td>Doe</td>
  <td>80</td>
</tr>
</table>

</body>
</html>
```

An HTML Table with 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:

```
html>

<head>
<style>
table, th, td {
  border: 1px solid black;
  border-collapse: collapse;
}
th, td {
  padding: 15px;
}
</style>
</head>

<body>

<table style="width:100%">
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
  <tr>
    <td>John</td>
    <td>Doe</td>
    <td>80</td>
  </tr>
</table>
```

```
<p>Try to change the padding to 5px.</p>
```

```
</body>
```

```
</html>
```

HTML Table Headings

Table headings are defined with the `<th>` tag.

By default, all major browsers display table headings as bold and centered:

```
<html>
```

```
<head>
```

```
<style>
```

```
table, th, td {
```

```
    border: 1px solid black;
```

```
    border-collapse: collapse;
```

```
}
```

```
th, td {
```

```
    padding: 5px;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<table style="width:100%">
```

```
<tr>
```

```
<th>Firstname</th>
```

```
<th>Lastname</th>
```

```
<th>Points</th>
```

```
</tr>
```

```
<tr>
```

```
<td>Jill</td>
```

```
<td>Smith</td>
```

```
<td>50</td>
```

```
</tr>
```

```
<tr>
```

```
<td>Eve</td>
```

```
<td>Jackson</td>
```

```
<td>94</td>
```

```
</tr>
```

```
<tr>
```

```
<td>John</td>
```

```
<td>Doe</td>
```

```
<td>80</td>
```

```
</tr>
```

```
</table>
```

```
</body>
```

```
</html>
```

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

```
<html>

<head>
<style>
table, th, td {
  border: 1px solid black;
  border-collapse: collapse;
}
th, td {
  padding: 5px;
}
th {
  text-align: left;
}
</style>
</head>

<body>

<table style="width:100%">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Points</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
  <tr>
    <td>John</td>
    <td>Doe</td>
    <td>80</td>
  </tr>
</table>

</body>
</html>
```

An HTML Table with 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;
}

<html>

<head>
<style>
table, th, td {
  border: 1px solid black;
  padding: 5px;
}
table {
  border-spacing: 15px;
}
</style>
</head>
```

<body>

```
<table style="width:100%">
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
  <tr>
    <td>John</td>
    <td>Doe</td>
    <td>80</td>
  </tr>
</table>
```

<p>Try to change the border-spacing to 5px.</p>

</body>
</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 colspan="2">Name</th>
```

```

    <th colspan="2">Telephone</th>
</tr>
<tr>
    <td>Bill Gates</td>
    <td>555 77 854</td>
    <td>555 77 855</td>
</tr>
</table>

```

```
<html>
```

```
<head>
```

```
<style>
```

```
table, th, td {
```

```
    border: 1px solid black;
```

```
    border-collapse: collapse;
```

```
}
```

```
th, td {
```

```
    padding: 5px;
```

```
    text-align: left;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h2>Cell that spans two columns:</h2>
```

```
<table style="width:100%">
```

```
<tr>
```

```
<th>Name</th>
```

```
<th colspan="2">Telephone</th>
```

```
</tr>
```

```
<tr>
```

```
<td>Bill Gates</td>
```

```
<td>555 77 854</td>
```

```
<td>555 77 855</td>
```

```
</tr>
```

```
</table>
```

```
</body>
```

```
</html>
```

Table Cells that Span Many Rows

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

```
<html>
```

```
<head>
```

```
<style>
```

```
table, th, td {
```

```
    border: 1px solid black;
```

```

border-collapse: collapse;
}
th, td {
padding: 5px;
text-align: left;
}
</style>
</head>

<body>

<h2>Cell that spans two rows:</h2>
<table style="width:100%">
<tr>
<th>First Name:</th>
<td>Bill Gates</td>
</tr>
<tr>
<th rowspan="2">Telephone:</th>
<td>555 77 854</td>
</tr>
<tr>
<td>555 77 855</td>
</tr>
</table>

</body>
</html>

```

An HTML Table With a Caption

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

```

<html>

<head>
<style>
table, th, td {
border: 1px solid black;
border-collapse: collapse;
}
th, td {
padding: 5px;
text-align: left;
}
</style>
</head>

<body>

<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>

</body>
</html>

```

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

Different Styles for Different Tables

Most of the examples above use a style attribute (width="100%") to define the width of each table.

This makes it easy to define different widths for different tables.

The styles in the <head> section, however, define a style for all tables in a page.

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>Points</th>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
</table>

```

Now you can define a different style for this table:

```

<html>

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

```



```
padding: 5px;
text-align: left;
}
table#t01 {
width: 100%;
background-color: #f1f1c1;
}
</style>
</head>
```

```
<body>
```

```
<table style="width:100%">
<tr>
<th>First Name</th>
<th>Last Name</th>
<th>Points</th>
</tr>
<tr>
<td>Jill</td>
<td>Smith</td>
<td>50</td>
</tr>
<tr>
<td>Eve</td>
<td>Jackson</td>
<td>94</td>
</tr>
<tr>
<td>John</td>
<td>Doe</td>
<td>80</td>
</tr>
</table>
```

```
<br>
```

```
<table id="t01">
<tr>
<th>First Name</th>
<th>Last Name</th>
<th>Points</th>
</tr>
<tr>
<td>Jill</td>
<td>Smith</td>
<td>50</td>
</tr>
<tr>
<td>Eve</td>
<td>Jackson</td>
<td>94</td>
```

```
</tr>
<tr>
  <td>John</td>
  <td>Doe</td>
  <td>80</td>
</tr>
</table>
```

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

And add more styles:

```
<html>

<head>
<style>
table {
  width:100%;
}
table, th, td {
  border: 1px solid black;
  border-collapse: collapse;
}
th, td {
  padding: 5px;
  text-align: left;
}
table#t01 tr:nth-child(even) {
  background-color: #eee;
}
table#t01 tr:nth-child(odd) {
  background-color:#fff;
}
table#t01 th {
  background-color: black;
  color: white;
}
</style>
</head>
```

```
<body>
```

```
<table>
  <tr>
    <th>First Name</th>
    <th>Last Name</th>
    <th>Points</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
```

```
<tr>
  <td>Eve</td>
  <td>Jackson</td>
  <td>94</td>
</tr>
<tr>
  <td>John</td>
  <td>Doe</td>
  <td>80</td>
</tr>
</table>
```

```
<br>
```

```
<table id="t01">
  <tr>
    <th>First Name</th>
    <th>Last Name</th>
    <th>Points</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
  <tr>
    <td>John</td>
    <td>Doe</td>
    <td>80</td>
  </tr>
</table>
```

```
</body>
```

```
</html>
```

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

Unordered HTML Lists

An unordered list starts with the `` tag. Each list item starts with the `` tag.

The list items will be marked with bullets (small black circles).

```
<html>
<body>

<h2>Unordered List with Default Bullets</h2>

<ul>
<li>Apples</li>
<li>Bananas</li>
<li>Lemons</li>
<li>Oranges</li>
</ul>

</body>
</html>
```

Unordered HTML Lists - The Style Attribute

A style attribute can be added to an unordered list, to define the style of the marker:

Style	Description
<code>list-style-type:disc</code>	The list items will be marked with bullets (default)
<code>list-style-type:circle</code>	The list items will be marked with circles
<code>list-style-type:square</code>	The list items will be marked with squares
<code>list-style-type:none</code>	The list items will not be marked

Disc:

```
html>
<body>

<h2>Unordered List with Disc Bullets</h2>
```

```
<ul style="list-style-type:disc">
<li>Apples</li>
<li>Bananas</li>
<li>Lemons</li>
<li>Oranges</li>
</ul>
```

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

Circle:

```
<html>
<body>

<h2>Unordered List with Circle Bullets</h2>
```

```
<ul style="list-style-type:circle">
  <li>Apples</li>
  <li>Bananas</li>
  <li>Lemons</li>
  <li>Oranges</li>
</ul>
```

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

Square:

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

```
<h2>Unordered List with Square Bullets</h2>
```

```
<ul style="list-style-type:square">
  <li>Apples</li>
  <li>Bananas</li>
  <li>Lemons</li>
  <li>Oranges</li>
</ul>
```

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

None:

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

Using a type attribute `<ul type="disc">`, instead of `<ul style="list-style-type:disc">`, also works. But in HTML5, the type attribute is not valid in unordered lists, only in ordered list.

Ordered HTML Lists

An ordered list starts with the `` tag. Each list item starts with the `` tag.

The list items will be marked with numbers.

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

```
<h2>Ordered List</h2>
```

```
<ol>
  <li>Apples</li>
  <li>Bananas</li>
  <li>Lemons</li>
  <li>Oranges</li>
```

```
</ol>
```

```
</body>
```

```
</html>
```

Ordered HTML Lists - The Type Attribute

A type attribute can be added to an ordered list, to define the type of the 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:

```
<html>
```

```
<body>
```

```
<h2>Ordered List with Numbers</h2>
```

```
<ol type="1">
```

```
<li>Apples</li>
```

```
<li>Bananas</li>
```

```
<li>Lemons</li>
```

```
<li>Oranges</li>
```

```
</ol>
```

```
</body>
```

```
</html>
```

Upper Case:

```
<html>
```

```
<body>
```

```
<h2>Ordered List with Letters</h2>
```

```
<ol type="A">
```

```
<li>Apples</li>
```

```
<li>Bananas</li>
```

```
<li>Lemons</li>
```

```
<li>Oranges</li>
```

```
</ol>
```

```
</body>
```

```
</html>
```

Lower Case:

```
<ol type="a">
```

```
<li>Coffee</li>
```

```
<li>Tea
```

```
<li>Milk</li>
</ol>
```

Roman Upper Case:

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

Roman Lower Case:

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

HTML Description Lists

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

The <dl> tag defines a description list.

The <dt> tag defines the term (name), and the <dd> tag defines the data (description).

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

```
<h2>A Description List</h2>
```

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

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

Nested HTML Lists

List can be nested (lists inside lists).

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

```
<h2>A Nested List</h2>
```

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

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

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

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

Horizontal Lists

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

One popular way, is to style a list to display horizontally:

```
<html>

<head>
<style>
ul#menu li {
  display:inline;
}
</style>
</head>

<body>

<h2>Horizontal List</h2>

<ul id="menu">
<li>Apples</li>
<li>Bananas</li>
<li>Lemons</li>
<li>Oranges</li>
</ul>

</body>
</html>
```

With a little extra style, you can make it look like a menu:

```
<html>

<head>
<style>
ul#menu {
  padding: 0;
}

```



```
ul#menu li {
  display: inline;
}

ul#menu li a {
  background-color: black;
  color: white;
  padding: 10px 20px;
  text-decoration: none;
  border-radius: 4px 4px 0 0;
}
```

```
ul#menu li a:hover {
  background-color: orange;
}
</style>
</head>
```

```
<body>
```

```
<h2>Horizontal List</h2>
```

```
<ul id="menu">
  <li><a href="html_tables.asp">Tables</a></li>
  <li><a href="html_lists.asp">Lists</a></li>
  <li><a href="html_blocks.asp">Blocks</a></li>
  <li><a href="html_classes.asp">Classes</a></li>
</ul>
```

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

HTML List Tags

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

HTML Block Elements

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

```
<div style="background-color:black; color:white; margin:20px; 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>
```

```
<p>
```

Standing on the River Thames, London has been a major settlement for two millennia, its history going back to its founding by the Romans, who named it Londinium.

```
</p>
```

```
</div>
```

```
</body>
```

```
</html>
```

HTML Block Elements and Inline Elements

Most HTML elements are defined as block level elements or inline elements.

