Best practices when using sectioning elements

BEST PRACTICE 1: ALWAYS ADD A HEADING TO EXPLICIT SECTIONING CONTENT

Mainly for accessibility reasons, it's always better to include a heading (a `<h1>`, `<h2>`...`<h6>`) in each sectioning element (`<section>`, `<article>`, `<nav>`, `<aside>`), but also after the `<body>` element (called a "sectioning root").

Here are some examples:

Good (heading in each explicit section):

```
<section>
  <h1>Blog post of April 2015</h1>
  ...
</section>
```

Good (heading in a `<header>` does not change anything)

```
<section>
  <header>
    <h1>Blog post of April 2015</h1>
    <p>Posted by Michel Buffa...</p>
  </header>
  ...
</section>
```

Bad (there is no Hx after the <section> -> no heading):
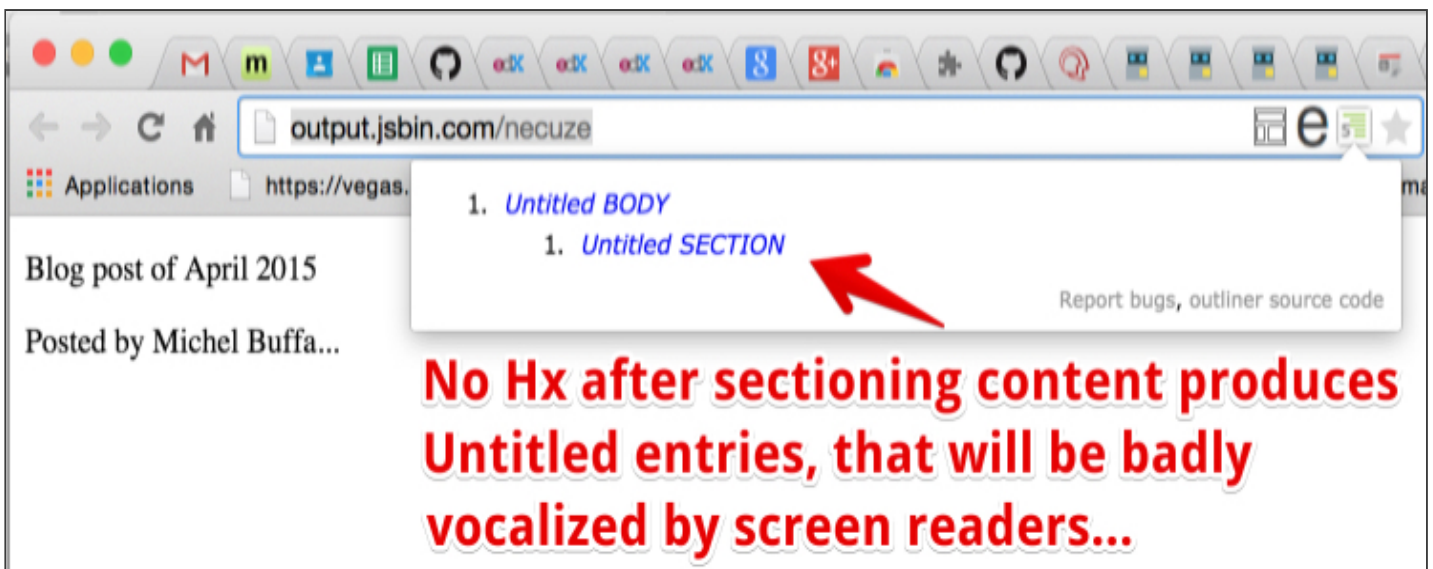
```
<section>
  <header>
    <p class="article title">Blog post of April 2015</p>
    <p>Posted by Michel Buffa...</p>
  </header>
  ...
</section>
```

The last example is bad for accessibility reasons. A screen reader that vocalizes the page will just say "Entering section", while in the previous two good examples it would say "entering section with heading Blog Posts of April 2015". You can also check if your headings and sectioning elements are ok by using a browser extension that displays the outline of the document (just search for "html5 outliner" in your browser's extension search engine).

For the course screenshots we used the Google Chrome HTML5 outliner extension.

The outline of the last example looks like this:

Notice that `<body>` is also a sectioning element, it's called a "sectioning root", and would also need a heading.

