

Using HTML5 Elements in Non-HTML5 Browsers

Sections and headings elements should work in most non-HTML5 browsers. Though unsupported, they don't need a special DOM interface and they only need a specific CSS styling as unknown elements are styled as `display:inline` by default:

```
1 section, article, aside, footer, header, nav, hgroup {  
2   display:block;  
3 }
```

Of course the web developer can style them differently, but keep in mind that in a non-HTML5 browser, the default styling is different from what is expected for such elements. Also note that the `<time>` element has not been included, because the default styling for it in a non-HTML5 browser is the same as the one in an HTML5-compatible one.

This method has its limitation though, as some browsers do not allow styling of unsupported elements. That is the case of the Internet Explorer (version 8 and earlier), which need a specific script to allow this:

```
1 <!--[if lt IE 9]>  
2   <script>  
3     document.createElement("header" );  
4     document.createElement("footer" );  
5     document.createElement("section");  
6     document.createElement("aside" );  
7     document.createElement("nav" );  
8     document.createElement("article");  
9     document.createElement("hgroup" );  
10    document.createElement("time" );  
11  </script>  
12 <![endif]-->
```

This script means that, in the case of Internet Explorer (8 and earlier), scripting should be enabled in order to display HTML5 sectioning and headings elements properly. If not, they won't be displayed, which may be problematic as these elements are likely defining the structure of the whole page. That's why an explicit `<noscript>` element should be added for this case:

```
1 <noscript>  
2   <strong>Warning !</strong>  
3   Because your browser does not support HTML5, some elements are simulated using JS  
4   Unfortunately your browser has disabled scripting. Please enable it in order to dis  
5 </noscript>
```

This leads to the following code to allow the support of the HTML5 sections and headings elements in non-HTML5 browsers, even for Internet Explorer (8 and older), with a proper fallback for the case where this latter browser is configured not to use scripting:

```
1 <!--[if lt IE 9]>
2   <script>
3     document.createElement("header" );
4     document.createElement("footer" );
5     document.createElement("section");
6     document.createElement("aside" );
7     document.createElement("nav" );
8     document.createElement("article");
9     document.createElement("hgroup" );
10    document.createElement("time" );
11  </script>
12  <noscript>
13    <strong>Warning !</strong>
14    Because your browser does not support HTML5, some elements are simulated using JS
15    Unfortunately your browser has disabled scripting. Please enable it in order to c
16  </noscript>
17 <![endif]-->
```

Conclusion

The new sections and headings elements introduced in HTML5 bring the ability to describe the structure and the outline of a web document in a standard way. They bring a big advantage for people having HTML5 browsers and needing the structure to help them understand the page, for instance people needing the help of some assistive technology. These new semantic elements are simple to use and, with very few burdens, can be made to work also in non-HTML5 browsers. Therefore they should be used without restrictions.

HTML5 Documentation

HTML

[Audio/Video](#) [Canvas](#) [WebGL](#) [SVG](#) [MathML](#) [WebForms](#) [AppCache](#)
[Microformats](#) [SemanticTags](#)

JavaScript

[Storage](#) [IndexedDB](#) [WebSockets](#) [WebWorkers](#) [Events](#) [Drag/Drop](#)
[ProtocolHandler](#) [Geolocation](#) [Focus](#)

CSS

[NewSelectors](#) [Typography](#) [Visual](#) [Effects](#)