Block level elements normally start (and end) with a new line, when displayed in a browser.

Examples: `<h1>`, `<p>`, ``, `<table>`

Inline elements are normally displayed without line breaks.

Examples: ``, `<td>`, `<a>`, ``

The HTML `<div>` Element

The HTML `<div>` element is a block level element that can be used as a container for other HTML elements.

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

Because it is a block level element, the browser will display line breaks before and after it.

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

The HTML `` Element

The HTML `` element is an inline element that can be used as a container for text.

The `` element has no special meaning. It has no required attributes, but `style` and `class` are common.

Unlike `<div>`, which is formatted with line breaks, the `` element does not have any automatic formatting.

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

```
<html>
```

```
<body>
```

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

```
</body>
```

```
</html>
```

HTML Grouping Tags

Tag Description

<div> Defines a section in a document (block-level)

 Defines a section in a document (inline)

HTML Classes

Classing HTML elements, makes it possible to define CSS styles for classes of elements.

Equal styles for equal classes, or different styles for different classes.

```
<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 city of England. It is the most populous city in the United Kingdom, with a metropolitan area of over 13 million inhabitants.</p>
```

```
<p>Standing on the River Thames, London has been a major settlement for two millennia, its history going back to its founding by the Romans, who named it Londinium.</p>
</div>
```

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

Classing Block Elements

The HTML <div> element is a block level element. It can be used as a container for other HTML elements.

Classing <div> elements, makes it possible to define equal styles for equal <div> elements:

```
<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 city of England. It is the most populous city in the United Kingdom, with
a metropolitan area of over 13 million inhabitants.</p>

<p>Standing on the River Thames, London has been a major settlement for two millennia, its
history going back to its founding by the Romans, who named it Londinium.</p>
</div>

<div class="cities">
<h2>Paris</h2>

<p>Paris is the capital and most populous city of France.</p>

<p>Situating on the Seine River, it is at the heart of the Île-de-France region, also known as the
région parisienne.</p>

<p>Within its metropolitan area is one of the largest population centers in Europe, with over 12
million inhabitants.</p>
</div>

<div class="cities">
<h2>Tokyo</h2>

<p>Tokyo is the capital of Japan, the center of the Greater Tokyo Area, and the most populous
metropolitan area in the world.</p>

<p>It is the seat of the Japanese government and the Imperial Palace, and the home of the Japanese
Imperial Family.</p>

<p>The Tokyo prefecture is part of the world's most populous metropolitan area with 38 million
people and the world's largest urban economy.</p>
</div>

</body>
</html>

```

Classing Inline Elements

The HTML `` element is an inline element that can be used as a container for text.

Classing `` elements makes it possible to design equal styles for equal `` elements.

```

<html>
<head>

```

```
<style>
span.red {
  color:red;
}
</style>
</head>

<body>

<h1>My <span class="red">Important</span> Heading</h1>

</body>
</html>
```

HTML Layouts

Websites often display content in multiple columns (like a magazine or newspaper).

HTML Layout Using <div> Elements

The <div> element is often used as a layout tool, because it can easily be positioned with CSS.

```
<html>

<head>
<style>
#header {
  background-color:black;
  color:white;
  text-align:center;
  padding:5px;
}
#nav {
  line-height:30px;
  background-color:#eeeeee;
  height:300px;
  width:100px;
  float:left;
  padding:5px;
}
#section {
  width:350px;
  float:left;
  padding:10px;
}
#footer {
  background-color:black;
  color:white;
  clear:both;
  text-align:center;
  padding:5px;
```

```
}
</style>
</head>

<body>

<div id="header">
<h1>City Gallery</h1>
</div>

<div id="nav">
London<br>
Paris<br>
Tokyo<br>
</div>

<div id="section">
<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>
<p>
Standing on the River Thames, London has been a major settlement for two millennia,
its history going back to its founding by the Romans, who named it Londinium.
</p>
</div>

<div id="footer">
Copyright © W3Schools.com
</div>

</body>
</html>
The CSS:

<style>
#header {
  background-color:black;
  color:white;
  text-align:center;
  padding:5px;
}
#nav {
  line-height:30px;
  background-color:#eeeeee;
  height:300px;
  width:100px;
  float:left;
  padding:5px;
}
#section {
```

```

width:350px;
float:left;
padding:10px;
}
#footer {
background-color:black;
color:white;
clear:both;
text-align:center;
padding:5px;
}
</style>

```

Website Layout Using HTML5

HTML5 offers new semantic elements that define different parts of a web page:

HTML5 Semantic Elements

header Defines a header for a document or a section
nav Defines a container for navigation links
section Defines a section in a document
article Defines an independent self-contained article
aside Defines content aside from the content (like a sidebar)
footer Defines a footer for a document or a section
details Defines additional details
summary Defines a heading for the details element

This example uses <header>, <nav>, <section>, and <footer> to create a multiple column layout:

```

<html>

<head>
<style>
header {
background-color:black;
color:white;
text-align:center;
padding:5px;
}
nav {
line-height:30px;
background-color:#eeeeee;
height:300px;
width:100px;
float:left;
padding:5px;
}
section {
width:350px;
float:left;
padding:10px;
}
footer {

```

```
background-color:black;
color:white;
clear:both;
text-align:center;
padding:5px;
}
</style>
</head>
```

```
<body>
```

```
<header>
<h1>City Gallery</h1>
</header>
```

```
<nav>
London<br>
Paris<br>
Tokyo<br>
</nav>
```

```
<section>
<h1>London</h1>
```

```
<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>
<p>
Standing on the River Thames, London has been a major settlement for two millennia,
its history going back to its founding by the Romans, who named it Londinium.
```

```
</p>
</section>
```

```
<footer>
Copyright © W3Schools.com
</footer>
```

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

The CSS

```
<style>
header {
background-color:black;
color:white;
text-align:center;
padding:5px;
}
nav {
line-height:30px;
background-color:#eeeeee;
height:300px;
```



```

width:100px;
float:left;
padding:5px;
}
section {
width:350px;
float:left;
padding:10px;
}
footer {
background-color:black;
color:white;
clear:both;
text-align:center;
padding:5px;
}

```

HTML Layout Using Tables

Note The <table> element was not designed to be a layout tool.
The purpose of the <table> element is to display tabular data.

Layout can be achieved using the <table> element, because table elements can be styled with CSS:

```

<html>
<head>
<style>
table.lamp {
width:100%;
border:1px solid #d4d4d4;
}
table.lamp th, td {
padding:10px;
}
table.lamp th {
width:40px;
}
</style>
</head>

<body>

<table class="lamp">
<tr>
<th>

</th>
<td>
The table element was not designed to be a layout tool.
</td>
</tr>
</table>

```

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

The CSS

```
<style>
table.lamp {
  width:100%;
  border:1px solid #d4d4d4;
}
table.lamp th, td {
  padding:10px;
}
table.lamp td {
  width:40px;
}
</style>
```

What is Responsive Web Design?

- RWD stands for Responsive Web Design
- RWD can deliver web pages in variable sizes
- RWD is a must for tablets and mobile devices

Creating Your Own Responsive Design

One way to create a responsive design, is to create it yourself:

```
<html lang="en-US">
<head>
<style>
.city {
float: left;
margin: 5px;
padding: 15px;
width: 300px;
height: 300px;
border: 1px solid black;
}
</style>
</head>

<body>

<h1>W3Schools Demo</h1>
<h2>Resize this responsive page!</h2>
<br>

<div class="city">
<h2>London</h2>
<p>London is the capital city of England.</p>
<p>It is the most populous city in the United Kingdom,
```

```
with a metropolitan area of over 13 million inhabitants.</p>
</div>
```

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

```
<div class="city">
<h2>Tokyo</h2>
<p>Tokyo is the capital of Japan, the center of the Greater Tokyo Area,
and the most populous metropolitan area in the world.</p>
</div>
```

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

Using Bootstrap

Another way to create a responsive design, is to use an already existing CSS framework.

Bootstrap is the most popular HTML, CSS, and JS framework for developing responsive webs.

Bootstrap helps you to develop sites that look nice at any size; screen, laptop, tablet, or phone:

```
<html>
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"
href="http://maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css">
</head>
```

```
<body>
```

```
<div class="container">
<div class="jumbotron">
  <h1>W3Schools Demo</h1>
  <p>Resize this responsive page!</p>
</div>
</div>
```

```
<div class="container">
<div class="row">
  <div class="col-md-4">
    <h2>London</h2>
    <p>London is the capital city of England.</p>
    <p>It is the most populous city in the United Kingdom,
with a metropolitan area of over 13 million inhabitants.</p>
  </div>
  <div class="col-md-4">
    <h2>Paris</h2>
    <p>Paris is the capital and most populous city of France.</p>
```

```
</div>
<div class="col-md-4">
  <h2>Tokyo</h2>
  <p>Tokyo is the capital of Japan, the center of the Greater Tokyo Area,
  and the most populous metropolitan area in the world.</p>
</div>
</div>
</div>

</body>
</html>
```

HTML Forms

HTML forms are used to pass data to a server.

An HTML form can contain input elements like text fields, checkboxes, radio-buttons, submit buttons and more. A form can also contain select lists, textarea, fieldset, legend, and label elements.

The `<form>` tag is used to create an HTML form:

`input type="text">` defines a one-line input field that a user can enter text into:

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

How the HTML code above looks 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.

Password Field

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

```
<form>
Password: <input type="password" name="pwd">
</form>
```

How the HTML code above looks in a browser:

Password:

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

Radio Buttons

`<input type="radio">` defines a radio button. Radio buttons let a user select ONLY ONE of a limited number of choices:

```
<form>
<input type="radio" name="sex" value="male">Male<br>
<input type="radio" name="sex" value="female">Female
```

</form>

How the HTML code above looks in a browser:

Male
 Female

Checkboxes

<input type="checkbox"> defines a checkbox. Checkboxes let a user select ZERO or MORE options of a limited number of choices.

<form>

<input type="checkbox" name="vehicle" value="Bike">I have a bike

<input type="checkbox" name="vehicle" value="Car">I have a car

</form>

How the HTML code above looks in a browser:

I have a bike
 I have a car

Submit Button

<input type="submit"> defines a submit button.

A submit button is used to send form data to a server. The data is sent to the page specified in the form's action attribute. The file defined in the action attribute usually does something with the received input:

```
<form name="input" action="demo_form_action.asp" method="get">
```

```
Username: <input type="text" name="user">
```

```
<input type="submit" value="Submit">
```

```
</form>
```

How the HTML code above looks in a browser:

Username:(-----)

If you type some characters in the text field above, and click the "Submit" button, the browser will send your input to a page called "demo_form_action.asp". The page will show you the received input.

HTML Form Tags

= Tag added in HTML5.

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
<keygen> Defines a key-pair generator field (for forms)
<output> Defines the result of a calculation

HTML Iframes

Iframe Syntax

The syntax for adding an iframe is:

```
<iframe src="URL"></iframe>
```

The src attribute specifies the URL (web address) of the iframe page.

Iframe - Set Height and Width

Use the height and width attributes to specify the size.

The attribute values are specified in pixels by default, but they can also be in percent (like "80%").

```
<html>
```

```
<body>
```

```
<iframe src="demo_iframe.htm" width="200" height="200"></iframe>
```

```
</body>
```

```
</html>
```

Iframe - Remove the Border

The frameborder attribute specifies whether or not to display a border around the iframe.

Set the attribute value to "0" to remove the border:

```
<html>
```

```
<body>
```

```
<iframe src="demo_iframe.htm" frameborder="0"></iframe>
```

```
</body>
```

```
</html>
```

Use iframe as a Target for a Link

An iframe can be used as the target frame for a link.

