

---

**BEGINNER'S**

# HTML CHEAT SHEET



<b>Main root</b>	<b>2</b>
<b>Document metadata</b>	<b>2</b>
<b>Sectioning root</b>	<b>3</b>
<b>Content sectioning</b>	<b>3</b>
<b>Text content</b>	<b>4</b>
<b>Inline text semantics</b>	<b>6</b>
<b>Image and multimedia</b>	<b>8</b>
<b>Scripting</b>	<b>9</b>
<b>Demarcating edits</b>	<b>9</b>
<b>Table content</b>	<b>9</b>
<b>Forms</b>	<b>11</b>
<b>Interactive elements</b>	<b>12</b>

# Main root

`<html> ... </html>`

The HTML `<html>` element represents the root (top-level element) of an HTML document, so it is also referred to as the root element. All other elements must be descendants of this element.

Example:

```
<!DOCTYPE html>
<html lang="en">
  <head>...</head>
  <body>...</body>
</html>
```

# Document metadata

`<head> ... </head>`

The HTML `<head>` element contains machine-readable information (metadata) about the document, like its title, scripts, and style sheets.

`<link>`

The HTML External Resource Link element (`<link>`) specifies relationships between the current document and an external resource. This element is most commonly used to link to stylesheets, but is also used to establish site icons (both "favicon" style icons and icons for the home screen and apps on mobile devices) among other things.

`<meta>`

The HTML `<meta>` element represents metadata that cannot be represented by other HTML meta-related elements, like `<base>`, `<link>`, `<script>`, `<style>` or `<title>`

`<style> ... </style>`

The HTML `<style>` element contains style information for a document, or part of a document.

`<title> ... </title>`

The HTML Title element (`<title>`) defines the document's title that is shown in a browser's title bar or a page's tab.

Example:

```
<!DOCTYPE html>
<html lang="en">
  <head>...</head>
  <body>...</body>
</html>
```

## Sectioning root

**<body> ... </body>**

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

Example:

```
<html>
  <head>
    <title>Document title</title>
  </head>
  <body>
    <p>This is a paragraph</p>
  </body>
</html>
```

## Content sectioning

**<address> ... </address>**

The HTML `<address>` element indicates that the enclosed HTML provides contact information for a person or people, or for an organization.

**<article> ... </article>**

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).

**<aside> ... </aside>**

The HTML `<aside>` element represents a portion of a document whose content is only indirectly related to the document's main content.

**<footer> ... </footer>**

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.

### `<header> ... </header>`

The HTML `<header>` element represents introductory content, typically a group of introductory or navigational aids. It may contain some heading elements but also a logo, a search form, an author name, and other elements.

### `<h1> to <h6> ... </h6>`

The HTML `<h1>`–`<h6>` elements represent six levels of section headings. `<h1>` is the highest section level and `<h6>` is the lowest.

### `<main> ... </main>`

The HTML `<main>` element represents the dominant content of the `<body>` of a document. 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.

### `<nav> ... </nav>`

The HTML `<nav>` element represents a section of a page whose purpose is to provide navigation links, either within the current document or to other documents. Common examples of navigation sections are menus, tables of contents, and indexes.

### `<section> ... </section>`

The HTML `<section>` element represents a standalone section — which doesn't have a more specific semantic element to represent it — contained within an HTML document.

Example:

```
<address>
  <a href="mailto:jim@rock.com">jim@rock.com</a><br>
  <a href="tel:+13115552368">(311) 555-2368</a>
</address>
```

## Text content

### `<blockquote> ... </blockquote>`

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.

### `<dd> ... </dd>`

The HTML `<dd>` element provides the description, definition, or value for the preceding term (`<dt>`) in a description list (`<dl>`).

`<div> ... </div>`

The HTML Content Division element (`<div>`) is the generic container for flow content. It has no effect on the content or layout until styled using CSS.

`<dl> ... </dl>`

The HTML `<dl>` element represents a description list. The element encloses a list of groups of terms (specified using the `<dt>` element) and descriptions (provided by `<dd>` elements). Common uses for this element are to implement a glossary or to display metadata (a list of key-value pairs).

`<dt> ... </dt>`

The HTML `<dt>` element specifies a term in a description or definition list, and as such must be used inside a `<dl>` element.

`<figcaption> ... </figcaption>`

The HTML `<figcaption>` or Figure Caption element represents a caption or legend describing the rest of the contents of its parent `<figure>` element.

`<figure> ... </figure>`

The HTML `<figure>` (Figure With Optional Caption) element represents self-contained content, potentially with an optional caption, which is specified using the (`<figcaption>`) element.

`<hr>`

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 within a section.

`<li> ... </li>`

The HTML `<li>` element is used to represent an item in a list.

`<ol> ... </ol>`

The HTML `<ol>` element represents an ordered list of items, typically rendered as a numbered list.

`<p> ... </p>`

The HTML `<p>` element represents a paragraph.

`<pre> ... </pre>`

The HTML `<pre>` element represents preformatted text which is to be presented exactly as written in the HTML file.

`<ul> ... </ul>`

The HTML `<ul>` element represents an unordered list of items, typically rendered as a bulleted list.

Examples:

```
<dl>
  <dt>Denim (semigloss finish)</dt>
  <dd>Ceiling</dd>

  <dt>Denim (eggshell finish)</dt>
  <dt>Evening Sky (eggshell finish)</dt>
  <dd>Layered on the walls</dd>
</dl>

<figure>
  
  <figcaption>An elephant at sunset</figcaption>
</figure>

<ol>
  <li>Mix flour, baking powder, sugar, and salt.</li>
  <li>In another bowl, mix eggs, milk, and oil.</li>
  <li>Stir both mixtures together.</li>
  <li>Fill muffin tray 3/4 full.</li>
  <li>Bake for 20 minutes.</li>
</ol>
```

## Inline text semantics

**<a> ... </a>**

The HTML `<a>` element (or anchor element), with its `href` attribute, creates a hyperlink to web pages, files, email addresses, locations in the same page, or anything else a URL can address.

**<abbr> ... </abbr>**

The HTML Abbreviation element (`<abbr>`) represents an abbreviation or acronym; the optional title attribute can provide an expansion or description for the abbreviation.

**<br>**

The HTML `<br>` element 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.

**<cite> ... </cite>**

The HTML Citation element (`<cite>`) is used to describe a reference to a cited creative work, and must include the title of that work.

`<code> ... </code>`

The HTML `<code>` element displays its contents styled in a fashion intended to indicate that the text is a short fragment of computer code.

`<em> ... </em>`

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

`<mark> ... </mark>`

The HTML Mark Text element (`<mark>`) represents text which is marked or highlighted for reference or notation purposes, due to the marked passage's relevance or importance in the enclosing context.

`<small> ... </small>`

The HTML `<small>` element represents side-comments and small print, like copyright and legal text, independent of its styled presentation. By default, it renders text within it one font-size small, such as from small to x-small.

`<span> ... </span>`

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.

`<strong> ... </strong>`

The HTML Strong Importance Element (`<strong>`) indicates that its contents have strong importance, seriousness, or urgency. Browsers typically render the contents in bold type.

`<time> ... </time>`

The HTML `<time>` element represents a specific period in time.

Examples:

`<blockquote>`

```
  <p>It was a bright cold day in April, and the clocks were striking
  thirteen.</p>
```

```
  <footer>
```

```
    First sentence in <cite><a
href="http://www.george-orwell.org/1984/0.html"><em>Nineteen
Eighty-Four</em></a></cite> by George Orwell (Part 1, Chapter 1).
```

```
  </footer>
```

```
</blockquote>
```

```
<p>You can use <abbr title="Cascading Style Sheets">CSS</abbr> to style your
<abbr title="HyperText Markup Language">HTML</abbr>.</p>
```



# Image and multimedia

## `<audio> ... </audio>`

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. It can also be the destination for streamed media, using a `MediaStream`.

## `<img>`

The HTML `<img>` element embeds an image into the document.

## `<track>`

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 or Timed Text Markup Language (TTML).

## `<video> ... </video>`

The HTML Video element (`<video>`) embeds a media player which supports video playback into the document. You can use `<video>` for audio content as well, but the `<audio>` element may provide a more appropriate user experience.

