

# HTML CHEAT SHEET

HTML (Hypertext Markup Language) has come a long way since Tim Berners-Lee invented it back in 1991. Today HTML5 is the standard version and it's supported by all modern web browsers. Our **HTML cheat sheet** gives you a full list of all the HTML elements, including descriptions, code examples and live previews. Simply scroll down to browse all HTML tags [alphabetically](#) or browse tags by their [category](#).

<a href="#">a element</a>	<a href="#">details element</a>	<a href="#">legend element</a>	<a href="#">section element</a>
<a href="#">abbr element</a>	<a href="#">dfn element</a>	<a href="#">li element</a>	<a href="#">select element</a>
<a href="#">address element</a>	<a href="#">dialog element</a>	<a href="#">link element</a>	<a href="#">small element</a>
<a href="#">area element</a>	<a href="#">div element</a>	<a href="#">main element</a>	<a href="#">source element</a>
<a href="#">article element</a>	<a href="#">dl element</a>	<a href="#">map element</a>	<a href="#">span element</a>
<a href="#">aside element</a>	<a href="#">dt element</a>	<a href="#">mark element</a>	<a href="#">strong element</a>
<a href="#">audio element</a>	<a href="#">em element</a>	<a href="#">menu element</a>	<a href="#">style element</a>
<a href="#">b element</a>	<a href="#">embed element</a>	<a href="#">meta element</a>	<a href="#">sub element</a>
<a href="#">base element</a>	<a href="#">fieldset element</a>	<a href="#">meter element</a>	<a href="#">summary element</a>
<a href="#">bdi element</a>	<a href="#">figcaption element</a>	<a href="#">noscript element</a>	<a href="#">sup element</a>
<a href="#">bdo element</a>	<a href="#">figure element</a>	<a href="#">object element</a>	<a href="#">table element</a>
<a href="#">blockquote element</a>	<a href="#">footer element</a>	<a href="#">ol element</a>	<a href="#">tbody element</a>
<a href="#">br element</a>	<a href="#">form element</a>	<a href="#">optgroup element</a>	<a href="#">td element</a>
<a href="#">button element</a>	<a href="#">h1 to h6 element</a>	<a href="#">option element</a>	<a href="#">textarea element</a>
<a href="#">canvas element</a>	<a href="#">head element</a>	<a href="#">output element</a>	<a href="#">tfoot element</a>
<a href="#">caption element</a>	<a href="#">header element</a>	<a href="#">p element</a>	<a href="#">th element</a>
<a href="#">cite element</a>	<a href="#">hgroup element</a>	<a href="#">param element</a>	<a href="#">thead element</a>
<a href="#">code element</a>	<a href="#">hr element</a>	<a href="#">pre element</a>	<a href="#">time element</a>
<a href="#">col element</a>	<a href="#">html element</a>	<a href="#">progress element</a>	<a href="#">title element</a>
<a href="#">colgroup element</a>	<a href="#">i element</a>	<a href="#">q element</a>	<a href="#">tr element</a>
<a href="#">data element</a>	<a href="#">iframe element</a>	<a href="#">rp element</a>	<a href="#">track element</a>
<a href="#">datalist element</a>	<a href="#">img element</a>	<a href="#">rt element</a>	<a href="#">ul element</a>
<a href="#">dd element</a>	<a href="#">input element</a>	<a href="#">ruby element</a>	<a href="#">var element</a>
<a href="#">del element</a>	<a href="#">ins element</a>	<a href="#">s element</a>	<a href="#">video element</a>
	<a href="#">kbd element</a>	<a href="#">samp element</a>	<a href="#">wbr element</a>
	<a href="#">label element</a>	<a href="#">script element</a>	

# LIST OF HTML ELEMENTS BY CATEGORY

An HTML element (or tag) is an individual component of an HTML document. Here below is a full list of HTML elements listed by category.

<a href="#">Basic HTML Elements Metadata HTML Elements Content</a>	<a href="#">Content Grouping HTML Elements Text Level Semantic HTML Elements</a>	<a href="#">Table HTML Elements Form HTML Elements Content Embedding HTML Elements</a>	<a href="#">Images &amp; Multimedia HTML Elements Global HTML Attributes New Tags in HTML5</a>
--	--	--	--

## HTML TAG

Specifies an html document. The HTML `<html>` element (or HTML root element) represents the root of an HTML document. All other elements must be descendants of this element. Since the `<html>` element is the first in a document other than comments, it is called the root element. Although this tag can be implied, or not required, with HTML, it is required to be opened and closed in XHTML.

**Attributes** (modifiers)

xmlns + [global attributes](#)

### Code example

```
<!DOCTYPE html>
<html>
  <head>...</head>
  <body>...</body>
</html>
```

[⤴ Back to categories](#)

## BASE TAG

Specifies URL which non-absolute URLs are relative to. The HTML `<base>` element specifies the base URL to use for all relative URLs contained within a document. There can be only one `<base>` element in a document.

**Attributes** (modifiers)

`href` | `target` (`_self` | `_blank` | `_parent` | `_top`) + [global attributes](#)

### Code example

```
<base  
href="http://www.DIGITAL.COM/page.html"  
>
```

## HEAD TAG

First element of the HTML document. Collection of metadata for the Document. The HTML `<head>` element provides general information (metadata) about the document, including its title and links to its scripts and style sheets.

**Attributes** (modifiers)

[Global attributes](#)

### Code example

```
<html>  
  <head>  
    <title>Document title</title>  
  </head>  
</html>
```

## LINK TAG

Other resources related to the document. The HTML `<link>` element specifies relationships between the current document and an external resource. Possible uses for this element include defining a relational framework for navigation. This Element is most used to link to style sheets.

**Attributes** (modifiers)

href | rel | media | hreflang | type | sizes | crossorigin | integrity + [global attributes](#)

**Code example**

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

## META TAG

Document metadata that can't be expressed with other elements. The HTML `<meta>` element represents any metadata information that cannot be represented by one of the other HTML meta-related elements (`<base>`, `<link>`, `<script>`, `<style>` or `<title>`). Depending on the attributes set, the kind of metadata can be one of the following: If `name` is set, it is document-level metadata, applying to the whole page. If `http-equiv` is set, it is a pragma directive, i.e. information normally given by the web server about how the web page should be served. If `charset` is set, it is a charset declaration, i.e. the charset used for the serialized form of the webpage. If `itemprop` is set, it is user-defined metadata, transparent for the user-agent as the semantics of the metadata is user-specific.

**Attributes** (modifiers)

charset | content | http-equiv | name + [global attributes](#)

**Code example**

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

## STYLE TAG

Embed style information in the documents. The HTML `<style>` element contains style information for a document, or part of a document. By default, the style instructions written inside that element are expected to be CSS.

**Attributes** (modifiers)

media | type | title + [global attributes](#)

**Code example**

```
<style type="text/css">
body {
  color:red;
}
</style>
```

## TITLE TAG

Document title or name. The HTML `<title>` element defines the title of the document, shown in a browser's title bar or on the page's tab. It can only contain text, and any contained tags are ignored.