The target attribute of the link must refer to the name attribute of the iframe:

```
<html>
```

```
<body>
```

```
<iframe width="100%" height="300px" src="demo_iframe.htm" name="iframe_a"></iframe>
```

```
<p><a href="http://www.w3schools.com" target="iframe_a">W3Schools.com</a></p>
```

```
<p>When the target of a link matches the name of an iframe, the link will open in the iframe.</p>
```

</body>

</html>

Sorted by Color Name

Colors sorted by HEX values

Click on a color name (or a hex value) to view the color as the background-color along with different text colors:

Color Name	HEX	Color	Shades	Mix
AliceBlue	#F0F8FF		Shades	Mix
AntiqueWhite	#FAEBD7		Shades	Mix
Aqua	#00FFFF		Shades	Mix
Aquamarine	#7FFFD4		Shades	Mix
Azure	#F0FFFF		Shades	Mix
Beige	#F5F5DC		Shades	Mix
Bisque	#FFE4C4		Shades	Mix
Black	#000000		Shades	Mix
BlanchedAlmond	#FFEBCD		Shades	Mix
Blue	#0000FF		Shades	Mix
BlueViolet	#8A2BE2		Shades	Mix
Brown	#A52A2A		Shades	Mix
BurlyWood	#DEB887		Shades	Mix
CadetBlue	#5F9EA0		Shades	Mix
Chartreuse	#7FFF00		Shades	Mix
Chocolate	#D2691E		Shades	Mix
Coral	#FF7F50		Shades	Mix
CornflowerBlue	#6495ED		Shades	Mix
Cornsilk	#FFF8DC		Shades	Mix
Crimson	#DC143C		Shades	Mix
Cyan	#00FFFF		Shades	Mix
DarkBlue	#00008B		Shades	Mix
DarkCyan	#008B8B		Shades	Mix
DarkGoldenRod	#B8860B		Shades	Mix
DarkGray	#A9A9A9		Shades	Mix
DarkGreen	#006400		Shades	Mix
DarkKhaki	#BDB76B		Shades	Mix
DarkMagenta	#8B008B		Shades	Mix
DarkOliveGreen	#556B2F		Shades	Mix
DarkOrange	#FF8C00		Shades	Mix
DarkOrchid	#9932CC		Shades	Mix
DarkRed	#8B0000		Shades	Mix
DarkSalmon	#E9967A		Shades	Mix
DarkSeaGreen	#8FBC8F		Shades	Mix
DarkSlateBlue	#483D8B		Shades	Mix
DarkSlateGray	#2F4F4F		Shades	Mix
DarkTurquoise	#00CED1		Shades	Mix
DarkViolet	#9400D3		Shades	Mix
DeepPink	#FF1493		Shades	Mix
DeepSkyBlue	#00BFFF		Shades	Mix
DimGray	#696969		Shades	Mix
DodgerBlue	#1E90FF		Shades	Mix

FireBrick	#B22222	Shades	Mix	
FloralWhite	#FFFAF0	Shades	Mix	
ForestGreen	#228B22	Shades	Mix	
Fuchsia	#FF00FF	Shades	Mix	
Gainsboro	#DCDCDC	Shades	Mix	
GhostWhite	#F8F8FF	Shades	Mix	
Gold	#FFD700	Shades	Mix	
GoldenRod	#DAA520	Shades	Mix	
Gray	#808080	Shades	Mix	
Green	#008000	Shades	Mix	
GreenYellow	#ADFF2F	Shades	Mix	
HoneyDew	#F0FFF0	Shades	Mix	
HotPink	#FF69B4	Shades	Mix	
IndianRed	#CD5C5C	Shades	Mix	
Indigo	#4B0082	Shades	Mix	
Ivory	#FFFFFF	Shades	Mix	
Khaki	#F0E68C	Shades	Mix	
Lavender	#E6E6FA	Shades	Mix	
LavenderBlush	#FFF0F5	Shades	Mix	Mix
LawnGreen	#7CFC00	Shades	Mix	
LemonChiffon	#FFFACD	Shades	Mix	Mix
LightBlue	#ADD8E6	Shades	Mix	
LightCoral	#F08080	Shades	Mix	
LightCyan	#E0FFFF	Shades	Mix	
LightGoldenRodYellow	#FAFAD2	Shades	Mix	Mix
LightGray	#D3D3D3	Shades	Mix	
LightGreen	#90EE90	Shades	Mix	
LightPink	#FFB6C1	Shades	Mix	
LightSalmon	#FFA07A	Shades	Mix	
LightSeaGreen	#20B2AA	Shades	Mix	Mix
LightSkyBlue	#87CEFA	Shades	Mix	
LightSlateGray	#778899	Shades	Mix	
LightSteelBlue	#B0C4DE	Shades	Mix	
LightYellow	#FFFFE0	Shades	Mix	
Lime	#00FF00	Shades	Mix	
LimeGreen	#32CD32	Shades	Mix	
Linen	#FAF0E6	Shades	Mix	
Magenta	#FF00FF	Shades	Mix	
Maroon	#800000	Shades	Mix	
MediumAquaMarine	#66CDAA	Shades	Mix	Mix
MediumBlue	#0000CD	Shades	Mix	
MediumOrchid	#BA55D3	Shades	Mix	
MediumPurple	#9370DB	Shades	Mix	
MediumSeaGreen	#3CB371	Shades	Mix	
MediumSlateBlue	#7B68EE	Shades	Mix	
MediumSpringGreen	#00FA9A	Shades	Mix	Mix
MediumTurquoise	#48D1CC	Shades	Mix	
MediumVioletRed	#C71585	Shades	Mix	
MidnightBlue	#191970	Shades	Mix	
MintCream	#F5FFFA	Shades	Mix	
MistyRose	#FFE4E1	Shades	Mix	
Moccasin	#FFE4B5	Shades	Mix	

NavajoWhite	#FFDEAD	Shades	Mix
Navy	#000080	Shades	Mix
OldLace	#FDF5E6	Shades	Mix
Olive	#808000	Shades	Mix
OliveDrab	#6B8E23	Shades	Mix
Orange	#FFA500	Shades	Mix
OrangeRed	#FF4500	Shades	Mix
Orchid	#DA70D6	Shades	Mix
PaleGoldenRod	#EEE8AA	Shades	Mix
PaleGreen	#98FB98	Shades	Mix
PaleTurquoise	#AFEEEE	Shades	Mix
PaleVioletRed	#DB7093	Shades	Mix
PapayaWhip	#FFEFD5	Shades	Mix
PeachPuff	#FFDAB9	Shades	Mix
Peru	#CD853F	Shades	Mix
Pink	#FFC0CB	Shades	Mix
Plum	#DDA0DD	Shades	Mix
PowderBlue	#B0E0E6	Shades	Mix
Purple	#800080	Shades	Mix
Red	#FF0000	Shades	Mix
RosyBrown	#BC8F8F	Shades	Mix
RoyalBlue	#4169E1	Shades	Mix
SaddleBrown	#8B4513	Shades	Mix
Salmon	#FA8072	Shades	Mix
SandyBrown	#F4A460	Shades	Mix
SeaGreen	#2E8B57	Shades	Mix
SeaShell	#FFF5EE	Shades	Mix
Sienna	#A0522D	Shades	Mix
Silver	#C0C0C0	Shades	Mix
SkyBlue	#87CEEB	Shades	Mix
SlateBlue	#6A5ACD	Shades	Mix
SlateGray	#708090	Shades	Mix
Snow	#FFFAFA	Shades	Mix
SpringGreen	#00FF7F	Shades	Mix
SteelBlue	#4682B4	Shades	Mix
Tan	#D2B48C	Shades	Mix
Teal	#008080	Shades	Mix
Thistle	#D8BFD8	Shades	Mix
Tomato	#FF6347	Shades	Mix
Turquoise	#40E0D0	Shades	Mix
Violet	#EE82EE	Shades	Mix
Wheat	#F5DEB3	Shades	Mix
White	#FFFFFF	Shades	Mix
WhiteSmoke	#F5F5F5	Shades	Mix
Yellow	#FFFF00	Shades	Mix
YellowGreen	#9ACD32	Shades	Mix

In the Stone Age

In the stone age, when computers only supported 256 different colors, a list of 216 "Web Safe Colors" was suggested as a Web standard, reserving 40 fixed system colors.

This 216 cross-browser color palette was created to ensure that all computers would display colors correctly:

000000	000033	000066	000099	0000CC	0000FF
003300	003333	003366	003399	0033CC	0033FF
006600	006633	006666	006699	0066CC	0066FF
009900	009933	009966	009999	0099CC	0099FF
00CC00	00CC33	00CC66	00CC99	00CCCC	00CCFF
00FF00	00FF33	00FF66	00FF99	00FFCC	00FFFF
330000	330033	330066	330099	3300CC	3300FF
333300	333333	333366	333399	3333CC	3333FF
336600	336633	336666	336699	3366CC	3366FF
339900	339933	339966	339999	3399CC	3399FF
33CC00	33CC33	33CC66	33CC99	33CCCC	33CCFF
33FF00	33FF33	33FF66	33FF99	33FFCC	33FFFF
660000	660033	660066	660099	6600CC	6600FF
663300	663333	663366	663399	6633CC	6633FF
666600	666633	666666	666699	6666CC	6666FF
669900	669933	669966	669999	6699CC	6699FF
66CC00	66CC33	66CC66	66CC99	66CCCC	66CCFF
66FF00	66FF33	66FF66	66FF99	66FFCC	66FFFF
990000	990033	990066	990099	9900CC	9900FF
993300	993333	993366	993399	9933CC	9933FF
996600	996633	996666	996699	9966CC	9966FF
999900	999933	999966	999999	9999CC	9999FF
99CC00	99CC33	99CC66	99CC99	99CCCC	99CCFF
99FF00	99FF33	99FF66	99FF99	99FFCC	99FFFF
CC0000	CC0033	CC0066	CC0099	CC00CC	CC00FF
CC3300	CC3333	CC3366	CC3399	CC33CC	CC33FF
CC6600	CC6633	CC6666	CC6699	CC66CC	CC66FF
CC9900	CC9933	CC9966	CC9999	CC99CC	CC99FF
CCCC00	CCCC33	CCCC66	CCCC99	CCCCCC	CCCCFF
CCFF00	CCFF33	CCFF66	CCFF99	CCFFCC	CCFFFF
FF0000	FF0033	FF0066	FF0099	FF00CC	FF00FF
FF3300	FF3333	FF3366	FF3399	FF33CC	FF33FF
FF6600	FF6633	FF6666	FF6699	FF66CC	FF66FF
FF9900	FF9933	FF9966	FF9999	FF99CC	FF99FF
FFCC00	FFCC33	FFCC66	FFCC99	FFCCCC	FFCCFF
FFFF00	FFFF33	FFFF66	FFFF99	FFFFCC	FFFFFF

HTML JavaScripts

The HTML <script> Tag

The <script> tag is used to define a client-side script, such as a JavaScript.

The <script> element either contains scripting statements or it points to an external script file through the src attribute.

Common uses for JavaScript are image manipulation, form validation, and dynamic changes of content.

The script below writes Hello JavaScript! into an HTML element with id="demo":

```
<html>
<body>

<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML = "Hello JavaScript!";
</script>

</body>
</html>
```

The HTML <noscript> Tag

The <noscript> tag is used to provide an alternate content for users that have disabled scripts in their browser or have a browser that doesn't support client-side scripting.

