

Brainstorms & Raves

Articles: Semantics, HTML, XHTML, and Structure

Introduction

Good **HTML** structure is based on logic, order, and using semantically correct markup. If you have a heading use the heading element, beginning with the **H1** element. If you have a paragraph, use a paragraph element. If you have a list, use a list item element. If you're quoting a few lines, use a blockquote element. Those elements provide meaning to the content, making them semantically correct, in addition to being solid HTML structure.

Once your HTML structure is in place with the appropriate markup, add **CSS** for visual presentation. Start with good HTML structure and then add the CSS, preferably with an external style sheet, for the visual presentation that you have in mind.

The information below is meant as a jumping off point, written for clients and others who want to know at least enough to understand what's going on within their pages and how to create a solid, semantically correct webpage.

A Few Basic Elements

Headings

W3C info: [7.5.5 Headings: The H1, H2, H3, H4, H5, H6 elements \(http://www.w3.org/TR/REC-html40/struct/global.html#edef-H1\)](http://www.w3.org/TR/REC-html40/struct/global.html#edef-H1)

Use heading elements for headings: **H1**, **H2**, **H3**, **H4**, **H5**, **H6**. Don't use CSS, the **STRONG** element or other markup to fake your headings. See my article about this, [Don't Fake Your Markup: Accessibility Issues for CSS \(http://brainstormsandraves.com/archives/2002/05/14/dont_fake_your_markup_accessibility_issues_for_css/\)](http://brainstormsandraves.com/archives/2002/05/14/dont_fake_your_markup_accessibility_issues_for_css/).

For example, this:

```
<h1>Our Company</h1>
<h2>Our Services</h2>
```

Becomes this:

Our Company
Our Services

Paragraphs: the **P** Element

Use the **P** element for paragraphs. Don't use the `
` element to instead provide paragraph-like breaks.

W3C info about **P** (<http://www.w3.org/TR/html401/struct/text.html#h-9.3.2>):

"The **P** element represents a paragraph. It cannot contain block-level elements (including **P** itself).

"We discourage authors from using empty **P** elements. User agents should ignore empty **P** elements."

[9.3.1 Paragraphs: the **P** element \(http://www.w3.org/TR/html401/struct/text.html#h-9.3.1\)](http://www.w3.org/TR/html401/struct/text.html#h-9.3.1)

Line Breaks: the **BR** Element

BR at **W3C**: [9.3.2 Controlling line breaks \(http://www.w3.org/TR/html401/struct/text.html#h-9.3.2\)](http://www.w3.org/TR/html401/struct/text.html#h-9.3.2)

The **BR** element is one of the more abused HTML elements around. When other HTML elements and CSS are understood and used appropriately, forced line breaks with the **BR** element can be minimized. For example:

- use the `P` element for paragraphs, not the `BR` element.
- Use list item elements for lists (`UL`, `OL`, `LI`) rather than marking up a series of items with the `BR` element, and remember that you can use CSS to hide the HTML bullets if you wish.
- Use CSS to set widths, margins, and padding for text rather than using a multitude of `
` tags for text line breaks.

Use this element judiciously.

Emphasis: the `EM` Element

`EM` (<http://www.w3.org/TR/html401/struct/text.html#edef-EM>) at W3C: "Indicates emphasis."

Example: `I'm emphasizing this.`

Use the `EM` element for emphasis. If you're after italicized text aside from emphasized text or citations use CSS (`font-style:italic`) rather than `EM` or `I`. Use the `CITE` element for citing a source.

Emphasis: the `STRONG` Element

`STRONG` (<http://www.w3.org/TR/html401/struct/text.html#edef-STRONG>) at W3C: "Indicates stronger emphasis."

Example: `I'm emphasizing this even stronger.`

Citation: the `CITE` Element

`CITE` (<http://www.w3.org/TR/html401/struct/text.html#edef-CITE>) at W3C: "Contains a citation or a reference to other sources."

