All HTML5 Elements

	Element	Description
Basic elements Basic elements are the backbone of any HTML document. You'll see these elements in the source code for all web pages, following the doctype declaration, which is on the first line of the page. The doctype specifies which version of (X)HTML that page is using. Elements comprising the contents of a Web page are placed between the opening <html> tag and the closing </html> tag. The <html> element is also known as the root element.	`<html>`	The **HTML `<html>` element** (or *HTML root element*) represents the root of an HTML document. All other elements must be descendants of this element.
Document metadata Metadata contains information about the page. This includes information about styles, scripts and data to help software (search engines, browsers, etc.) use and render the page. Metadata for styles and scripts may be defined in the page or link to another file that has the information.	`<base>`	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.
	`<head>`	The **HTML `<head>` element** provides general information (metadata) about the document, including its title and links to its scripts and style sheets.
	`<link>`	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.
	`<meta>`	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>`).
	`<style>`	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.
	`<title>`	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.
Content sectioning Content sectioning elements allow you to organize the document content into logical pieces. Use the sectioning elements to create a broad outline for your page content, including header and footer navigation, and heading elements to identify sections of content.	`<address>`	The **HTML `<address>` element** supplies contact information for its nearest `<article>` or `<body>` ancestor; in the latter case, it applies to the whole document.
	`<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). 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.
	`<aside>`	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.
	`<footer>`	The **HTML `<footer>` element** represents a footer for its nearestsectioning content or sectioning root element. A footer typically contains information about the author of the section, copyright data or links to related documents.
	`<header>`	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.
	`<h1>`,`<h2>`,`<h3>`, `<h4>`,`<h5>`, `<h6>`	**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(just like the fixed sider bar of this page on the right).
	`<hgroup>`	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.
	`<nav>`	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.
Text content Use HTML text content elements to organize blocks or sections of content placed between the opening <body> and closing </body> tags. Important for accessibility and SEO, these elements identify the purpose or structure of that content.	`<dd>`	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.
	`<div>`	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.
	`<dl>`	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).
	`<dt>`	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.

	Element	Description
	`<figcaption>`	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.
	`<figure>`	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.
	`<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 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.
	`<li>`	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.
	`<main>`	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).
	`<ol>`	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.
	`<p>`	The **HTML `<p>` element** (or *HTML Paragraph Element*) represents a paragraph of text.
	`<pre>`	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.
	`<ul>`	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.
Inline text semantics Use the HTML inline text semantic to define the meaning, structure, or style of a word, line, or any arbitrary piece of text.	`<a>`	The *HTML `Anchor` Element* (**`<a>`**) 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.
	`<abbr>`	The **HTML `<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.
	`<b>`	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.
	`<bdi>`	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.
	`<bdo>`	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.
	` `	The HTML element *line break* ` ` 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>`	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.
	`<code>`	The *HTML Code Element* (**`<code>`**) represents a fragment of computer code. By default, it is displayed in the browser's default monospace font.
	`<data>`	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.

	Element	Description
	`<dfn>`	The *HTML Definition Element* (**<dfn>**) represents the defining instance of a term.
	`<em>`	The HTML element *emphasis* **** marks text that has stress emphasis. The `<em>` element can be nested, with each level of nesting indicating a greater degree of emphasis.
	`<i>`	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.
	`<kbd>`	The *HTML Keyboard Input Element* (**<kbd>**) represents user input and produces an inline element displayed in the browser's default monospace font.
	`<mark>`	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.
	`<q>`	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.
	`<rp>`	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 Taiwainese 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.
	`<rt>`	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.
	`<rtc>`	The **HTML <rtc> Element** embraces semantic annotations of characters presented in a ruby of `<rb>` elements used inside of `<ruby>` element. `<rb>`elements can have both pronunciation (`<rt>`) and semantic (`<rtc>`) annotations.
	`<ruby>`	The **HTML <ruby> Element** represents a ruby annotation. Ruby annotations are for showing pronunciation of East Asian characters.
	`<s>`	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.
	`<samp>`	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).
	`<small>`	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.
	`<span>`	The **HTML ** 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>`	The HTML Strong Element (`<strong>`) gives text strong importance, and is typically displayed in bold.
	`<sub>`	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.
	`<sup>`	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.
	`<time>`	Technical review completed.
	`<u>`	The *HTML Underline Element* (`<u>`) renders text with an underline, a line under the baseline of its content.
	`<var>`	The HTML Variable Element (`<var>`) represents a variable in a mathematical expression or a programming context.
	`<wbr>`	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.
Image and multimedia HTML supports various multimedia resources such as images, audio, and video.	`<area>`	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.
	`<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.