The <noscript> element can contain all the elements that you can find inside the <body> element of a normal HTML page.

The content inside the <noscript> element will only be displayed if scripts are not supported, or are disabled in the user's browser.

```
<html>
<body>

<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML = "Hello JavaScript!";
</script>

<noscript>Sorry, your browser does not support JavaScript!</noscript>

<p>A browser without support for JavaScript will show the text written inside the noscript
element.</p>

</body>
</html>
```

A Taste of JavaScript (From Our JavaScript Tutorial)

Here are some examples of what JavaScript can do:
JavaScript can change HTML content:

```
<html>
<body>

<h1>My First JavaScript</h1>

<p>JavaScript can change the content of an HTML element:</p>
```

```
<button type="button" onclick="myFunction()">Click Me!</button>
```

```
<p id="demo">This is a demonstration.</p>
```

```
<script>
function myFunction() {
  document.getElementById("demo").innerHTML = "Hello JavaScript!";
}
</script>
```

```
</body>
```

```
</html>
```

JavaScript can change HTML styles:

```
<html>
```

```
<body>
```

```
<h1>My First JavaScript</h1>
```

```
<p id="demo">JavaScript can change the style of an HTML element.</p>
```

```
<script>
function myFunction() {
  document.getElementById("demo").style.fontSize = "25px";
}
</script>
```

```
<button type="button" onclick="myFunction()">Click Me!</button>
```

```
</body>
```

```
</html>
```

JavaScript can change HTML attributes:

```
document.getElementById("image").src = "picture.gif";
```

The HTML <base> Element

The <base> tag specifies the base URL/target for all relative URLs in a page:

Example

```
<head>
```

```
<base href="http://www.w3schools.com/images/" target="_blank">
```

```
</head>
```

The HTML <link> Element

The <link> tag defines the relationship between a document and an external resource.

The <link> tag is most used to link to style sheets:

Example

```
<head>
```

```
<link rel="stylesheet" type="text/css" href="mystyle.css">
```

</head>

The HTML <style> Element

The <style> tag is used to define style information for an HTML document.

Inside the <style> element you specify how HTML elements should render in a browser:

Example

```
<head>
<style>
body {background-color:yellow;}
p {color:blue;}
</style>
</head>
```

The HTML <meta> Element

Metadata is data (information) about data.

The <meta> tag provides metadata about the HTML document. Metadata will not be displayed on the page, but will be machine parsable.

Meta elements are typically used to specify page description, keywords, author of the document, last modified, and other metadata.

The metadata can be used by browsers (how to display content or reload page), search engines (keywords), or other web services.

<meta> tags always go inside the <head> element.

<meta> Tags - Examples of Use

Define keywords for search engines:

Example

```
<meta name="keywords" content="HTML, CSS, XML, XHTML, JavaScript">
```

Define a description of your web page:

Example

```
<meta name="description" content="Free Web tutorials on HTML and CSS">
```

Define the author of a page:

Example

```
<meta name="author" content="Hege Refsnes">
```

Refresh document every 30 seconds:

Example

```
<meta http-equiv="refresh" content="30">
```

The HTML <script> Element

The <script> tag is used to define a client-side JavaScript.

The script below writes Hello JavaScript! into an HTML element with id="demo":

```
<html>
<head>
```

```
<script>
function myFunction() {
  document.getElementById("demo").innerHTML = "Hello JavaScript!";
}
</script>
</head>
```

```
<body>
```

```
<h1>My Web Page</h1>
```

```
<p id="demo">A Paragraph</p>
```

```
<button type="button" onclick="myFunction()">Try it</button>
```

```
</body>
```

```
</html>
```

HTML Tutorial

HTML HOME HTML Introduction HTML Editors HTML Basic HTML Elements HTML Attributes HTML Headings HTML Paragraphs HTML Styles HTML Formatting HTML Quotations HTML Computercode HTML Comments HTML CSS HTML Links HTML Images HTML Tables HTML Lists HTML Blocks HTML Classes HTML Layout HTML Responsive HTML Forms HTML Iframes HTML Color Names HTML Color Values HTML Color Shades HTML JavaScript HTML Head HTML Entities HTML Symbols HTML Charset HTML URL Encode HTML XHTML HTML5

HTML5 Intro HTML5 Support HTML5 Elements HTML5 Semantics HTML5 Migration HTML5 Syntax

HTML Forms

HTML Form Elements HTML Input Types HTML Input Attributes

HTML Graphics

HTML Canvas HTML SVG

HTML Media

HTML Media HTML Video HTML Audio HTML Plug-ins HTML YouTube

HTML APIs

HTML Geolocation HTML Drag/Drop HTML Local Storage HTML App Cache HTML Web Workers HTML SSE

HTML Examples

HTML Examples HTML Quiz HTML5 Quiz HTML Certificate HTML5 Certificate HTML

Summary

HTML References

HTML Tag List HTML Attributes HTML Events HTML Canvas HTML Audio/Video HTML Doctypes HTML Colornames HTML Colorpicker HTML Colormixer HTML Character Sets HTML URL Encode HTML Lang Codes HTTP Messages HTTP Methods Keyboard Shortcuts

HTML Head

« Previous

Next Chapter »

Examples

Try it Yourself - Examples

<title> - Define a title for an HTML document

Use the <title> tag to define a title for a document.

<base> - Default URL and target for all links

Use the <base> tag to specify a default URL and a default target for all links on a page.

<meta> - Provide metadata for an HTML document

Use <meta> elements to specify a description, keywords, author, and character set of a document.

The HTML <head> Element

The <head> element is a container for all the head elements. Elements inside <head> can include scripts, instruct the browser where to find style sheets, provide meta information, and more.

The following tags can be added to the head section: <title>, <style>, <meta>, <link>, <script>, <noscript>, and <base>.

The HTML <title> Element

The <title> tag defines the title of the document.

The <title> element is required in all HTML/XHTML documents.

The <title> element:

- defines a title in the browser toolbar
- provides a title for the page when it is added to favorites
- displays a title for the page in search engine results

A simplified HTML document:

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Title of the document</title>
</head>

<body>
The content of the document.....
</body>

</html>
```

The HTML <base> Element

The <base> tag specifies the base URL/target for all relative URLs in a page:

Example

```
<head>
<base href="http://www.w3schools.com/images/" target="_blank">
</head>
```

The HTML <link> Element

The <link> tag defines the relationship between a document and an external resource.

The <link> tag is most used to link to style sheets:

Example

```
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
```

The HTML <style> Element

The <style> tag is used to define style information for an HTML document.

Inside the <style> element you specify how HTML elements should render in a browser:

Example

```
<head>
<style>
body {background-color:yellow;}
p  {color:blue;}
</style>
</head>
```

The HTML <meta> Element

Metadata is data (information) about data.

The <meta> tag provides metadata about the HTML document. Metadata will not be displayed on the page, but will be machine parsable.

Meta elements are typically used to specify page description, keywords, author of the document, last modified, and other metadata.

The metadata can be used by browsers (how to display content or reload page), search engines (keywords), or other web services.

<meta> tags always go inside the <head> element.

<meta> Tags - Examples of Use

Define keywords for search engines:

Example

```
<meta name="keywords" content="HTML, CSS, XML, XHTML, JavaScript">
```

Define a description of your web page:

Example

```
<meta name="description" content="Free Web tutorials on HTML and CSS">
```

Define the author of a page:

Example

```
<meta name="author" content="Hege Refsnes">
```

Refresh document every 30 seconds:

Example

```
<meta http-equiv="refresh" content="30">
```

The HTML <script> Element

The <script> tag is used to define a client-side JavaScript.

The script below writes Hello JavaScript! into an HTML element with id="demo":

Example

```
<script>
function myFunction {
    document.getElementById("demo").innerHTML = "Hello JavaScript!";
}
</script>
```

Try it Yourself »

Note To learn all about JavaScript, visit our JavaScript Tutorial!

HTML head Elements

Tag Description

<head>	Defines information about the document
<title>	Defines the title of a document
<base>	Defines a default address or a default target for all links on a page
<link>	Defines the relationship between a document and an external resource
<meta>	Defines metadata about an HTML document
<script>	Defines a client-side script
<style>	Defines style information for a document

HTML Entities

Some characters are reserved in HTML.

If you use the less than (<) or greater than (>) signs in your text, the browser might mix them with tags.

Character entities are used to display reserved characters in HTML.

A character entity looks like this:

&entity_name;

OR

&#entity_number;

To display a less than sign we must write: < or <

Non Breaking Space

A common character entity used in HTML is the non breaking space ().

Remember that browsers will always truncate spaces in HTML pages. If you write 10 spaces in your text, the browser will remove 9 of them. To add real spaces to your text, you can use the character entity.

Some Other Useful HTML Character Entities

Result	Description	Entity Name	Entity Number
	non-breaking space	 	
<	less than	<	<
>	greater than	>	>

& ampersand & &
 ¢ cent ¢ ¢
 £ pound £ £
 ¥ yen ¥ ¥
 € euro € €
 © copyright © ©
 ® registered trademark ® ®
 Entity names are case sensitive.

Combining Diacritical Marks

A diacritical mark is a "glyph" added to a letter.

Some diacritical marks, like grave (`) and acute (´) are called accents.

Diacritical marks can appear both above and below a letter, inside a letter, and between two letters.

Diacritical marks can be used in combination with alphanumeric characters, to produce a character that is not present in the character set (encoding) used in the page.

Here are some examples:

Mark	Character	Construct	Result
`	a	à	à
'	a	á	á
^	a	â	â
~	a	ã	ã
`	O	Ò	Ò
'	O	Ó	Ó
^	O	Ô	Ô
~	O	Õ	Õ

HTML Symbol Entities

HTML entities were described in the previous chapter.

Many mathematical, technical, and currency symbols, are not present on a normal keyboard.

To add these symbols to an HTML page, you can use an HTML entity name.

If no entity name exists, you can use an entity number; a decimal (or hexadecimal) reference.

Note : - If you use an HTML entity name or a hexadecimal number, the character will always display correctly.

This is independent of what character set (encoding) your page uses!

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

```
<p>I will display &euro;</p>
<p>I will display &#8364;</p>
<p>I will display &#x20AC;</p>
```

</body>

</html>

output result

I will display €

I will display €

I will display €

Some Mathematical Symbols Supported by HTML

Char	Number	Entity	Description
∀	∀	∀	FOR ALL
∂	∂	∂	PARTIAL DIFFERENTIAL
∃	∃	∃	THERE EXISTS
∅	∅	∅	EMPTY SETS
∇	∇	∇	NABLA
∈	∈	∈	ELEMENT OF
∉	∉	∉	NOT AN ELEMENT OF
⊃	∋	∋	CONTAINS AS MEMBER
∏	∏	∏	N-ARY PRODUCT
∑	∑	∑	N-ARY SUMMATION

Some Greek Letters Supported by HTML

Char	Number	Entity	Description
Α	Α	Α	GREEK CAPITAL LETTER ALPHA
Β	Β	Β	GREEK CAPITAL LETTER BETA
Γ	Γ	Γ	GREEK CAPITAL LETTER GAMMA
Δ	Δ	Δ	GREEK CAPITAL LETTER DELTA
Ε	Ε	Ε	GREEK CAPITAL LETTER EPSILON
Ζ	Ζ	Ζ	GREEK CAPITAL LETTER ZETA

Some Other Entities Supported by HTML

Char	Number	Entity	Description
©	©	©	COPYRIGHT SIGN
®	®	®	REGISTERED SIGN
€	€	€	EURO SIGN
™	™	™	TRADEMARK
←	←	←	LEFTWARDS ARROW
↑	↑	↑	UPWARDS ARROW
→	→	→	RIGHTWARDS ARROW
↓	↓	↓	DOWNWARDS ARROW
♠	♠	♠	BLACK SPADE SUIT
♣	♣	♣	BLACK CLUB SUIT
♥	♥	♥	BLACK HEART SUIT
♦	♦	&diamonds;	BLACK DIAMOND SUIT

URL Encoding Examples

Character	ASCII encoding	UTF-8 encoding
€	%80	%E2%82%AC
£	%A3	%C2%A3
©	%A9	%C2%A9
®	%AE	%C2%AE
À	%C0	%C3%80

Á %C1 %C3%81
Â %C2 %C3%82
Ã %C3 %C3%83
Ä %C4 %C3%84
Å %C5 %C3%85

HTML5 Introduction

What is New in HTML5?

The DOCTYPE declaration for HTML5 is very simple:

```
<!DOCTYPE html>
```

The character encoding (charset) declaration is also very simple:

```
<meta charset="UTF-8">
```

HTML5 Example:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Title of the document</title>
</head>

<body>
Content of the document.....
</body>

</html>
```

The default character encoding in HTML5 is UTF-8

New HTML5 Elements

The most interesting new elements are:

New semantic elements like <header>, <footer>, <article>, and <section>.