Example: `<cite>Designing with Web Standards</cite>` is an excellent book by Jeffrey Zeldman.

Citation: The `BLOCKQUOTE` Element

`BLOCKQUOTE` (<http://www.w3.org/TR/html401/struct/text.html#h-9.2.2>) at W3C: "`BLOCKQUOTE` is for long quotations (block-level content)." (See note below about [the `Q` element](http://brainstormsandraves.com/articles/semantics/structure/#qnote) (<http://brainstormsandraves.com/articles/semantics/structure/#qnote>)). Don't use `BLOCKQUOTE` just for text indenting. If you want to indent text, use CSS margins, padding, or a combination of both. Likewise, if you don't want to use `BLOCKQUOTE` because of its block-level characteristics, you can use CSS to change them, too.

Include the cited source and title within the `BLOCKQUOTE` element:

1. Link to the citation:

Use the `CITE` attribute for the source's `URI`: `cite="uri"`.

```
<blockquote cite="http://w3c.org/" title="Article title, author, date">I'm citing the W3C here.</blockquote>
```

From [W3C regarding the cite attribute](http://www.w3.org/TR/html401/struct/text.html#edef-cite-Q) (<http://www.w3.org/TR/html401/struct/text.html#edef-cite-Q>):

"The value of this attribute is a URI that designates a source document or message. This attribute is intended to give information about the source from which the quotation was borrowed."

[BLOCKQUOTE Attribution definitions: CITE](http://www.w3.org/TR/html401/struct/text.html#edef-cite-Q) (<http://www.w3.org/TR/html401/struct/text.html#edef-cite-Q>)

2. Use the `TITLE` attribute for the source title: `title="text"`:

```
<blockquote cite="http://w3c.org/" title="Article title, author, date">I'm citing the W3C here.</blockquote>
```

For example, this markup:

```
<blockquote cite="http://w3c.org/" title="Article title, author, date">"I'm citing the W3C here."</blockquote>
```

Becomes this:

"I'm citing the W3C here."

It's also recommended to include a visual reference to your citation, as always, perhaps like this:

"I'm citing the W3C here."

[Article title, author, date \(http://w3c.org/\)](http://w3c.org/)

Note: When you hover your mouse over the quotation in some browsers that the title information will appear.

Note: While `BLOCKQUOTE` is intended for longer quotations and `Q` is intended for shorter quotations, the `Q` element is not supported by browsers and even causes problems for some alternative devices, so its use *at the moment* is not recommended. Hopefully next generation browsers will support it. For more information:

- [The Q tag \(http://diveintomark.org/archives/2002/05/04/the_q_tag\)](http://diveintomark.org/archives/2002/05/04/the_q_tag) by Mark Pilgrim
- [The Q Element \(http://www.brain-stream.com/development/test-suites/html401/winie55/q-element.html\)](http://www.brain-stream.com/development/test-suites/html401/winie55/q-element.html) by B.K. DeLong
- [Fixing quotes with Javascript \(http://simon.incutio.com/archive/2003/04/03/fixingQuotesWithJavascript\)](http://simon.incutio.com/archive/2003/04/03/fixingQuotesWithJavascript) by Simon Willison

List Elements: `UL`, `OL`, `LI`

[W3C info about the List element \(http://www.w3.org/TR/html401/struct/lists.html\)](http://www.w3.org/TR/html401/struct/lists.html)

NOTE: It's important to note that the W3C's examples on the above page are outdated for `XHTML` since they don't include closing tags in their samples. It's **REQUIRED** to include closing tags for all list items and other tags for XHTML. It's a good practice anyway, whether using XHTML or HTML.

When you have a list of something, use the list element tags. While there are a few choices of list style types, these can be replaced with images using CSS. It's also possible to hide bullets completely, change indenting, use lists inline or block, and more.