Examples:

```

```

```
<figure>
  <figcaption>Listen to the T-Rex:</figcaption>
  <audio
    controls
    src="/media/examples/t-rex-roar.mp3">
    Your browser does not support the
    <code>audio</code> element.
  </audio>
</figure>
```

```
<video controls width="250"
  src="/media/examples/friday.mp4">

  <track default kind="captions"
    srclang="en"
```

```
src="/media/examples/friday.vtt"/>
```

```
Sorry, your browser doesn't support embedded videos.  
</video>
```

## Scripting

**<script> ... </script>**

The HTML `<script>` element is used to embed or reference executable code; this is typically used to embed or refer to JavaScript code.

Example:

```
<!-- HTML4 -->  
<script type="text/javascript" src="javascript.js"></script>  
  
<!-- HTML5 -->  
<script src="javascript.js"></script>
```

## Demarcating edits

**<del> ... </del>**

The HTML `<del>` element represents a range of text that has been deleted from a document.

**<ins> ... </ins>**

The HTML `<ins>` element represents a range of text that has been added to a document.

Example:

```
<p>"You're late!"</p>  
<del>  
  <p>"I apologize for the delay."</p>  
</del>  
<ins cite="../howtobeawizard.html" datetime="2018-05">  
  <p>"A wizard is never late ..."</p>  
</ins>
```

## Table content

**<caption> ... </caption>**

The HTML Table Caption element (`<caption>`) specifies the caption (or title) of a table, and if used is always the first child of a `<table>`.

### `<table> ... </table>`

The HTML `<table>` element represents tabular data — that is, information presented in a two-dimensional table comprised of rows and columns of cells containing data.

### `<tbody> ... </tbody>`

The HTML Table Body element (`<tbody>`) encapsulates a set of table rows (`<tr>` elements), indicating that they comprise the body of the table (`<table>`).

### `<td> ... </td>`

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

### `<tfoot> ... </tfoot>`

The HTML `<tfoot>` element defines a set of rows summarizing the columns of the table.

### `<th> ... </th>`

The HTML `<th>` element defines a cell as header of a group of table cells. The exact nature of this group is defined by the `scope` and `headers` attributes.

### `<thead> ... </thead>`

The HTML `<thead>` element defines a set of rows defining the head of the columns of the table.

### `<tr> ... </tr>`

The HTML `<tr>` element defines a row of cells in a table. The row's cells can then be established using a mix of `<td>` (data cell) and `<th>` (header cell) elements.

Example:

```
<table>
  <thead>
    <tr>
      <th colspan="2">The table header</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>The table body</td>
      <td>with two columns</td>
    </tr>
  </tbody>
</table>
```

# Forms

## `<button> ... </button>`

The HTML `<button>` element represents a clickable button, which can be used in forms or anywhere in a document that needs simple, standard button functionality.

## `<datalist> ... </datalist>`

The HTML `<datalist>` element contains a set of `<option>` elements that represent the values available for other controls.

## `<fieldset> ... </fieldset>`

The HTML `<fieldset>` element is used to group several controls as well as labels (`<label>`) within a web form.

## `<form> ... </form>`

The HTML `<form>` element represents a document section that contains interactive controls for submitting information to a web server.

## `<input>`

The HTML `<input>` element is used to create interactive controls for web-based forms in order to accept data from the user; a wide variety of types of input data and control widgets are available, depending on the device and user agent.

## `<label> ... </label>`

The HTML `<label>` element represents a caption for an item in a user interface.

## `<legend> ... </legend>`

The HTML `<legend>` element represents a caption for the content of its parent `<fieldset>`.

## `<optgroup> ... </optgroup>`

The HTML `<optgroup>` element creates a grouping of options within a `<select>` element.

## `<option> ... </option>`

The HTML `<option>` element is used to define an item contained in a `<select>`, an `<optgroup>`, or a `<datalist>` element. As such, `<option>` can represent menu items in popups and other lists of items in an HTML document.

## `<progress> ... </progress>`

The HTML `<progress>` element displays an indicator showing the completion progress of a task, typically displayed as a progress bar.

`<select> ... </select>`

The HTML `<select>` element represents a control that provides a menu of options

`<textarea> ... </textarea>`

The HTML `<textarea>` element represents a multi-line plain-text editing control, useful when you want to allow users to enter a sizeable amount of free-form text, for example a comment on a review or feedback form.

Example:

```
<form action="" method="get" class="form-example">
  <div class="form-example">
    <label for="name">Enter your name: </label>
    <input type="text" name="name" id="name" required>
  </div>
  <div class="form-example">
    <label for="email">Enter your email: </label>
    <input type="email" name="email" id="email" required>
  </div>
  <div class="form-example">
    <input type="submit" value="Subscribe!">
  </div>
</form>
```

## Interactive elements

`<details> ... </details>`

The HTML Details Element (`<details>`) creates a disclosure widget in which information is visible only when the widget is toggled into an "open" state.

`<summary> ... </summary>`

The HTML Disclosure Summary element (`<summary>`) element specifies a summary, caption, or legend for a `<details>` element's disclosure box.

Example:

```
<details>
  <summary>Details</summary>
  Something small enough to escape casual notice.
</details>
```

Source: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element>

# HTML CheatSheet

## Attributes

### SYNTAX

```
<tag attributename="value" />
```

- lowercase attributes, quote values

### Global attributes

accesskey, class, contenteditable, data-\*, dir, draggable, hidden, id, lang, spellcheck, style, tabindex, title

```
<div id="demo" class="big" dir="ltr" lang="en" style="color: red;" tabindex="0" title="Tooltip" contenteditable="true" spellcheck="true" data-htmlcheat="99">Hello World!</div>
```

### Internationalization: dir, lang, xml:lang

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

...

```
  <p dir="rtl">Right to left (Arabic)</p>
```

...

```
</html>
```

### Link: download, href, hreflang, media, rel, target, type

```
<a href="https://htmlg.com/" target="_blank" rel="e" Link</a>
```

### Image: src, alt, height, ismap, longdesc, src, srcset, usemap, width

```

```

### All attributes

#### accept

form, input

#### accept-charset

form

#### accesskey

Global attribute

#### action

form

#### align

applet, caption, col, colgroup, hr, iframe, img, table, tbody, td, tfoot, th, thead, tr

#### alt

applet, area, img, input

#### async

script

#### autocomplete

form, input

#### autofocus

button, input, keygen, select, textarea

#### autoplay

## Head Tags

```
<!doctype html>
<html lang="en" class="no-js">
<head>
  <meta charset="utf-8">
  <meta http-equiv="x-ua-compatible" content="ie=
```

## Tags

### Div Section

```
<div>Block element</div>
```

### Headings

```
<h1>Page title</h1>
```

```
<h2>Subheading</h2>
```

```
<h3>Tertiary heading</h3>
```

```
<h4>Quaternary heading</h4>
```

### Paragraph

```
<p style="text-align: center;">text</p>
```

### Image

```

```

### Outbound Link

Get your online assignment help at [Copycrafter](#) professional writing service.

```
<a href="https://htmlg.com/" target="_blank" rel='
```

### Mailto link

```
<a href="mailto:me@ruwix.com?Subject=Hi%20mate" to
```

### Inner anchor (jump on page)

```
<a href="#footer">Jump to footnote</a>
```

```
<br />
```

```
<a name="footer"></a>Footnote content
```

### Bold text

```
<strong>Bold text</strong>
```

### Italic text

```
<em>Italic text</em>
```

### Underlined text

```
<span style="text-decoration: underline;">Underlin
```

### Iframe

```
<iframe src="link.html" width="200" height="200" </iframe>
```

### Abbreviation

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

### Comment

```
<!-- HTML
Comment -->
```

### Horizontal Line

```
<hr />
```

### Line break

```
<br />
```

### Quotation

```
<q>Success is a journey not a destination.</q>
```

```

<meta name="viewport" content="width=device-wid
<link rel="canonical" href="https://htmlcheatsh
<title>HTML CheatSheet</title>
<meta name="description" content="A brief page
<meta name="keywords" content="html,cheatsheet"
<meta property="fb:admins" content="YourFaceboo
<meta property="og:title" content="HTML CheatSh
<meta property="og:type" content="website" />
<meta property="og:url" content="https://htmlch
<meta property="og:image" content="https://html
<meta property="og:description" content="A brie
<link rel="apple-touch-icon" href="apple-touch-
<link rel="alternate" hreflang="es" href="https
<link rel="stylesheet" href="/styles.css">
<script src="/script.js"></script>
</head>

