CIS 4004: Web Based Information Technology Summer 2014

Inside HTML5 – Part 3

Instructor : Dr. Mark Llewellyn
markl@cs.ucf.edu
HEC 236, 407-823-2790
http://www.cs.ucf.edu/courses/cis4004/sum2014

Department of Electrical Engineering and Computer Science
University of Central Florida

Inside HTML5 – Part 3

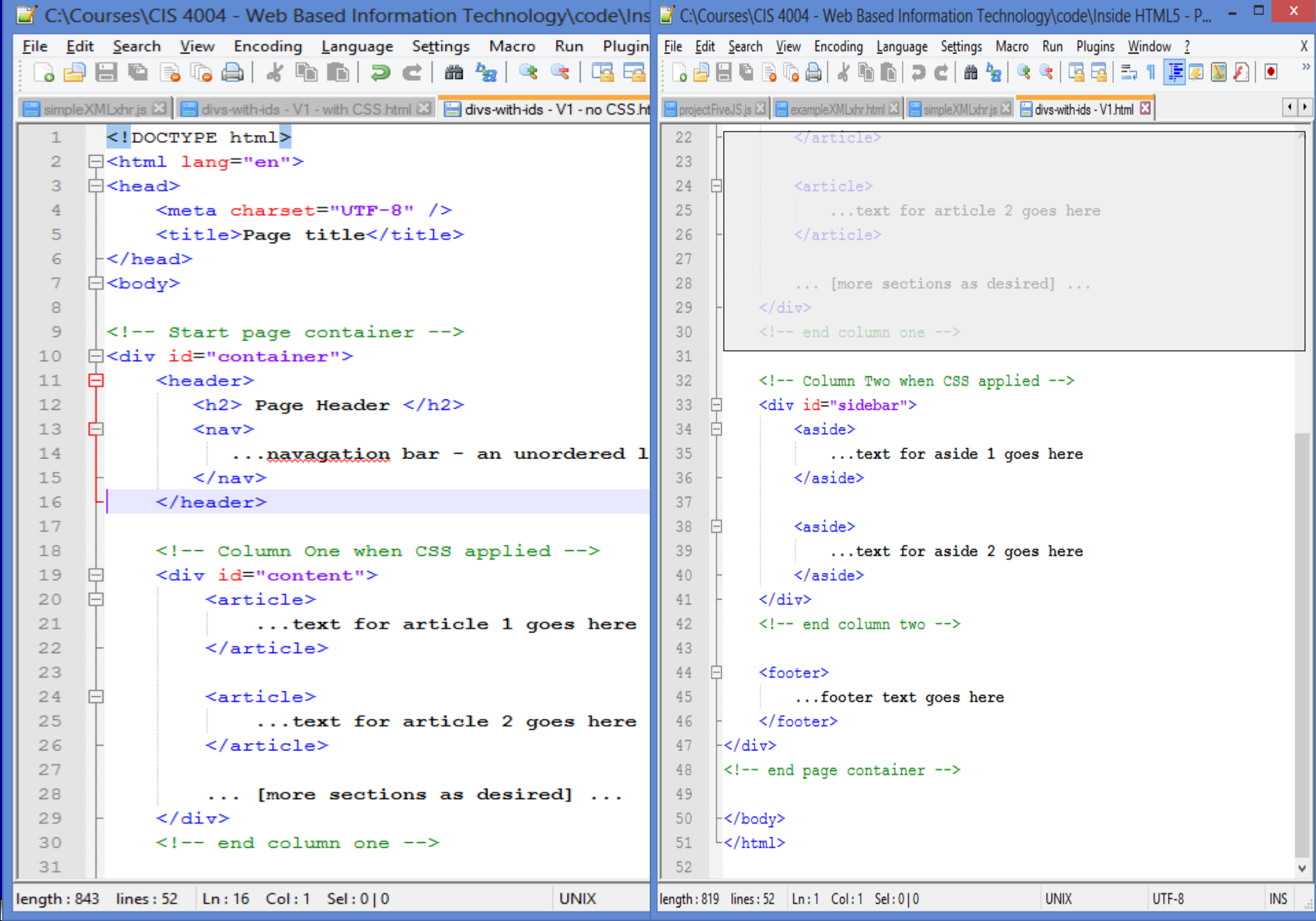
- Before looking at some of the new elements of HTML5 such as the form element, I want to wrap up a few loose ends with how some XHTML, HTML4, and HTML5 semantics merge in HTML5.
- In Inside HTML5 – Part 1, we looked at several HTML5 structural elements, such as `h1-h6`, `div`, `article`, `section`, `header`, `footer`, `nav`, and `aside`.
- The `div` and `h1-h6` elements are the only ones that predate HTML5. Until HTML5, the `div` element was the de facto choice for surrounding chunks of content such as a page's header, footer, main content, and sidebars so that they could be styled with CSS.