For example, this markup:

```
<ul>
<li>Lemons</li>
<li>Limes</li>
<li>Oranges</li>
</ul>
```

Becomes this:

- Lemons
- Limes
- Oranges

There are many helpful tutorials around about list items. Check WebsiteTips.com's CSS section, [CSS and Lists \(http://websites tips.com/css/#lists\)](http://websites tips.com/css/#lists).

Examples of variations in how list items are treated using CSS:

1. my website, [Brainstorms and Raves \(http://brainstormsandraves.com/\)](http://brainstormsandraves.com/), uses list item tags for the top-of-the-page navigation where visually you'll see what appears to be horizontally arranged graphic buttons.
2. I've used list item tags for the sidebar column lists of links, even though you don't visually see any bulleted lists.
3. I've also used list item tags for the [main archive listing \(http://brainstormsandraves.com/archives/\)](http://brainstormsandraves.com/archives/) for categories and monthly archives. Once again, the bullets are hidden.

They're each treated differently using CSS but they all use the same list item tags.

Definition List Elements: DL, DT, DD

[W3C information about definition list elements \(http://www.w3.org/TR/REC-html40/struct/lists.html#edef-DL\)](http://www.w3.org/TR/REC-html40/struct/lists.html#edef-DL)

Use definition list elements for terms and their corresponding descriptions. Another possible use is for marking up dialogues, with each `DT` element naming a speaker followed by the `DD` element containing the speaker's words.

For example, this:

```
<d1>
<dt>Salad</dt>
<dd>A dish of raw leafy green vegetables, often tossed with pieces of other raw or cooked vegetables, fruit, cheese,
or other ingredients and served with a dressing.</dd>
<dt>Vegetable</dt>
<dd>A plant cultivated for an edible part, such as the root of the beet, the leaf of spinach, or the flower buds of
broccoli or cauliflower.</dd>
<dt>Grains</dt>
<dd>Small, dry, one-seeded fruit of cereal grasses, having the fruit and the seed walls united, such as wheat.</dd>
</d1>
```

Becomes this:

Salad

A dish of raw leafy green vegetables, often tossed with pieces of other raw or cooked vegetables, fruit, cheese, or other ingredients and served with a dressing.

Vegetable

A plant cultivated for an edible part, such as the root of the beet, the leaf of spinach, or the flower buds of broccoli or cauliflower.

Grains

Small, dry, one-seeded fruit of cereal grasses, having the fruit and the seed walls united, such as wheat.

Misc. Elements, Attributes

The `CODE` Element

[CODE \(http://www.w3.org/TR/html401/struct/text.html#edef-CODE\)](http://www.w3.org/TR/html401/struct/text.html#edef-CODE) at W3C: "Designates a fragment of computer code"

For example, this:

```
<code>
<lt;style type="text/css">
body {margin:0;}
</style>
</code>
```

Becomes this:

```
<style type="text/css">
body {margin:0;}
</style>
```

Note: most browsers will render content within the `CODE` element in the browser's default monospace font. You can also use CSS, of course, to designate fonts, colors, and more.

Note: If you use reserved HTML within your code snippet or elsewhere within your HTML markup, such as the ampersand &, less-than sign <, or greater-than sign >, you must escape them with their [character entities \(http://www.blooberry.com/indexdot/html/tagpages/c/charents.htm\)](http://www.blooberry.com/indexdot/html/tagpages/c/charents.htm), such as what is shown in the example above.

The PRE Element

W3C info: [Preformatted text: The PRE element \(http://www.w3.org/TR/html401/struct/text.html#edef-PRE\)](http://www.w3.org/TR/html401/struct/text.html#edef-PRE)

Preformats text with fixed-width font, preserves the given white-space and line-breaks, usually disables automatic word-wrapping. Used in the code examples on this page, for example:

```
<pre>
<ul>
<li> </li>
<li> </li>
<li> </li>
</ul>
</pre>
```

Like the BR element, though, the PRE element has been abused and used to force line breaks for text formatting when CSS and other elements can be used appropriately instead. Just like I mention about the BR element, use the PRE element judiciously.