```

## HTML5 Page Structure

header, nav, main, article, section, aside, footer, address

```

<header>
  <div id="logo">HTML</div>
  <nav>
    <ul>
      <li><a href="/">Home</a>
      <li><a href="/link">Page</a>
    </ul>
  </nav>
</header>
<main role="main">
  <article>
    <h2>Title 1</h2>
    <p>Content 1</p>
  </article>
  <article>
    <h2>Title 2</h2>
    <p>Content 2</p>
  </article>
</main>
<section>
  A group of related content
</section>
<aside>
  Sidebar
</aside>
<footer>
  <p>&copy; HTML CheatSheet</p>
  <address>
    Contact <a href="mailto:me@htmlg.com">me</a>
  </address>
</footer>

```

**Free HTML Templates**

## Color Picker

CE5937

1

color: #CE5937; Text color

background-color:#CE5937; Background

border: 3px solid #CE5937; Box border

```

<blockquote cite="https://ruwix.com/">
The Rubik's Cube is the World's best selling puzz:
</blockquote>

```

### Video

```

<video width="200" height="150" controls>
  <source src="vid.mp4" type="video/mp4">
  <source src="vid.ogg" type="video/ogg">
  No video support.
</video>

```

### Audio

```

<audio controls>
  <source src="sound.ogg" type="audio/ogg">
  <source src="sound.mp3" type="audio/mpeg">
  No audio support.
</audio>

```

## Structures

### Table

```

<table><caption>Phone numbers</caption>
  <thead>
    <tr>
      <th>Name</th>
      <th colspan="2">Phone</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>John</td>
      <td>577854</td>
      <td>577855</td>
    </tr>
    <tr>
      <td>Jack</td>
      <td>577856</td>
      <td>577857</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <td>&nbsp;</td>
      <td>Personal</td>
      <td>Office</td>
    </tr>
  </tfoot>
</table>

```

### Unordered list

```

<ul>
  <li>First</li>
  <li>Second</li>
  <li>Third</li>
</ul>

```

### Definition list

```

<dl>
  <dt>HTML</dt>
  <dd>Hypertext Markup Language</dd>
  <dt>CSS</dt>
  <dd>Cascading Style Sheets </dd>
</dl>

```

### Form

```

<form action="/action.php" method="post">
  Name: <input name="name" type="text" /> <br />

```

text-shadow: 1px 2px 2px #C|

Text shadow

box-shadow: 2px 2px 7px 1px

Box shadow

## Characters

á à â ä ã å &

< > " ' < > « »

☞ \$ ¢ £ ¥ € ₹ ₪

元 © ® ☂ ☕ ☎ ☃ ☠

☢ ♠ ♣ ♥ ♦ ▲ △ ○

■ § ¶ ☎ □ ☑ ☒ ✓

✂ ☺ ☻ 🎵 🎶 🚩 ✉ ✎

☀ ☆ ★ “ ” ♥ † ►

▶ ← ↑ → ↓ ↖ ↗ ↘

✓ ™ @ ✓ °C °F + ×

÷ = ≠ ¼ ½ ¾ ⅓ ⅔

! ? - — ∑ Ω i №

○ 😊 😄 😁 😊 😞 ☁ 🌙 🌻

🍷 🍺 🍔 🦷 🏠 🌲 🐵 🐱

🐾 👁 👍 👎 👤 👨 👩 👪 👤

💔 🍷 💪 💰 💾 📁 🗡 🔑

👉 🚧 🛠 🛒

## Create Image

Source:

Descriptio

Style:

Width:  Height:

Float:  px

Generate image

## Blank Page

```
Age: <input max="99" min="1" name="age" step="1" value="1" />
<select name="gender">
  <option selected="selected" value="male">Male</option>
  <option value="female">Female</option>
</select><br />
<input checked="checked" name="newsletter" type="checkbox" />
<textarea cols="20" name="comments" rows="5"></textarea>
<label><input name="terms" type="checkbox" value="terms" />
<input type="submit" value="Submit" />
</form>
```

## Gibberish text

Lorem ipsum

Cupcake ipsum

Random English

Chinese

Spanish

## Create iframe

URL:

Width:  Height:

Scrolling:  Border:

Generate iframe

## Create Table

Cols:  Rows:

Border:  Border Collapse:

Width:  Cellpadd:

Generate table

Div Tables

## Create Link

URL:

Text:

Title:

Target:

Generate link

## Create List

List type:

Unordered:  Ordered:

Generate list



```
<!DOCTYPE html>
<html Lang="en">
  <head>
    <meta charset="utf-8">
    <title>Page Title</title>
    <meta name="description" content="Roughly 155 c
    <link rel="stylesheet" type="text/css" href="my
    <script src="https://ajax.googleapis.com/ajax/l
    <script src="script.js"></script>
  </head>
  <body>
    <!-- Content -->
  </body>
</html>
```

[Download Package](#)

## Robots.txt

### Example

```
User-agent: *
Disallow: /dont-index-this-folder/
Sitemap: https://htmlcheatsheet.com/sitemap.xml
```

### Ban all robots

```
User-agent: *
Disallow: /
```

## Open Graph

```
<!doctype html>
<html xmlns:og="http://ogp.me/ns#">
<head>
<title>The Rock (1996)</title>
<meta property="og:title" content="Cheat Sheet" />
<meta property="og:type" content="website" />
<meta property="og:url" content="https://htmlcheats
<meta property="og:image" content="https://htmlchea
```

### Optional

```
<meta property="og:audio" content="https://htmlchea
<meta property="og:description" content="A brief de
<meta property="og:determiner" content="the" />
<meta property="og:locale" content="en_US" />
<meta property="og:locale:alternate" content="es_ES
<meta property="og:site_name" content="HTML CheatSh
<meta property="og:video" content="https://htmlchea
```

### Force HTTPS

```
RewriteEngine on
RewriteCond %{HTTPS} !on
RewriteRule (.*) https://%{HTTP_HOST}%{REQUEST_UR
```

### Force www

```
RewriteEngine on
RewriteCond %{HTTP_HOST} ^htmlg\.com [NC]
RewriteRule ^(.*)$ http://www.htmlg.com/$1 [L,R=30
```

### Force non-www

```
RewriteEngine on
RewriteCond %{HTTP_HOST} ^www\.htmlg\.com [NC]
RewriteRule ^(.*)$ http://htmlg.com/$1 [L,R=301]
```

### Custom Error Pages

```
ErrorDocument 500 "Sorry, something went wrong!"
ErrorDocument 401 https://htmlg.com/404/
ErrorDocument 404 404error.html
```

### Redirect Entire Site

```
Redirect 301 / https://htmlg.com/
```

### Permanent Page Redirect

```
Redirect 301 /oldlink.html https://htmlg.com/help,
Redirect 301 /oldlink https://htmlg.com/about/
```

### Alias Directory

```
RewriteEngine On
RewriteRule ^source_directory/(.*) target_director
```

### Remove .php Extension

```
RewriteEngine On
RewriteCond %{SCRIPT_FILENAME} !-d
RewriteRule ^([^.]+)$ $1.php [NC,L]
```

### Block IP Address

```
Order deny,allow
Allow from all
Deny from 123.123.123.123
Deny from 123.123.123.123
```

### Allow Access From Only One IP

```
# Require all denied
# Require ip 123.123.123.123
```

## Useful Links

### RGB color codes

- [Word doc to HTML](#)
- [Div tables](#)
- [HTML cleaner](#)
- [HTML blog](#)
- [HTML editor](#)
- [Basic concepts](#)
- [HTML validator](#)
- [W3 schools](#)

### Can I use?

# HTML 5 TAGS

Below is a full list of all the HTML 5 Tags / elements and a description of what they are used for.