**Attributes** (modifiers)

[Global attributes](#)

### Code example

```
<title>This is the page title</title>
```

[↗ Back to categories](#)

## ADDRESS TAG

The HTML `<address>` element supplies contact information for its nearest `<article>` or `<body>` ancestor; in the latter case, it applies to the whole document.

**Attributes** (modifiers)

[Global attributes](#)

### Code example

```
<address>Review Squirrel<br>99 Elizabeth
Street<br> Sydney<br> Australia </address>
```

## ARTICLE TAG

Section of the page content, such as a blog or forum post. The HTML `<article>` element represents a self-contained composition in a document, page, application, or site, which is intended to be independently distributable or reusable (e.g., in syndication). This could be a forum post, a magazine or newspaper article, a blog entry, an object, or any other independent item of content. Each `<article>` should be identified, typically by including a heading (`<h1>`-`<h6>` element) as a child of the `<article>` element.

**Attributes** (modifiers)

[Global attributes](#)

## Code example

```
<article class="ureview">
  <p>I love this tool.</p>
  <footer>
    <p>
      Posted on <time datetime="2016-05-16
19:00">May 16</time> by Matt.
    </p>
  </footer>
</article>
```

## ASIDE TAG

Content related to surrounding elements that doesn't belong inline, such as a advertising or quotes. The HTML `<aside>` element represents a section of the page with content connected tangentially to the rest, which could be considered separate from that content. These sections are often represented as sidebars or inserts. They often contain the definitions on the sidebars, such as definitions from the glossary; there may also be other types of information, such as related advertisements; the biography of the author; web applications; profile information or related links on the blog.

**Attributes** (modifiers)

[Global attributes](#)

## Code example

```
<article>
  <p>
    The Disney movie <em>The Little
Mermaid</em> was
```

```
    first released to theatres in 1989.
</p>
<aside>
    The movie earned $87 million during its
initial release.
</aside>
<p>
    More info about the movie...
</p>
</article>
```

## BODY TAG

Main content of the document. The HTML `<body>` Element represents the content of an HTML document. There can be only one `<body>` element in a document.

### Attributes (modifiers)

onafterprint | onbeforeprint | onbeforeunload | onblur | onerror |  
onfocus | onhashchange | onlanguagechange | onload | onmessage |  
onoffline | ononline | onpopstate | onredo | onresize | onstorage |  
onundo | onunload + [global attributes](#)

### Code example

```
<html>
<head>
  <title>Here goes the title of the
document</title>
</head>
<body>
  Here goes the he content of the
document.....
</body>
</html>
```

## FOOTER TAG

Footer of the current section. The HTML `<footer>` element represents a footer for its nearest sectioning content or sectioning root element. A footer typically contains information about the author of the section, copyright data or links to related documents.

**Attributes** (modifiers)

[Global attributes](#)

## Code example

```
<footer>Some copyright info goes here</footer>
```

## H1 TO H6 TAG

Heading for the current section. Heading elements implement six levels of document headings, <h1> is the most important and <h6> is the least. A heading element briefly describes the topic of the section it introduces. Heading information may be used by user agents, for example, to construct a table of contents for a document automatically.

**Attributes** (modifiers)

[Global attributes](#)

## Code example

```
<h1>Heading level 1</h1>
<h2>Heading level 2</h2>
<h3>Heading level 3</h3>
<h4>Heading level 4</h4>
<h5>Heading level 5</h5>
<h6>Heading level 6</h6>
```

## HEADER TAG

Navigation or introductory elements for the current section. The HTML <header> element represents a group of introductory or navigational aids. It may contain some heading elements but also other elements like a logo, wrapped section's header, a search form, and so on.

**Attributes** (modifiers)

[Global attributes](#)

## Code example

```
<header>Logo</header>
```



## HGROUP TAG

The HTML `<hgroup>` Element (HTML Headings Group Element) represents the heading of a section. It defines a single title that participates in the outline of the document as the heading of the implicit or explicit section that it belongs to. Its text for the outline algorithm is the text of the first HTML Heading Element of highest rank (i.e., the first `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>` or `<h6>` with the smallest number among its descendants) and the rank is the rank of this very same HTML Heading Element. Therefore this element groups several headings, contributing only the main one to the outline of the document. It allows associating secondary titles, like subheadings, alternative titles, or even taglines, with the main heading, without polluting the outline of the document.

**Attributes** (modifiers)

[Global attributes](#)

### Code example

```
<hgroup>
  <h1>Main title</h1>
  <h2>Secondary title</h2>
</hgroup>
```

## NAV TAG

A section of a page that links to other pages. The HTML `<nav>` element (HTML Navigation Element) represents a section of a page that links to other pages or to parts within the page: a section with navigation links.

**Attributes** (modifiers)

[Global attributes](#)

### Code example

```
<nav>
  <ul>
    <li><a href="#">Home</a></li>
    <li><a href="#">About us</a></li>
    <li><a href="#">Contact us</a></li>
  </ul>
</nav>
```

## SECTION TAG

Contains of elements grouped by theme, for example a chapter or tab box. The HTML `<section>` element represents a generic section of a document, i.e., a thematic grouping of content, typically with a heading. Each `<section>` should be identified, typically by including a heading (`<h1>-<h6>` element) as a child of the `<section>` element.

**Attributes** (modifiers)

[Global attributes](#)

### Code example

```
<section><h1>Heading</h1><p>Bunch of awesome  
content</p></section>
```

## BLOCKQUOTE TAG

Quote from another source. The HTML `<blockquote>` Element (or HTML Block Quotation Element) indicates that the enclosed text is an extended quotation. Usually, this is rendered visually by indentation (see Notes for how to change it). A URL for the source of the quotation may be given using the `cite` attribute, while a text representation of the source can be given using the `<cite>` element.

**Attributes** (modifiers)

`cite` + [global attributes](#)

### Code example

```
<blockquote cite="https://DIGITAL.COM">  
<p>This is a quote taken from Review Squirrel.  
</p></blockquote>
```

[⤴ Back to categories](#)

## BR TAG

Line break. The HTML element line break `<br>` produces a line break in text (carriage-return). It is useful for writing a poem or an address, where the division of lines is significant. Do not use `<br>` to increase the gap between lines of text; use the CSS margin property or the `<p>` element.

**Attributes** (modifiers)

[Global attributes](#)

## Code example

```
<p>Review Squirrel<br>99 Elizabeth  
Street<br>Australia</p>
```

## DD TAG

Description, definition, or value, part of a term- description group in a description list. The HTML `<dd>` element (HTML Description Element) indicates the description of a term in a description list (`<dl>`) element. This element can occur only as a child element of a description list and it must follow a `<dt>` element.

**Attributes** (modifiers)

[Global attributes](#)

## Code example

```
<dl>  
  <dt>Review Squirrel</dt>  
  <dd>Helps you find the best tools for  
  running a small business website</dd>  
</dl>
```

## DIV TAG

Container or section with no semantic meaning. The HTML `<div>` element (or HTML Document Division Element) is the generic container for flow content, which does not inherently represent anything. It can be used to group elements for styling purposes (using the class or id attributes), or because they share attribute values, such as lang. It should be used only when no other semantic element (such as `<article>` or `<nav>`) is appropriate.