FAQ

Why do some elements start on a new line and others don't? And why do some indent or have padding?

Most HTML 4 elements are either blocklevel or inline. Blocklevel elements by default start on a new line, such as heading elements (H1, H2, H3, etc.), BLOCKQUOTE, P, and DIV. For lists of blocklevel and inline elements, see the WDG Group's [HTML 4.0 Block-Level Elements \(http://www.htmlhelp.com/reference/html40/block.html\)](http://www.htmlhelp.com/reference/html40/block.html) and [HTML 4.0 Inline Elements \(http://www.htmlhelp.com/reference/html40/inline.html\)](http://www.htmlhelp.com/reference/html40/inline.html) and W3C's [Appendix A. A sample style sheet for HTML 4.0 \(http://www.w3.org/TR/REC-CSS2/sample.html\)](http://www.w3.org/TR/REC-CSS2/sample.html) by Todd Farner.

In case you're wondering, [XHTML 1.0 \(http://www.w3.org/TR/xhtml1/\)](http://www.w3.org/TR/xhtml1/) is a "reformulation of HTML 4 in XML 1.0." See the explanation below, [What's the difference between HTML 4 and XHTML 1? \(http://brainstormsandraves.com/articles/semantics/structure/print_test.shtml#difference\)](http://brainstormsandraves.com/articles/semantics/structure/print_test.shtml#difference).

How can I prevent blocklevel elements from indenting, having padding, or starting on a new line?

Using CSS you can designate blocklevel elements to be inline if you wish. Here's an example for BLOCKQUOTE:

```
blockquote {display:inline}
```

You can designate your heading elements to have as much or as little padding and margin as you wish, too. The example below provides 0 margin and 0 padding top and bottom and 4px padding right and left.

```
h1 {margin:0;padding:0 4px;}
```

What if I want one image to display inline but not the others?

There are several ways to do this. If you're using that image in your topmast, for example, it might be fine to designate your image information within the topmast DIV:

```
#topmast img {display:inline;}
```

Then your topmast would be something like this:

```
<div id="topmast">

</div>
```

You can also designate an image to have a class:

```
img.logo {display:inline;}
```

Then your image would be something like this:

```

```

What's the difference between HTML 4 and XHTML 1? How do I create a webpage with XHTML?

W3C info: [Differences with HTML 4 \(http://www.w3.org/TR/xhtml1/#diffs\)](http://www.w3.org/TR/xhtml1/#diffs)

My tutorial: [Fast and Easy XHTML \(http://www.websitetips.com/info/xhtml1/basics.shtml\)](http://www.websitetips.com/info/xhtml1/basics.shtml)

via Sizefactory.com: [Standards-compliant XHTML \(http://www.sizefactory.com/xhtml/\)](http://www.sizefactory.com/xhtml/)

In general terms, your markup must be in lowercase, you must use proper nesting, proper ending tags, and ampersands (&) in URLs must be written as `&`. If you already used valid markup with HTML 4, the differences will be quite minor and won't take long to learn.

Even when I don't follow semantic markup my pages still validate. Don't standards mean you have to use semantically correct markup?

Semantics and standards aren't necessarily the same, although using good semantics is recommended. The W3C validator doesn't know if you've used other tags to fake your headings or not, for example.

So why bother with using semantic markup? What's the big deal?

1. Longevity;
2. search engines rank keyword weight based on semantic structure;
3. your markup will make sense to anyone else, whether today or later;
4. See this article by Jason Kottke: [Standards don't necessarily have anything to do with being semantically correct \(http://www.kottke.org/03/08/030826standards_do.html\)](http://www.kottke.org/03/08/030826standards_do.html)

Copyright © 2000-2005 [Shirley E. Kaiser, M.A., SKDesigns](http://brainstormsandraves.com/). All rights reserved.

<http://brainstormsandraves.com/articles/>

Page last modified 03/01/05