## NEW HTML 5 TAGS

<code>&lt;article&gt;</code>	self-contained composition that is independently distributable	<code>&lt;menuitem&gt;</code>	specifies a command that a user can invoke from a popup menu
<code>&lt;aside&gt;</code>	section of page that consists of content tangentially related to content around it	<code>&lt;mark&gt;</code>	marked text
<code>&lt;audio&gt;</code>	sound content	<code>&lt;meter&gt;</code>	measurement in defined range
<code>&lt;bdi&gt;</code>	span of text to be isolated from surroundings for bidirectional formatting purposes	<code>&lt;nav&gt;</code>	navigation links
<code>&lt;canvas&gt;</code>	area that can be used to draw graphics via JavaScript	<code>&lt;output&gt;</code>	represents results of calculation
<code>&lt;command&gt;</code>	user invokable command	<code>&lt;progress&gt;</code>	progress of any kind of task
<code>&lt;datalist&gt;</code>	dropdown list	<code>&lt;rb&gt;</code>	marks the base text component of a ruby annotation.
<code>&lt;datatemplate&gt;</code>	data template	<code>&lt;rp&gt;</code>	parenthesized ruby text
<code>&lt;details&gt;</code>	details of an element	<code>&lt;rt&gt;</code>	ruby text
<code>&lt;dialog&gt;</code>	specifies that part of an application is interactive	<code>&lt;rtc&gt;</code>	marks a ruby text container for ruby text components in a ruby annotation
<code>&lt;embed&gt;</code>	embedded content	<code>&lt;ruby&gt;</code>	ruby annotations
<code>&lt;figcaption&gt;</code>	caption of figure element	<code>&lt;section&gt;</code>	section in a document
<code>&lt;figure&gt;</code>	group of media content	<code>&lt;source&gt;</code>	media resources
<code>&lt;footer&gt;</code>	footer for section or page	<code>&lt;summary&gt;</code>	header of a detail element
<code>&lt;header&gt;</code>	header for section or page	<code>&lt;template&gt;</code>	declares HTML fragments that can be cloned and inserted in the document by script.
<code>&lt;hgroup&gt;</code>	group of headings for section	<code>&lt;time&gt;</code>	date/time
<code>&lt;keygen&gt;</code>	generated key in a form	<code>&lt;track&gt;</code>	specifies a text track for media such as video and audio
<code>&lt;main&gt;</code>	specifies the main content area of an HTML document	<code>&lt;video&gt;</code>	video
		<code>&lt;wbr&gt;</code>	possible line break

## OLD UNSUPPORTED TAGS

<code>&lt;acronym&gt;</code>	acronym	<code>&lt;isindex&gt;</code>	provides searchable index related to current document
<code>&lt;applet&gt;</code>	applet	<code>&lt;dir&gt;</code>	directory list
<code>&lt;basefont&gt;</code>	base font	<code>&lt;noembed&gt;</code>	no embed section
<code>&lt;bgsound&gt;</code>	background sound	<code>&lt;noframes&gt;</code>	no frame section
<code>&lt;big&gt;</code>	big text	<code>&lt;s&gt;</code>	strikethrough text
<code>&lt;center&gt;</code>	centered text	<code>&lt;strike&gt;</code>	strikethrough text
<code>&lt;fn&gt;</code>	footnotes	<code>&lt;tt&gt;</code>	teletype text
<code>&lt;font&gt;</code>	text font, size, and color	<code>&lt;u&gt;</code>	underlined text
<code>&lt;frame&gt;</code>	sub window	<code>&lt;xmp&gt;</code>	preformatted text
<code>&lt;frameset&gt;</code>	set of frames		

## EXISTING HTML 5 TAGS

<!--...-->	comment	<label>	label for a form control
<!doctype>	document type	<legend>	title in a fieldset
<a>	hyperlink	<li>	list item
<abbr>	abbreviation	<link>	resource reference
<address>	address element	<map>	image map
<area>	Image map area	<menu>	menu list
<b>	bold text	<meta>	meta information
<base>	base URL for all links in page relative to document root	<noscript>	no script section
<bdo>	text direction	<object>	embedded object
<blockquote>	long quotation	<ol>	ordered list
<body>	body element	<optgroup>	option group
 	single line break	<option>	option in a drop-down list
<button>	push button	<p>	paragraph
<caption>	table caption	<param>	parameter for an object
<cite>	citation	<pre>	preformatted object
<code>	code text	<q>	short quotation
<col>	attributes for columns	<samp>	sample computer code
<colgroup>	groups of columns	<script>	script
<dd>	definition of description	<select>	selectable list
<del>	deleted text	<small>	small text
<div>	generic block-level element	<span>	inline generic container
<dfn>	defining instance of a term	<strong>	strong text
<dl>	definition list	<style>	style definition
<dt>	definition term	<sub>	subscripted text
<em>	emphasized text	<sup>	superscripted text
<fieldset>	logically group items in a form	<table>	table
<form>	defines a form	<tbody>	table body
<h1> to <h6>	header 1 to header 6	<td>	table cell
<head>	document information	<textarea>	text area
<hr>	horizontal rule	<tfoot>	table footer
<html>	html document	<th>	table header
<i>	italic text	<thead>	wraps row containing table headers
<iframe>	inline sub window	<title>	document title
<img>	image	<tr>	table row
<input>	input field	<ul>	unordered list
<ins>	inserted text	<var>	variable
<kbd>	keyboard text		

## Existing Attributes In HTML4 & 5

A list of existing attributes that are support in both HTML4 & HTML 5

<b>onabort</b>	abort event	<b>onmouseout</b>	mouse cursor exited event
<b>onbeforeunload</b>	before unload event	<b>onmouseover</b>	mouse cursor entered event
<b>onclick</b>	mouse clicked event	<b>onmouseup</b>	mouse button released event
<b>ondblclick</b>	mouse double-clicked event	<b>onmove</b>	move event
<b>ongesturechange</b>	gesture change event	<b>onorientationchange</b>	device orientation changed event
<b>ongestureend</b>	end of gesture event	<b>onpaste</b>	paste event
<b>ongesturestart</b>	start of gesture event	<b>onresize</b>	resize event
<b>onkeydown</b>	key pressed down event	<b>ontouchcancel</b>	touch canceled event
<b>onkeypress</b>	key pressed event	<b>ontouchend</b>	touch ended event
<b>onkeyup</b>	key released event	<b>ontouchmove</b>	touch move event
<b>onmousedown</b>	mouse button pressed down event	<b>ontouchstart</b>	touch started event
<b>onmousemove</b>	mouse moved event		

## Event Handler Content Attributes

HTML 4 added the ability to let events trigger actions in a browser, like starting a JavaScript when a user clicks on an element.

Below are event handler attributes that can be added to HTML elements to define event actions.

<b>onbeforeonload</b>	before onload event	<b>onloadeddata</b>	media data is loaded
<b>oncanplay</b>	media can start play	<b>onloadedmetadata</b>	duration of media element is loaded
<b>oncanplaythrough</b>	media can be played to end	<b>onloadstart</b>	browser starts to load media data
<b>oncontextmenu</b>	context menu is triggered	<b>onmessage</b>	element is invalid
<b>ondrag</b>	element is dragged	<b>onmousewheel</b>	mouse wheel is being rotated
<b>ondragend</b>	at the end of drag operation	<b>onpause</b>	media data is paused
<b>ondragenter</b>	element dragged on drop target	<b>onplay</b>	media data is going to start playing
<b>ondragleave</b>	element leaves valid drop target	<b>onplaying</b>	media data has started playing
<b>ondragover</b>	element is dragged over drop target	<b>onprogress</b>	browser is fetching media data
<b>ondragstart</b>	at the start of drag operation	<b>onratechange</b>	media data's pplaying rate has changed
<b>ondrop</b>	dragged element is being dropped	<b>onscroll</b>	element's scrollbar is being scrolled
<b>ondurationchange</b>	length of media is changed	<b>onseeked</b>	element's seeking attribute is not true
<b>onemptied</b>	media resource element becomes empty	<b>onseeking</b>	element's seeking attribute is true
<b>onended</b>	media has reached end	<b>onstalled</b>	there is error in fetching media data
<b>onerror</b>	error occurs	<b>onstorage</b>	document loads
<b>onformchange</b>	form changes	<b>onsuspend</b>	browser has stopped fetching media data
<b>onforminput</b>	form gets user input	<b>ontimeupdate</b>	media changes its playing position
<b>onhaschange</b>	document has changed	<b>onvolumechange</b>	media changes volume, also when mute
<b>oninput</b>	message is triggered	<b>onwaiting</b>	media has stopped playing
<b>oninvalid</b>	element gets user input		

# HTML 5 Canvas

The HTML <canvas> element is used to draw graphics, on the fly, via scripting (usually JavaScript). The <canvas> element is only a container for graphics. You must use a script to actually draw the graphics.