**Attributes** (modifiers)

[Global attributes](#)

## Code example

```
<div><p>Any kind of content here. Such as <p>,
<table>. You name it!</p></div>
```

## DL TAG

An association list consisting of zero or more name-value groups (a description list). The HTML `<dl>` element (or HTML Description List Element) encloses a list of pairs of terms and descriptions. Common uses for this element are to implement a glossary or to display metadata (a list of key-value pairs). Prior to HTML5, `<dl>` was known as a Definition List.

**Attributes** (modifiers)

[Global attributes](#)

## Code example

```
<dl>
  <dt>Review Squirrel</dt>
  <dd>Helps you find the best tools for
running a small business website</dd>
</dl>
```

## DT TAG

Term, or name, part of a term-description group in a description list. The HTML `<dt>` element (or HTML Definition Term Element) identifies a term in a definition list. This element can occur only as a child element of a `<dl>`. It is usually followed by a `<dd>` element; however, multiple `<dt>` elements in a row indicate several terms that are all defined by the immediate next `<dd>` element.

**Attributes** (modifiers)

[Global attributes](#)

## Code example

```
<dl>
  <dt>Review Squirrel</dt>
  <dd>Helps you find the best tools for
running a small business website</dd>
</dl>
```

## FIGCAPTION TAG

Caption or legend for the figure element. The HTML `<figcaption>` element represents a caption or a legend associated with a figure or an illustration described by the rest of the data of the `<figure>` element which is its immediate ancestor which means `<figcaption>` can be the first or last element inside a `<figure>` block. Also, the HTML Figcaption Element is optional; if not provided, then the parent figure element will have no caption.

**Attributes** (modifiers)

[Global attributes](#)

### Code example

```
<figure>
  
  <figcaption>A picture</figcaption>
</figure>
```

## FIGURE TAG

Contains elements related to single concept, such as an illustration or code example. The HTML `<figure>` element represents self-contained content, frequently with a caption (`<figcaption>`), and is typically referenced as a single unit. While it is related to the main flow, its position is independent of the main flow. Usually this is an image, an illustration, a diagram, a code snippet, or a schema that is referenced in the main text, but that can be moved to another page or to an appendix without affecting the main flow.

**Attributes** (modifiers)

[Global attributes](#)

### Code example

```
<figure>
  
  <figcaption>A picture</figcaption>
</figure>
```

## HR TAG

Paragraph-level thematic break. The HTML `<hr>` element represents a thematic break between paragraph-level elements (for example, a change of scene in a story, or a shift of topic with a section). In previous versions of HTML, it represented a horizontal rule. It may still be displayed as a horizontal rule in visual browsers, but is now defined in semantic terms, rather than presentational terms.

**Attributes** (modifiers)

[Global attributes](#)

### Code example

```
<p>This is the first paragraph of text.</p>
<hr><p>This is second paragraph of text.</p>
```

## LI TAG

List item. The HTML `<li>` element (or HTML List Item Element) is used to represent an item in a list. It must be contained in a parent element: an ordered list (`<ol>`), an unordered list (`<ul>`), or a menu (`<menu>`). In menus and unordered lists, list items are usually displayed using bullet points. In ordered lists, they are usually displayed with an ascending counter on the left, such as a number or letter.

**Attributes** (modifiers)

value + [global attributes](#)

### Code example

```
<ol>
  <li>first item</li>
  <li>second item</li>
```

```
<li>third item</li>
</ol>
```

## MAIN TAG

Specifies the main content area of an HTML document. The HTML `<main>` element represents the main content of the `<body>` of a document or application. The main content area consists of content that is directly related to, or expands upon the central topic of a document or the central functionality of an application. This content should be unique to the document, excluding any content that is repeated across a set of documents such as sidebars, navigation links, copyright information, site logos, and search forms (unless the document's main function is as a search form).

**Attributes** (modifiers)

[Global attributes](#)

### Code example

```
<main>
  <h1>Apples</h1>
  <p>The apple is the pomaceous fruit of the
apple tree.</p>
  <article>
    <p>The apple is the pomaceous fruit of the
apple tree.</p>
  </article>
</main>
```

## OL TAG

Ordered list. The HTML `<ol>` Element (or HTML Ordered List Element) represents an ordered list of items. Typically, ordered-list items are displayed with a preceding numbering, which can be of any form, like numerals, letters or Romans numerals or even simple bullets. This numbered style is not defined in the HTML description of the page, but in its associated CSS, using the `list-style-type` property. There is no limitation to the depth and overlap of lists defined with the `<ol>` and `<ul>` elements.

**Attributes** (modifiers)

start | reversed | type + [global attributes](#)

## Code example

```
<ol>
  <li>first item</li>
  <li>second item</li>
  <li>third item</li>
</ol>
```

## P TAG

Paragraph content. The HTML `<p>` element (or HTML Paragraph Element) represents a paragraph of text. Paragraphs are usually represented in visual media as blocks of text that are separated from adjacent blocks by vertical blank space and/or first-line indentation. Paragraphs are block-level elements.

**Attributes** (modifiers)

[Global attributes](#)

## Code example

```
<p>This is the first paragraph of text.</p>
<p>This is second paragraph of text.</p>
```

## PRE TAG

A block of preformatted text. The HTML `<pre>` element (or HTML Preformatted Text) represents preformatted text. Text within this element is typically displayed in a non-proportional ("monospace") font exactly as it is laid out in the file. Whitespace inside this element is displayed as typed.

**Attributes** (modifiers)

[Global attributes](#)

## Code example

```
<pre>
body {
  background-color:red;
}
</pre>
```



## UL TAG

Unordered list. The HTML `<ul>` element (or HTML Unordered List Element) represents an unordered list of items, namely a collection of items that do not have a numerical ordering, and their order in the list is meaningless. Typically, unordered-list items are displayed with a bullet, which can be of several forms, like a dot, a circle or a squared. The bullet style is not defined in the HTML description of the page, but in its associated CSS, using the `list-style-type` property.

**Attributes** (modifiers)

[Global attributes](#)

### Code example

```
<ul>
  <li>first item</li>
  <li>second item</li>
  <li>third item</li>
</ul>
```

[⤴ Back to categories](#)

## A TAG

Hyperlink (a hypertext anchor). The HTML Anchor Element (`<a>` tag) defines a hyperlink to a location on the same page or any other page on the Web. It can also be used (in an obsolete way) to create an anchor point - a destination for hyperlinks within the content of a page, so that links aren't limited to connecting simply to the top of a page.

**Attributes** (modifiers)

`href` | `hreflang` | `media` | `rel` | `target` (`_self` | `_blank` | `_parent` | `_top`) | `type` | `download` | `ping` | `referrerpolicy` + [global attributes](#)

### Code example

```
<a href="https://DIGITAL.COM">Review  
Squirrel</a>
```

## ABBR TAG

Abbreviation or acronym. The `<abbr>` element (or HTML Abbreviation Element) represents an abbreviation and optionally provides a full description for it. If present, the title attribute must contain this full description and nothing else.

