

Don't go class-crazy.

It's easy to go overboard in assigning class names to elements. In many cases, other types of selectors, such as contextual or attribute selectors, may be used instead. For example, instead of labeling every `h1` element in a sidebar as `class="sideread"`, a contextual selector could be used, like this:

```
div#sidebar h1 {font: Verdana 1.2em bold #444;}
```

Lists

Humans are natural list-makers, so it makes sense that mechanisms for creating lists of information have been part of HTML since its birth. This section looks at the types of lists defined in (X)HTML:

- Unordered information
- Ordered information
- Terms and definitions

Unordered Lists

Unordered lists are used for collections of related items that appear in no particular order. Most lists fall into this category. Just about any list of examples, components, thoughts, or options should be marked up as an unordered list. Most notably, unordered lists are the element of choice for navigational options. Unordered lists for navigation are discussed later in this section.

In (X)HTML, unordered lists are denoted with the `ul` element. The content of a `ul` is limited to one or more list items (`li`). List items may contain either block-level or inline elements, or both. Unordered lists and their list items are block elements, so each will display starting on a new line.

ul

`...`

Attributes

Core (id, class, style, title), *Internationalization*, *Events*

Deprecated attributes

compact
type="disc|circle|square"

li

`...`

Attributes

Core (id, class, style, title), *Internationalization*, *Events*

Deprecated attributes

```
type="format"
value="number"
```

Unordered list syntax

This example shows the markup for a basic unordered list.

```
<ul>
  <li>Unordered information</li>
  <li>Ordered information</li>
  <li>Terms and definitions</li>
</ul>
```

In HTML 4.01, the end tags for list items are optional, but in XHTML, all end tags are required. It is good practice to close all elements regardless of the version of HTML you are using.

Unordered list presentation

By default, user agents insert a bullet before each list item in an unordered list. Leaving an unordered list unstyled (that is, applying no style sheet properties to it) is a reliable shortcut to having your information appear as an indented bulleted list.

But that sells the usefulness of the unordered list element short. By applying style properties, an unordered list may be presented however you like. You can change the shape of the bullets with the `list-style-type` property (this property replaces the deprecated `type` attribute that is discouraged from use). The `list-style-image` property allows you to use your own image as a bullet. Style properties for lists are discussed in Chapter 23.

And that's just the beginning. You can set lists to display horizontally, too. You can even use unordered list markup as the structure underlying a rich graphical navigation toolbar with rollover effects, all accomplished with Cascading Style Sheets. These techniques are outlined in Chapter 24.

Ordered Lists

Ordered lists are used for lists in which the sequence of the items is important, such as step-by-step instructions or endnotes. Ordered lists are indicated by the `ol` element and must include one or more list items (`li`). Like all lists, ordered lists and their list items are block-level elements.

`ol`

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

Attributes

Core (id, class, style, title), *Internationalization*, *Events*

Deprecated attributes

```
compact
start="number"
type="1|A|a|I|i"
```

Ordered lists have the same basic structures as unordered lists, as shown in this simple example.

```
<ol>
  <li>Get out of bed</li>
  <li>Take a shower</li>
  <li>Walk the dog</li>
</ol>
```

By default, user agents automatically number the list items in ordered lists. There is no need to add the number in the source.

Style sheets may be used to change the numbering system (`list-style-type`) as described in Chapter 23. The `list-style-type` property replaces the deprecated `type` attribute that specifies the numbering system for lists, as shown in Table 10-1.

Table 10-1. Values of the deprecated `type` attribute

Type value	Generated style	Sample sequence
1	Arabic numerals (default)	1, 2, 3, 4
A	Uppercase letters	A, B, C, D
a	Lowercase letters	a, b, c, d
I	Uppercase Roman numerals	I, II, III, IV
i	Lowercase Roman numerals	i, ii, iii, iv

Use the deprecated `start` attribute to start the counting of the list items at a particular number. This markup example creates an ordered list using lowercase letters that starts counting at 10.

```
<ol type="a" start="10">
  <li>See quirksmode.org/css/tests/</li>
  <li>According to the W3C Working Group</li>
  <li>See the XHTML 1.1 Working Document</li>
</ol>
```

The resulting list would look like this, because “j” is the tenth letter in the alphabet:

- j. See quirksmode.org/css/tests/
- k. According to the W3C Working Group
- l. See the XHTML 1.1 Working Document

There is a CSS alternative to the `start` attribute using the `counter-reset` property, but it is poorly supported by browsers at this time.