Inside HTML5 – Part 3

- The `div` element had no semantic meaning in HTML4 or XHTML and it still doesn't in HTML5.
- That's why HTML5 introduced the new `header`, `footer`, `article`, `section`, `aside`, and `nav` elements. These elements have a semantic meaning which the `div` element does not.
- The `div` element doesn't go away in HTML5, you should just have fewer occasions to use it than in the past.

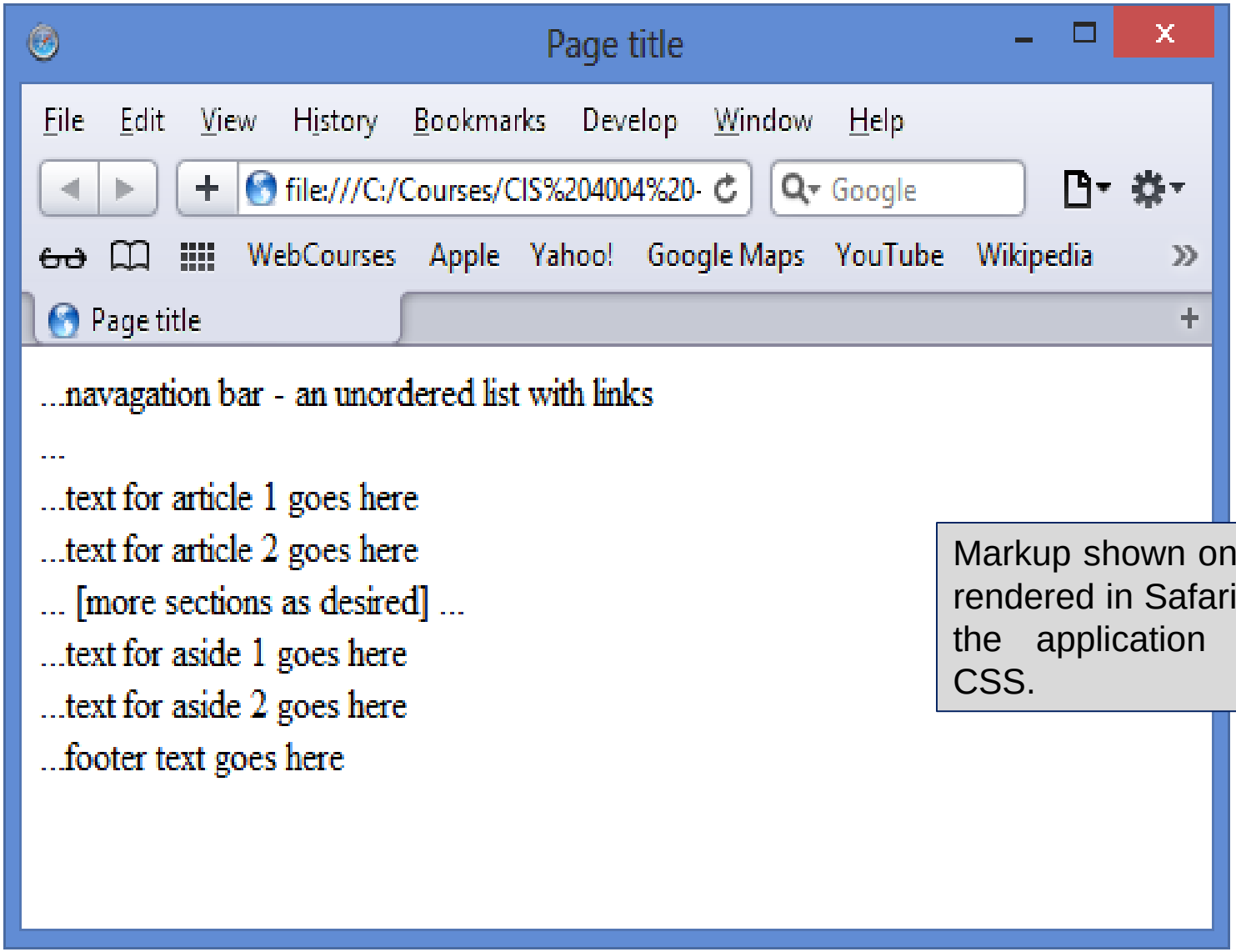
Inside HTML5 – Part 3

- As you will discover in your first project, the most common usage of the `div` element in HTML5 is for containing columar content for the purposes of positioning.
- You will group together content that you want to represent in each column and place this content inside a `div` element. Then that `div` element can be styled accordingly.
- In this case a `div` makes sense while a `section` element would not because semantically the column is not a section, but might contain many sections or articles inside the column. So the `div` is semantically correct in this context.
- The markup example on the next page illustrates this technique.