Canvas has several methods for drawing paths, boxes, circles, text, and adding images.

## CANVAS ELEMENT

### ATTRIBUTES

Name	Type	Default
Width	Unsigned Long	300
Height	Unsigned Long	150

### METHODS

Return	Name
String	toDataURL([Optional] string type, [Variadic] any args)
Object	getContext9 string contextId)

## 2D CONTEXT

### ATTRIBUTES

Name	Type
Canvas	HTMLCanvasObject [readonly]

### METHODS

Return	Name
Void	save()
Void	restore()

## TRANSFORMATION

### METHODS

Return	Name
Void	scale(float x, float y)
Void	rotate (float x, float y)
Void	transform(float m11, float m12, float m21, float m22, float dx, float dy)
Void	setTransform(float m11, float m12, float m21, float m22, float dx, float dy)

## IMAGE DRAWING

### METHODS

Return	Name
Void	drawImage(Object image, float dx, float dy, [Optional] float dw, float dh)

Argument "image" can be of type HTMLImageElement, HTMLCanvasElement or HTMLVideoElement

Void	drawImage(float sx, float sy, float sw, float sh, float dx, float dy, float dw, float dh)
------	---

## COLORS, STYLES AND SHADOWS

### ATTRIBUTES

Name	Type	Default
strokeStyle	any	black
fillStyle	any	black
shadowOffsetX	float	0.0
shadowOffsetY	float	0.0
shadowBlur	float	0.0
shadowColor	string	transparent black

### METHODS

Return	Name
CanvasGradient	createLinearGradient(float x0, float y0, float r0, float x1, float y1, float r1)
CanvasPattern	createPattern(Object image, string repetition)

Argument "image" can be of type HTMLImageElement, HTMLCanvasElement or HTMLVideoElement  
"repetition" supports any of the following values: [repeat (default), repeat-x, repeat-y, no-repeat]

### CANVASGRADIENT INTERFACE

Return	Name
Void	addColorStop(float offset, string color)

# HTML 5 Canvas

The HTML <canvas> element is used to draw graphics, on the fly, via scripting (usually JavaScript). The <canvas> element is only a container for graphics. You must use a script to actually draw the graphics.

Canvas has several methods for drawing paths, boxes, circles, text, and adding images.

## PATHS

### METHODS

Return	Name
Void	beginPath()
Void	closePath()
Void	fill()
Void	stroke()
Void	clip()
Void	moveTo( float x, float y)
Void	lineTo( float x, float y)
Void	quadraticCurveTo( float cpx, float cpy, float x, float y)
Void	bezierCurveTo( float cp1x, float cp1y, float cp2x, float cp2y, float x, float y)
Void	arcTo( float x1, float y1, float x2, float y2, float radius )
Void	arc( float x1, float y1, float radius ) float startAngle, float endAngle, boolean anticlockwise )
Void	rect( float x, float w, float h) isPointInPath( float x, float y)

## PIXEL MANIPULATION

### METHODS

Return	Name
ImageData	createImageData( float sw, float sh)
ImageData	createImageData( ImageData)
ImageData	getImageData( ImageData imagedata, float dx, float dy, [Optional] float dirtyX, float dirtyY, float dirtyWidth, float dirtyHeight)

### ImageData interface

width	unsigned long	[readonly]
height	unsigned long	[readonly]
data	CanvasPixelArray	[readonly]

## TEXT

### ATTRIBUTES

Name	Type	Default
font	String	10px sans-serif
textAlign	String	start

Supports any of the following values:  
[start, end, left, right, center]

textBaseline String alphabetic

Supports an of the following values:  
[top, hanging, middle, alphabetic, ideographic, bottom]

### METHODS

Return	Name
Void	fillText( string text, float x, float y, [Optional] float maxWidth)
Void	strokeText( string text, float x, float y, [Optional] float maxWidth)
TextMetrics	measureText( string text)

### TextMetrics interface

width float [readonly]

## RECTANGLES

### METHODS

Return	Name
Void	clearRect( float x, float y, float w, float h)
Void	fillRect( float x, float y, float w, float h)
Void	strokeRect( float x, float y, float w, float h)

# HTML 5 Canvas

The HTML <canvas> element is used to draw graphics, on the fly, via scripting (usually JavaScript). The <canvas> element is only a container for graphics. You must use a script to actually draw the graphics.

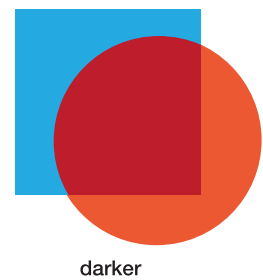
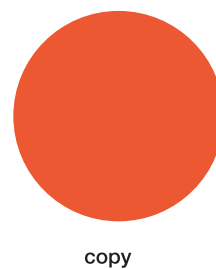
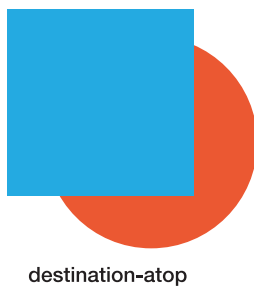
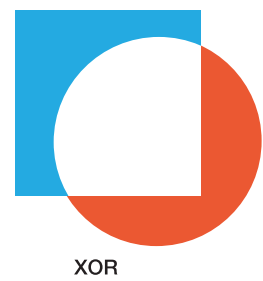
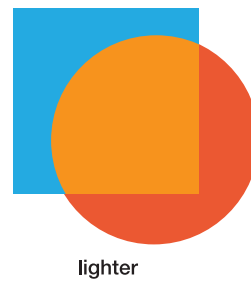
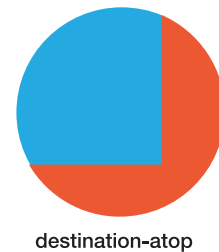
Canvas has several methods for drawing paths, boxes, circles, text, and adding images.

## CANVAS ELEMENT

### ATTRIBUTES

Name	Type	Default
globalAlpha	Float	1
globalCompositeOperation	String	Source-over

Supports any of the following values:





# HTML 5 Canvas

The HTML <canvas> element is used to draw graphics, on the fly, via scripting (usually JavaScript). The <canvas> element is only a container for graphics. You must use a script to actually draw the graphics.

Canvas has several methods for drawing paths, boxes, circles, text, and adding images.

## LINE STYLES

### ATTRIBUTES

Name	Type	Default
lineWidth	Float	1
lineCap	String	Butt

Supports any of the following values:

butt



rounded



square



lineJoin

String

miter

Supports any of the following values:

round



bevel



miter



miterLimit

Float

10

html-cheat-sheet.html

# Beginner's HTML CHEAT SHEET

## Document Summary

**<html>** ... **</html>**

The tag shows up at the beginning and end of an HTML document (known as the root element). It indicates that the webpage is written in HTML5, and all other page markup comes in between these beginning and ending tags.

**<head>** ... **</head>**

The contains information that specific page, including the title tags, meta data, and links to scripts and style sheets.

**<title>** ... **</title>**

The title tag is the title for that page, useful for both search engines (when they scan and index pages) and users (showing up up in a browser's title bar) by explicitly stating the primary topic of each page.

**<body>** ... **</body>**

Body tags include all content that will be shown to users, including everything they'll see & read.