**Attributes** (modifiers)

[Global attributes](#)

### Code example

```
<p>This is <abbr title="Hypertext Markup Language">HTML</abbr></p>
```

## B TAG

Stylistically separated text of equal importance, such as a product name. The HTML `<b>` Element represents a span of text stylistically different from normal text, without conveying any special importance or relevance. It is typically used for keywords in a summary, product names in a review, or other spans of text whose typical presentation would be boldfaced. Another example of its use is to mark the lead sentence of each paragraph of an article.

**Attributes** (modifiers)

[Global attributes](#)

### Code example

```
<p>This article describes several <b>text-level</b> elements. It explains their usage in an <b>HTML</b> document. </p>
```

## BDI TAG

Defines directional formatting for content. The HTML `<bdi>` Element (or Bi-Directional Isolation Element) isolates a span of text that might be formatted in a different direction from other text outside it.

**Attributes** (modifiers)

`dir` + [global attributes](#)

## Code example

```
<p dir="ltr">This arabic word  
<bdi>ARABIC_PLACEHOLDER</bdi> is automatically  
displayed right-to-left.</p>
```

## BDO TAG

Defines directional formatting for content. The HTML `<bdo>` Element (or HTML bidirectional override element) is used to override the current directionality of text. It causes the directionality of the characters to be ignored in favor of the specified directionality.

**Attributes** (modifiers)

dir + [global attributes](#)

## Code example

```
<p><bdo dir="rtl">This text will go right to  
left.</bdo></p>
```

## CITE TAG

Title of a referenced piece of work. The HTML Citation Element (`<cite>`) represents a reference to a creative work. It must include the title of a work or a URL reference, which may be in an abbreviated form according to the conventions used for the addition of citation metadata.

**Attributes** (modifiers)

[Global attributes](#)

## Code example

```
<blockquote cite="https://DIGITAL.COM">  
<p>This is a quote taken from Review  
Squirrel</p></blockquote>
```

## CODE TAG

Fragment of computer code. The HTML Code Element (<code>) represents a fragment of computer code. By default, it is displayed in the browser's default monospace font.

**Attributes** (modifiers)

[Global attributes](#)

## Code example

```
<p>Regular text. <code>This is code.</code>  
Regular text.</p>
```

## DEL TAG

Text that has been removed during document editing. The HTML Deleted Text Element (<del>) represents a range of text that has been deleted from a document. This element is often (but need not be) rendered with strike-through text.

**Attributes** (modifiers)

cite | datetime + [global attributes](#)

## Code example

```
<p><del>This text has been deleted</del>, here  
is the rest of the paragraph.</p><del ><p  
>This paragraph has been deleted.</p ></del >
```

## DFN TAG

Defining instance of a term. The HTML Definition Element (<dfn>) represents the defining instance of a term.

**Attributes** (modifiers)

[Global attributes](#)

## Code example

```
<p><dfn id="def-internet">The Internet</dfn>  
is a global system of interconnected networks  
that use the Internet Protocol Suite (TCP/IP)  
to serve billions of users worldwide.</p>
```

## EM TAG

Text that should be emphasized. The HTML element emphasis `<em>` marks text that has stress emphasis. The `<em>` element can be nested, with each level of nesting indicating a greater degree of emphasis.

**Attributes** (modifiers)

[Global attributes](#)

### Code example

```
<p>In HTML 5, what was previously called  
<em>block-level</em> content is now called  
<em>flow</em> content.</p>
```

## I TAG

Text in a alternate voice or mood, such as a technical term. The HTML `<i>` Element represents a range of text that is set off from the normal text for some reason, for example, technical terms, foreign language phrases, or fictional character thoughts. It is typically displayed in italic type.

**Attributes** (modifiers)

[Global attributes](#)

### Code example

```
<p>The Latin phrase <i>Veni, vidi, vici</i> is  
often mentioned in music, art, and  
literature</p>
```

## INS TAG

Text that has been inserted during document editing. The HTML `<ins>` Element (or HTML Inserted Text) HTML represents a range of text that has been added to a document.

**Attributes** (modifiers)

`cite` | `datetime` + [global attributes](#)

## Code example

```
<ins>This text has been inserted</ins>
```

## KBD TAG

Example input (usually keyboard) for a program. The HTML Keyboard Input Element (`<kbd>`) represents user input and produces an inline element displayed in the browser's default monospace font.

**Attributes** (modifiers)

[Global attributes](#)

## Code example

```
<p>Save the document by pressing  
<kbd>Ctrl</kbd> + <kbd>S</kbd></p>
```

## MARK TAG

Text highlighted for referencing elsewhere. The HTML Mark Element (`<mark>`) represents highlighted text, i.e., a run of text marked for reference purpose, due to its relevance in a particular context. For example it can be used in a page showing search results to highlight every instance of the searched-for word.

**Attributes** (modifiers)

[Global attributes](#)

## Code example

```
<p>The <mark> element is used to  
<mark>highlight</mark> text</p>
```

## Q TAG

Phrasing content quoted from another source. The HTML Quote Element (`<q>`) indicates that the enclosed text is a short inline quotation. This element is intended for short quotations that don't require paragraph breaks; for long quotations use `<blockquote>` element.

**Attributes** (modifiers)

cite + [global attributes](#)

## Code example

```
<p>Everytime Kenny is killed, Stan will  
announce  
  <q  
  cite="http://en.wikipedia.org/wiki/Kenny_McCor  
  mick#Cultural_impact">  
    Oh my God, you/they killed Kenny!  
  </q>.  
</p>
```

## RP TAG

Contains semantically meaningless markup for browsers that don't understand ruby annotations. The HTML `<rp>` element is used to provide fall-back parenthesis for browsers non-supporting ruby annotations. Ruby annotations are for showing pronunciation of East Asian characters, like using Japanese furigana or Taiwanese bopomofo characters. The `<rp>` element is used in the case of lack of `<ruby>` element support its content has what should be displayed in order to indicate the presence of a ruby annotation, usually parentheses.

**Attributes** (modifiers)

[Global attributes](#)

## Code example

```
<ruby>  
  漢 <rp>(</rp><rt>Kan</rt><rp>)</rp>  
  字 <rp>(</rp><rt>ji</rt><rp>)</rp>  
</ruby>
```

## RT TAG

Annotation of preceding text. The HTML `<rt>` Element embraces pronunciation of characters presented in a ruby annotations, which are used to describe the pronunciation of East Asian characters. This element is always used inside a `<ruby>` element.

**Attributes** (modifiers)

[Global attributes](#)

## Code example

```
<ruby>
  漢 <rp>(</rp><rt>Kan</rt><rp>)</rp>
  字 <rp>(</rp><rt>ji</rt><rp>)</rp>
</ruby>
```

## RUBY TAG

Contains text with annotations, such as pronunciation hints.

Commonly used in East Asian text. The HTML `<ruby>` Element represents a ruby annotation. Ruby annotations are for showing pronunciation of East Asian characters.

**Attributes** (modifiers)

[Global attributes](#)

## Code example

```
<ruby>
  漢 <rp>(</rp><rt>Kan</rt><rp>)</rp>
  字 <rp>(</rp><rt>ji</rt><rp>)</rp>
</ruby>
```

## S TAG

Strikethrough text that is outdated or no longer accurate. The HTML Strikethrough Element (`<s>`) renders text with a strikethrough, or a line through it. Use the `<s>` element to represent things that are no longer relevant or no longer accurate. However, `<s>` is not appropriate when indicating document edits; for that, use the `<del>` and `<ins>` elements, as appropriate.