C:\Courses\CIS 4004 - Web Based Information Technology\code\Ins
File Edit Search View Encoding Language Settings Macro Run Plugin
simpleXMLxhr.js
divs-with-ids - V1 - with CSS.html
divs-with-ids - V1 - no CSS.ht
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8" />
5 <title>Page title</title>
6 </head>
7 <body>
8
9 <!-- Start page container -->
10
11 <header>
12 <h2> Page Header </h2>
13 <nav>
14 ...navagation bar - an unordered l
15 </nav>
16 </header>
17
18 <!-- Column One when CSS applied -->
19
20 <article>
21 ...text for article 1 goes here
22 </article>
23
24 <article>
25 ...text for article 2 goes here
26 </article>
27
28 ... [more sections as desired] ...
29
30 <!-- end column one -->
31
length : 843 lines : 52 Ln : 16 Col : 1 Sel : 0 | 0 UNIX
C:\Courses\CIS 4004 - Web Based Information Technology\code\Inside HTML5 - P...
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
projectFiveJS.js
exampleXMLxhr.html
simpleXMLxhr.js
divs-with-ids - V1.html
22 </article>
23
24 <article>
25 ...text for article 2 goes here
26 </article>
27
28 ... [more sections as desired] ...
29
30 <!-- end column one -->
31
32 <!-- Column Two when CSS applied -->
33
34 <aside>
35 ...text for aside 1 goes here
36 </aside>
37
38 <aside>
39 ...text for aside 2 goes here
40 </aside>
41
42 <!-- end column two -->
43
44 <footer>
45 ...footer text goes here
46 </footer>
47
48 <!-- end page container -->
49
50 </body>
51 </html>
52
length : 819 lines : 52 Ln : 1 Col : 1 Sel : 0 | 0 UNIX UTF-8 INS

Inside HTML5 – Part 3

- The markup on the previous page represents a basic template for a very common two-column web page layout.
- While the markup shown doesn't really include anything useful that can be rendered (see next page), I've included a screen shot on the following page of this same markup styled so that you can see the visual layout that is represented by this markup using the `div` element as a generic container for the main structural elements of the page.
- The actual markup that created this rendering is available on the course website if you'd like to look at it, but it doesn't contain anything special. (`divs-with-ids-V1.html`)



Markup shown on page 5 rendered in Safari without the application of any CSS.

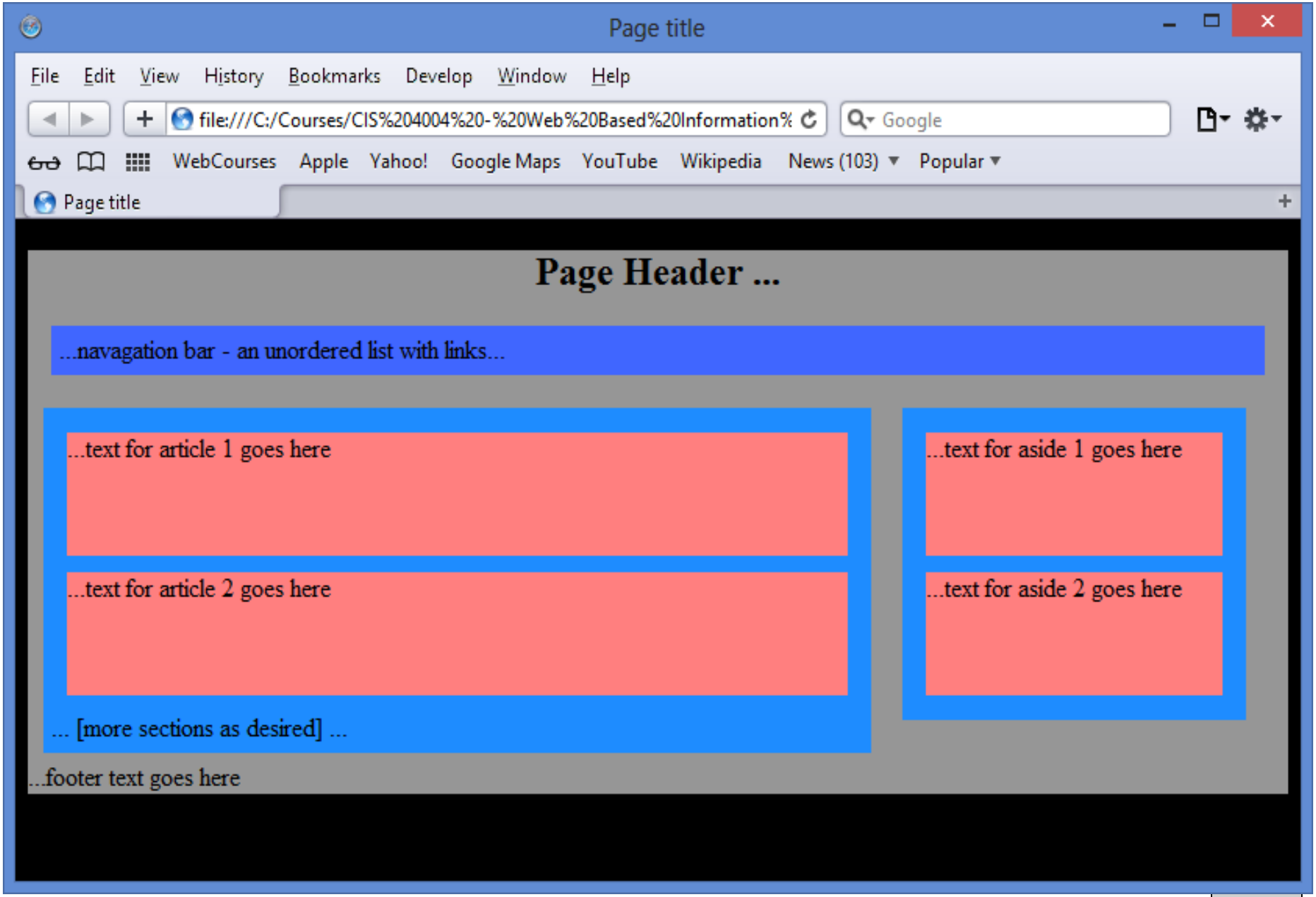
C:\Courses\CIS 4004 - Web Based Information Technology\code\Inside HTML5 - Part 3\divs-with-ids - V1 - with CSS.html - Notepad++

