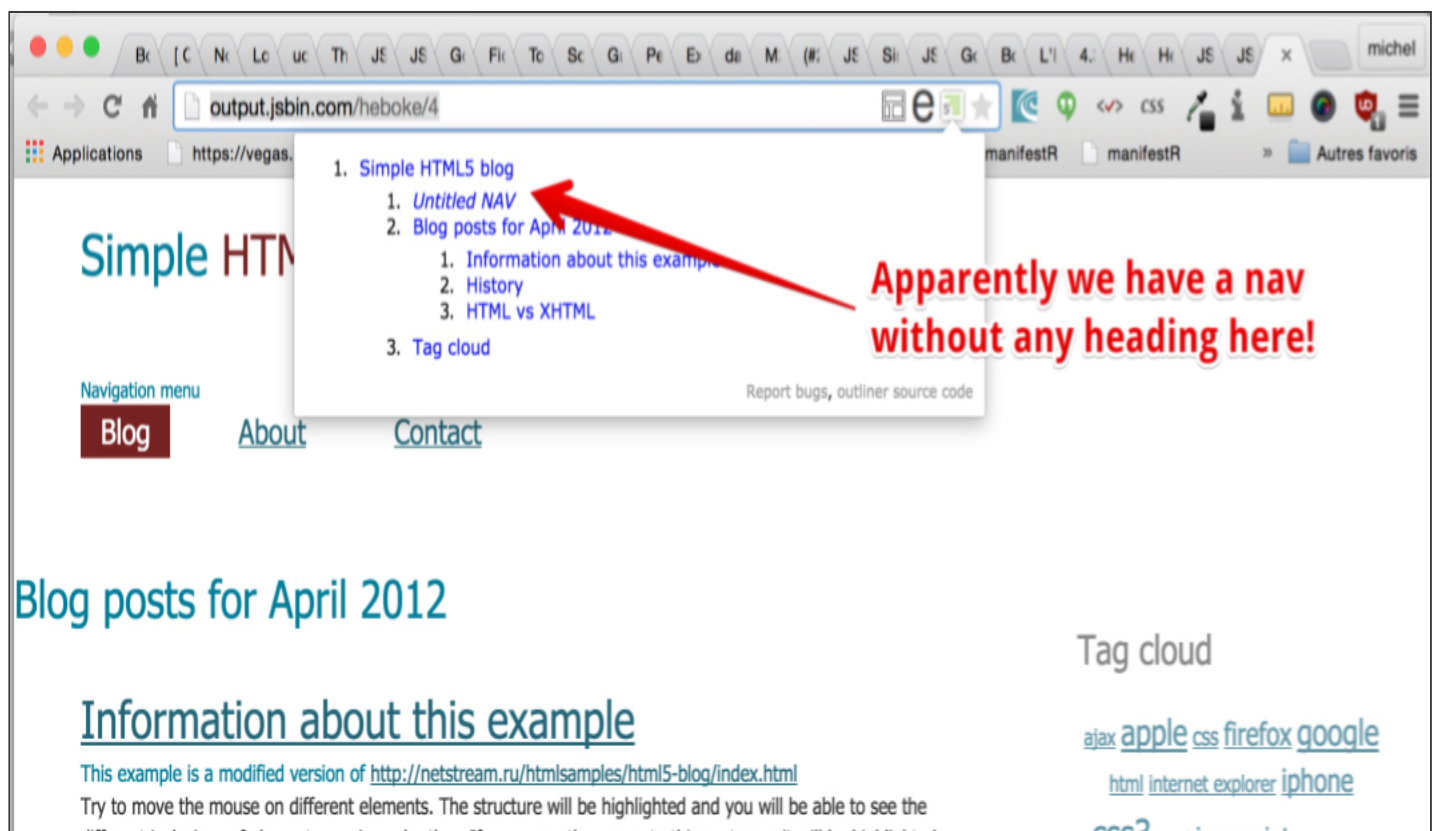# Let's fix the blog example, taking into account best practices

Let's go back to our blog example and see what can be improved:

- Do we have a heading after each sectioning element?