New form controls like number, date, time, calendar, and range.

New graphic elements: <svg> and <canvas>.

New multimedia elements: <audio> and <video>.

New HTML5 API's (Application Programming Interfaces)

The most interesting new API's are:

- HTML Geolocation
- HTML Drag and Drop
- HTML Local Storage
- HTML Application Cache

HTML Web Workers

HTML SSE

Note:- Local storage is a powerful replacement for cookies.

Elements Removed in HTML5

The following HTML4 elements have been removed from HTML5:

Element	Use instead
<acronym>	<abbr>
<applet>	<object>
<basefont>	CSS
<big>	CSS
<center>	CSS
<dir>	
	CSS
<frame>	
<frameset>	
<noframes>	
<strike>	CSS
<tt>	CSS

HTML5 History

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

Version	Year
Tim Berners-Lee	1989
HTML	1991
HTML+	1993
HTML 2.0	1995
HTML 3.2	1997
HTML 4.01	1999
XHTML	2000
HTML5	2012

Define HTML5 Elements as Block Elements

HTML5 defines 8 new semantic HTML elements. All these are block level elements.

To secure correct behavior in older browsers, you can set the CSS display property to block:

Example

```
header, section, footer, aside, nav, main, article, figure {  
  display: block;  
}
```

Adding New Elements to HTML

You can add any new element to HTML with a browser trick:

This example adds a new element called <myHero> to HTML, and defines a display style for it:

```
<html>
```

```

<head>
  <title>Styling the article element</title>
  <script>document.createElement("myHero")</script>
  <style>
    myHero {
      display:block;
      background-color:#ddd;
      padding: 50px;
      font-size: 30px;
    }
  </style>
</head>

<body>

<h1>My First Heading</h1>

<p>My first paragraph.</p>

<myHero>My First Hero</myHero>

</body>
</html>

```

Problem With Internet Explorer

You could use the solution described above, for all new HTML5 elements, but:

Note :- internet Explorer 8 and earlier, does not allow styling of unknown elements.

An HTML5 Skeleton

```

<!DOCTYPE html>
<html lang="en">
<head>
<title>HTML5</title>
<meta charset="utf-8">

<!--[if lt IE 9]>
<script src="http://html5shiv.googlecode.com/svn/trunk/html5.js">
</script>
<![endif]-->

<style>
body {font-family: Verdana, sans-serif; font-size:0.8em;}
header, nav, section, article, footer
{border:1px solid grey; margin:5px; padding:8px;}
nav ul {margin:0; padding:0;}
nav ul li {display:inline; margin:5px;}
</style>
</head>

```

```

<body>

<header>
<h1>HTML5 Skeleton</h1>
</header>

<nav>
<ul>
  <li><a href="html5_semantic_elements.asp">HTML5 Semantic</a></li>
  <li><a href="html5_form_elements.asp">HTML5 Forms</a></li>
  <li><a href="html5_canvas.asp">HTML5 Graphics</a></li>
</ul>
</nav>

<section>

<h2>Famous Cities</h2>

<article>
<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>
</article>

<article>
<h2>Paris</h2>
<p>Paris is the capital and most populous city of France.</p>
</article>

<article>
<h2>Tokyo</h2>
<p>Tokyo is the capital of Japan, the center of the Greater Tokyo Area,
and the most populous metropolitan area in the world.</p>
</article>

</section>

<footer>
<p>&copy; 2014 W3Schools. All rights reserved.</p>
</footer>

</body>
</html>

```

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 the document

<aside>	Defines content aside from the page content
<bdi>	Defines a part of text that might be formatted in a different direction from other text
<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, like illustrations, diagrams, photos, code listings, etc.
<footer>	Defines a footer for the document or a section
<header>	Defines a header for the document or a section
<main>	Defines the main content of a document
<mark>	Defines marked or 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 in the document
<progress>	Defines 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 the document
<summary>	Defines a visible heading for a <details> element
<time>	Defines a date/time
<wbr>	Defines a possible line-break

Read more about HTML5 Semantics.

New Form Elements

Tag	Description
<datalist>	Defines pre-defined options for input controls
<keygen>	Defines a key-pair generator field (for forms)
<output>	Defines the result of a calculation

Read more about HTML5 Form Elements.

New Input Types

New Input Attributes

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

autocomplete
autofocus
form

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

Read more about HTML5 Input Types.

Read more about HTML5 Input Attributes.
HTML5 - New Attribute Syntax

HTML5 allows 4 different syntaxes for attributes.

This example demonstrates the different syntaxes used in an `<input>` tag:

Type	Example
Empty	<code><input type="text" value="John Doe" disabled></code>
Unquoted	<code><input type="text" value=John></code>
Double-quoted	<code><input type="text" value="John Doe"></code>
Single-quoted	<code><input type="text" value='John Doe'></code>

In HTML5, all 4 syntaxes may be used, depending on what is needed for the attribute.

HTML5 Graphics

Tag	Description
<code><canvas></code>	Defines graphic drawing using JavaScript
<code><svg></code>	Defines graphic drawing using SVG

Read more about HTML5 Canvas.

Read more about HTML5 SVG.

New Media Elements

Tag	Description
<code><audio></code>	Defines sound or music content
<code><embed></code>	Defines containers for external applications (like plug-ins)
<code><source></code>	Defines sources for <code><video></code> and <code><audio></code>
<code><track></code>	Defines tracks for <code><video></code> and <code><audio></code>
<code><video></code>	Defines video or movie content

HTML5 Semantic Elements

What are Semantic Elements?

A semantic element clearly describes its meaning to both the browser and the developer.

Examples of non-semantic elements: `<div>` and `` - Tells nothing about its content.

Examples of semantic elements: <form>, <table>, and - Clearly defines its content.

New Semantic Elements in HTML5

Many web sites contain HTML code like: <div id="nav"> <div class="header"> <div id="footer"> 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 Semantic Elements

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 Web site's home page could be split into sections for introduction, content, and contact information.

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

```
<section>
<h1>WWF</h1>
<p>
```

The World Wide Fund for Nature (WWF) is an international organization working on issues regarding the conservation, research and restoration of the environment, formerly named the World Wildlife Fund. WWF was founded in 1961.

```
</p>
</section>
```

```
<section>
<h1>WWF's Panda symbol</h1>
<p>
```

The Panda has become the symbol of WWF. The well-known panda logo of WWF originated from a panda named Chi Chi that was transferred from the Beijing Zoo to the London Zoo in the same year of the establishment of WWF.

```
</p>  
</section>
```

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

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

```
<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>
```

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

Nesting Semantic Elements

In the HTML5 standard, the <article> element defines a complete, self-contained block of related elements.

The <section> element is defined as a block of related elements.

Can we use the definitions to decide how to nest elements? No, we cannot!

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.

Note :- Newspaper: The sports articles in the sports section, 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:

```
<html>
<body>

<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>

</body>
</html>
```

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 can have several <footer> elements in one document.

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

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

</body>
</html>
```

HTML5 <nav> Element

The <nav> element defines a set of navigation links.

The <nav> element is intended for large blocks of navigation links. However, not all links in a document should be inside a <nav> element!

Example

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

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

</body>
</html>
```

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.

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

<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>

</body>
</html>
```

HTML5 <figure> and <figcaption> Elements

In books and newspapers, it is common to have captions with images.

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

With HTML5, images and captions can be grouped together in <figure> elements:

```
<html>
<body>

<p>The Pulpit Rock is a massive cliff 604 metres (1982 feet) above Lysefjorden, opposite the
Kjerag plateau, in Forsand, Ryfylke, Norway. The top of the cliff is approximately 25 by 25 metres
(82 by 82 feet) square and almost flat, and is a famous tourist attraction in Norway.</p>

<figure>

```

```
<figcaption>Fig.1 - The Pulpit Rock, Norway.</figcaption>
</figure>
```

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

The `` element defines the image, the `<figcaption>` element defines the caption.

Why Semantic HTML5 Elements?

With HTML4, developers used their own favorite attribute names to style page elements:

header, top, bottom, footer, menu, navigation, main, container, content, article, sidebar, topnav, ...

This made it impossible for search engines to identify the correct web page content.

With HTML5 elements like: `<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 goes to our complete HTML5 Reference.

Tag	Description
<code><article></code>	Defines an article
<code><aside></code>	Defines content aside from the page content
<code><details></code>	Defines additional details that the user can view or hide
<code><figcaption></code>	Defines a caption for a <code><figure></code> element
<code><figure></code>	Specifies self-contained content, like illustrations, diagrams, photos, code listings, etc.
<code><footer></code>	Defines a footer for a document or section
<code><header></code>	Specifies a header for a document or section
<code><main></code>	Specifies the main content of a document
<code><mark></code>	Defines marked/highlighted text
<code><nav></code>	Defines navigation links
<code><section></code>	Defines a section in a document
<code><summary></code>	Defines a visible heading for a <code><details></code> element
<code><time></code>	Defines a date/time

Migration from HTML4 to HTML5

This chapter is entirely about how to migrate from a typical HTML4 page to a typical HTML5 page.

This chapter demonstrates how to convert an existing HTML4 page into an HTML5 page, without destroying anything of the original content or structure.

Note You can migrate to HTML5 from HTML4, and XHTML, using the same recipe.

Typical HTML4	Typical HTML5
<div id="header">	<header>
<div id="menu">	<nav>
<div id="content">	<section>
<div id="post">	<article>
<div id="footer">	<footer>

Typical HTML4 Page

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>HTML4</title>
```

```
<style>
body {
  font-family: Verdana, sans-serif; font-size: 0.8em;
}
div#header, div#footer, div#content, div#post {
  border: 1px solid grey;
  margin: 5px; margin-bottom: 15px; padding: 8px;
  background-color: white;
}
div#header, div#footer {
  color: white; background-color: #444; margin-bottom: 5px;
}
div#content {
  background-color: #ddd;
}
div#menu ul {
  margin: 0; padding: 0;
}
div#menu ul li {
  display: inline; margin: 5px;
}
</style>
</head>
<body>
```

```
<div id="header">
<h1>Monday Times</h1>
</div>
```

```
<div id="menu">
<ul>
  <li>News</li>
  <li>Sports</li>
  <li>Weather</li>
```

```
</ul>
</div>
```

```
<div id="content">
<h2>News Section</h2>
```

```
<div id="post">
<h2>News Article</h2>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
</div>
```

```
<div id="post">
<h2>News Article</h2>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
</div>
```

```
</div>
```

```
<div id="footer">
<p>&copy; 2014 Monday Times. All rights reserved.</p>
</div>
```

