WA2256 Responsive Mobile Web Development with HTML5, CSS3, JavaScript, and jQuery Mobile

Student Labs

Web Age Solutions Inc.

Table of Contents

Lab 1 - Create a Basic HTML5 Layout....................3
Lab 2 - New HTML5 Semantic Elements....................10
Lab 3 - Add More Semantic Value to Your Page with HTML5....................16
Lab 4 - Improve Your Forms Using HTML5....................20
Lab 5 - Drawing Using the Canvas Element....................39
Lab 6 - Dynamic Drawing Using the Canvas Element....................46
Lab 7 - Intro to CSS3....................57
Lab 8 - Applying Basic Styles....................61
Lab 9 - Modify Text Styles....................71
Lab 10 - Control Element Spacing with Box Properties....................81
Lab 11 - Media Queries and Responsive Design....................93
Lab 12 - Intro to JavaScript....................106
Lab 13 - Basic JavaScript Syntax....................110
Lab 14 - JavaScript Functions....................121
Lab 15 - Arrays in JavaScript....................128
Lab 16 - Objects in JavaScript....................139
Lab 17 - Getting Started with jQuery Mobile....................153
Lab 18 - Exploring Mobile UI Patterns....................159
Lab 19 - More on Selectors....................171
Lab 20 - Dynamic Style Class Assignment....................178
Lab 21 - DOM Manipulation....................181
Lab 22 - Lists....................185
Lab 23 - Buttons....................190
Lab 24 - Forms....................196
Lab 25 - Geolocation....................202
Lab 26 - Getting Started With Bootstrap....................211
Lab 27 - Simple Components....................215

Lab 1 - Create a Basic HTML5 Layout

In this lab, you will create an HTML5 template which will serve as the starting point for creating new HTML files in upcoming labs.

This lab will help you understand what a very simple HTML5 document looks like. Specifically, you will learn to declare the DOCTYPE and character encoding.

When creating HTML5 files, you will use Notepad as a text editor. Of course, you're free to use another text editor if you'd prefer.

Also, we'll be using Google Chrome as a web browser, since it provides the best support for HTML5 in a Windows environment.

Note: On the Mac, Safari provides the best support for HTML5. Feel free to experiment with other web browsers, such as Firefox 4 and Internet Explorer 9. However, we can't guarantee that all the HTML5 pages you create will render properly in those other browsers. If you want to see how well your browser supports HTML5, navigate to **http://www.html5test.com** from within the associated browser.

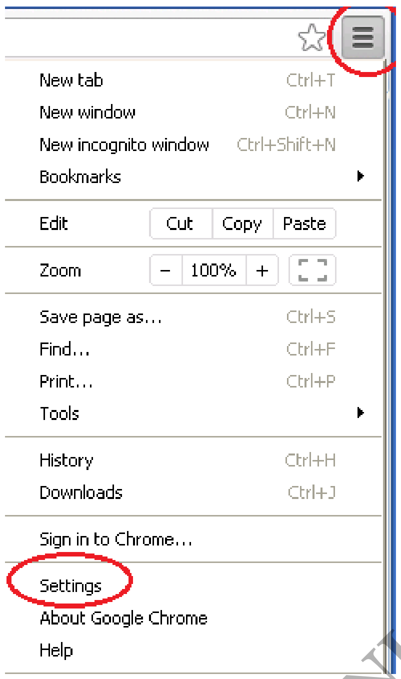
Note: These labs were written using Chrome 23.0.x. If you're using another version of Chrome, you may see slight differences in the GUI.

Part 1 - Clear the Browser Cache and Location Settings

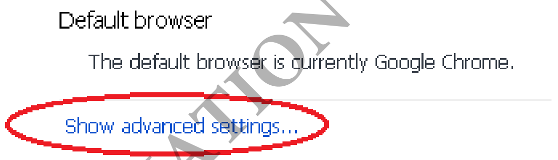
In this part, you will clear Chrome's cache and location settings. This will help ensure that any cached files and stored location settings will not interfere with any of the labs.

__1. Open Chrome by selecting **Start->All Programs->Google Chrome->Google Chrome**.

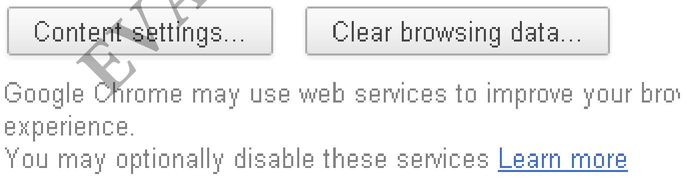
__2. Click the icon to the right of the location bar and select **Settings** from the drop-down menu. Older Chrome versions will show Options instead of Settings.



__3. Under Settings expand Show advanced settings...


Default browser
The default browser is currently Google Chrome.
Show advanced settings...