- Did we use sectioning elements or implicit sections?

- Can we embed a table of contents?

The blog example is online at: http://jsbin.com/heboke/19/edit, let's see what the Google Chrome HTML5 extension shows:

Notice also that in this example, we used H1s after each sectioning element, and we still get a hierarchy, some H1s are inside an `<article>` that is in a `<section>` (this corresponds to the third example given in the "heading and sectioning elements" part of the course):

```
<section>
  <header>
    <h1>Blog posts for April 2012</h1>
  </header>
  <article>
    <header>
      <h1><a href="">Information about this example</a></h1>
      This example is a modified version of <a
        href="http://netstream.ru/htmlsamples/html5-
        blog/index.html">http://netstream.ru/htmlsamples/html5-
        blog/index.html</a>
    </header>
    ...
  </article>
</section>
```

With this technique, parts of the document can be moved more easily, or integrated inside an RSS stream, without the need to renumber the headings.

Beware that this technique will require you to use some CSS styling, and may confuse some screen readers that do not yet take into account this way of computing the heading hierarchy. A simple fix is to use an H1 right after the `<body>` and use only H2...H6 inside `<section>`, `<article>`, `<nav>` and `<aside>`.

LET'S FIX THE MISSING HEADING

We need to add a heading in the `<nav>` element. This will both fix the outline of the document by removing the untitled entry, and will also make screen readers happy as they will better vocalize the structure of the page (it will say "entering nav" followed by the vocalization of the heading content).

```
<nav>
  <header>
    <h1>Navigation menu</h1>
```
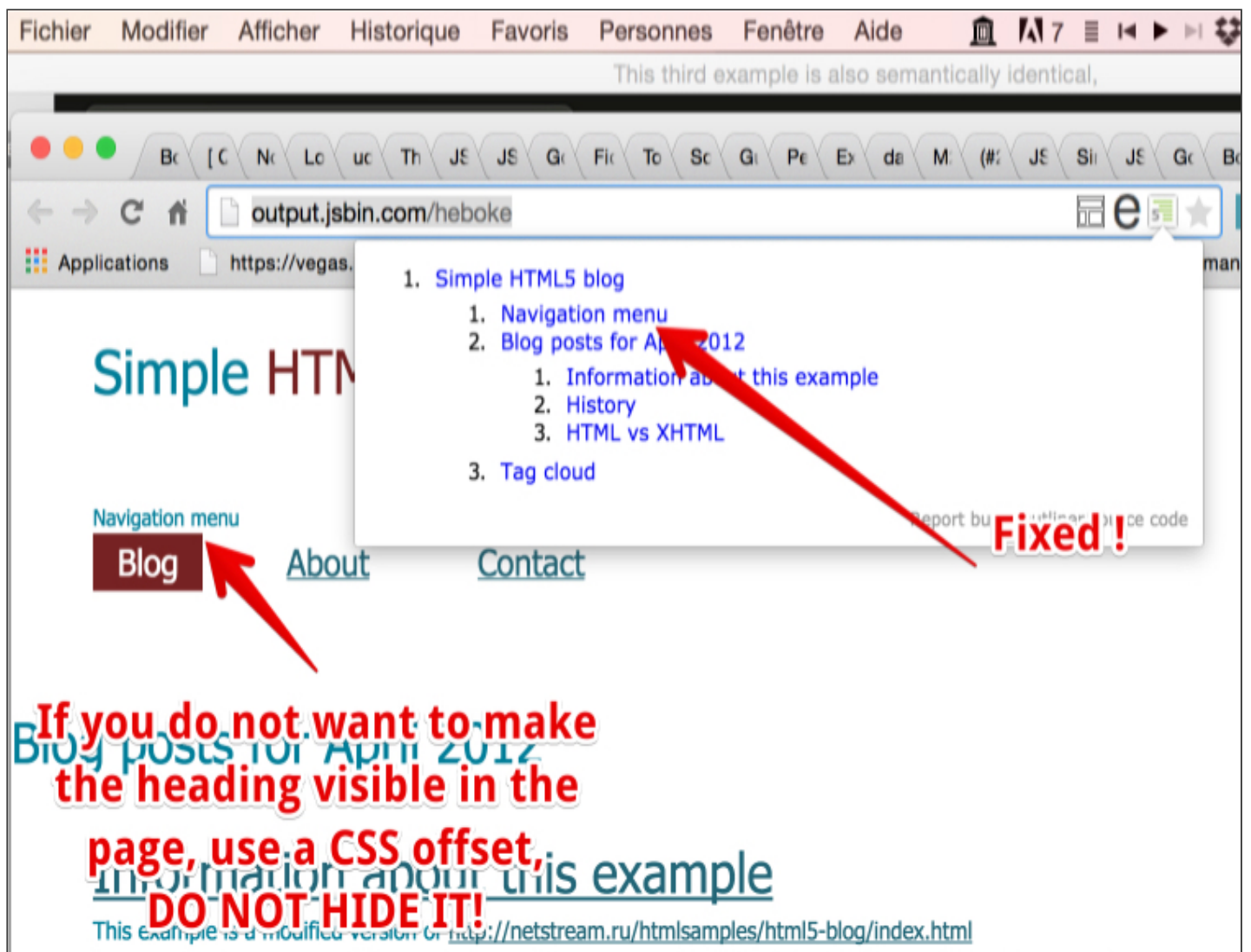
```
   </header>
   <ul>
     <li><span>Blog</span></li>
     <li><a href="">About</a></li>
     <li><a href="">Contact</a></li>
   </ul>
10. </nav>
```