Final good version:

```
      <body>
        <h1>Example Blog</h1>
        <section>
          <header>
            <h2>Blog post of April 2015</h2>
            <p>Posted by Michel Buffa...</p>
          </header>
          <p>Content of the blog post...</p>
        </section>
10.   </body>
```

In red, the sectioning root (`<body>`) and the sectioning elements (`<section>` here...), each have a heading.

To sum up:

- Always use a heading element after a sectioning element, for example `<section>` **`<Hx>...</Hx>`**`...</section>`, and after `<body>`, where **x** can be 1..6,

- Or, use a `<header>` element, like in `<section>`**`<header><Hx>...</Hx>.....</header>`**`...</section>`

MORE ABOUT THE `<HEADER>` ELEMENT

The `<header>` element is just a container. It is not taken into account for defining new sections of a document nor does it affect the hierarchy levels.

You can use heading elements `<h1>...<h6>` in a `<header>` but be careful if you use more than one, as the rules explained in the previous part of the course will apply and may generate implicit "sections" in the header.

This example has two headings in the `<header>`:

```
      <section>
        <header>
```

```
            <h1>Some text in a h1 in a header of a section</h1>
            <h2>This a h2 in the header...</h2>
        </header>
    </section>
```
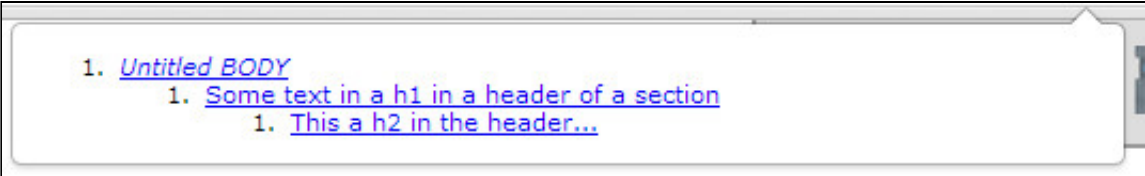
Here is the resulting table of contents, notice the two subsections that appear, one for the H1, one for the H2:


```
1. Untitled BODY
        1. Some text in a h1 in a header of a section
            1. This a h2 in the header...
```

Indeed, HTML does not have a dedicated mechanism for marking up subheadings, alternative titles or taglines.

If you do not want the subtitles to be included in the table of contents, just use standard markup, for example `<p>` elements, like in the next example. Of course, CSS rules can be applied to change colors, sizes, etc.

```
<header>
  <h1>HTML 5.1 Nightly</h1>
  <p>A vocabulary and associated APIs for HTML and XHTML</p>
  <p>Editor's Draft 9 May 2013</p>
</header>
```

BEST PRACTICE 2: TRY NOT TO RELY ON IMPLICIT SECTIONING, USE `<SECTION>`, `<ARTICLE>`, ETC. INSTEAD OF JUST `<H1>...<H6>`

The example below defines several implicit "sections" by using `<Hx>` directly (at lines 7 and 9):

Ok version (no explicit sections everywhere):

```
<body>
<h4>Apples</h4>
<p>Apples are fruit.</p>
<section>
   <h2>Taste</h2>
   <p>They taste lovely.</p>
   <h6>Sweet</h6>
   <p>Red apples are sweeter than green ones.</p>
   <h1>Color</h1>
10.   <p>Apples come in various colors.</p>
 </section>
 </body>
```