	Element	Description
	`<map>`	The **HTML `<map>` element** is used with `<area>` elements to define an image map (a clickable link area).
	`<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.
	`<video>`	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.
Embedded content In addition to regular multimedia content, HTML can include a variety of other content, even if it's not always easy to interact with.	`<embed>`	The **HTML `<embed>` Element** represents an integration point for an external application or interactive content (in other words, a plug-in).
	`<object>`	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.
	`<param>`	The **HTML `<param>` Element** (or *HTML Parameter Element)* defines parameters for `<object>`.
	`<source>`	Editorial review completed.
Scripting In order to create dynamic content and Web applications, HTML supports the use of scripting languages, most prominently JavaScript. Certain elements support this capability.	`<canvas>`	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.
	`<noscript>`	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.
	`<script>`	The *HTML Script Element* (**<script>**) is used to embed or reference an executable script within an HTML or XHTML document.
Demarcating edits These elements let you provide indications that specific parts of the text have been altered.	`<del>`	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.
	`<ins>`	The **HTML `<ins>` Element** (or *HTML Inserted Text*) HTML represents a range of text that has been added to a document.
Table content The elements here are used to create and handle tabular data.	`<caption>`	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.
	`<col>`	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.
	`<colgroup>`	The *HTML Table Column Group Element* (**<colgroup>**) defines a group of columns within a table.
	`<table>`	The **HTML Table Element** (`<table>`) represents tabular data: information expressed via two dimensions or more.
	`<tbody>`	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.
	`<td>`	The *Table cell* HTML element (**<td>**) defines a cell of a table that contains data. It participates in the *table model*.
	`<tfoot>`	The *HTML Table Foot Element* (`<tfoot>`) defines a set of rows summarizing the columns of the table.
	`<th>`	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.
	`<thead>`	The *HTML Table Head Element* (`<thead>`) defines a set of rows defining the head of the columns of the table.
	`<tr>`	The HTML element *table row* `<tr>` defines a row of cells in a table. Those can be a mix of `<td>` and `<th>` elements.
Forms	`<button>`	Technical review completed.

	Element	Description
HTML provides a number of elements which can be used together to create forms which the user can fill out and submit to the Web site or application. There's a great deal of further information about this available in the HTML forms guide.	`<datalist>`	The *HTML Datalist Element* (**<datalist>**) contains a set of `<option>`elements that represent the values available for other controls.
	`<fieldset>`	The **HTML `<fieldset>` element** is used to group several controls as well as labels (`<label>`) within a web form.
	`<form>`	The **HTML `<form>` element** represents a document section that contains interactive controls to submit information to a web server.
	`<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.
	`<label>`	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.
	`<legend>`	The **HTML `<legend>` Element** (or *HTML Legend Field Element*) represents a caption for the content of its parent `<fieldset>`.
	`<meter>`	The **HTML `<meter>` Element** represents either a scalar value within a known range or a fractional value.
	`<optgroup>`	In a Web form, the **HTML `<optgroup>` element** creates a grouping of options within a `<select>` element.
	`<option>`	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.
	`<output>`	The **HTML `<output>` element** represents the result of a calculation or user action.
	`<progress>`	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.
	`<select>`	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.
	`<textarea>`	The **HTML `<textarea>` element** represents a multi-line plain-text editing control.
Interactive elements HTML offers a selection of elements which help to create interactive user interface objects.	`<details>`	The *HTML Details Element* (**<details>**) is used as a disclosure widget from which the user can retrieve additional information.
	`<dialog>`	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.
	`<menu>`	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.
	`<menuitem>`	The **HTML `<menuitem>` element** represents a command that a user is able to invoke through a popup menu. This includes context menus, as well as menus that might be attached to a menu button.
	`<summary>`	The HTML *summary* element (`<summary>`) is used as a summary, caption, or legend for the content of a `<details>` element.
Web Components Web Components is an HTML-related technology which makes it possible to, essentially, create and use custom elements as if it were regular HTML. In addition, you can even create custom versions of standard HTML elements, as well.	`<content>`	The **HTML `<content>`** element is used inside of Shadow DOM as aninsertion point. It is not intended to be used in ordinary HTML. It is used with Web Components. It has now been replaced by the **<slot>** element.
	`<element>`	The **HTML `<element>` element** is used to define new custom DOM elements.
	`<shadow>`	The **HTML `<shadow>`** element is used as a shadow DOM insertion point. You might use it if you have created multiple shadow roots under a shadow host. It is not useful in ordinary HTML. It is used with Web Components.
Note: The elements for Web Components are are defined in the World Wide Web Consortium (W3C) Web Components collection of specifications rather than the HTML specification. In addition, the Web Components specification has not been finalized and is subject to change.	`<template>`	The **HTML template element `<template>`** is a mechanism for holding client-side content that is not to be rendered when a page is loaded but may subsequently be instantiated during runtime using JavaScript.
Obsolete and deprecated elements Warning: These are old HTML elements which are deprecated and should not be used. You should never use them in new projects, and should replace them in old projects as soon as you can. They are listed here for informational purposes only.	`<acronym>`	The HTML Acronym Element (`<acronym>`) allows authors to clearly indicate a sequence of characters that compose an acronym or abbreviation for a word.
	`<applet>`	The HTML Applet Element (`<applet>`) identifies the inclusion of a Java applet.
	`<basefont>`	The HTML basefont element (`<basefont>`) establishes a default font size for a document. Font size then can be varied relative to the base font size using the `<font>` element.