**Attributes** (modifiers)

[Global attributes](#)

## Code example

```
<p>Planet earth is <s>flat</s> round</p>
```



## SAMP TAG

Sample output of a program. The HTML `<samp>` element is an element intended to identify sample output from a computer program. It is usually displayed in the browser's default monotype font (such as Lucida Console).

**Attributes** (modifiers)

[Global attributes](#)

### Code example

```
<p>Regular text. <samp>This is sample text.
</samp> Regular text.</p>
```

## SMALL TAG

Small text, such as fine print. The HTML Small Element (`<small>`) makes the text font size one size smaller (for example, from large to medium, or from small to x-small) down to the browser's minimum font size. In HTML5, this element is repurposed to represent side-comments and small print, including copyright and legal text, independent of its styled presentation.

**Attributes** (modifiers)

[Global attributes](#)

### Code example

```
<p>This is the first sentence. <small>This
whole sentence is in small letters.</small>
</p>
```

## SPAN TAG

Container with no semantic meaning. The HTML `<span>` element is a generic inline container for phrasing content, which does not inherently represent anything. It can be used to group elements for styling purposes (using the class or id attributes), or because they share attribute values, such as lang. It should be used only when no other semantic element is appropriate. `<span>` is very much like a

<div> element, but <div> is a block-level element whereas a <span> is an inline element.

**Attributes** (modifiers)

[Global attributes](#)

### Code example

```
<p><span>Some text</span></p>
```

## STRONG TAG

Text that is important. The HTML Strong Element (<strong>) gives text strong importance, and is typically displayed in bold.

**Attributes** (modifiers)

[Global attributes](#)

### Code example

```
<p>When doing x it is  
<strong>imperative</strong> to do y before  
proceeding.</p>
```

## SUB TAG

Subscript text. The HTML Subscript Element (<sub>) defines a span of text that should be displayed, for typographic reasons, lower, and often smaller, than the main span of text.

**Attributes** (modifiers)

[Global attributes](#)

### Code example

```
<p>The chemical formula of water is  
H<sub>2</sub>O</p>
```

## SUP TAG

Superscript text. The HTML Superscript Element (<sup>) defines a span of text that should be displayed, for typographic reasons, higher, and often smaller, than the main span of text.

**Attributes** (modifiers)

[Global attributes](#)

### Code example

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

## TIME TAG

Time defined in a machine readable format. The HTML <time> element represents either a time on a 24-hour clock or a precise date in the Gregorian calendar (with optional time and timezone information). This element is intended to be used presenting dates and times in a machine readable format. This can be helpful for user agents to offer any event scheduling for user's calendar.

**Attributes** (modifiers)

datetime + [global attributes](#)

### Code example

```
<p>The concert starts at <time>20:00</time>.  
</p>
```

## VAR TAG

Mathematical or programming variable. The HTML Variable Element (<var>) represents a variable in a mathematical expression or a programming context.

**Attributes** (modifiers)

[Global attributes](#)

### Code example

```
<p> A simple equation: <var>x</var> =  
<var>y</var> + 2 </p>
```

## WBR TAG

Opportunity for a line break. The HTML element word break opportunity `<wbr>` represents a position within text where the browser may optionally break a line, though its line-breaking rules would not otherwise create a break at that location.

**Attributes** (modifiers)

[Global attributes](#)

### Code example

```
<p>http://this<wbr>.is<wbr>.a<wbr>.really<wbr>
.long<wbr>.example<wbr>.com/With<wbr>/deeper<w
br>/level<wbr>/pages<wbr>/deeper<wbr>/level<wb
r>/pages<wbr>/deeper<wbr>/level<wbr>/pages<wbr
>/deeper<wbr>/level<wbr>/pages<wbr>/deeper<wbr
>/level<wbr>/pages</p>
```

[⤴ Back to categories](#)

## CAPTION TAG

Title of a table. The HTML `<caption>` Element (or HTML Table Caption Element) represents the title of a table. Though it is always the first descendant of a `<table>`, its styling, using CSS, may place it elsewhere, relative to the table.

**Attributes** (modifiers)

[Global attributes](#)

### Code example

```
<table>
  <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>
```

## COL TAG

Columns in a table. The HTML Table Column Element (<col>) defines a column within a table and is used for defining common semantics on all common cells. It is generally found within a <colgroup> element.

**Attributes** (modifiers)

span + [global attributes](#)

### Code example

```
<table>
  <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>
```

## COLGROUP TAG

Defines a group of columns in a table. The HTML Table Column Group Element (<colgroup>) defines a group of columns within a table.

**Attributes** (modifiers)

span + [global attributes](#)

### Code example

```
<table>
  <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>
```

## TABLE TAG

Table of multi-dimensional data. The HTML Table Element (<table>) represents tabular data: information expressed via two dimensions or more.

**Attributes** (modifiers)

summary + [global attributes](#)

### Code example

```
<table>
  <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>
```

# TBODY TAG

Contains rows that hold the table's data. The HTML Table Body Element (<tbody>) defines one or more <tr> element data-rows to be the body of its parent <table> element (as long as no <tr> elements are immediate children of that table element.) In conjunction with a preceding <thead> and/or <tfoot> element, <tbody> provides additional semantic information for devices such as printers and displays. Of the parent table's child elements, <tbody> represents the content which, when longer than a page, will most likely differ for each page printed; while the content of <thead> and <tfoot> will be the same or similar for each page printed. For displays, <tbody> will enable separate scrolling of the <thead>, <tfoot>, and <caption> elements of the same parent <table> element. Note that unlike the <thead>, <tfoot>, and <caption> elements however, multiple <tbody> elements are permitted (if consecutive), allowing the data-rows in long tables to be divided into different sections, each separately formatted as needed.

## Attributes (modifiers)

[Global attributes](#)

## Code example

```
<table>
<thead>
  <tr> ...header information... </tr>
</thead>
<tfoot>
  <tr> ...footer information... </tr>
</tfoot>
<tbody>
  <tr> ...first row of block one data...
</tr>
  <tr> ...second row of block one data...
</tr>
</tbody>
<tbody>
  <tr> ...first row of block two data...
</tr>
  <tr> ...second row of block two data...
</tr>
  <tr> ...third row of block two data...
</tr>
</tbody>
</table>
```

## TD TAG

Table cell. The Table cell HTML element (<td>) defines a cell of a table that contains data. It participates in the table model.

**Attributes** (modifiers)

colspan | rowspan | headers + [global attributes](#)

### Code example

```
<table>
  <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>
```

## TFOOT TAG

Contains rows with summary of data. The HTML Table Foot Element (<tfoot>) defines a set of rows summarizing the columns of the table.