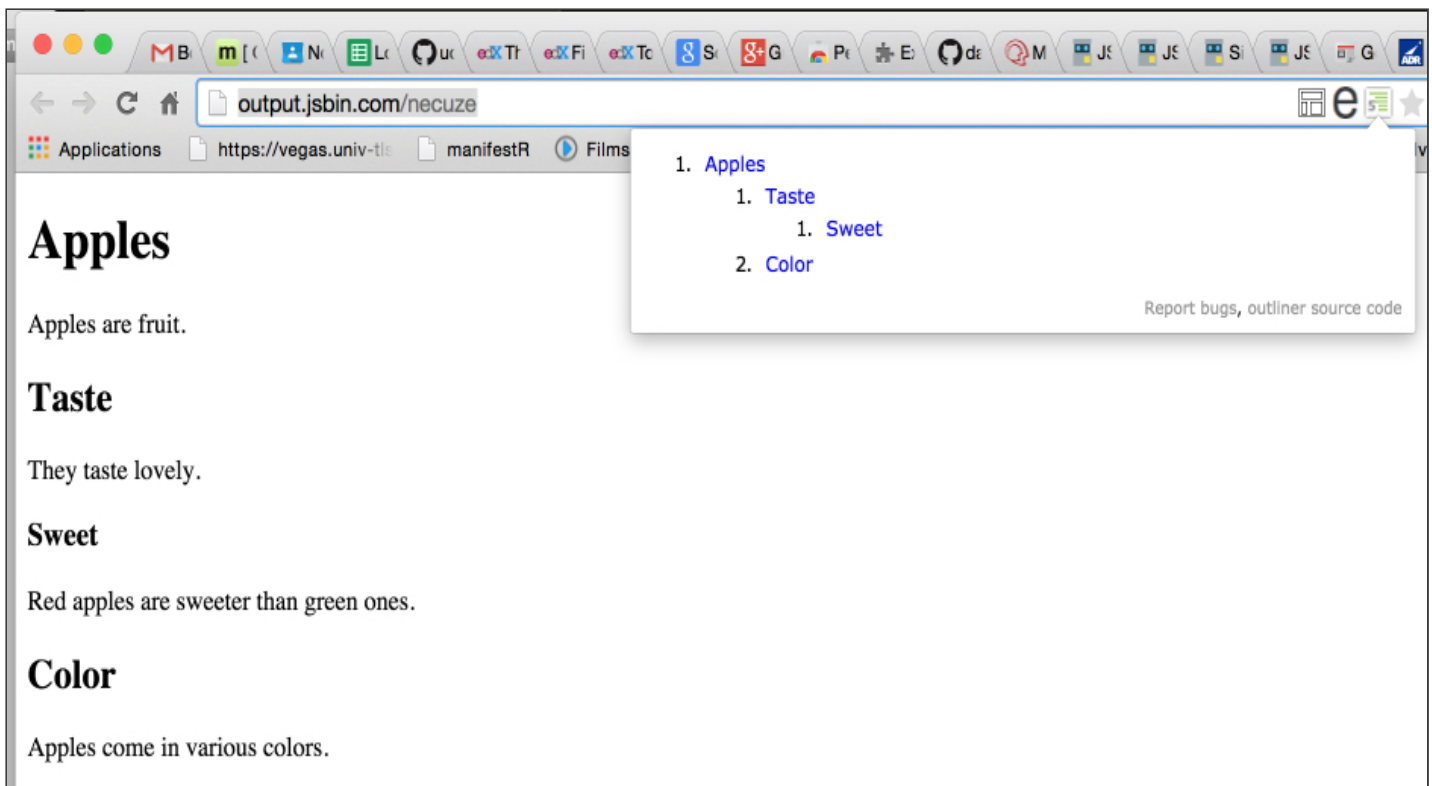**Better version (best practice): this example can also be written like this:**

```
<body>
<h1>Apples</h1>
<p>Apples are fruit.</p>
<section>
   <h2>Taste</h2>
   <p>They taste lovely.</p>
   <section>
     <h3>Sweet</h3>
     <p>Red apples are sweeter than green ones.</p>
10.   </section>
 </section>
 <section>
   <h2>Color</h2>
   <p>Apples come in various colors.</p>
 </section>
 </body>
```

Both of the examples above are semantically identical and produce the same outline:

The third example *is also semantically identical*, and might be easier to maintain (e.g. if sections are often moved around in editing), as it uses only H1s right after sectioning elements. Be sure to read the warning message in the explanations at the end of this page.

```
     <body>
     <h1>Apples</h1>
     <p>Apples are fruit.</p>
     <section>
        <h1>Taste</h1>
        <p>They taste lovely.</p>
        <section>
           <h1>Sweet</h1>
           <p>Red apples are sweeter than green ones.</p>
10.     </section>
     </section>
     <section>
        <h1>Color</h1>
        <p>Apples come in various colors.</p>
```

```
    </section>
  </body>
```

This final example would need explicit CSS style rules to be rendered well in legacy browsers. Legacy browsers without CSS support would render all the headings as top-level headings (same default size).

IMPORTANT WARNING ABOUT THE THIRD EXAMPLE JUST ABOVE!

As stated the article HTML5 Document Outline and on the W3C HTML Wiki, it is not recommended today to follow the third example, as many browsers do not implement the outline algorithm that "sort of renumbers" H1s when located inside sectioning content.