### Example

```
<html>
  <head>
    <title>My Beautiful Website</title>
  </head>
  <body>
    ...
  </body>
</html>
```

## Document Information

**<base/>**

The Base URL (for example: www.YourSite.com) is useful for specifying all relative links in a document (especially if you have many internal links).

**<meta/>**

Meta data spells out additional information about the page, including the page's description, author, published date, keywords and other typically 'hidden' page information.

**<link/>**

Can be used to a create relationships with external pages or documents, including style sheets.

**<style>** ... **</style>**

This element includes document style information, typically defaulting to CSS.

**<script>** ... **</script>**

This element includes all scripting information, or links to external scripts. You can also include this element in the body to dynamically generate content.

### Example

```
<html>
  <head>
    <meta charset="utf-8">
    <base href="http://mybeautifulwebsite.com" target="_blank" />
    <title>My Beautiful Website</title>
    <link rel="stylesheet" href="/css/master.css">
    <script type="text/javascript">
      var MyVar = 0;
```

```

63     var myVal = 0;
64     </script>
65 </head>
66 <body>
67
68     </body>
69 </html>

```

## Document Structure

**<h1..h6> ... </h1..h6>**

All six levels of Headings, with 1 being the most important on a page and 6 being the least. These elements are used to describe content sections on a page.

**<div> ... </div>**

A generic container used to denote a page section or

**<span> ... </span>**

An inline section or block container, typically used for grouping styling elements.

**<p> ... </p>**

This foundational tag is used to organize paragraphs of text.

**<br/>**

Creates a line break (or old-school carriage-return), useful for writing blocks of text that need to be on different lines (think addresses, etc.)

**<hr/>**

Creates a horizontal rule, a sectional break in an HTML page. Typically used to denote a change in topic or section of a page.

### Example

```

90 <div>

```

```

91   <h1>Ways to make your cat happy</h1>
92   <p>You have a <span>mini-lion</span> at
93   home and you want to make it as happy as pos-
94   sible.</p>

```

```

95   <hr/>

```

```

96   <h2>Feed your cat well.</h2>

```

```

97   <p>The right diet is <span>extremely</span>
98   important for the wellbeing of your cat.<br/>
99   Obesity is a common source of problems among
100  domesticated animals.</p>
101 </div>

```

## Ways to make your cat happy

You have a **mini-lion** at home and you want to make it as happy as possible.

### Feed your cat well.

The right diet is **extremely** important for the wellbeing of your cat.

Obesity is a common source of problems among domesticated animals.

## Text (+ formatting)

**<strong> ... </strong>**

Exactly like it sounds, indicating STRONG emphasis, displayed bold in most browsers.

**<b> ... </b>**

Another way to create bold text, however it's more for drawing attention as opposed to emphasizing extra emphasis like the previous tag.

**<em> ... </em>**

The emphasis tags also are like they sound, emphasizing text or phrases displayed as italics in most browsers.

**<i> ... </i>**

Another way to add italics to text, however without the added emphasis (similar to the bold tags above) and instead used to denote things like thoughts or names.

**<tt> ... </tt>**

An older tag used to display fixed-width, typewriter-esque text. No longer supported in HTML5.

**<q> ... </q>**

These are used for shorter quotations.

**<abbr> ... </abbr>**

Abbreviation These help denote abbreviations, while also making the full form available.

**<acronym> ... </acronym>**

Acronym text Similar to the previous abbreviation tag, but this time with acronyms. Not support in HTML5.

**<address> ... </address>**

A helpfully obvious tag that displays the author's contact information.

**<dfn> ... </dfn>**

Definition Used to create an inline definition in the body text.

**<code> ... </code>**

Code text Displays code snippets (like the one over to the left), typically showing up monospaced.

128	<code>&lt;strike&gt; ... &lt;/strike&gt;</code>	<code>&lt;font&gt; ... &lt;/font&gt;</code>
129	This tag creates strike through text (or text with a line through it). Another older tag not commonly supported.	Font tag Old-school way to colour fonts. No longer part of HTML5.
130		
131		
132	<code>&lt;cite&gt; ... &lt;/cite&gt;</code>	<code>&lt;sub&gt; ... &lt;/sub&gt;</code>
133	Tags used to cite or reference information, useful for quotes and statements in a document.	Subscript text A helpful way to subscript text (shrinking it and placing it a half line lower than the previous text).
134		
135		
136	<code>&lt;pre&gt; ... &lt;/pre&gt;</code>	<code>&lt;sup&gt; ... &lt;/sup&gt;</code>
137	Pre-formatted, 'monospace' text laid out with whitespace inside the element intact.	Superscript text Similar to the previous subscript, however this time placing it a half line higher than the previous text.
138		
139		
140	<code>&lt;del&gt; ... &lt;/del&gt;</code>	<code>&lt;small&gt; ... &lt;/small&gt;</code>
141	Helps denote a previously deleted section of text.	Small size text Historically used to simply reduce text size, in HTML5 it also refers to information that may no longer be valid, accurate or relevant.
142		
143	<code>&lt;ins&gt; ... &lt;/ins&gt;</code>	<code>&lt;bdo&gt; ... &lt;/bdo&gt;</code>
144	Represents a section of text that's been inserted into the document.	The bi-directional override tag will identify which should text should be read opposite from the preceeding text.
145		
146		
147	<code>&lt;blockquote&gt; ... &lt;/blockquote&gt;</code>	
148	Reserved for long paragraphs of quotations, often cited.	
149		
150		

### Example

```

151
152
153 <p><strong>Keep your cat healthy.</strong>
154 Felines can catch a <em>variety of ill-
155 nesses</em> outside and inside the house. It
156 is important to bring your cat to the veteri-
157 narian at least <strike>one</strike> two
158 times a year for a check-up.</p>
159
160 <blockquote>
161     Your cat should be neutered to prevent un-
162 wanted babies and lower the risks of feline
163 HIV.
164     <cite>- Dr. Tac Nam. Cat Professor</cite>
165 </blockquote>
166 <pre>
167     This is a preformatted text which is a
168 block type element.
169 </pre>
170 <p>While this line has a <code>code
171 tag</code> in it, codes that can be embedded
172 inline.</p>
173

```

Keep your cat healthy. Felines can catch a variety of illnesses outside and inside the house. It is important to bring your cat to the veterinarian at least two times a year for a check-up.

*Your cat should be neutered to prevent unwanted babies and lower the risks of feline HIV.*

- Dr. Tac Nam. Cat Professor

This is a preformatted text which is a block type element.

While this line has a `code tag` in it, codes that can be embedded inline.

### Links (+ formatting)

174		
175		
176	<code>&lt;a href=""&gt; ... &lt;/a&gt;</code>	<code>&lt;a name="name"&gt; ... &lt;/a&gt;</code>
177	Anchor text for hyperlink.	An anchor that's useful for bringing users to specific document elements.
178		
179	<code>&lt;a href="mailto:"&gt; ... &lt;/a&gt;</code>	<code>&lt;a href="#name"&gt; ... &lt;/a&gt;</code>
180	A link used to pull up an outgoing message to a specific email addresses.	An anchor link that brings users specifically to a div element.
181		
182		
183	<code>&lt;a href="tel://###-###"&gt; ... &lt;/a&gt;</code>	
184	A link to make phone numbers clickable, especially useful for mobile users.	
185		
186		
187		

### Images (+ formatting)

188		
189		
190	<code>&lt;img /&gt;</code>	<code>border=""</code>
191	An image tag to include and display image files.	Explains the border thickness (if any).
192		

193	<b>src="url"</b>	<b>vspace=""</b>
194	Exactly like it sounds. The URL or file of the image to display.	Denote spacing on top or the bottom of image.
195		
196		<b>hspace=""</b>
197	<b>alt="text"</b>	Denote spacing on top or the bottom of image.
198	Alternative text that helps explain the image content to both search engines and users.	<b>&lt;map&gt; ... &lt;/map&gt;</b>
199		Helps you tell users that this is an interactive image with clickable areas.
200		
201	<b>height=""</b>	<b>&lt;map name=""&gt; ... &lt;/map&gt;</b>
202	The ability to specify image height in pixels or percentages.	Name of the map associated between the image and the map.
203		
204	<b>width=""</b>	<b>&lt;area /&gt;</b>
205	The ability to specify width in pixels or percentages.	Specify the area of image map.
206		
207	<b>align=""</b>	
208	The alignment of the image (relative to other text elements on the page).	
209		