File Edit Search View Encoding Language Settings Macro Run Plugins Window ?

simpleXMLxhr.js | divs-with-ids - V1 - with CSS.html | divs-with-ids - V1 - no CSS.html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8" />
5      <title>Page title</title>
6      <style>
7      <!--
8          #container { background-color: rgb(150,150,150); }
9          #content { background-color: rgb(12%,55%,100%); padding:5px; margin:5px 10px; position:relative; float:left
10         #sidebar {background-color: rgb(12%,55%,100%); padding:5px; margin:5px 10px; position:relative; float:left;
11         #navigation {background-color: rgb(25%,40%,100%); padding:5px; margin:15px;}
12         aside, article {background-color: hsl(0,100%,75%); margin:10px; height:75px;}
13         footer {clear:left; }
14         body {background-color:black;}
15         h2 {text-align:center;}
16         p {font-size:13pt; font-weight:bold;}
17
18     -->
19     </style>
20 </head>
21 <body>
22
23 <!-- Start page container -->
24 <div id="container">
25     <header>
26       <h2> Page Header ... </h2>
27       <nav id="navigation">
28         ...navagation bar - an unordered list with links...
29       </nav>
30     </header>
```

Hyper Text Markup Language file | length : 1513 lines : 66 | Ln : 1 Col : 16 Sel : 0 | 0 | UNIX | UTF-8 | INS


Page title
File Edit View History Bookmarks Develop Window Help
file:///C:/Courses/CIS%204004%20-%20Web%20Based%20Information%
Google
WebCourses Apple Yahoo! Google Maps YouTube Wikipedia News (103) Popular
Page title
Page Header ...
...navagation bar - an unordered list with links...
...text for article 1 goes here
...text for article 2 goes here
... [more sections as desired] ...
...text for aside 1 goes here
...text for aside 2 goes here
...footer text goes here

Inside HTML5 – Part 3

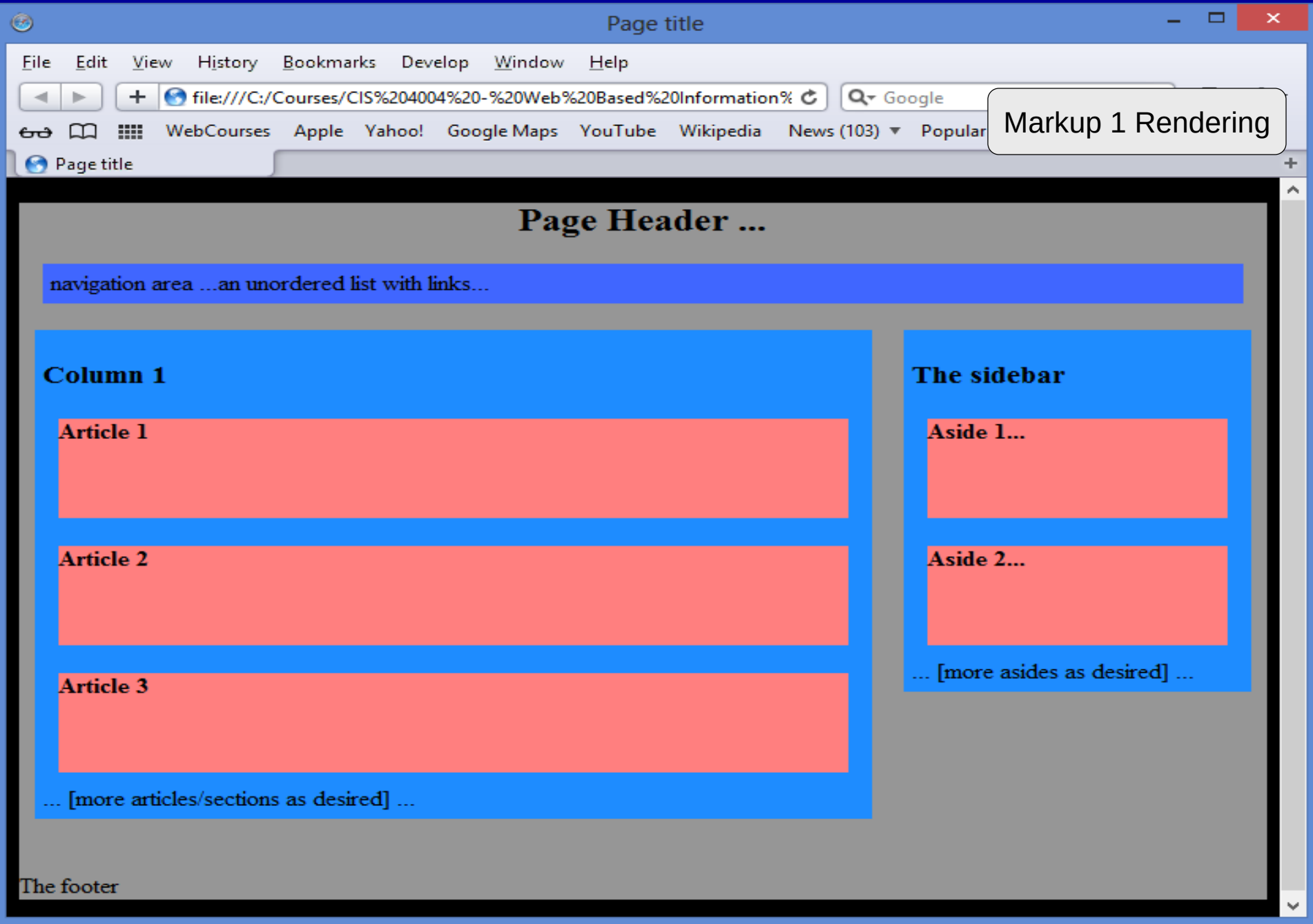
- Compare the two pieces of markup shown on the next page and their respective renderings on the following pages.
- What is the difference between the two pages?

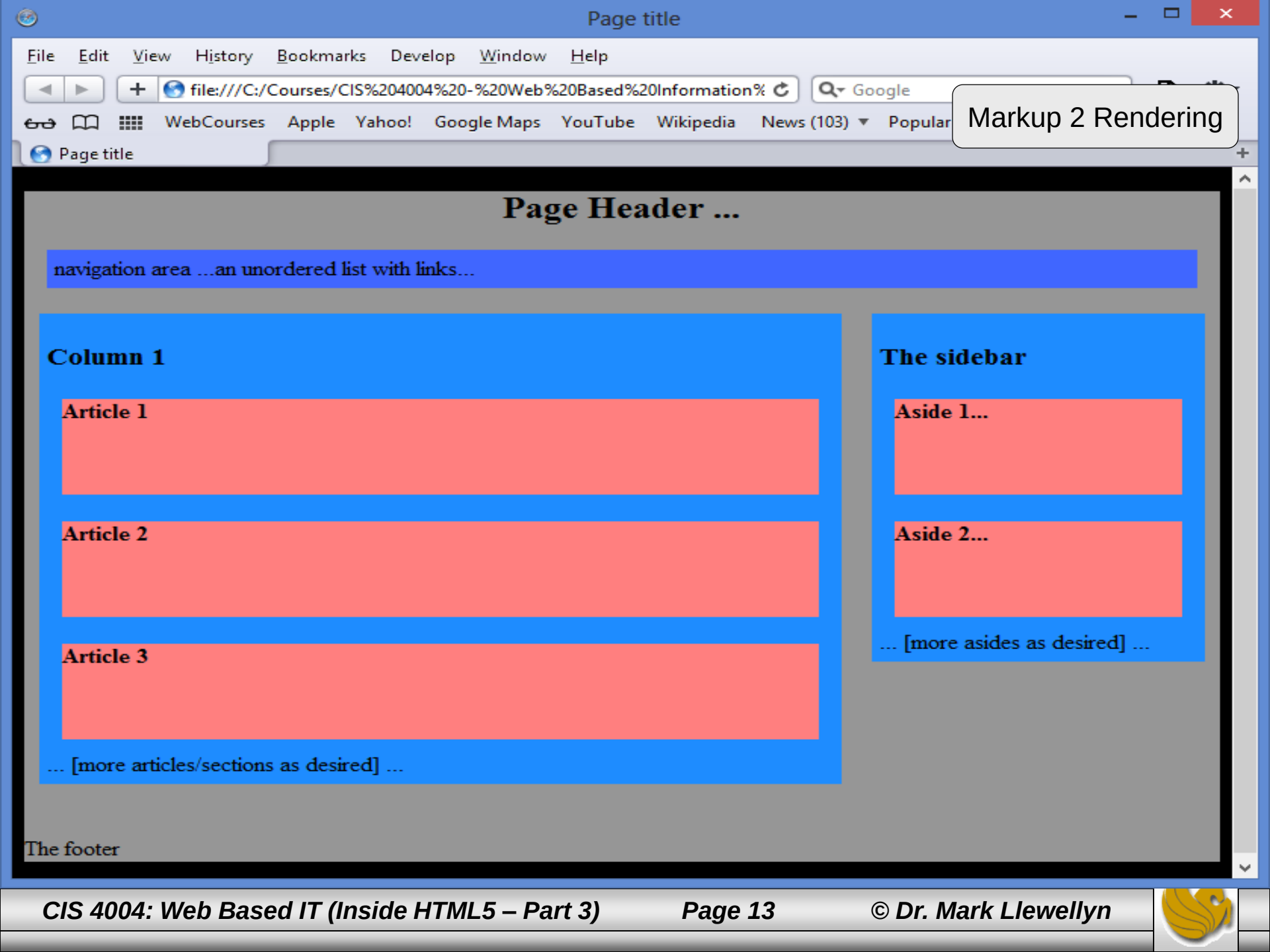
Markup 1

