

## 1) Important tags in html

--> For commenting out sections of HTML or placing comments within HTML. Anything between `<!--` and `-->` won't be rendered as text when the page loads in a web browser, but will still be visible if looking at the plain 'ol HTML file in a text editor.

`<!DOCTYPE html>` Basically tells the browser what kind of document we're loading, in this case good 'ol HTML. This guy likes to be the first element in the document, even before the opening `<html>` tag. This page, for example, starts things off with `<!DOCTYPE html>`

`<a>` a for anchor, also known as a hyperlink, or simply a link. To make an actual link using the `<a>` tag, use the href attribute: `<a href=http://eastmanreference.com>Link</a>`

`<abbr>` Abbr is an abbreviation for abbreviation. Use it with the title attribute to define abbreviations. `<abbr title="Cascading Style Sheets">CSS</abbr>`. Safari 12.0 hovering the mouse over CSS brings up a little tool tip generated by the abbreviation tag.

`<address>` Contact information for the author. Alternatively, this could be general contact information for the website.

`<area>` Establishes areas in an image map. `<area>` is always used inside of `<map>`.

`<article>` The `<article>` tag is new in HTML5. This tag can be used to contain blog entries, forum posts, etc.

`<aside>` New in HTML5. The `<aside>` element is for making sidebars that are placed next to the main content.

`<audio>` New in HTML5. The `<audio>` tag can be used to place an audio stream within an HTML document.

`<b>` For bolding or otherwise styling text without conveying any additional meaning such as with `<em>` and `<strong>` and etc.

`<base>` Use to set a base URL.

`<bdi>` Short for bidirectional isolation. Useful for working with languages that use right to left text.

`<bdo>` Short for Bidirectional Override. Use this to override the current direction of text. Useful for languages that use right to left text.

`<blockquote>` For quoting from an external source. Typically rendered as indented text.

`<body>` Establishes the body of an HTML document.

`<br>` Single line break. Think of br as short for break.

`<button>` Specifies button that can be clicked. Commonly used with forms.

<canvas> Used for rendering graphics on the fly, typically in conjunction with a scripting language.

<caption> Placed just after <table>, used for table caption.

<cite> For citing.

<code> For code examples. A little bit like <samp> and <kbd>, but specifically for code.

<col> Specify properties of a <col> within a <colgroup>

<colgroup> Essentially a group of columns.

<data> Used for script friendly data.

<datalist> New in HTML5. Specify predefined data for an <input>.

<dd> For description lists. Use with <dt> and <dl>.

<del> Mark text as deleted without actually deleting it. This will typically be rendered as a strikethrough.

<details> New in HTML5. Can be used for additional details which can be hidden or revealed.

<dfn> Use when first defining a term.

<dialog> Dialog box.

<div> A divider or a kind of generic container.

<dl> Description list. Use with <dd> and <dt>.

<dt> Description list. Use with and <dl>.

<em> Emphasis, which I believe often defaults to italics. For something stronger, use <strong>.

<embed> New in HTML5. A container for external content.

<fieldset> Group related items in a form.

<figure> Define self contained content such as photos or diagrams.

<footer> The footer of a document, below the main content.

<form> Forms for user input.

<h1> Level 1 heading, the headline or title of a page.

<h2> Level 2 heading, the subtitles of a page.

<h3> Level 3 heading.

<h4> Level 4 heading.

<h5> Level 5 heading.

<h6> Level 6 heading.

<head> The head section. Used mostly for metadata.

<header> Not to be confused with <head>, <header> typically contains introductory content and layout that goes above the body.

<hgroup> Used for grouping <h1>, <h2>, <h3>, etc., tags together. Useful for multilevel headings.

<hr> This can be used for separating text within a paragraph. Typically renders as a horizontal line running across the page.

<html> The root level tag of an HTML document. All other HTML tags go inside the HTML tag.

<i> The <i> element has changed a little in HTML5. It can still make text italicized, and I think that is often the default behavior, however, this depends on the stylesheet.

<iframe> Makes it possible to embed another page within a page. In HTML5 this is known as nested browsing.

<img> For displaying images on a webpage. Example: <img src=""image.png"">

<input> Input is used with various attributes to create form input elements.

<ins> Useful for denoting text that has been added to an HTML file since the original version. Updates to a blog post for example. This is typically rendered as underlined text, although this can really be rendered just about any way you want with a stylesheet.

<kbd> Used for keyboard input. This is different from form element text input.