### Example

```

212 
213
214 <map name="planetmap">
215   <area shape="rect" coords="0,0,82,126" href="sun.htm" alt="Sun">
216   <area shape="circle" coords="90,58,3" href="mercur.htm" alt="Mercury">
217   <area shape="circle" coords="124,58,8" href="venus.htm" alt="Venus">
218 </map>
219

```

## Lists (+ formatting)

223	<b>&lt;ol&gt; ... &lt;/ol&gt;</b>	<b>&lt;dl&gt; ... &lt;/dl&gt;</b>
224	Create numbered (ordered) lists showing sequential order, preference or priority.	Reserved specifically for list items definitions.
225		
226		<b>&lt;dt&gt; ... &lt;/dt&gt;</b>
227	<b>&lt;ul&gt; ... &lt;/ul&gt;</b>	The definition of a single term inline with body content.
228	Display a bulleted (unordered) list without any extra emphasis on order of importance.	<b>&lt;dd&gt; ... &lt;/dd&gt;</b>
229		The description for the defined term.
230		
231	<b>&lt;li&gt; ... &lt;/li&gt;</b>	
232	Specifies each list item to be bulleted or numbered.	
233		

### Example

```

235 <ol>
236   <li>January</li>
237   <li>February</li>
238   <li>March</li>
239 </ol>
240
241 <ul>
242   <li>Tomato</li>
243   <li>Lettuce</li>
244   <li>Cheese</li>
245 </ul>
246
247 <dl>
248   <dt>Coffee</dt>
249   <dd>Black hot drink</dd>
250   <dt>Milk</dt>
251   <dd>White cold drink</dd>
252 </dl>

```

## Forms (formatting and attributes)

251	<b>&lt;form&gt; ... &lt;/form&gt;</b>	<b>accept-charset</b>
252	The form element creates a form, spelling out how the form will operate based on it's attributes.	Identifies the character encodings upon the form submission.
253		
254		<b>target</b>
255	<b>action="url"</b>	Tells where to display the form response after being submitted, generally one of the following: <code>_blank</code> , <code>_self</code> , <code>_par</code>
256	The form action URL specifies where data is to be sent when a site visitor submits the form.	
257		

258		ent, _top
259	<b>method=""</b>	
260	The method attribute refers to the HTTP method (Get, Post), which dictates how to send the form data.	<b>&lt;fieldset&gt; ... &lt;/fieldset&gt;</b>
261		Identifies the group of all fields on the form.
262		
263	<b>enctype=""</b>	<b>&lt;label&gt; ... &lt;/label&gt;</b>
264	This attribute dictates how the form-data is to be encoded when submitting information back to the web sever (for method="post" only).	A simple field label, telling the user what to enter in each field.
265		
266		
267	<b>autocomplete</b>	<b>&lt;legend&gt; ... &lt;/legend&gt;</b>
268	Dictates whether a form should have autocomplete on or off.	The form legend acts as a caption for the fieldset element.
269		
270		<b>&lt;input /&gt;</b>
271	<b>novalidate</b>	The form input attribute defines the type of field information to receive from a user.
272	Dictates whether the form should not be validated when submitted.	
273		
274		

### Input type attributes

277	<b>type=""</b>	<b>placeholder=""</b>
278	Specifies the field input type, typically including text, password, date-time, checkbox, password submit etc.	Provides a helpful hint to the user, describing what the <input> element value should be.
279		
280		
281	<b>name=""</b>	<b>pattern=""</b>
282	Describes the name of the form.	Identifies a regular expression that the <input> element gets checked against, making sure the user entered the correct information.
283		
284	<b>value=""</b>	
285	Describes the value or input field information.	
286		<b>min=""</b>
287	<b>size=""</b>	The minimum value allowed for each <input> element.
288	Specifies the input element width in characters.	
289		<b>max=""</b>
290	<b>maxlength=""</b>	The maximum value allowed for each <input> element.
291	Identifies the maximum input element character numbers allowed.	<b>autofocus</b>
292		Says to make sure that the <input> element comes into focus after the page loads.
293	<b>required</b>	
294	Another helpfully explicit tag, making sure the <input> element is completely filled out prior to the user submitting the form.	<b>disabled</b>
295		Disables an <input> element on the form.
296		
297		
298	<b>step=""</b>	<b>&lt;textarea&gt; ... &lt;/textarea&gt;</b>
299	Identifies the legal number intervals for an input field.	Specifies a large text input for longer messages.
300		
301	<b>width=""</b>	<b>&lt;select&gt; ... &lt;/select&gt;</b>
302	Specifies the width (in pixels) of an <input> element.	Describes a drop-down box for user's to select one from a variety of options.
303		
304	<b>height=""</b>	
305	Dictates the height (again, in pixels) of an <input> element.	
306		

### Select attributes

309	<b>name=""</b>	<b>autofocus</b>
310	The name for a drop down combination box.	Specifies that a drop-down list automatically comes into focus after a page loads.
311		
312	<b>size=""</b>	<b>&lt;optgroup&gt; ... &lt;/optgroup&gt;</b>
313	Specifies the number of available, visible options in a drop-down.	Specifies the entire grouping of available options.
314		
315		
316	<b>multiple</b>	<b>&lt;option&gt; ... &lt;/option&gt;</b>
317	Allows for multiple selections to be made at one time.	Defines one of the available options in the drop-down list.
318		
319	<b>required</b>	
320	Requires that a value is selected before a user can submit a form.	
321		
322		

### Option attributes

### Option attributes

- value=""** `<button> ... </button>`  
Explains the option value available for selection. Defines the clickable button for users to submit options.
- selected**  
Defines the default selected option for users.

### Example

```

331 <form action="action_page.php" method="post">
332 <fieldset>
333 <legend>Personal information:</legend>
334 First name:<br>
335 <input type="text" name="firstname" value="Mickey" placeholder="First Name"><br>
336 Last name:<br>
337 <input type="text" name="lastname" value="Mouse" placeholder="Last Name"><br><br>
338 Favorite car brand:<br>
339 <select>
340 <option value="volvo">Volvo</option>
341 <option value="saab">Saab</option>
342 <option value="mercedes">Mercedes</option>
343 <option value="audi">Audi</option>
344 </select>
345 <textarea name="description"></textarea>
346 <input type="submit" value="Submit">
347 </fieldset>
348 </form>

```

### Tables (+ formatting)

- <table> ... </table>** `<tr> ... </tr>`  
The table tag identifies and contains all table related content. Contains the information to be included in a single row of the table.
- <caption> ... </caption>** `<th> ... </th>`  
The caption is a description of what the table is, and what it contains. Contains the actual information or data in a single header item.
- <thead> ... </thead>** `<td> ... </td>`  
The table headers describe the type of information contained in each column underneath. Contains the actual information or data in a single table cell.
- <tbody> ... </tbody>** `<colgroup> ... </colgroup>`  
The table body contains the table's data or information. Groups a single (or multiple) columns for formatting purposes.
- <tfoot> ... </tfoot>** `<col />`  
Table footers describe all footer content. Defines a single column of information inside a table.

### Example

```

373 <table>
374 <colgroup>
375 <col span="2" style="background-color:#181a1f">
376 <col style="background-color:#2c323c">
377 </colgroup>
378 <tr>
379 <th>ISBN</th>
380 <th>Title</th>
381 <th>Price</th>
382 </tr>
383 <tr>
384 <td>3476896</td>
385 <td>My first HTML</td>
386 <td>$53</td>

```

ISBN	TITLE	PRICE
3476896	My first HTML	\$53
5869207	My first CSS	\$49

```

388 </tr>
389 <tr>
390 <td>5869207</td>
391 <td>My first CSS</td>
392 <td>$49</td>
393 </tr>
394 </table>

```

## Objects and iFrames

**<object> ... </object>**

The object tag describes an embedded file type, including audio, video, PDFs, additional pages, and more.