```
<!-- Start page container -->
<div id="container">
    <header>
        <h2> Page Header ... </h2>
        <nav id="navigation">
            navigation area ...an unordered list with links...
        </nav>
    </header>

    <!-- Column One when CSS applied -->
    <div id="content">
       <p> Column 1</p>
        <article>
            <p> Article 1 </p>
        </article>

        <article>
            <p> Article 2 </p>
        </article>

        <article>
            <p> Article 3 </p>
        </article>

        ... [more articles/sections as desired] ...
    </div>
```

Markup 2

```
<!-- Start page container -->
<div id="container">
    <header>
        <h2> Page Header ... </h2>
        <nav id="navigation">
            navigation area ...an unordered list with links...
        </nav>
    </header>

    <!-- Column One when CSS applied -->
    <div id="content">
       <h3> Column 1</h3>
        <article>
            <h4> Article 1 </h4>
        </article>

        <article>
            <h4> Article 2 </h4>
        </article>

        <article>
            <h4> Article 3 </h4>
        </article>

        ... [more articles/sections as desired] ...
    </div>
```



Page title
File Edit View History Bookmarks Develop Window Help
file:///C:/Courses/CIS%204004%20-%20Web%20Based%20Information%
Google
WebCourses Apple Yahoo! Google Maps YouTube Wikipedia News (103) Popular
Page title
Markup 1 Rendering
Page Header ...
navigation area ...an unordered list with links...
Column 1
Article 1
Article 2
Article 3
... [more articles/sections as desired] ...
The sidebar
Aside 1...
Aside 2...
... [more asides as desired] ...
The footer


Page title
File Edit View History Bookmarks Develop Window Help
file:///C:/Courses/CIS%204004%20-%20Web%20Based%20Information%
Google
WebCourses Apple Yahoo! Google Maps YouTube Wikipedia News (103) Popular
Page title
Markup 2 Rendering
Page Header ...
navigation area ...an unordered list with links...
Column 1
Article 1
Article 2
Article 3
... [more articles/sections as desired] ...
The sidebar
Aside 1...
Aside 2...
... [more asides as desired] ...
The footer

Inside HTML5 – Part 3

- What is different about the two pages is that markup 2 uses semantically correct HTML5 and markup 1 does not.
- Notice in markup1 that the column and article headers use paragraph elements to delineate them. In markup 2 the column and article headers are using proper heading elements.
- From the renderings of the two pages, you can't really see any difference between the two. However, there is a huge difference from an accessibility point of view.
- I've put both of these pages on the course website. I encourage you to use them as examples for what comes next.

More Accessible Web Pages With WAI-ARIA

- The accessibility of your web pages improves simply by using semantically correct HTML5. However, you can still do more with ARIA.
- Web Accessibility Initiatives Accessible Rich Internet Applications (WAI-ARIA), also known just as ARIA, is a draft specification (http://w3.org/TR/wai-aria) that improves the accessibility of web applications and web pages.
- ARIA enables developers and content authors to develop rich Internet applications and content that can be recognized and used by assistive technology.
- More often than not, assistive technology does not know what a widget is and rarely are widgets accessible with a keyboard.

More Accessible Web Pages With WAI-ARIA

- When web page content is updated via JavaScript or an AJAX call, assistive technology does not know that the content has been updated, and thus cannot inform the user.
- Although we will not consider all the possible solutions that ARIA offers, I wanted you to see the Landmark Roles section of ARIA and how you can add these new roles to your HTML5 documents.
- To get a feel for how assistive technology works and how semantically correct HTML5 can enhance a user's experience on your web site who utilizes such technology, I suggest that you watch the following videos:

  http://www.youtube.com/watch?v=AmUPhEVWu_E and

  http://www.youtube.com/watch?v=izrC4R7SsH4&list=TLZmcv_cCIrLVISIgCXuXHuNknKQHIG1Fo

More Accessible Web Pages With WAI-ARIA

- Landmark Roles are regions of the web pages used as navigational landmarks. There are more than 50 of them listed in the specification. To see them all go to: (http://w3.org/TR/wai-aria/roles - landmark_roles).

- Here is a list of the more commonly used landmark roles:

```
role="article"
role="banner"
role="complementary"
role="contentinfo"
role="form"
role="heading"
role="main"
role="navigation"
role="search"
```

More Accessible Web Pages With WAI-ARIA

- Landmark roles are added to your markup quite easily by simply appending the role to a semantically correct element.
- For example:

```
<form role="search">

    . . .