Here is the fixed result:

A common remark from Web designers is: "we do not want a heading content displayed systematically after a `<nav>`, or an `<aside>` element..."

BEST PRACTICE 1: In order to NOT display the heading content on screen the recommended technique is described in this article by Steve Faulkner**. Do not use `display:none` or `visibility:hidden` in your CSS stylesheet, as in that case the heading content will never be vocalized by screen readers, and more generally by assistive technologies.**

BEST PRACTICE 2: it is not advised to include interactive content (links, controls etc) that is hidden offscreen (it is in fact a violation of the W3C WCAG 2.0 Guidelines**). All interactive content must have a visible focus indicator (and be on screen when focused).**

EMBEDDING A TABLE OF CONTENTS AND ADDING A `<MAIN>` ELEMENT

We saw in the previous section how to embed a table of contents using some JavaScript code borrowed from the Google Chrome HTML5 outliner extension.

Let's add this piece of code (we removed the JS details from this extract):

```
<aside>
 <h1>
  <a href="javascript:(function(){...});"
     title="TableDeMatiere">
    Click here for displaying the table of contents!
  </a>
 </h1>
</aside>
```

We also added a `<main>` element to identify the main content of the page composed of the big section with all blog posts:

```
<main>
 <section>
   <header>
      <h2>Blog posts for April 2012</h2>
   </header>
   ...
 </main>
```

USE H1 AS TOP LEVEL HEADINGS ONLY, USE H2…H6 IN SECTIONING CONTENT

As stated by the article HTML5 Document Outline and by the W3C HTML Wiki , using nested H1s is a risky behavior today, as browsers do not implement correctly the "outline algorithm".

The blog example uses nested H1's. if you check it with the W3C conformance checker, it will issues a warning: "*Consider using the h1 element as a top-level heading only (all h1 elements are treated as top-level headings by many screen readers and other tools).*"

So, while this is just a warning, we do prefer to use H1s only as top level elements, and replace the H1s we had after `<section>`, `<article>`, `<nav>` and `<aside>` elements respectively by a H2s and H3s.

Extract from source code:

```
<nav>
 <header>
  <h2>Navigation menu</h2>
 </header>
    . . .
</nav>
```

FINAL, FIXED EXAMPLE

Full online example: http://jsbin.com/heboke/24/edit