Element	Description
`<big>`	The HTML Big Element (`<big>`) makes the text *font size* one size bigger (for example, from small to medium, or from large to x-large) up to the browser's maximum font size.
`<blink>`	The HTML Blink Element (`<blink>`) is a non-standard element causing the enclosed text to flash slowly.
`<center>`	The HTML Center Element (`<center>`) is a block-level element that can contain paragraphs and other block-level and inline elements. The entire content of this element is centered horizontally within its containing element (typically, the `<body>`).
`<command>`	The `command` element represents a command which the user can invoke.
`<content>`	The **HTML `<content>`** element is used inside of Shadow DOM as aninsertion point. It is not intended to be used in ordinary HTML. It is used with Web Components. It has now been replaced by the **<slot>** element.
`<dir>`	The *HTML directory element* (`<dir>`) represents a directory, namely a collection of filenames.
`<font>`	The *HTML Font Element* (`<font>`) defines the font size, color and face for its content.
`<frame>`	`<frame>` is an HTML element which defines a particular area in which another HTML document can be displayed. A frame should be used within a `<frameset>`.
`<frameset>`	`<frameset>` is an HTML element which is used to contain `<frame>`elements.
`<isindex>`	**`<isindex>`** is an obsolete HTML element that puts a text field in a page for querying the document.
`<keygen>`	The HTML `<keygen>` element exists to facilitate generation of key material, and submission of the public key as part of an HTML form. This mechanism is designed for use with Web-based certificate management systems. It is expected that the `<keygen>` element will be used in an HTML form along with other information needed to construct a certificate request, and that the result of the process will be a signed certificate.
`<listing>`	The *HTML Listing Element* (`<listing>`) renders text between the start and end tags without interpreting the HTML in between and using a monospaced font. The HTML 2 standard recommended that lines shouldn't be broken when not greater than 132 characters.
`<marquee>`	The HTML `<marquee>` element is used to insert a scrolling area of text.
`<nextid>`	**`<nextid>`** is an obsolete HTML element that served to enable the NeXT web designing tool to generate automatic NAME labels for its anchors. It was generated by that web editing tool automatically and was not to be adjusted or entered by hand. This element has the distinction of being the first element to become one of the "Lost Tags" by being eliminated from the official public DTD's of the HTML versions. It is also probably one of the least understood of all of the early HTML elements.
`<noembed>`	The **`<noembed>`** element is a deprecated and non-standard way to provide alternative, or "fallback", content for browsers that do not support the `<embed>` element or do not support embedded content an author wishes to use.
`<plaintext>`	The *HTML Plaintext Element* (`<plaintext>`) renders everything following the start tag as raw text, without interpreting any HTML. There is no closing tag, since everything after it is considered raw text.
`<spacer>`	`<spacer>` is an HTML element which is used for inserting white spaces to web pages. It was created by NetScape for achieving same effect as a single-pixel layout GIF image, which was something web designers used to use to add white spaces to web pages, without actually using a GIF. However `<spacer>` is not supported by any major browser and same effects can be created with various CSS rules. In Mozilla applications, support for this element was removed in Gecko 2.0. Therefore usage of `<spacer>` is unnecessary.
`<strike>`	The *HTML Strikethrough Element* (`<strike>`) renders text with a strikethrough, or a line through it.
`<tt>`	The HTML Teletype Text Element (`<tt>`) produces an inline element displayed in the browser's default monotype font. This element was intended to style text as it would display on a fixed width display, such as a teletype. It probably is more common to display fixed width type using the `<code>` element.
`<xmp>`	The *HTML Example Element* (`<xmp>`) renders text between the start and end tags without interpreting the HTML in between and using a monospaced font. The HTML2 specification recommended that it should be rendered wide enough to allow 80 characters per line.