</form>
```

- This signifies that the particular `<form>` element (there might be several on a page) is used for searching.
- You can see from the list on the previous page that some have obvious pairings with new HTML5 elements.
- The markup on the next page illustrates the use of roles.

C:\Courses\CIS 4004 - Web Based Information Technology\code\Inside HTML5 - P...

File Edit Search View Encoding Language Settings Macro Run Plugins Window

simpleXMLxhr.js | divs-with-ids - V1 - with CSS.html | divs-with-ids - V1 - no CSS.html

```
22
23      <!-- Start page container -->
24      <div id="container">
25          <header role="banner">
26              <h2> Page Header ... </h2>
27              <nav id="navigation" role="navigation">
28                  navigation area ...an unordered list wi
29              </nav>
30          </header>
31
32          <!-- Column One when CSS applied -->
33          <div id="content" role="main">
34              <h3> Column 1</h3>
35              <article>
36                  <h4> Article 1 </h4>
37              </article>
38
39              <article>
40                  <h4> Article 2 </h4>
41              </article>
42
43              <article>
44                  <h4> Article 3 </h4>
45              </article>
46
47              ... [more articles/sections as desired] ...
48          </div>
49          <!-- end column one -->
50
51          <!-- Column Two when CSS applied -->
```

length : 1669 lines : 74 Ln : 1 Col : 1 Sel : 0 | 0 UNIX UTF

C:\Courses\CIS 4004 - Web Based Information Technology\CIS ...

File Edit Search View Encoding Language Settings Macro Run Plugin

divs-with-ids - V1.html | divs-with-ids - V2 - ARIA.html