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

Change to HTML5 Doctype

Change the doctype, from the HTML4 doctype:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

to the HTML5 doctype:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>HTML5</title>
<style>
body {
font-family: Verdana, sans-serif; font-size: 0.8em;
```



```

}
div#header,div#footer,div#content,div#post {
  border:1px solid grey;
  margin:5px;margin-bottom:15px;padding:8px;
  background-color:white;
}
div#header,div#footer {
  color:white;background-color:#444;margin-bottom:5px;
}
div#content {
  background-color:#ddd;
}
div#menu ul {
  margin:0;padding:0;
}
div#menu ul li {
  display:inline; margin:5px;
}
</style>
</head>
<body>

<div id="header">
<h1>Monday Times</h1>
</div>

<div id="menu">
<ul>
  <li>News</li>
  <li>Sports</li>
  <li>Weather</li>
</ul>
</div>

<div id="content">
<h2>News Section</h2>

<div id="post">
<h2>News Article</h2>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
</div>

<div id="post">
<h2>News Article</h2>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum

```

```
lurum hurum turum.</p>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
</div>
```

```
</div>
```

```
<div id="footer">
<p>&copy; 2014 Monday Times. All rights reserved.</p>
</div>
```

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

Change to HTML5 Encoding

Change the encoding information, from HTML4:

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
```

to HTML5:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>HTML5</title>
<style>
body {
  font-family: Verdana, sans-serif; font-size: 0.8em;
}
div#header, div#footer, div#content, div#post {
  border: 1px solid grey;
  margin: 5px; margin-bottom: 15px; padding: 8px;
  background-color: white;
}
div#header, div#footer {
  color: white; background-color: #444; margin-bottom: 5px;
}
div#content {
  background-color: #ddd;
}
div#menu ul {
  margin: 0; padding: 0;
}
div#menu ul li {
  display: inline; margin: 5px;
}
</style>
</head>
<body>

<div id="header">
```

```
<h1>Monday Times</h1>
</div>
```

```
<div id="menu">
<ul>
  <li>News</li>
  <li>Sports</li>
  <li>Weather</li>
</ul>
</div>
```

```
<div id="content">
<h2>News Section</h2>
```

```
<div id="post">
<h2>News Article</h2>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
</div>
```

```
<div id="post">
<h2>News Article</h2>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
</div>
```

```
</div>
```

```
<div id="footer">
<p>&copy; 2014 Monday Times. All rights reserved.</p>
</div>
```

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

Add The Shiv

HTML5 semantic elements are supported in all modern browsers.

In addition, you can "teach" older browsers how to handle "unknown elements".

Add the shiv for Internet Explorer support:

```
<!DOCTYPE html>
```

```
<html lang="en">
<head>
<meta charset="utf-8">
<title>HTML5</title>

<!--[if lt IE 9]>
<script src="http://html5shiv.googlecode.com/svn/trunk/html5.js">
</script>
<![endif]-->

<style>
body {
  font-family: Verdana,sans-serif;font-size:0.8em;
}
div#header,div#footer,div#content,div#post {
  border:1px solid grey;
  margin:5px;margin-bottom:15px;padding:8px;
  background-color:white;
}
div#header,div#footer {
  color:white;background-color:#444;margin-bottom:5px;
}
div#content {
  background-color:#ddd;
}
div#menu ul {
  margin:0;padding:0;
}
div#menu ul li {
  display:inline; margin:5px;
}
</style>
</head>
<body>

<div id="header">
<h1>Monday Times</h1>
</div>

<div id="menu">
<ul>
  <li>News</li>
  <li>Sports</li>
  <li>Weather</li>
</ul>
</div>

<div id="content">
<h2>News Section</h2>

<div id="post">
<h2>News Article</h2>
```

```
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
</div>
```

```
<div id="post">
<h2>News Article</h2>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
</div>
```

```
</div>
```

```
<div id="footer">
<p>&copy; 2014 Monday Times. All rights reserved.</p>
</div>
```

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

\Add CSS for HTML5 Semantic Elements

Look at your existing CSS styles:

```
div#header,div#footer,div#content,div#post {
  border:1px solid grey;margin:5px;margin-bottom:15px;padding:8px;background-color:white;
}
div#header,div#footer {
  color:white;background-color:#444;margin-bottom:5px;
}
div#content {
  background-color:#ddd;
}
div#menu ul {
  margin:0;padding:0;
}
div#menu ul li {
  display:inline; margin:5px;
}
```

Duplicate with equal CSS styles for HTML5 semantic elements:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
```

```
<title>HTML5</title>

<!--[if lt IE 9]>
<script src="http://html5shiv.googlecode.com/svn/trunk/html5.js">
</script>
<![endif]-->

<style>
body {
  font-family: Verdana, sans-serif; font-size: 0.8em;
}

div#header, div#footer, div#content, div#post {
  border: 1px solid grey;
  margin: 5px; margin-bottom: 15px; padding: 8px;
  background-color: white;
}
div#header, div#footer {
  color: white; background-color: #444; margin-bottom: 5px;
}
div#content {
  background-color: #ddd;
}
div#menu ul {
  margin: 0; padding: 0;
}
div#menu ul li {
  display: inline; margin: 5px;
}

header, footer, section, article {
  border: 1px solid grey;
  margin: 5px; margin-bottom: 15px; padding: 8px;
  background-color: white;
}
header, footer {
  color: white; background-color: #444; margin-bottom: 5px;
}
section {
  background-color: #ddd;
}
nav ul {
  margin: 0; padding: 0;
}
nav ul li {
  display: inline; margin: 5px;
}
</style>
</head>
<body>

<div id="header">
```

```
<h1>Monday Times</h1>
</div>
```

```
<div id="menu">
<ul>
  <li>News</li>
  <li>Sports</li>
  <li>Weather</li>
</ul>
</div>
```

```
<div id="content">
<h2>News Section</h2>
```

```
<div id="post">
<h2>News Article</h2>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
</div>
```

```
<div id="post">
<h2>News Article</h2>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
</div>
```

```
</div>
```

```
<div id="footer">
<p>&copy; 2014 Monday Times. All rights reserved.</p>
</div>
```

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

Change to HTML5 <header> and <footer>

Change the <div> elements with id="header" and id="footer":

```
<div id="header">
  <h1>Monday Times</h1>
</div>
```

```
.
.
.
```

```
<div id="footer">
  <p>&amp;copy; 2014 W3Schools. All rights reserved.</p>
</div>
```

to HTML5 semantic <header> and <footer> elements:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>HTML5</title>

<!--[if lt IE 9]>
<script src="http://html5shiv.googlecode.com/svn/trunk/html5.js">
</script>
<![endif]-->

<style>
body {
  font-family: Verdana, sans-serif; font-size: 0.8em;
}

div#header, div#footer, div#content, div#post {
  border: 1px solid grey;
  margin: 5px; margin-bottom: 15px; padding: 8px;
  background-color: white;
}
div#header, div#footer {
  color: white; background-color: #444; margin-bottom: 5px;
}
div#content {
  background-color: #ddd;
}
div#menu ul {
  margin: 0; padding: 0;
}
div#menu ul li {
  display: inline; margin: 5px;
}

header, footer, section, article {
  border: 1px solid grey;
  margin: 5px; margin-bottom: 15px; padding: 8px;
  background-color: white;
}
header, footer {
  color: white; background-color: #444; margin-bottom: 5px;
}
section {
  background-color: #ddd;
}
nav ul {
```



```
margin:0;padding:0;
}
nav ul li {
display:inline; margin:5px;
}
</style>
</head>
<body>
```

```
<header>
<h1>Monday Times</h1>
</header>
```

```
<div id="menu">
<ul>
<li>News</li>
<li>Sports</li>
<li>Weather</li>
</ul>
</div>
```

```
<div id="content">
<h2>News Section</h2>
```

```
<div id="post">
<h2>News Article</h2>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
</div>
```

```
<div id="post">
<h2>News Article</h2>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
</div>
```

```
</div>
```

```
<footer>
<p>&copy; 2014 Monday Times. All rights reserved.</p>
</footer>
```

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

Change to HTML5 <nav>

Change the <div> element with id="menu":

```
<div id="menu">
  <ul>
    <li>News</li>
    <li>Sports</a></li>
    <li>Weather</li>
  </ul>
</div>
```

to an HTML5 semantic <nav> element:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>HTML5</title>

<!--[if lt IE 9]>
<script src="http://html5shiv.googlecode.com/svn/trunk/html5.js">
</script>
<![endif]-->

<style>
body {
  font-family: Verdana, sans-serif; font-size: 0.8em;
}

div#header, div#footer, div#content, div#post {
  border: 1px solid grey;
  margin: 5px; margin-bottom: 15px; padding: 8px;
  background-color: white;
}
div#header, div#footer {
  color: white; background-color: #444; margin-bottom: 5px;
}
div#content {
  background-color: #ddd;
}
div#menu ul {
  margin: 0; padding: 0;
}
div#menu ul li {
  display: inline; margin: 5px;
}

header, footer, section, article {
  border: 1px solid grey;
  margin: 5px; margin-bottom: 15px; padding: 8px;
  background-color: white;
}
header, footer {
```

```
    color:white;background-color:#444;margin-bottom:5px;
}
section {
    background-color:#ddd;
}
nav ul {
    margin:0;padding:0;
}
nav ul li {
    display:inline; margin:5px;
}
</style>
</head>
<body>
```

```
<header>
<h1>Monday Times</h1>
</header>
```

```
<nav>
<ul>
  <li>News</li>
  <li>Sports</li>
  <li>Weather</li>
</ul>
</nav>
```

```
<div id="content">
<h2>News Section</h2>
```

```
<div id="post">
<h2>News Article</h2>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
</div>
```

```
<div id="post">
<h2>News Article</h2>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
</div>
```

```
</div>
```

```
<footer>
<p>&copy; 2014 Monday Times. All rights reserved.</p>
</footer>
```

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

Change to HTML5 <section>

Change the <div> element with id="content":

```
<div id="content">
```

```
.
.
.
```

```
</div>
```

to an HTML5 semantic <section> element:

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

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

```
<title>HTML5</title>
```

```
<!--[if lt IE 9]>
```

```
<script src="http://html5shiv.googlecode.com/svn/trunk/html5.js">
```

```
</script>
```

```
<![endif]-->
```

```
<style>
```

```
body {
```

```
    font-family: Verdana, sans-serif; font-size: 0.8em;
```

```
}
```

```
div#header, div#footer, div#content, div#post {
```

```
    border: 1px solid grey;
```

```
    margin: 5px; margin-bottom: 15px; padding: 8px;
```

```
    background-color: white;
```

```
}
```

```
div#header, div#footer {
```

```
    color: white; background-color: #444; margin-bottom: 5px;
```

```
}
```

```
div#content {
```

```
    background-color: #ddd;
```

```
}
```

```
div#menu ul {
```

```
    margin: 0; padding: 0;
```

```
}
```

```
div#menu ul li {
```

```
    display: inline; margin: 5px;
```

```
}
```

```

header,footer,section,article {
  border:1px solid grey;
  margin:5px;margin-bottom:15px;padding:8px;
  background-color:white;
}
header,footer {
  color:white;background-color:#444;margin-bottom:5px;
}
section {
  background-color:#ddd;
}
nav ul {
  margin:0;padding:0;
}
nav ul li {
  display:inline; margin:5px;
}
</style>
</head>
<body>

<header>
<h1>Monday Times</h1>
</header>

<nav>
<ul>
  <li>News</li>
  <li>Sports</li>
  <li>Weather</li>
</ul>
</nav>

<section>
<h2>News Section</h2>

<div id="post">
<h2>News Article</h2>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
</div>

<div id="post">
<h2>News Article</h2>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>