**Attributes** (modifiers)

[Global attributes](#)

### Code example

```
<table>
  <thead>
    <tr> ...header information... </tr>
  </thead>
  <tfoot>
    <tr> ...footer information... </tr>
  </tfoot>
  <tbody>
```



```
    <tr> ...first row of block one data...
</tr>
    <tr> ...second row of block one data...
</tr>
</tbody>
<tbody>
    <tr> ...first row of block two data...
</tr>
    <tr> ...second row of block two data...
</tr>
    <tr> ...third row of block two data...
</tr>
</tbody>
</table>
```

## TH TAG

Table heading. The HTML element table header cell `<th>` defines a cell as a header for a group of cells of a table. The group of cells that the header refers to is defined by the `scope` and `headers` attribute.

### Attributes (modifiers)

`colspan` | `rowspan` | `scope` | `headers` + [global attributes](#)

### Code example

```
<table>
  <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>
```

## THEAD TAG

Contains rows with table headings. The HTML Table Head Element (<thead>) defines a set of rows defining the head of the columns of the table.

**Attributes** (modifiers)

[Global attributes](#)

## Code example

```
<table>
  <thead>
    <tr> ...header information... </tr>
  </thead>
  <tfoot>
    <tr> ...footer information... </tr>
  </tfoot>
  <tbody>
    <tr> ...first row of block one data...
  </tr>
    <tr> ...second row of block one data...
  </tr>
  </tbody>
  <tbody>
    <tr> ...first row of block two data...
  </tr>
    <tr> ...second row of block two data...
  </tr>
    <tr> ...third row of block two data...
  </tr>
  </tbody>
</table>
```

## TR TAG

A row of cells in a table. The HTML element table row <tr> defines a row of cells in a table. Those can be a mix of <td> and <th> elements.

**Attributes** (modifiers)

[Global attributes](#)

## Code example

```
<table>
  <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>
```

[↗ Back to categories](#)

## BUTTON TAG

A button. The HTML `<button>` Element represents a clickable button.

### Attributes (modifiers)

autofocus | disabled | form | formaction | formenctype | formmethod | formnovalidate | formtarget (\_self | \_blank | \_parent | \_top) | name | type | value + [global attributes](#)

### Code example

```
<button name="button">I am a button. Click me!
</button>
```

## DATA TAG

Allows for machine-readable data to be provided. The HTML `<data>` Element links a given content with a machine-readable translation. If the content is time- or date-related, the `<time>` must be used.

### Attributes (modifiers)

value + [global attributes](#)

### Code example

```
<ul>
  <li><data value="3967381398">Mini
  Ketchup</data></li>
  <li><data value="3967381399">Jumbo
  Ketchup</data></li>
  <li><data value="3967381400">Mega Jumbo
  Ketchup</data></li>
</ul>
```

## DATALIST TAG

Define sets of options. The HTML Datalist Element (<datalist>) contains a set of <option> elements that represent the values available for other controls.

**Attributes** (modifiers)

[Global attributes](#)

### Code example

```
<label>Choose a browser from this list:
<input list="browsers" name="myBrowser" />
</label>
<datalist id="browsers">
  <option value="Chrome">
  <option value="Firefox">
  <option value="Internet Explorer">
  <option value="Opera">
  <option value="Safari">
  <option value="Microsoft Edge">
</datalist>
```

## FIELDSET TAG

Set of form controls grouped by theme. The HTML <fieldset> element is used to group several controls as well as labels (<label>) within a web form.

**Attributes** (modifiers)

disabled | form | name + [global attributes](#)

### Code example

```
<form action="test.php" method="post">
  <fieldset>
    <legend>Title</legend>
    <input type="radio" id="radio"> <label
for="radio">Click me</label>
  </fieldset>
</form>
```

## FORM TAG

Used to create an HTML form for user input. The HTML `<form>` element represents a document section that contains interactive controls to submit information to a web server. It is possible to use the `:valid` and `:invalid` CSS pseudo-classes to style a `<form>` element.

### Attributes (modifiers)

action | autocomplete | name | novalidate | accept-charset | enctype | method | target ( \_self | \_blank | \_parent | \_top) + [global attributes](#)

### Code example

```
<form action="" method="post">
  <fieldset>
    <legend>Title</legend>
    <input type="radio" id="radio"> <label
for="radio">Click me</label>
  </fieldset>
</form>
```

## INPUT TAG

Generic form input. The HTML element `<input>` is used to create interactive controls for web-based forms in order to accept data from the user. How an `<input>` works varies considerably depending on the value of its type attribute.

### Attributes (modifiers)

accept | alt | auto-complete | autofocus | checked | disabled | form | formaction | formenctype | formmethod | formnovalidate | formtarget | height | list | max | maxlength | min | multiple | name | pattern | placeholder | readonly | required | size | src | step | type | value | width | webkitdirectory | global | spellcheck | selectionDirection + [global attributes](#)

## Code example

```
<input type="text" value="Type here">
```

## LABEL TAG

Caption for a form control. The HTML Label Element (<label>) represents a caption for an item in a user interface. It can be associated with a control either by placing the control element inside the <label> element, or by using the for attribute. Such a control is called the labeled control of the label element. One input can be associated with multiple labels.

**Attributes** (modifiers)

for + [global attributes](#)

## Code example

```
<label>Click me <input type="text"></label>
```

## LEGEND TAG

Define a name for a fieldset. The HTML <legend> Element (or HTML Legend Field Element) represents a caption for the content of its parent <fieldset>.

**Attributes** (modifiers)

[Global attributes](#)

## Code example

```
<form action="" method="post">
  <fieldset>
    <legend>Title</legend>
    <input type="radio" id="radio"> <label
for="radio">Click me</label>
  </fieldset>
</form>
```

## METER TAG

Control for entering a numeric value in a known range. The HTML `<meter>` Element represents either a scalar value within a known range or a fractional value.

**Attributes** (modifiers)

high | low | max | min | optimum | value | form + [global attributes](#)

### Code example

```
<p>Heat the oven to <meter min="200" max="500" value="350">350 degrees</meter>.</p>
```

## OPTGROUP TAG

Group of option. In a Web form, the HTML `<optgroup>` element creates a grouping of options within a `<select>` element.

**Attributes** (modifiers)

disabled | label + [global attributes](#)

### Code example

```
<select>
  <optgroup label="Group 1">
    <option>Option 1.1</option>
  </optgroup>
  <optgroup label="Group 2">
    <option>Option 2.1</option>
    <option>Option 2.2</option>
  </optgroup>
  <optgroup label="Group 3" disabled>
    <option>Option 3.1</option>
    <option>Option 3.2</option>
    <option>Option 3.3</option>
  </optgroup>
</select>
```

## OPTION TAG

Single option within a select control. In a Web form, the HTML `<option>` element is used to create a control representing an item within a `<select>`, an `<optgroup>` or a `<datalist>` HTML5 element.

**Attributes** (modifiers)

disabled | label | selected | value + [global attributes](#)

## Code example

```
<select name="select">
  <option value="value1">Value 1</option>
  <option value="value2" selected>Value
2</option>
  <option value="value3">Value 3</option>
</select>
```

## OUTPUT TAG

Contains the results of a calculation. The HTML `<output>` element represents the result of a calculation or user action.

