

## HTML Elements:

Types of elements:

- **Void elements** - Have no closing tags, so have no text associated with them: **area, base, br, col, embed, hr, img, input, keygen, link, meta, param, source, track, wbr**  
May be "closed" with an ending /, i.e.: `<br />`, but is not necessary in HTML5.
- **Raw text elements** - can have text associated with them, with restrictions - may not contain it's closing tag in part or in whole.  
**script, style**
- **Escape-able raw text elements** - same as raw text elements + can have text and "character references", but must not include an ambiguous ampersand (&).  
**textarea, title**
- **Foreign elements** - e.g.: Elements from the MathML and SVG name-spaces.
- **Normal Elements** - All other HTML elements. Can have text, "character references" and comments, but text may not contain the character '<' or an ambiguous ampersand. Some normal elements may have additional restrictions.

## Global Attributes:

- The following attributes may be placed on any tag. Enumerated attribute are Booleans, i.e.: (' / 'true' or 'false' or sometimes "yes" / "no")

<b>class</b>	The list of space separated classes that the element belongs to. Mostly used to specify the CSS properties of an element, or for use with the <b>getElementsByClassName()</b> method.
<b>data-*</b>	Custom data attributes used to store custom data associated with the element. Accessible via the DOM <code>&lt;element&gt;.dataset</code> object. e.g: <b>data-info='xyz'</b> → <code>element.dataset.info = 'xyz'</code> <b>data-foo-bar</b> → <code>element.dataset.fooBar</code> (hyphens → CamelCase)
<b>hidden</b>	(enumerated), when present hides the element and all children from being displayed.
<b>id</b>	Items unique identifier, must be unique amongst all IDs. Must be at least one character, with no spaces.
<b>style</b>	Sets the CSS styling attributes for the element.
<b>title</b>	Advisory information for the target element. Meaning varies depending on target. Usually used for mouse-over tool-tips. <i>&lt;&lt; Not advisable to always use, since often requires a pointer to reveal (so keyboard only or touch only cannot access this resource. &gt;&gt;</i>

## Document meta-data:

- Usually inside of a `<head>` element.

<b>&lt;title&gt;</b>	Sets the documents title or name. Must be in the head element and only one title element per document.
<b>&lt;link&gt;</b>	Allows one to link the document to other resources. Often used for linking to style-sheets ex: <code>&lt;link rel=stylesheet href='styles.css'&gt;</code> Attributes: <b>href, crossorigin, rel, media, hreflang, type, sizes</b>

**<meta>** Allows one to set additional meta data on the document that cannot be expressed using <title>, <base>, <link>, <style> and <script>  
 Attributes: **name, http-equiv, content, charset.**  
 ex: `<meta charset='utf-8'>`

**<style>** Embeds CSS style information in the document.  
 Attributes:  
**media, type, scoped**  
 ex: `<style>`  
     `/* C style comments, not HTML comments */`  
     `body { color: black; background: white; }`  
`</style>`

## Grouping Content:

**<p>** Represents a paragraph.

**<hr>** (void) Creates a horizontal rule across it's containing element.

**<pre>** Represents a block of preformatted text. Text is displayed as is, ignoring any HTML elements. Useful for displaying source code.

**<ol>** Ordered (numbered) lists

**<ul>** Unordered (bullet-ted) lists

**<li>** List element - Doesn't need to be closed, next <li> auto-closes current element. ex:  
`<ol>`  
   `<li> Item 1 </li>`  
   `<li> Item 2`  
   `<li> Item 3`  
`</ol>`

**<div>** Groups block-elements, Has no special meaning, represents its children.

## Text Level Semantics:

**<a>** Represents a hyperlink *if* it contains an **href** attribute, otherwise a placeholder for a link.  
 Attributes:

- href** A valid URL.
- target** Where to open the link, one of:
  - `_blank` New window or tab
  - `_self` (default) Same frame as the link
  - `_parent` Parent frame
  - `_top` Full body of the window.
  - `named` In the *named* frame.
- download** Boolean indicating the link is for downloading a resource.
- rel** type of link, a set of space separated tokens.
- type** the MIME-type of the linked resource.

ex: `<a href='relative/url.html'>` Relative to source page `</a>`  
`<a href='/full/path.html'>` Starts at base dir on same server `</a>`  
`<a href='http://full.com/path'>` Full URL `</a>`  
`<a href='#xyz'>` Moves to element with id 'xyz' within document `</a>`

**<sub>** Subscript text.

**<sup>** Superscript text.

**<i>** *Italicized* text.

**<b>** **Bold** text.

**<u>** Underlines text. May be confused for a hyperlink.

**<span>** Groups in-line objects, otherwise no special meaning, like **<div>**

## Tabular Data:

<b>&lt;table&gt;</b>	Represents a table
<b>&lt;tbody&gt;</b>	The tables main body (can have more than one), auto-created if none specified.
<b>&lt;thead&gt;</b>	A tables heading section.
<b>&lt;tfoot&gt;</b>	A tables footer section.
<b>&lt;tr&gt;</b>	A row of cells (columns) in the table.
<b>&lt;th&gt;</b>	Represents a header cell in a table (usually bold/centered.)
<b>&lt;td&gt;</b>	Represents a data cell in a table.
Attributes:	
<b>colspan</b>	The number of columns this cell spans.
<b>rowspan</b>	The number of rows this cell spans.
Deprecated in HTML5 (use CSS text-align/vertical-align):	
<b>align</b>	Align text horizontally (left/right/center/justify/char)
<b>valign</b>	Align text vertically (top/middle/bottom/baseline)

Table example:

A	B	C
1	X	Y
	Z	
2		3

```
<table border=1>
  <tbody>
    <tr>
      <th> A <th> B <th> C
    <tr>
      <td rowspan=2> 1 <td> X <td> Y
    <tr>
      <td colspan=2> Z
    <tr>
      <td colspan=2> 1 <td> 3
    </tbody>
  </table>
```

## Forms:

<b>&lt;form&gt;</b>	A collection of form-associated elements, some of which represent editable values that can be submitted to a server for processing.
Common Attributes:	
<b>name</b>	Represents the forms name within the forms collection.
<b>autocomplete</b>	'on'/default = indicates from controls in the form will have their autofill field name set to 'on', 'off' = off.
<b>action</b>	a valid URL specifying the submission target
<b>method</b>	'GET' or 'POST'
<b>enctype</b>	'application/x-www-form-urlencoded'/'multipart/form-data' / 'text/plain' encoding types.

<b>&lt;input&gt;</b>	Represents a typed data field, useually with a form control to allow the user to edit the data.
Attributes:	
<b>maxlength</b>	The maximum input length.
<b>size</b>	sizes the element based on number of characters (default 20)
<b>readonly</b>	Boolean controlling whether the element can be edited
<b>required</b>	Boolean specifying the element is required
<b>multiple</b>	Boolean indicating the user may specify more than one value (email / files)

**pattern** A regular expression against which the value is checked.  
**min/max** allowed range of values for the element.  
**step** granularity of the value that is expected.  
**list** used to identify an element that lists predefined options to the user, must be the ID of a datalist element.  
**name** form elements name.  
**value** default value for the form element.  
**checked** Boolean setting checkboxes/radio buttons as being 'checked'  
**placeholder** a short hint to aid data entry when the control has no value.