```

```
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum  
lurum hurum turum.</p>  
</div>
```

```
</section>
```

```
<footer>  
<p>&copy; 2014 Monday Times. All rights reserved.</p>  
</footer>
```

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

Change to HTML5 <article>

Change all <div> element with class="post":

```
<div class="post">  
  <h2>News Article</h2>  
  <p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum  
  lurum hurum turum.</p>  
</div>
```

to HTML5 semantic <article> elements:

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
<meta charset="utf-8">  
<title>HTML5</title>  
  
<!--[if lt IE 9]>  
<script src="http://html5shiv.googlecode.com/svn/trunk/html5.js">  
</script>  
<![endif]-->
```

```
<style>  
body {  
  font-family: Verdana, sans-serif; font-size: 0.8em;  
}  
  
div#header, div#footer, div#content, div#post {  
  border: 1px solid grey;  
  margin: 5px; margin-bottom: 15px; padding: 8px;  
  background-color: white;  
}  
div#header, div#footer {  
  color: white; background-color: #444; margin-bottom: 5px;  
}  
div#content {  
  background-color: #ddd;  
}  
div#menu ul {
```

```

    margin:0;padding:0;
}
div#menu ul li {
    display:inline; margin:5px;
}

header,footer,section,article {
    border:1px solid grey;
    margin:5px;margin-bottom:15px;padding:8px;
    background-color:white;
}
header,footer {
    color:white;background-color:#444;margin-bottom:5px;
}
section {
    background-color:#ddd;
}
nav ul {
    margin:0;padding:0;
}
nav ul li {
    display:inline; margin:5px;
}
</style>
</head>
<body>

<header>
<h1>Monday Times</h1>
</header>

<nav>
<ul>
    <li>News</li>
    <li>Sports</li>
    <li>Weather</li>
</ul>
</nav>

<section>
<h2>News Section</h2>

<article>
<h2>News Article</h2>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
</article>

```

```
<article>
<h2>News Article</h2>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
</article>

</section>

<footer>
<p>&copy; 2014 Monday Times. All rights reserved.</p>
</footer>

</body>
</html>
```

Remove these "no longer needed" <style> elements:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>HTML5</title>

<!--[if lt IE 9]>
<script src="http://html5shiv.googlecode.com/svn/trunk/html5.js">
</script>
<![endif]-->

<style>
body {
  font-family: Verdana, sans-serif; font-size: 0.8em;
}
header, footer, section, article {
  border: 1px solid grey;
  margin: 5px; margin-bottom: 15px; padding: 8px;
  background-color: white;
}
header, footer {
  color: white; background-color: #444; margin-bottom: 5px;
}
section {
  background-color: #ddd;
}
nav ul {
  margin: 0; padding: 0;
}
nav ul li {
  display: inline; margin: 5px;
```



```

}
</style>
</head>
<body>

<header>
<h1>Monday Times</h1>
</header>

<nav>
<ul>
  <li>News</li>
  <li>Sports</li>
  <li>Weather</li>
</ul>
</nav>

<section>
<h2>News Section</h2>

<div id="post">
<h2>News Article</h2>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
</div>

<div id="post">
<h2>News Article</h2>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum ipsum lurum hurum turum ipsum
lurum hurum turum.</p>
</div>

</section>

<footer>
<p>&copy; 2014 Monday Times. All rights reserved.</p>
</footer>

</body>
</html>

```

A Typical HTML5 Page

Finally you can remove the `<head>` tags. They are not needed in HTML5:

```
<!DOCTYPE html>
<html lang="en">
<title>HTML</title>
<meta charset="utf-8">

<!--[if lt IE 9]>
<script src="http://html5shiv.googlecode.com/svn/trunk/html5.js">
</script>
<![endif]-->

<style>
body {
  font-family: Verdana,sans-serif;font-size:0.8em;
}
header,footer,section,article {
  border:1px solid grey;
  margin:5px;margin-bottom:15px;padding:8px;
  background-color:white;
}
header,footer {
  color:white;background-color:#444;margin-bottom:5px;
}
section {
  background-color:#ddd;
}
nav ul {
  margin:0;padding:0;
}
nav ul li {
  display:inline; margin:5px;
}
</style>

<body>

<header>
<h1>Monday Times</h1>
</header>

<nav>
<ul>
  <li>News</li>
  <li>Sports</li>
  <li>Weather</li>
</ul>
</nav>

<section>
<h2>News Section</h2>

<article>
```

```
<h2>News Article</h2>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum
ipsum lurum hurum turum ipsum lurum hurum turum.</p>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum
ipsum lurum hurum turum ipsum lurum hurum turum.</p>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum
ipsum lurum hurum turum ipsum lurum hurum turum.</p>
</article>
```

```
<article>
<h2>News Article</h2>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum
ipsum lurum hurum turum ipsum lurum hurum turum.</p>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum
ipsum lurum hurum turum ipsum lurum hurum turum.</p>
<p>Ipsum lurum hurum turum ipsum lurum hurum turum
ipsum lurum hurum turum ipsum lurum hurum turum.</p>
</article>
```

```
</section>
```

```
<footer>
<p>&copy; 2014 Monday Times. All rights reserved.</p>
</footer>
```

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

The Difference Between `<article>` `<section>` and `<div>`

There is a confusing (lack of) difference in the HTML5 standard, between `<article>` `<section>` and `<div>`.

In the HTML5 standard, the `<section>` element is defined as a block of related elements.

The `<article>` element is defined as a complete, self-contained block of related elements.

The `<div>` element is defined as a block of children elements.

How to interpret that?

In the example above, we have used `<section>` as a container for related `<articles>`.

But, we could have used `<article>` as a container for articles as well.

Here are some different examples:

```
<article> in <article>:
<article>
```

```
<h2>Famous Cities</h2>
```

```
<article>
```

```
<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>
```

```
</article>
```

```
<article>
```

```
<h2>Paris</h2>
```

```
<p>Paris is the capital and most populous city of France.</p>
```

```
</article>
```

```
<article>
```

```
<h2>Tokyo</h2>
```

```
<p>Tokyo is the capital of Japan, the center of the Greater Tokyo Area, and the most populous metropolitan area in the world.</p>
```

```
</article>
```

```
</article>
```

```
div> in <article>:
```

```
<article>
```

```
<h2>Famous Cities</h2>
```

```
<div class="city">
```

```
<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>
```

```
<div class="city">
```

```
<h2>Paris</h2>
```

```
<p>Paris is the capital and most populous city of France.</p>
```

```
</div>
```

```
<div class="city">
```

```
<h2>Tokyo</h2>
```

```
<p>Tokyo is the capital of Japan, the center of the Greater Tokyo Area, and the most populous metropolitan area in the world.</p>
```

```
</div>
```

```
</article>
```

```
div> in <section> in <article>:
```

```
<article>
```

```
<section>
```

```
<h2>Famous Cities</h2>
```

```
<div class="city">
```

```
<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>
```

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

```
<div class="city">
<h2>Tokyo</h2>
<p>Tokyo is the capital of Japan, the center of the Greater Tokyo Area,
and the most populous metropolitan area in the world.</p>
</div>
</section>
```

```
<section>
<h2>Famous Countries</h2>
```

```
<div class="country">
<h2>England</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>
```

```
<div class="country">
<h2>France</h2>
<p>Paris is the capital and most populous city of France.</p>
</div>
```

```
<div class="country">
<h2>Japan</h2>
<p>Tokyo is the capital of Japan, the center of the Greater Tokyo Area,
and the most populous metropolitan area in the world.</p>
</div>
</section>
```

```
</article>
```

Try it Yourself »

« Previous
Next Chapter »
Search w3schools.com:

Select Language ▼
WEB HOSTING
UK Reseller Hosting
WEB BUILDING
XML Editor - Free Trial! FREE Website BUILDER Free HTML5 Templates
W3SCHOOLS EXAMS
HTML, CSS, JavaScript, PHP, jQuery, XML, and ASP Certifications
SHARE THIS PAGE

facebook

WEB RESOURCES

Web Statistics Web Validation

HTML5 Syntax Style

Many developers are uncertain about the syntax style to use in HTML5.

Between 2000 and 2010, most web developers converted from HTML to XHTML.

XHTML syntax was easy. Developers were forced to write valid and "well-formed" code.

HTML5 is a bit more sloppy when it comes to code validation.

With HTML5, you have to create your own best practice syntax style.

Note Always keep your style smart, tidy, clean, and well-formed.

Make it Easy for Yourself

A consequent use of style, makes it easier for others to understand and use your HTML.

In the future, programs like XML readers, may want to read your HTML.

Using a well-formed "close to XHTML" syntax, can be very smart.

You can never be sure that different styles will produce the same result:

```
objects = getElementsByTagName("DIV")
```

```
objects = getElementsByTagName("div")
```

Lower Case Element Names

HTML5 allows mixing uppercase and lowercase letters in element names.

At W3Schools, we feel it's natural to use lowercase element names:

Mixing styles is not good

Developers are used to lowercase from XHTML

Lowercase look clean

Lowercase are easier to write

Looking bad:

```
<SECTION>
```

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

```
</SECTION>
```

Looking very bad:

```
<Section>
```

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

</SECTION>

Looking good:

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

Close HTML Elements

HTML5 allows unclosed HTML elements.

However, at W3Schools, we feel it more natural to always close HTML elements:

Looking bad:

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

Looking good:

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

Close Empty HTML Elements

HTML5 allows unclosed empty HTML elements.

This is allowed:

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

This is also allowed:

```
<meta charset="utf-8" />
```

The slash (/) closes the element from an XML readers point of view. It might be a good idea to keep it.

Lower Case Attribute Names

HTML5 allows mixing uppercase and lowercase letters in attribute names.

At W3Schools, we feel it's natural to use lowercase attribute names:

- Mixing styles is not good
- Developers are used to lowercase from XHTML
- Lowercase look clean
- Lowercase are easier to write

Looking bad:

```
<div CLASS="table table-striped table-bordered">
```

Looking good:

```
<div class="table table-striped table-bordered">
```

Quote Attribute Values

HTML5 allows unquoted attribute values. It saves space, and is easier to write.

However, at W3Schools, we feel it more natural to always quote attribute values:

- Mixing styles is not good
- Quoted values are easy to read
- Quotes allow multiple values

This will not work:

```
<div class=table table-striped table-bordered>
```

This will work:

```
<div class="table table-striped table-bordered">
```

HTML5 Form Elements

HTML5 has the following new form elements:

- <datalist>
- <keygen>
- <output>

HTML5 <datalist> Element

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

The <datalist> element is used to provide an "autocomplete" feature on <input> elements. Users will see a drop-down list of pre-defined options as they input data.

Use the <input> element's list attribute to bind it together with a <datalist> element.

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

```
<form action="demo_form.asp" method="get">
```

```
<input list="browsers" name="browser">
```

```
<datalist id="browsers">
```

```
<option value="Internet Explorer">
```

```
<option value="Firefox">
```

```
<option value="Chrome">
```

```
<option value="Opera">
```

```
<option value="Safari">
```

```
</datalist>
```

```
<input type="submit">
```

```
</form>
```


<p>Note: The datalist tag is not supported in Internet Explorer 9 and earlier versions, or in Safari.</p>

</body>
</html>

HTML5 <keygen> Element

The purpose of the <keygen> element is to provide a secure way to authenticate users.