```
51          <!-- Column Two when CSS applied -->
52          <div id="sidebar" role="complementary">
53              <h3> The sidebar</h3>
54              <aside>
55                  <h4> Aside 1...</h4>
56              </aside>
57
58              <aside>
59                  <h4> Aside 2...</h4>
60              </aside>
61              ... [more asides as desired] ...
62          </div>
63          <!-- end column two -->
64
65          <footer role="contentinfo">
66              <p> </p>
67              <p>The footer</p>
68          </footer>
69      </div>
70      <!-- end page container -->
71
72      </body>
73      </html>
```

Ln : 31 Col : 1 Sel : 0 | 0 UNIX UTF-8

More Accessible Web Pages With WAI-ARIA

- Here is a brief description of some of the most commonly used landmark roles. For more information, I encourage you to do more research using the links to WAI-ARIA on the previous pages.

- `role="banner"` – A region that contains mostly site-oriented content, rather than page-specific content. Site-oriented content typically includes things such as the logo or identity of the site sponsor, and a site-specific search tool. A banner usually appears at the top of the page and typically spans the full width of the page. Usage: add it to your page-level masthead (typically a `header` element), and use it only once on each page.

More Accessible Web Pages With WAI-ARIA

- `role="navigation"` – A collection of navigational elements (usually links) for navigating the document or related documents. Usage: This mirrors the HTML5 nav element, so add it to each `nav` element. This roles can used multiple times per page.

- `role="main"` – The main content of a document. Usage: Add it to the container that contains the main section of content. Often this will be a `div` element, but it could be an `article` or `section`, too. Except in rare circumstances, your page should have only one area marked with `main`.

More Accessible Web Pages With WAI-ARIA

- `role="complementary"` – A supporting section of the document, designed to be complementary to the main content…but that remains meaningful when separated from the main content. The complementary role indicates that the contained content is relevant to the main content. Usage: This mirrors the HTML5 `aside` element, so add it to an `aside` or a `div` that contains all complementary content. You can include more than one `complementary` role on a given page.

- `role="contentinfo"` – A large perceivable region that contains information about the parent document. Examples of information included in this region of the page are copyrights and links to privacy statements. Usage: Add it once on a page to your page-level footer, typically a `footer` element.

Things To Try Yourself With WAI-ARIA

- I encourage you to try some assistive technology and see how HTML5 markup with ARIA can enhance the experience.
- NVDA (Windows, free download at: http://www.nvds-project.org/).
- VoiceOver (free as part of Mac OSX and IOS4+).
- JAWS (Windows, free trial for 30 days available at: http://www.freedomscientific.com).
- These three are among the most advanced screen readers available today. I strongly suggest you try one and test some of your markup in a screen reader as part of your normal development process.

More Accessible Web Pages With WAI-ARIA

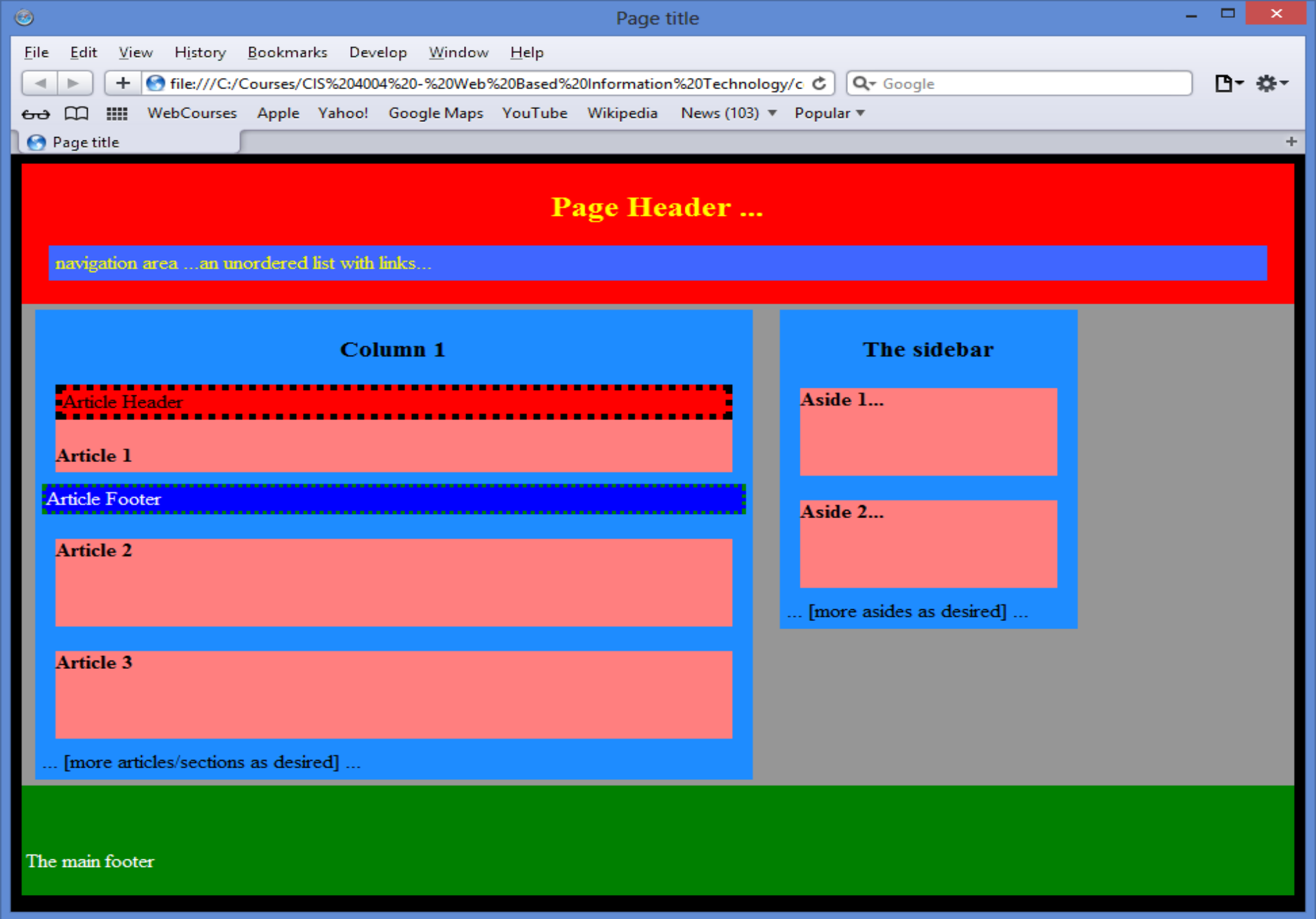
- When you use semantically correct HTML5 markup and augment it with the ARIA roles, one day assistive technology will be able to navigate easily to certain areas of the page content.
- However, at the moment, there is limited screen reader support not only for HTML5 but also for ARIA elements.
- HTML5 validation accepts ARIA roles. HTML4 validation does not, you will get validation errors with roles inserted into the markup. However, STOP USING HTML4!
- The use of roles also provides you with a pretty nifty CSS hook that adds to your arsenal of CSS selectors…see next page.

More Accessible Web Pages With WAI-ARIA

- Suppose that you have several headers or footers on a page, but you want to style the main page header and the main footer differently. You target them in the CSS using their roles as shown below:

creating-generic-containers – without CSS copy.html

```
/* to style all headers */
header {background-color:red; border: 5px dotted black;}

/* to style the main header - which might have a logo in it */
header[role="banner"] {background-color:red; border:5px solid red;
                       color:yellow;}

/* to style all footers */
footer {background-color:blue; border: 3px dotted green; color:white;}

/* to style the main footer - likely containing copyright info */
footer[role="contentinfo"] {clear:left; background-color:green;
                            border: 3px solid blue;}
```



Page title
File Edit View History Bookmarks Develop Window Help
file:///C:/Courses/CIS%204004%20-%20Web%20Based%20Information%20Technology/c
Google
WebCourses Apple Yahoo! Google Maps YouTube Wikipedia News (103) Popular
Page title
Page Header ...
navigation area ...an unordered list with links...
Column 1
Article Header
Article 1
Article Footer
Article 2
Article 3
... [more articles/sections as desired] ...
The sidebar
Aside 1...
Aside 2...
... [more asides as desired] ...
The main footer