Definition Lists

Use a *definition list* for lists that consist of term and definition pairs.

Definition lists are marked up as `dl` elements. The content of a `dl` is some number of terms (indicated by the `dt` element) and definitions (indicated by the `dd` element). The `dt` (term) element may contain only inline content, but a `dd` may include block-level or inline elements. All three elements used in definition lists are block-level elements and will start on a new line by default.

`dl`

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

Attributes

Core (id, class, style, title), *Internationalization*, *Events*
compact

`dd`

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

Attributes

Core (id, class, style, title), *Internationalization*, *Events*

`dt`

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

Attributes

Core (id, class, style, title), *Internationalization*, *Events*

The markup structure for definition lists is a little different from the lists discussed so far. The entire list, made up of `dt` and `dd` elements, is contained within the `dl` element, as shown here.

```
<dl>
  <dt>em</dt>
  <dd>Indicates emphasized text. em elements are nearly always rendered in
  italics.</dd>

  <dt>strong</dt>
  <dd>Denotes strongly emphasized text. Strong elements are nearly always
  rendered in bold text.</dd>

  <dt>abbr</dt>
  <dd>Indicates an abbreviated form.</dd>

  <dt>acronym</dt>
  <dd>Indicates an acronym.</dd>
</dl>
```

Terms and definitions are not required to appear in alternating order. In other words, it is fine to introduce two terms and apply one definition, or supply two or more definition elements for a single term. The HTML 4.01 Recommendation provides an informal example of definition list dialogues, where the speaker corresponds to the term and the spoken words correspond to the definition. Many semantic (X)HTML experts consider this particular example to be an abuse of the semantics of definitional lists and thus it should be avoided.

The presentation of definition lists should be controlled with style sheet properties. By default, user agents generally display the definitions on an indent.

Nesting Lists

List elements may be nested within other lists. For example, you can add an unordered list within a definition list, or a numbered list as an item within an unordered list. This example shows just one variation. The resulting list is shown in Figure 10-2.

```
<ol>
<li>Mix Marinade
  <ul>
    <li>2 slices ginger <em>(smashed)</em></li>
    <li>1 T. rice wine or sake</li>
    <li>1 t. salt</li>
    <li>2 T. peanut oil</li>
  </ul>
</li>
<li>Saute the seasonings</li>
<li>Add fish sauce</li>
</ol>
```

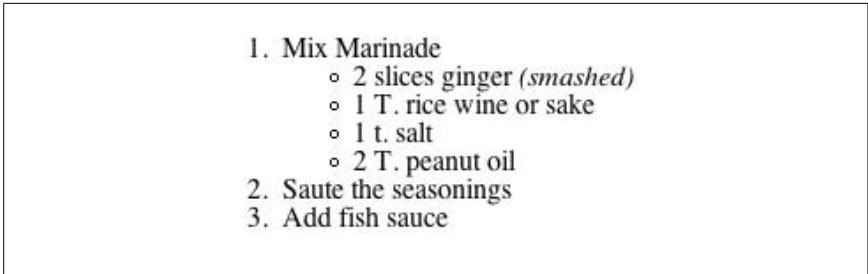
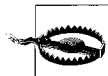
- 
1. Mix Marinade
 - 2 slices ginger (*smashed*)
 - 1 T. rice wine or sake
 - 1 t. salt
 - 2 T. peanut oil
 2. Saute the seasonings
 3. Add fish sauce

Figure 10-2. Nested lists



Note that in order for the list markup to be valid, `ul` and `ol` elements may contain only `li` elements. That means the nested list must be contained within a list item (`li`) and may not be a child of the `ul` or `ol` element. Authors should also be careful to close all elements so they are nested properly and do not overlap.

When unstyled unordered lists (`ul`) are nested within each other, browsers automatically display a different bullet for each consecutive level, usually `disc`, then