**Attributes** (modifiers)

form | for | name + [global attributes](#)

## Code example

```
<form
oninput="result.value=parseInt(a.value)+parseI
nt(b.value)">
  <input type="range" name="b" value="50" />
+
  <input type="number" name="a" value="10"
/> =
  <output name="result">60</output>
</form>
```

## PROGRESS TAG

Control for displaying progress of a task. The HTML `<progress>` Element is used to view the completion progress of a task. While the specifics of how it's displayed is left up to the browser developer, it's typically displayed as a progress bar. Javascript can be used to manipulate the value of progress bar.

**Attributes** (modifiers)

max | value + [global attributes](#)

## Code example



```
<progress value="70" max="100">70 %</progress>
```

## SELECT TAG

Control for selecting from multiple options. The HTML select (<select>) element represents a control that presents a menu of options. The options within the menu are represented by <option> elements, which can be grouped by <optgroup> elements. Options can be pre-selected for the user.

### Attributes (modifiers)

autofocus | size | disabled | form | multiple | name | required + [global attributes](#)

### Code example

```
<select name="select">
  <option value="value1">Value 1</option>
  <option value="value2" selected>Value
2</option>
  <option value="value3">Value 3</option>
</select>
```

## TEXTAREA TAG

Multiline free-form text input. The HTML <textarea> element represents a multi-line plain-text editing control.

### Attributes (modifiers)

autocomplete | autofocus | cols | disabled | dirname | form | name | readonly | required | rows | maxlength | minlength | placeholder | wrap | selectionDirection | selectionEnd | selectionStart | spellcheck + [global attributes](#)

### Code example

```
<textarea name="textarea" rows="10"
cols="50">Write something here</textarea>
```

## EMBED TAG

Integration point for an external (typically non-HTML) application or interactive content. The HTML `<embed>` Element represents an integration point for an external application or interactive content (in other words, a plug-in).

**Attributes** (modifiers)

height | src | type | width + [global attributes](#)

### Code example

```
<embed type="video/quicktime" src="movie.mov"
width="640" height="480">
```

## IFRAME TAG

Nested browser frame. The HTML Inline Frame Element (`<iframe>`) represents a nested browsing context, effectively embedding another HTML page into the current page. In HTML 4.01, a document may contain a head and a body or a head and a frameset, but not both a body and a frameset. However, an `<iframe>` can be used within a normal document body. Each browsing context has its own session history and active document. The browsing context that contains the embedded content is called the parent browsing context. The top-level browsing context (which has no parent) is typically the browser window.

**Attributes** (modifiers)

allowfullscreen | src | name | sandbox | seamless | width | height | srcdoc | referrerpolicy + [global attributes](#)

### Code example

```
<iframe src="http:www.example.com/iframe-
example" width="400" height="300">
  <p>Your browser does not support iframes.
</p>
</iframe>
```

## IMG TAG

An image. The HTML `<img>` element represents an image in the document.

**Attributes** (modifiers)

`alt` | `src` | `height` | `ismap` | `usemap` | `width` | `crossorigin` | `longdesc` | `referrerpolicy` | `sizes` | `srcset` + [global attributes](#)

### Code example

```

```

## OBJECT TAG

External resource such as an image, iframe or plugin. The HTML Embedded Object Element (`<object>`) represents an external resource, which can be treated as an image, a nested browsing context, or a resource to be handled by a plugin.

**Attributes** (modifiers)

`data` | `height` | `type` | `usemap` | `width` | `form` | `name` | `typemustmatch` | `usemap` + [global attributes](#)

### Code example

```
<object data="move.swf" type="application/x-shockwave-flash">  
  <param name="foo" value="bar">  
</object>
```

## PARAM TAG

Parameters for the parent object. The HTML `<param>` Element (or HTML Parameter Element) defines parameters for `<object>`.

**Attributes** (modifiers)

`name` | `value` + [global attributes](#)

### Code example

```
<object data="move.swf" type="application/x-shockwave-flash">  
  <param name="foo" value="bar">  
</object>
```

## SOURCE TAG

Alternative sources for parent video or audio elements. The HTML `<source>` element specifies multiple media resources for either the `<picture>`, the `<audio>` or the `<video>` element. It is an empty element. It is commonly used to serve the same media content in multiple formats supported by different browsers.

**Attributes** (modifiers)

media | src | type | sizes | srcset + [global attributes](#)

### Code example

```
<video controls poster="/images/sample.gif">
  <source src="sample.mp4" type="video/mp4">
  <source src="sample.ogv" type="video/ogv">
  <track kind="captions"
src="sampleCaptions.vtt" srclang="en">
  <track kind="descriptions"
src="sampleDescriptions.vtt" srclang="en">
  <track kind="chapters"
src="sampleChapters.vtt" srclang="en">
  <track kind="subtitles"
src="sampleSubtitles_de.vtt" srclang="de">
  <track kind="subtitles"
src="sampleSubtitles_en.vtt" srclang="en">
  <track kind="subtitles"
src="sampleSubtitles_ja.vtt" srclang="ja">
  <track kind="subtitles"
src="sampleSubtitles_oz.vtt" srclang="oz">
  <track kind="metadata" src="keyStage1.vtt"
srclang="en" label="Key Stage 1">
  <track kind="metadata" src="keyStage2.vtt"
srclang="en" label="Key Stage 2">
  <track kind="metadata" src="keyStage3.vtt"
srclang="en" label="Key Stage 3">
</video>
```

## AREA TAG

Hyperlink with some text and a corresponding area on an image map, or a dead area on an image map. The HTML `<area>` element defines a hot-spot region on an image, and optionally associates it

with a hypertext link. This element is used only within a `<map>` element.

#### **Attributes** (modifiers)

alt | coords | download | href | hreflang | media | rel | referrerpolicy | shape | target | type + [global attributes](#)

### Code example

```
<map name="primary">
  <area shape="circle" coords="200,250,25"
href="another.htm" />
  <area shape="default" nohref />
</map>
```

[↗ Back to categories](#)

## AUDIO TAG

Sound or audio stream. The HTML `<audio>` element is used to embed sound content in documents. It may contain one or more audio sources, represented using the `src` attribute or the `<source>` element; the browser will choose the most suitable one.

#### **Attributes** (modifiers)

autoplay | buffered | preload | loop | controls | src | muted | played | volume + [global attributes](#)

### Code example

```
<audio
src="http://developer.mozilla.org/@api/deki/files/2926/=AudioTest_(1).ogg" autoplay>
  Your browser does not support the
<code>audio</code> element.
</audio>
```

## MAP TAG

Image map for adding hyperlinks to parts of an image. The HTML `<map>` element is used with `<area>` elements to define an image map

(a clickable link area).

**Attributes** (modifiers)

name + [global attributes](#)

## Code example

```
<map name="example-map-1">
  <area shape="circle" coords="200,250,25"
href="another.htm" />
  <area shape="default" />
</map>
```

## TRACK TAG

Specifies external timing track for media element. The HTML `<track>` element is used as a child of the media elements—`<audio>` and `<video>`. It lets you specify timed text tracks (or time-based data), for example to automatically handle subtitles. The tracks are formatted in WebVTT format (.vtt files) — Web Video Text Tracks.