### INPUT types:

\* = remember these

**\*hidden** An arbitrary string that is hidden.  
**\*text** Text with no line breaks (default type).  
**\*password** (password) Text with no line breaks (sensitive information)  
**email** An e-mail address or list of email addresses  
**number** A numerical value.  
**range** A numerical value where the exact value is unimportant.  
**color** A sRGB color with 8bit color (a color well)  
**\*checkbox** A set or zero or more values from a predefined list (checkbox)  
**\*radio** An enumerated value (radio button)  
**\*file** Zero or more files each with a MIME type and optionally a files name (a label and a button)  
**\*submit** Initiates the form submission.  
**\*reset** Resets all the form elements to their default values.  
**\*button** n/a

**<button>** Represents a button labeled by its contents.  
**type** attribute:  
**submit** submits form,  
**reset** reset form,  
**button** does nothing

**<select>** Represents a control for selecting amongst a set of options.  
 Attributes:  
**multiple** Boolean controls for selecting zero or more options.  
**size** gives the number of option to display  
**required** Boolean indicating that a selection is required  
**autofocus, disabled, form, name**

**<option>** Represents an option in a select or as part of a list of suggestions in a datalist.  
 Attributes: **disabled, label, selected, value**

**<textarea>** Represents a multi-line plain text edit control for the elements 'raw value'. The contents of the control represent the controls default value.  
 Attributes:  
**cols** the expected maximum number of characters per line.  
**rows** The number of lines to show.  
**wrap** 'soft' = text is not to be wrapped when submitted but may be wrapped when viewed. 'hard' = text is wrapped when submitted (cols must be specified)  
**maxlength** maximum allowed length  
**autocomplete, autofocus, dirname, disabled, form, name, placeholder, readonly, required**

**<progress>** Represents the completion progress of a task.

Attributes: **value, max**

**<meter>** Represents a scalar measurement within a known range, or a fractional value, e.g. disk usage. A guage.

Attributes: **value, min, max, low, high, optimum**

## **Embedded Content:**

**<img>** Represents an image.

Attributes:

**alt** A text description of the image for those who cannot see it

**src** A valid URL of the image location, must be present

**crossorigin** A CORS setting attribute to allow images from third party sites that allow cross-origin access to be used with `<canvas>`

**usemap** If present the image has an associated image map

**ismap** Boolean indicating the image has an ancestor `<a>` with an href attribute.

**width, height** Width and height of the image in CSS pixels (96 ppi)

**<iframe>** Represents a nested browsing context.

Attributes: **src, srcdoc, name, sandbox, seamless, width, height**

**<embed>** Allows integrating external (non-HTML, e.g. flash) resources or interactive content.

Attributes: **src, type, width, height**

ex: `<embed src='vid.swf'>`

**<video>** Represents a video, or audio file with captions to be played.

Attributes:

**src, crossorigin, poster, preload, autoplay, mediagroup, loop, muted, controls, width, height**

ex: `<video src="vid.mp4" autoplay controls></video>`

**<audio>** Represents a sound or audio stream

Attributes:

**src, crossorigin, preload, autoplay, mediagroup, loop, muted, controls**

**<source>** Specified multiple alternative resources to media elements. It does not represent anything by itself.

Attributes: **src, type, media**

ex:

```
<video controls autoplay>
```

```
<source src='video.mp4' type='video/mp4; codecs="avc1.42E01E, mp4a.40.2"'>
```

```
<source src='video.ogv' type='video/ogg; codecs="theora, vorbis"'>
```

```
</video>
```

**<canvas>** Provides a resolution-dependent bitmap canvas which can be used for rendering graphs, game graphics, art or other images on the fly.

Attributes: **width, height**

## Interactive Elements:

**<dialog>** Represents a part of an application that a user interacts with to perform a task, e.g. a dialog box, inspector or window.  
Attributes:  
    **open** Boolean indicating the dialog element is active and the user can interact with it.

## Scripting:

**<script>** Includes scripting and data source into the document.  
Attributes:

**src, type, charset, async, defer, crossorigin**

ex:

```
<script src="script.js" type="text/javascript"></script>
<script type="text/x-data">
  .. Some data ...
</script>
```

**<noscript>** The children of this element are used if scripting is disabled. In a **<head>** element, this element may contain: **<link>**, **<style>** and **<meta>** elements. May contain any valid text outside a head element.