Extra notes - HTML - Semantics

- Dr Nick Hayward

A brief introduction to semantic considerations in HTML markup.

Contents

Intro As we introduce semantic usage and considerations into our HTML markup, we need to ensure that we convey the correct meaning. i.e. the meaning expected for the contained content.

For example, we may often see the following elements mis-used and applied incorrectly for markup,

- `<p>` - paragraphs
- `<ul` - unordered list
- `<h1>` to `<h6>` - headings
- `<blockquote>` - blockquote

Correct usage If we consider such HTML element tags, we may consider correct, expected usage against usage for convenience or a required visual effect for rendering.

Using `<blockquote>` , for example, to simply help indent text, instead of CSS margins for example, or the perennial mis-use of a `<p>` .

```
<p> <p>
```

to simply add extra space between elements. These are common examples of bad, non-semantic HTML markup and usage.

A worse example is the use of the `<ul>` element to again indent text. This is not only semantically incorrect but invalid HTML. List groups for list items is how we should be using them.

Structure & Validation We may consider example use cases, both good and bad, for HTML lists.

Our first examples is as follows,

```
<li>nice</li>
<li>cannes</li>
<li>menton</li>
```

e.g. basic list items

This example markup for a list looks OK, but it fails in one obvious way. Without a correctly defined structural grouping, a collective parent element, it's still not valid markup.

So, *what's missing?*

We've closed each `<li>` element properly, and we've put each list item on its own line.

However, we're missing an outer layer of semantics. In effect, there is structure missing to define a containing element. Something that semantically denotes this list as a list.

a semantic point of view So, from the perspective of semantics, this list is clearly meant to act as a group of items that belong together.

Therefore, they should be denoted as such.

This is a good example of semantic markup - structuring items to clearly denote their meaning and purpose.

e.g.

```
<ul>
  <li>nice</li>
  <li>cannes</li>
  <li>menton</li>
</ul>
```

e.g. basic group

We might also choose to add one of the global attributes to the `<ul>` element,

- https://developer.mozilla.org/en-US/docs/Web/HTML/Global_attributes

benefits of structure & validation So, by simply providing a meaningful structure to our markup we gain flexibility as our project develops.

For example, it helps increase ease of CSS styling, creates properly structured documents, and improves general management of updates to markup. So, easier to understand and easier to maintain and update.

So, with structured, valid markup we can easily repurpose data into different representations of information.

example 1 e.g. a list grouping used as a standard list

```
<ul>
  <li>nice</li>
  <li>cannes</li>
  <li>menton</li>
  <li>antibes</li>
  <li>grasse</li>
</ul>
```

- example - basic group style

example 2 e.g. a list grouping used for a navigation menu, tabs…

```
<ul id="menutabs">
  <li><a href="nice">nice</a></li>
  <li><a href="cannes">cannes</a></li>
  <li><a href="menton">menton</a></li>
  <li><a href="antibes">antibes</a></li>
  <li><a href="grasse">grasse</a></li>
</ul>
```

example - basic menu tabs

Markup for Headings As we've seen with lists, for example, HTML is flexible in markup usage due to presentational versus structural considerations.

We might perceive headings from a purely presentational perspective, only caring about how they're rendered to the user.

e.g.

```
<span class="heading">Chapter 1</span>
```

However, there are further issues with this markup for a heading.

For example,

- Visual browsers (e.g., Firefox, Safari, Chrome &c.) will render the heading the same as normal text on the page when CSS is disabled or unavailable. Non-visual browsers won't know the difference between the heading and normal text.
- Search engines won't place greater importance over headings that are marked up with `<span>` .
- We can style it uniquely, but we're locked into the heading class when adding similar headings in the future.

We might try another option.

e.g.

```
<p><b>Chapter 1 - The Call</b></p>
```

However, we still encounter familiar issues.

For example,

- Visual browsers will render the text only in bold and the same size as the default.
- We can't style this heading uniquely from other text on the page.
- Search engines won't place greater importance over headings that are marked up with `<p>` and `<b>` elements.

correct use of headings So, mark it up correctly with structure and meaning.

e.g.

```
<h3>Chapter 1 - The Call</h3>
```

Benefits of this structure and markup include,

- Conveys meaning to the text contained within.
- Visual and non-visual browsers will treat the heading correctly regardless of any style that is associated with it.
- Easily styled uniquely with CSS.
- Search engines will place importance on keywords contained within heading elements.