<keygen> Used for authentication with a security certificate.

<label> For assigning labels to interface elements such as forms.

<legend> Useful for making forms easier to understand.

<li> This represents a list item with an <ol> (ordered list) or <ul> (unordered list).

<link> Used for linking to external resources. Typically seen in <head> referring to an external style sheet. Different from the <a> (anchor) tag.

<main> New to HTML5. These tags enclose the main content of an HTML document.

<map> Used when defining an image map.

<mark> This is essentially used for highlighting. Of course, whether or not highlighting is actually involved depends on the involved stylesheet.

`<menu>` Used for creating various kinds of menus. This is new to HTML 5.1.

`<menuitem>` Specify actions that can be taken with a menu.

`<meta>` Used for specifying various meta information about the document. Meta tags are used within the document `<head>`.

`<meter>` Used to represent a number when the minimum and maximum is known, for example 10%.

`<nav>` Used for specifying a navigational region within a document.

`<noscript>` For anyone using scripts within their pages, `<noscript>` is useful for specifying content that should be rendered when scripting isn't supported.

`<object>` For embedding objects within a document.

`<ol>` Ordered list (1. 2. 3.). Use with list item `<li>`.

`<optgroup>` For grouping objects.

`<option>` Used within the `<select>` tag to specify an option.

`<output>` Displays the output of an action, such as from a script or form.

`<p>` Paragraph. Enclose paragraphs in an article within `<p>` tags.

`<param>` Use when parameters need to be passed to an object embedded within the document.

`<pre>` For preformatted text. This can be useful for demonstrating code, especially if there are multiple lines.

`<progress>` Display the progress of an action, such as with a script.

`<q>` Quote from some source.

`<rb>` For Ruby annotations.

`<rp>` Specify text to be displayed when Ruby annotations are not supported by the browser.

`<rt>` Text component of a Ruby annotation.

`<rtc>` Ruby Text Container.

`<ruby>` Also used for Ruby annotation.

`<s>` For marking up text that is not longer relevant, applicable, or accurate.

`<samp>` Sample output one might get from a particular computer program.

`<script>` Scripts inside of an HTML document are enclosed in the script tags.

`<section>` A nonspecific means of breaking up content within a webpage into sections.

`<select>` Used with `<option>` for selecting a particular option.

`<small>` Typically used for small print. Useful for comments within the main content.

`<source>` Used with `<audio>` and `<video>` for specifying media source.

`<span>` Useful for applying styles to or around text, especially inline.

`<strong>` The enclosed text is of strong importance.

`<style>` For declaring style sheets within a document.

`<sub>` For displaying subscript. Useful for math, especially in conjunction with `<sup>` (superscript).

`<summary>` Used in conjunction with the `<details>` tag for specifying a summary.

`<sup>` For displaying superscript. Useful for math, especially in conjunction with `<sub>` (subscript).

`<table>` For making tables in an HTML page.

`<tbody>` Groups table rows.

`<td>` Table data; this represents a data cell within a table.

`<template>` Declares HTML snippets that can be used later.

`<textarea>` For multi-line text entry.

`<tfoot>` Table footer.

`<th>` Table header cell.

`<thead>` Table header row.

`<time>` Specifies the date/time within a document.

`<title>` The title of an HTML document.

`<tr>` A row within a table.

`<track>` Used to specify text tracks with `<audio>` and `<video>`.

`<u>` Useful for annotating text for various reasons. No longer specifically for underlining, although you can of course still do that with CSS.

`<ul>` Unordered list. Use with `<li>` (list item) to make unordered lists.

`<var>` Defines a variable within the content of an HTML page.

`<video>` For embedding video within an HTML page.

`<wbr>` The optional line wrap.

## **2) Difference between semantic and non semantic tags with example**

Semantic Elements:

A semantic element clearly describes its meaning to both the browser and the developer. These elements clearly defines its content. In other terms semantic tags are those tags whose name can easily let you know which type of content take place in it. Eg. <header>, <article> and <footer>.

Non-Semantic Elements:

Non-semantic element are the element which Tells nothing about its content. In other terms element whose name doesn't suggest you what they do. E.g. <div> and <span>.

## **3) Difference between inline and block level elements**

Block-level element

there will always be a new line after the closing tag. So, no matter how you organize your HTML, block-level elements always create a new line.

Inline element

there is never a new line. Therefore, no matter how you organize your inline-elements in the markup, they will always appear side-by-side.