The <keygen> tag specifies a key-pair generator field in a form.

When the form is submitted, two keys are generated, one private and one public.

The private key is stored locally, and the public key is sent to the server. The public key could be used to generate a client certificate to authenticate the user in the future.

OperaSafariChromeFirefoxInternet Explorer

Example

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

<form action="demo_keygen.asp" method="get">
Username: <input type="text" name="usr_name">
Encryption: <keygen name="security">
<input type="submit">
</form>

</body>
</html>
```

HTML5 <output> Element

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

OperaSafariChromeFirefoxInternet Explorer

Example

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

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

<form action="demo_form.asp" method="get"
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>
```

```
<input type="submit">
</form>
```

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

HTML5 New Form Elements

Tag Description

<datalist>	Specifies a list of pre-defined options for an <input> element
<keygen>	Specifies a key-pair generator field in a form
<output>	Represents the result of a calc

HTML5 Input Types

HTML5 New Input Types

HTML5 has several new input types for forms. These new features allow better input control and validation.

This chapter covers the new input types:

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

Input Type: number

The number type is used for input fields that should contain a numeric value.

You can set restrictions on the numbers.

Depending on browser support, the restrictions can apply to the input field.

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

```
<p>
```

Depending on browser support:

Numeric restrictions will apply in the input field.

```
</p>
```

```
<form action="demo_form.asp">
  Quantity (between 1 and 5):
  <input type="number" name="quantity" min="1" max="5">
  <input type="submit" value="Send">
</form>
```

<p>Note: type="number" is not supported in Internet Explorer 9 and earlier versions.</p>

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

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

= New in HTML5.

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

```
<p>
Depending on browser support:<br>
Numeric restrictions will apply in the input field.
</p>
```

```
<form action="demo_form.asp">
  Quantity:
  <input type="number" name="points"
  min="0" max="100" step="10" value="30">
  <input type="submit" value="Send">
</form>
```

<p>Note:
type="number" is not supported in Internet Explorer 9 and earlier.
</p>

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

Input Type: date

The date type is used for input fields that should contain a date.

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

Opera Safari Chrome Firefox Internet Explorer

Example

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

You can add restrictions to the input:

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

<form action="demo_form.asp">
  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>
  <input type="submit" value="Send">
</form>
```

```
<p><strong>Note:</strong>
type="date" is not supported in Internet Explorer 9 and earlier versions.</p>
```

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

The color type is used for input fields that should contain a color.

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

Opera Safari Chrome Firefox Internet Explorer

Example

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

Input Type: range

The range type is used for input fields that should contain a value within a range.

Depending on browser support, the input field can be displayed as a slider control.

Opera Safari Chrome Firefox Internet Explorer

Example

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

```
<p>
```

Depending on browser support:

The input type "range" can be displayed as a slider control.

</p>

```
<form action="demo_form.asp" method="get">
  Points:
  <input type="range" name="points" min="0" max="10">
  <input type="submit" value="Send">
</form>
```

<p>
Note:
type="range" is not supported in Internet Explorer 9 and earlier versions.
</p>

</body>
</html>

Input Type: month

The month type allows the user to select a month and year.

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

Example

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

Input Type: week

The week type allows the user to select a week and year.

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

Example

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

Input Type: time

The time type allows the user to select a time (no time zone).

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

Example

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

Input Type: datetime

The datetime type allows the user to select a date and time (with time zone).

Depending on browser support, a date picker can show up in the input field.
Opera Safari Chrome Firefox Internet Explorer

Example

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

Input Type: datetime-local

The datetime-local type allows the user to select a date and time (no time zone).

Depending on browser support, a date picker can show up in the input field.
Opera Safari Chrome Firefox Internet Explorer

Example

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

```
<form action="demo_form.asp">
  E-mail:
  <input type="email" name="email">
  <input type="submit" value="Send">
</form>
```

```
<p>
<b>Note:</b>
type="email" is not supported in Internet Explorer 9 and earlier.</p>
```

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

Input Type: search

The search type is used for search fields (a search field behaves like a regular text field).
OperaSafariChromeFirefoxInternet Explorer

Example

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

<form action="demo_form.asp">
  Search Google:
  <input type="search" name="googlesearch">
  <input type="submit" value="Send">
</form>
```

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

Input Type: tel

The tel type is used for input fields that should contain a telephone number.

The tel type is currently supported only in Safari 8.

OperaSafariChromeFirefoxInternet Explorer

Example

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<form action="demo_form.asp">
```

```
  Telephone:
```

```
  <input type="tel" name="usrtel">
```

```
  <input type="submit" value="Send">
```

```
</form>
```

```
<p><b>Note:</b> type="tel" is supported only in Safari 8.</p>
```

```
</body>
```

```
</html>
```

Input Type: url

The url type 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.

OperaSafariChromeFirefoxInternet Explorer

Example

```
<form>
```

```
  Add your homepage:
```

```
  <input type="url" name="homepage">
```

```
</form>
```

Html 5 canvas

Create a linear gradient. Fill rectangle with the gradient:

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<canvas id="myCanvas" width="200" height="100" style="border:1px solid #d3d3d3;">
```

```
Your browser does not support the HTML5 canvas tag.</canvas>
```

```
<script>
```

```
var c = document.getElementById("myCanvas");
```

```
var ctx = c.getContext("2d");
```

```
// Create gradient
var grd = ctx.createLinearGradient(0,0,200,0);
grd.addColorStop(0,"red");
grd.addColorStop(1,"white");
```

```
// Fill with gradient
ctx.fillStyle = grd;
ctx.fillRect(10,10,150,80);
```

```
</script>
```

```
</body>
```

```
</html>
```

Canvas - Images

To draw an image on a canvas, we will use the following method:

```
drawImage(image,x,y)
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<p>Image to use:</p>
```

```
<p>Canvas:</p>
```

```
<canvas id="myCanvas" width="250" height="300" style="border:1px solid #d3d3d3;">
Your browser does not support the HTML5 canvas tag.</canvas>
```

```
<script>
```

```
var c = document.getElementById("myCanvas");
```

```
var ctx = c.getContext("2d");
```

```
var img = document.getElementById("scream");
```

```
ctx.drawImage(img,10,10);
```

```
</script>
```

```
</body>
```

```
</html>
```

The HTML <video> Element

To show a video in HTML, use the <video> element:

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<video width="320" height="240" controls>
```

```
  <source src="movie.mp4" type="video/mp4">
```



```
<source src="movie.ogg" type="video/ogg">
Your browser does not support the video tag.
</video>
```

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

How it Works

The controls attribute adds video controls, like play, pause, and volume.

It is a good idea to always include width and height attributes.

If height and width are not set, the browser does not know the size of the video. The effect will be that the page will change (or flicker) while the video loads.

Text between the <video> and </video> tags will only display in browsers that do not support the <video> element.

Multiple <source> elements can link to different video files. The browser will use the first recognized format.

HTML <video> Autoplay

To start a video automatically use the autoplay attribute:

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

<video width="320" height="240" autoplay>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
  Your browser does not support the video tag.
</video>

</body>
</html>
```

HTML Video - Methods, Properties, and Events

HTML5 defines DOM methods, properties, and events for the <video> element.

This allows you to load, play, and pause videos, as well as setting duration and volume.

There are also DOM events that can notify you when a video begins to play, is paused, etc.
Example: Using JavaScript

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

```

<div style="text-align:center">
  <button onclick="playPause()">Play/Pause</button>
  <button onclick="makeBig()">Big</button>
  <button onclick="makeSmall()">Small</button>
  <button onclick="makeNormal()">Normal</button>
  <br>
  <video id="video1" width="420">
    <source src="mov_bbb.mp4" type="video/mp4">
    <source src="mov_bbb.ogv" type="video/ogg">
    Your browser does not support HTML5 video.
  </video>
</div>

```

```

<script>
var myVideo = document.getElementById("video1");

```

```

function playPause() {
  if (myVideo.paused)
    myVideo.play();
  else
    myVideo.pause();
}

```

```

function makeBig() {
  myVideo.width = 560;
}

```

```

function makeSmall() {
  myVideo.width = 320;
}

```

```

function makeNormal() {
  myVideo.width = 420;
}

```

```

</script>

```

```

<p>Video courtesy of <a href="http://www.bigbuckbunny.org/" target="_blank">Big Buck
Bunny</a>.</p>

```

```

</body>

```

```

</html>

```

HTML5 Video Tags

Tag	Description
<video>	Defines a video or movie
<source>	Defines multiple media resources for media elements, such as <video> and <audio>
<track>	Defines text tracks in media players

The HTML <audio> Element

To play an audio file in HTML, use the <audio> element:

```

<!DOCTYPE html>
<html>
<body>

<audio controls>
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
Your browser does not support the audio element.
</audio>

</body>
</html>

```

HTML Audio - How It Works

The controls attribute adds audio controls, like play, pause, and volume.

Text between the <audio> and </audio> tags will display in browsers that do not support the <audio> element.

Multiple <source> elements can link to different audio files. The browser will use the first recognized format.

HTML Audio - Browser Support

Currently, there are 3 supported file formats for the <audio> element: MP3, Wav, and Ogg:

Browser	MP3	Wav	Ogg
Internet Explorer	YES	NO	NO
Chrome	YES	YES	YES
Firefox	YES	YES	YES
Safari	YES	YES	NO
Opera	YES	YES	YES

HTML Audio - Media Types

File Format	Media Type
MP3	audio/mpeg
Ogg	audio/ogg
Wav	audio/wav

HTML Audio - Methods, Properties, and Events

HTML5 defines DOM methods, properties, and events for the <audio> element.

This allows you to load, play, and pause audios, as well as setting duration and volume.

There are also DOM events that can notify you when an audio begins to play, is paused, etc.

For a full DOM reference, go to our [HTML5 Audio/Video DOM Reference](#).

HTML5 Audio Tags

Tag	Description
<audio>	Defines sound content
<source>	Defines multiple media resources for media elements, such as <video> and <audio>

HTML Plug-ins

Helper applications are computer programs that extend the standard functionality of a web browser.

Helper applications are also called plug-ins.

Examples of well-known plug-ins are Java applets.

Plug-ins can be added to web pages with the `<object>` tag or the `<embed>` tag.

Plug-ins can be used for many purposes: display maps, scan for viruses, verify your bank id, etc.

Note To display video and audio: Use the `<video>` and `<audio>` tags.

The `<object>` Element

The `<object>` element is supported by all browsers.

The `<object>` element defines an embedded object within an HTML document.

It is used to embed plug-ins (like Java applets, PDF readers, Flash Players) in web pages.

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

<object width="100%" height="500px" data="snippet.html"></object>

</body>
</html>
```

Or images if you like:

Example

```
<object data="audi.jpeg"></object>
```

Note that the `<embed>` element does not have a closing tag. It can not contain alternative text.

Or images if you like:

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

<embed src="audi.jpeg">

</body>
</html>
```

Playing a YouTube Video in HTML

If you want to play a video in a web page, you can upload the video to YouTube and insert the proper HTML code, in your web page, to display the video:

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

```
<body>
```

```
<iframe width="420" height="345"  
src="http://www.youtube.com/embed/XGSy3_Czz8k">  
</iframe>
```

```
</body>
```

```
</html>
```

Example - Using <object>

```
<object width="420" height="315"  
data="http://www.youtube.com/v/XGSy3_Czz8k">  
</object>
```

Example - Using <embed>

```
<embed width="420" height="315"  
src="http://www.youtube.com/v/XGSy3\_Czz8k">
```