Markup for Tables A great example of poor usage of HTML markup is the `<table>` element in HTML.

The main issue is the use of nested tables and spacer elements and images.

If used correctly, in structured markup, tables are very useful. They can also impart a sense of semantic organisation to data, creating various interpretive information.

So, what is a table for? It should be use for structuring data, which imparts a sense of curated information.

So, use a table for *data*.

example 1

- simple table example (with issues) - columns and rows for *presentation*

```
<p>Travel Destinations</p>
<!-- basic table structure - minimal - rows and columns -->
<table>
  <tr>
    <td><b>Place</b></td>
    <td><b>Country</b></td>
    <td><b>Sights</b></td>
  </tr>
  <tr>
    <td>Nice</td>
    <td>France</td>
    <td>Cours Saleya</td>
  </tr>
  <tr>
    <td>Cannes</td>
    <td>France</td>
    <td>La Croisette</td>
  </tr>
  <tr>
    <td>Antibes</td>
    <td>France</td>
    <td>Picasso museum</td>
  </tr>
</table>
```

- example - basic table for presentation

example 2

- add semantic structure & elements to table caption - replace `<p>` with correct `<caption>` usage for a table...

```
<!-- basic table structure - minimal - add a caption -->
<table>
  <caption>Travel Destinations</caption>
  ...
```

Modern browsers, by default, will style text in a `<caption>` element above a table and centred. We can modify the styling, of course, but the browser is rendering the information for us based upon the underlying semantics of the element tag.

- example - basic table caption

example 3

- add a summary attribute to the table

```
<!-- basic table structure - minimal - add summary attribute -->
<table summary="structured table data for travel destinations...">
  <caption>Travel Destinations</caption>
  ...
```

We can then add further meaning and structure to the table with the use of a *summary* attribute on the table element. Again, this is processed by the browsers for semantics, and is particularly useful for non-visual browsers.

- example - basic table with summary

example 4

- add correct headers `<th>` to the table

```
<!-- basic table structure - minimal - add table headers -->
<table summary="structured table data for travel destinations...">
  <caption>Travel Destinations</caption>
  <tr>
    <th>Place</th>
    <th>Country</th>
    <th>Sights</th>
  </tr>
  ...
```

We can update our table with correct table headings instead of purely presentational elements, such as bold.

Visual browsers will usually render such headings in bold and centred, but again we can easily customise the styling of this structural element.

Benefits include:

- remove need for presentational markup, bold elements
- visual browsers will process, by default, both structural and presentation qualities of this table heading - again, difference between structured data and rendered information

Such heading elements can also be useful for non-visual browsers.

- example - basic table with headers

example 5

- table markup and accessibility markup

```
<!-- basic table structure - accessibility - add ids and headers -->
<table summary="structured table data for travel destinations...">
  <caption>Travel Destinations</caption>
  <tr>
    <th id="place">Place</th>
    <th id="country">Country</th>
    <th id="sights">Sights</th>
  </tr>
  <tr>
    <td headers="place">Nice</td>
    <td headers="country">France</td>
    <td headers="sights">Cours Saleya</td>
  </tr>
  ...
```

In this example, we're now creating a known relationship between the table's header, and its data. With this in place, a screen reader, for example, may read this table as follows,

- Place: Nice, Country: France, Sights: Cours Saleya

and so on throughout the table's data.

In effect, we've established a pattern to the output information that makes more sense to non-standard browsers and readers.

- example - basic table with accessibility

example 6

- add extra semantic markup for thead, tfoot, tbody

```
<!-- basic table structure - add head, foot, body -->
<table summary="structured table data for travel destinations...">
  <caption>Travel Destinations</caption>
  <thead>
    <tr>
      ...
    </tr>
  </thead>
  <tfoot>
    <tr>
      ...
    </tr>
  </tfoot>
  <tbody>
    <tr>
      ...
    </tr>
  </tbody>
</table>
```

The table head and foot elements customarily go above the table body to allow modern browsers, readers, &c. to load that data first, and then render the main table content.

Benefits include:

- better underlying structure to data
- greater ease for styling a table due to clear divisions in data and information
- structural and presentational markup now working together correctly…
- example - basic table with head, foot, body

Resources

- MDN - HTML - Global attributes
- MDN - HTML - Headings
- MDN - HTML - Lists
- MDN - HTML - Tables