



INTRODUCTION TO HTML5

HyperText Markup Language V.5 (HTML5)

HTML5 is the latest evolution of the standard that defines HTML. The term represents two different concepts:

- It is a new version of the *language* HTML, with new elements, attributes, and behaviors,
- A larger set of **technologies** that allows more diverse and powerful Web sites and applications

Some rules for HTML5 were established:

- New features should be based on HTML, CSS, DOM, and JavaScript
- Reduce the need for external plugins (like Flash)
- Better error handling
- More markup to replace scripting

New HTML5 Elements

The most interesting new elements are:

- New **semantic** elements like <header>, <footer>, <article>, and <section>.
- New form **control attributes** like number, date, time, calendar, and range.
- New **graphic** elements: <svg> and <canvas>.
- New **multimedia** elements: <audio> and <video>.

HTML5 - New Attribute Syntax

HTML5 allows four different syntaxes for attributes:

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

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

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

New Semantic/Structural Elements

HTML5 offers new elements for better document structure:

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

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

New Input Types

New Input Types	New Input Attributes
<ul style="list-style-type: none"> • Color • date • datetime • datetime-local • email • month • number • range • search • tel • time • url • week 	<ul style="list-style-type: none"> • autocomplete • autofocus • form • formaction • formenctype • formmethod • formnovalidate • formtarget • height and width • list • min and max • multiple • pattern (regexp) • placeholder • required • step

HTML5 Graphics

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

New Media Elements

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

Declaring that the document contains HTML5 mark-up with the HTML5 doctype

The doctype for HTML5 is very simple. To indicate that your HTML content uses HTML5, simply use: `<!DOCTYPE html>`

Doing so will cause even browsers that don't presently support HTML5 to enter into standards mode, which means that they'll interpret the long-established parts of HTML in an HTML5-compliant way while ignoring the new features of HTML5 they don't support.

This is much simpler than the former doctypes, and shorter, making it easier to remember and reducing the amount of bytes that must be downloaded.

At the very top of the page you will see the doctype declaration:

`<!DOCTYPE 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:

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

Use Correct Document Type

Always declare the document type as the first line in your document:

`<!DOCTYPE html>`

If you want consistency with lowercase tags, you can use:

`<!doctype html>`

Use Lower Case Element Names

HTML5 allows mixing uppercase and lowercase letters in element names.

We recommend using lowercase element names:

- Mixing uppercase and lowercase names is bad
- Developers are used to use lowercase names (as in XHTML)
- Lowercase look cleaner
- Lowercase are easier to write

Bad:
<pre><SECTION> <p>This is a paragraph.</p> </SECTION></pre>
Very Bad:
<pre><Section> <p>This is a paragraph.</p> </SECTION></pre>
Good:
<pre><section> <p>This is a paragraph.</p> </section></pre>

Close All HTML Elements

In HTML5, you don't have to close all elements (for example the <p> element).

We recommend closing all HTML elements:

Looking bad:
<pre><section> <p>This is a paragraph. <p>This is a paragraph. </section></pre>
Looking good:
<pre><section> <p>This is a paragraph.</p> <p>This is a paragraph.</p> </section></pre>

Close Empty HTML Elements

In HTML5, it is optional to close empty elements.

This is allowed:
<pre><meta charset="utf-8"></pre>
This is also allowed:
<pre><meta charset="utf-8" /></pre>

The slash (/) is required in XHTML and XML.

If you expect XML software to access your page, it might be a good idea to keep it.

Use Lower Case Attribute Names

HTML5 allows mixing uppercase and lowercase letters in attribute names.

We recommend using lowercase attribute names:

- Mixing uppercase and lowercase names is bad
- Developers are used to use lowercase names (as in XHTML)
- Lowercase look cleaner
- Lowercase are easier to write

Looking bad:
<code><div CLASS="menu"></code>
Looking good:
<code><div class="menu"></code>

Quote Attribute Values

HTML5 allows attribute values without quotes.

We recommend quoting attribute values:

- You have to use quotes if the value contains spaces
- Mixing styles is never good
- Quoted values are easier to read

This will not work, because the value contains spaces:
<code><table class=table striped></code>
This will work:
<code><table class="table striped"></code>

Image Attributes

Always use the **alt** attribute with images. It is important when the image cannot be viewed.

```

```

Always define image size. It reduces flickering because the browser can reserve space for images before they are loaded.

```

```

Spaces and Equal Signs

Spaces around equal signs is legal:

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

But space-less is easier to read, and group's entities better together:

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

Omitting <html> and <body>

In the HTML5, the <html> tag and the <body> tag can be omitted.

The following code will validate as HTML5:

Example
<pre><!DOCTYPE html> <head> <title>Page Title</title> </head> <h1>This is a heading</h1> <p>This is a paragraph.</p></pre>

Note: We do not recommend omitting the <html> and <body> tags.

The <html> element is the document root. It is the recommended place for specifying the page language:

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

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

Omitting <html> or <body> can crash DOM and XML software.

Omitting <body> can produce errors in older browsers (IE9).

Omitting <head>

In the HTML5, the <head> tag can also be omitted.

By default, browsers will add all elements before <body>, to a default <head> element.

You can reduce the complexity of HTML, by omitting the <head> tag:

Example
<pre> <!DOCTYPE html> <html> <title>Page Title</title> <body> <h1>This is a heading</h1> <p>This is a paragraph.</p> </body> </html> </pre>

HTML Comments

Short comments should be written on one line, with a space after <!-- and a space before -->:

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

Long comments, spanning many lines, should be written with <!-- and --> on separate lines:

```

<!--
  This is a long comment example. This is a long comment example. This is a long
  comment example.
-->

```

Long comments are easier to observe, if they are indented 2 spaces.

HTML5: The <script> tag

It's common practice to add a type attribute with a value of "text/javascript" to script elements as follows

```
<script type="text/javascript" src="scriptfile.js"></script>
```

HTML 5 removes extra information required and you can use simply following syntax

```
<script src="scriptfile.js"></script>
```


HTML5: The <link> tag

So far you were writing <link> as follows –

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

HTML 5 removes extra information required and you can use simply following syntax

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

Notes:

- Place the opening bracket on the same line as the selector.
- Use one space before the opening bracket.
- Use 2 spaces of indentation.
- Use colon plus one space between each property and its value.
- Use space after each comma or semicolon.
- Use semicolon after each property-value pair, including the last.
- Only use quotes around values if the value contains spaces.
- Place the closing bracket on a new line, without leading spaces.
- Avoid lines over 80 characters.