**Attributes** (modifiers)

default | kind | label | src | srclang + [global attributes](#)

## Code example

```
<video controls poster="/images/sample.gif">
  <source src="sample.mp4" type="video/mp4">
  <source src="sample.ogv" type="video/ogv">
  <track kind="captions"
src="sampleCaptions.vtt" srclang="en">
  <track kind="descriptions"
src="sampleDescriptions.vtt" srclang="en">
  <track kind="chapters"
src="sampleChapters.vtt" srclang="en">
  <track kind="subtitles"
src="sampleSubtitles_de.vtt" srclang="de">
  <track kind="subtitles"
src="sampleSubtitles_en.vtt" srclang="en">
  <track kind="subtitles"
src="sampleSubtitles_ja.vtt" srclang="ja">
  <track kind="subtitles"
src="sampleSubtitles_oz.vtt" srclang="oz">
  <track kind="metadata" src="keyStage1.vtt"
srclang="en" label="Key Stage 1">
```

```
<track kind="metadata" src="keyStage2.vtt"
srclang="en" label="Key Stage 2">
  <track kind="metadata" src="keyStage3.vtt"
srclang="en" label="Key Stage 3">
</video>
```

## VIDEO TAG

Used for playing videos or movies. Use the HTML `<video>` element to embed video content in a document. The video element contains one or more video sources. To specify a video source, use either the `src` attribute or the `<source>` element; the browser will choose the most suitable one.

### Attributes (modifiers)

`autoplay` | `controls` | `height` | `loop` | `poster` | `preload` | `src` | `width` | `buffered` | `crossorigin` | `muted` | `played` + [global attributes](#)

### Code example

```
<video src="videofile.webm" autoplay
poster="posterimage.jpg">
  Sorry, your browser doesn't support embedded
  videos,
  but don't worry, you can <a
href="videofile.webm">download it</a>
  and watch it with your favorite video
  player!
</video>
```

## DETAILS TAG

Contains additional information, such as the contents of an accordian view. The HTML Details Element (`<details>`) is used as a disclosure widget from which the user can retrieve additional information.

### Attributes (modifiers)

`open` + [global attributes](#)

### Code example

```
<details><summary>Some details</summary>
<p>More info about the details.</p></details>
```

## DIALOG TAG

The HTML `<dialog>` element represents a dialog box or other interactive component, such as an inspector or window. `<form>` elements can be integrated within a dialog by specifying them with the attribute `method="dialog"`. When such a form is submitted, the dialog is closed with a `returnValue` attribute set to the value of the submit button used.

**Attributes** (modifiers)

open + [global attributes](#)

### Code example

```
<dialog open>
  <p>Greetings, one and all!</p>
</dialog>
```

## MENU TAG

Set of commands. The HTML `<menu>` element represents a group of commands that a user can perform or activate. This includes both list menus, which might appear across the top of a screen, as well as context menus, such as those that might appear underneath a button after it has been clicked.

**Attributes** (modifiers)

label | type + [global attributes](#)

### Code example

```
<menu type="context" id="popup-menu">
  <menuitem>Action</menuitem>
  <menuitem>Another action</menuitem>
  <hr>
  <menuitem>Separated action</menuitem>
</menu>
```

## SUMMARY TAG



Caption of a details element. The HTML summary element (<summary>) is used as a summary, caption, or legend for the content of a <details> element.

**Attributes** (modifiers)

[Global attributes](#)

## Code example

```
<details><summary>Some details</summary>  
<p>More info about the details.</p></details>
```

## CANVAS TAG

Bitmap which is editable by client side scripts. The HTML <canvas> Element can be used to draw graphics via scripting (usually JavaScript). For example, it can be used to draw graphs, make photo compositions or even perform animations. You may (and should) provide alternate content inside the <canvas> block. That content will be rendered both on older browsers that don't support canvas and in browsers with JavaScript disabled.

**Attributes** (modifiers)

height | width + [global attributes](#)

## Code example

```
<canvas id="canvas" width="300"  
height="300">An alternative text describing  
what your canvas displays.</canvas>
```

## NOSCRIPT TAG

Contains elements that are part of the document only if scripting is disabled. The HTML <noscript> Element defines a section of html to be inserted if a script type on the page is unsupported or if scripting is currently turned off in the browser.

**Attributes** (modifiers)

[Global attributes](#)

## Code example

```
<noscript>
  <a
href="https://DIGITAL.COM/">External
Link</a>
</noscript>
```

## SCRIPT TAG

Inline or linked client side scripts. The HTML Script Element (<script>) is used to embed or reference an executable script within an HTML or XHTML document. Scripts without async or defer attributes, as well as inline scripts, are fetched and executed immediately, before the browser continues to parse the page.

### Attributes (modifiers)

async | type | defer | src | charset | integrity | text | language | defer | crossorigin + [global attributes](#)

### Code example

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

[⤴ Back to categories](#)

## GLOBAL ATTRIBUTES

Here below is a list of attributes supported by all HTML5 tags

Attribute	Description	Values
accesskey	Specifies a keyboard shortcut to access an element	character
class	Specifies a classname for an element (used to specify a class in a style sheet)	classname
contenteditable	Specifies if the user is allowed to edit the content or not	true   false

contextmenu	Specifies the context menu for an element	menu_id
dir	Specifies the text direction for the content in an element	ltr   rtl
draggable	Specifies whether or not a user is allowed to drag an element	true   false   auto
dropzone	Specifies what happens when dragged items/data is dropped in the element	copy   move   link
hidden	Specifies that the element is not relevant. Hidden elements are not displayed	hidden
id	Specifies a unique id for an element	id
lang	Specifies a language code for the content in an element.	language_code
spellcheck	Specifies if the element must have its spelling and grammar checked	true   false
style	Specifies an inline style for an element	style_definition
tabindex	Specifies the tab order of an element	number
title	Specifies extra information about an element	text

[⤴ Back to categories](#)

## NEW TAGS IN HTML5

Here below is a list of the new elements introduced in HTML5.

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

<article>	Specifies an article
<aside>	Specifies content aside from the page content
<bdi>	For bi-directional text formatting
<details>	Specifies details of an element
<dialog>	Specifies that part of an application is interactive.
<figcaption>	Specifies caption for the figure element.
<figure>	Specifies a group of media content, and their caption
<footer>	Specifies a footer for a section or page
<header>	Specifies a group of introductory or navigational aids, including hgroup elements
<main>	Specifies the main content area of an HTML document.
<mark>	Specifies marked text
<menuitem>	Specifies a command that a user can invoke from a popup menu.
<meter>	Specifies measurement within a predefined range
<nav>	Specifies navigation links
<progress>	Specifies progress of a task of any kind
<rp>	Used for the benefit of browsers that don't support ruby annotations
<rt>	Specifies the ruby text component of a ruby annotation.
<ruby>	Specifies a ruby annotation (used in East Asian typography)
<section>	Specifies a section
<summary>	Specifies a summary / caption for the <details> element
<time>	Specifies a date/time

<wbr> Specifies a line break opportunity for very long words and strings of text with no spaces.

[⤴ Back to categories](#)

Thanks to <https://developer.mozilla.org/en-US/docs/Web/HTML> for providing us with some of the definitions and code examples.