**height=""**

Describes the height of the object in pixels.

**width=""**

Describes the width of the object in pixels.

**type=""**

Describes which media type the data contains.

**usemap=""**

This is the name of a client-side image map within <object>.

**<iframe> ... </iframe>**

Contains an inline frame that allows you to embed external information into an existing document.

**iFrame Attributes**

**name=""**

The name of the <iframe>.

**src=""**

The URL source of the original document to embed inside the <iframe>.

**srcdoc=""**

This contains the actual HTML content to display inside the <iframe> on the current page.

**width=""**

Defines the width of your <iframe>.

**<param />**

Appending extra parameters help you customize the iframe content.

**<embed> ... </embed>**

The embed tag acts as a container for another external application or additional plug-in.

### Embed Attributes

**height=""**

Defines the height of the embedded content you're including.

**width=""**

Similar to previous, but this time defining the width of the embedded content.

**src=""**

The source URL of the external file you're embedding.

**type=""**

Describes the media type of embedded content to include.

### Example

```

<object width="400" height="400"></object>
<iframe src="demo_iframe.htm" width="200" height="200"></iframe>
<embed src="helloworld.swf" width="200" height="200"></embed>

```

## HTML5 New Tags

**<header> ... </header>**

Defines the header block for a document (or individual section).

**<footer> ... </footer>**

Identifies the footer block for the document (or an individual section).

**<main> ... </main>**

Describes the main content of a document.

**<article> ... </article>**

Identifies an article inside a document.

**<nav> ... </nav>**

Navigation links for the user in a document.

**<menuitem> ... </menuitem>**

The specific menu item that a user can raise from a popup menu.

**<meter> ... </meter>**

Describes the scalar measurement within a known array.

**<progress> ... </progress>**

Displays the progress of a task, usually used for progress bar.



452			
453	<b>&lt;aside&gt;</b> ... <b>&lt;/aside&gt;</b>	<b>&lt;rp&gt;</b> ... <b>&lt;/rp&gt;</b>	Displays text within browsers that do not support ruby annotations.
454	Specifies content contained in a document sidebar.		
455			
456	<b>&lt;section&gt;</b> ... <b>&lt;/section&gt;</b>	<b>&lt;rt&gt;</b> ... <b>&lt;/rt&gt;</b>	Displays East Asian typography character details.
457	Specifies a section block in the document.		
458			
459	<b>&lt;details&gt;</b> ... <b>&lt;/details&gt;</b>	<b>&lt;ruby&gt;</b> ... <b>&lt;/ruby&gt;</b>	Describes a Ruby annotation for East Asian typography.
460	Describes additional facts or information that the user can view or hide.		
461			
462	<b>&lt;dialog&gt;</b> ... <b>&lt;/dialog&gt;</b>	<b>&lt;summary&gt;</b> ... <b>&lt;/summary&gt;</b>	Contains a visible heading for a <details> element.
463	A dialog box or window.		
464			
465	<b>&lt;figcaption&gt;</b> ... <b>&lt;/figcaption&gt;</b>	<b>&lt;bdi&gt;</b> ... <b>&lt;/bdi&gt;</b>	Helps you format part of text in a different direction from other text.
466	The <figure> element caption that helps describes the figure.		
467			
468			
469	<b>&lt;figure&gt;</b> ... <b>&lt;/figure&gt;</b>	<b>&lt;time&gt;</b> ... <b>&lt;/time&gt;</b>	Identifies the time and date.
470	An independent content block featuring diagrams, photos, illustrations or more.		
471			
472			
473	<b>&lt;mark&gt;</b> ... <b>&lt;/mark&gt;</b>	<b>&lt;br&gt;</b>	A line-break within the content.
474	Displays a portion of highlighted text within the page content.		
475			
476			
477			

### Collective Character Objects

478			
479			
480	<b>&amp;#34;</b> <b>&amp;quot;</b> ; Quotation Marks - "	<b>&amp;#169;</b> <b>&amp;copy;</b> ; Copyright symbol - ©	
481			
482	<b>&amp;#38;</b> <b>&amp;amp;</b> ; Ampersand - &	<b>&amp;#64;</b> <b>&amp;Uuml;</b> ; @ Symbol - @	
483			
484	<b>&amp;#60;</b> <b>&amp;lt;</b> ; Less than sign - <	<b>&amp;#149;</b> <b>&amp;ouml;</b> ; Small bullet - •	
485			
486	<b>&amp;#62;</b> <b>&amp;gt;</b> ; Greater than sign - >	<b>&amp;#153;</b> <b>&amp;ucirc;</b> ; Trademark symbol - ™	
487	<b>&amp;#160;</b> <b>&amp;nbsp;</b> ; Non-breaking space		
488			
489			
490			
491			
492			Created by the folks at <a href="#">Hosting Facts</a>
493			

# 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 Sectioning HTML Elements</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

# HTML5 CHEAT SHEET



TAGS	MEANING	TAGS	MEANING	TAGS	MEANING
<a>	hyperlink	<h1>	heading level 1	<progress>	progress of a task
<abbr>	abbreviation	<h2>	heading level 2	<q>	short quotation
<address>	address element	<h2>	heading level 2	<rb>	marks the base text component of a ruby annotation.
<area>	area inside an image map	<h3>	heading level 3	<rp>	used for the benefit of browsers that don't support ruby annotations
<article>	article	<h4>	heading level 4		
		<h5>	heading level 5		
<aside>	content aside from the page content	<h6>	heading level 6	<rt>	ruby text component
<audio>	sound content	<head>	information about the document	<rtc>	marks a ruby text container for ruby text components in a ruby annotation.
<b>	bold text	<header>	group of introductory or navigational aids	<ruby>	ruby annotation
<base>	base URL for all the links in a page	<hr>	horizontal rule	<s>	Indicates text that's no longer accurate or relevant.
<bdi>	bi-directional text formatting	<html>	html document		
		<i>	italic text	<samp>	sample computer code
<bdo>	direction of text display	<iframe>	inline sub window (frame)	<script>	script
<blockquote>	long quotation	<img>	image	<section>	section
<body>	body element	<input>	input field	<select>	selectable list
 	single line break	<ins>	inserted text	<small>	small text
<button>	push button	<kbd>	keyboard text	<source>	media resources
<canvas>	define graphics				
<caption>	table caption	<keygen>	generates a key pair	<span>	section in a document
<cite>	citation	<label>	label for form control	<strong>	strong text
<code>	computer code text	<legend>	title in a fieldset	<style>	style definition
<col>	table columns	<li>	list item	<sub>	subscripted text
<colgroup>	groups of table columns	<link>	resource reference	<summary>	summary / caption for the <details> element
<data>	Allows machine-readable data to be provided	<main>	main content area of an HTML document.	<sup>	superscripted text
<datalist>	"autocomplete" dropdown list	<map>	image map	<table>	table
<dd>	definition description	<mark>	marked text	<tbody>	table body
<del>	deleted text	<menu>	menu list	<td>	table cell
<details>	details of an element	<menuitem>	command that user can invoke from popup menu	<textarea>	text area
				<tfoot>	table footer
<dfn>	definition term	<meta>	meta information	<th>	table header
<dialog>	part of an application is interactive.	<meter>	measurement within a predefined range	<thead>	table header
				<time>	date/time
<div>	section in a document, definition list.	<nav>	navigation links	<title>	document title
<dl>	definition list.	<noscript>	noscript section	<tr>	table row
<dt>	definition term	<object>	embedded object	<track>	text track for media such as video and audio
<em>	emphasized text	<ol>	ordered list	<u>	text with a non-textual annotation.
<embed>	external application or interactive content	<optgroup>	option group	<ul>	unordered list
<fieldset>	fieldset	<option>	option in a drop-down list	<var>	variable
<figcaption>	caption for the figure element.	<output>	types of output		
<figure>	group of media content, and their caption	<p>	paragraph	<video>	video
<footer>	footer section or page	<param>	parameter for an object		
<form>	specifies a form	<pre>	preformatted text	<wbr>	line break opportunity for very long words and strings of text with no spaces.

(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.