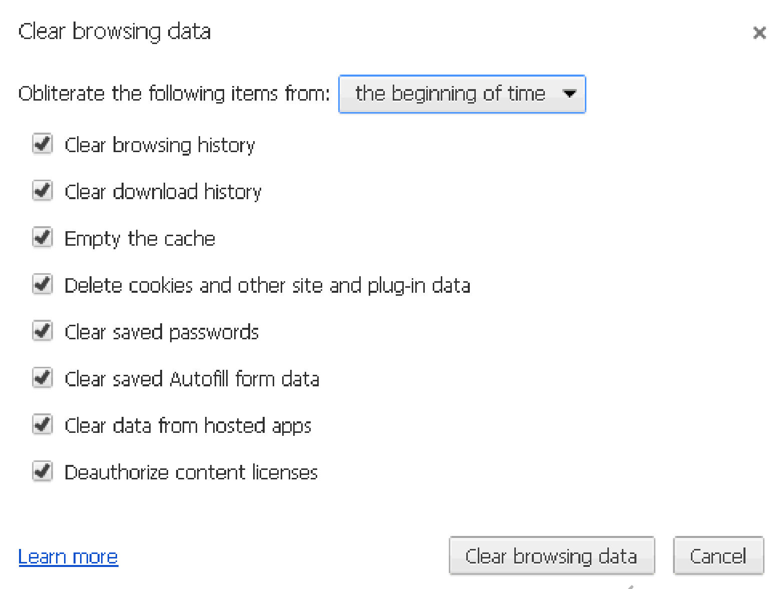
__4. Under Privacy click the **Clear browsing data** button.


Content settings...
Clear browsing data...
Google Chrome may use web services to improve your brow
experience.
You may optionally disable these services Learn more

__5. Check all the checkboxes.

__6. Select **the beginning of time** option from the **Obliterate the following items from** drop-down box.

__7. Click the **Clear browsing data** button.


Clear browsing data

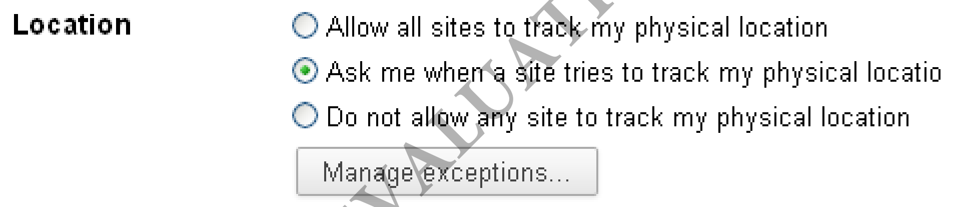
Obliterate the following items from: the beginning of time

- [✓] Clear browsing history
- [✓] Clear download history
- [✓] Empty the cache
- [✓] Delete cookies and other site and plug-in data
- [✓] Clear saved passwords
- [✓] Clear saved Autofill form data
- [✓] Clear data from hosted apps
- [✓] Deauthorize content licenses

Wait for the clearing process to complete.

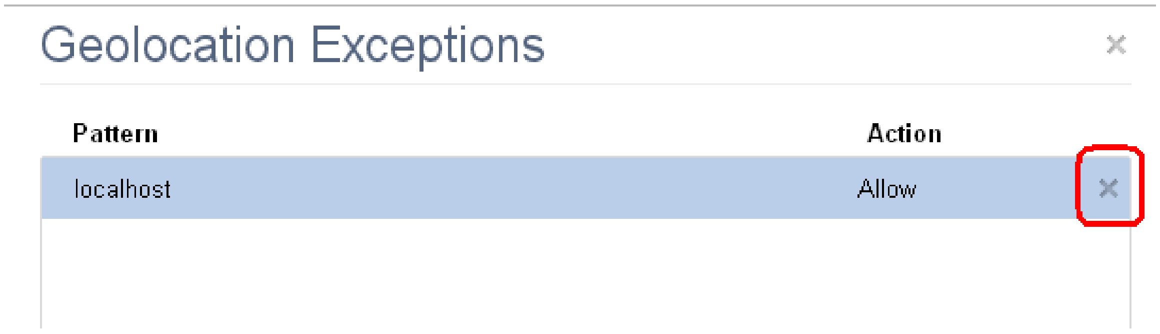
__8. Click the **Content settings** button, which is located to the left of the **Clear browsing data** button.

__9. Scroll down in the browser and click the **Manage exceptions** button under the **Location** section.


Location

- ○ Allow all sites to track my physical location
- ● Ask me when a site tries to track my physical locatio
- ○ Do not allow any site to track my physical location

__10. If any exceptions appear in the panel, select the exception and click the **X** to the right of it to delete the exception.



Pattern	Action
localhost	Allow

__11. Click the **X** in the upper right corner of the **Geolocation Exceptions** panel to close the panel.

__12. Click the **X** in the upper right corner of the **Content Settings** panel to close the panel.

__13. Close the browser.

Part 2 - Create the Template

In this part, you will create the HTML5 template. The template will consist of the following elements:

- DOCTYPE
- html
- head
- meta (used to specify character encoding)
- title
- body

__1. Run the following program to run a specialized editor. You do not need to use anything but a regular text editor for the labs but this program has some tools that can make it easier to work with the web pages you will be editing.

```
C:\Software\NotepadPlus\notepad++.exe
```

__2. If you are asked to install plug-ins simply click **Cancel**.

__3. If there is any opened file, close it.

__4. Enter the **DOCTYPE** declaration, followed by the **html** element:

```
<!DOCTYPE html>
<html>
```

The DOCTYPE declaration must be the very first element in your HTML5 document. It even must appear before the html element.

Note: For those of you who are familiar with XML and XHTML (an XML compliant version of HTML), the DOCTYPE element is used to indicate the document type definition (DTD) that an XML document adheres to. A DTD is used to specify the grammar of an XML document.

However, an HTML5 document is not XML, so the DOCTYPE element is not required for this purpose. Instead, it's used to trigger "no-quirks mode" (a.k.a. "standards mode"). "No-quirks mode" implies the browser renders the page according to the HTML5 and CSS standards. "Quirks mode" implies it doesn't and instead tries to maintain backward compatibility with legacy pages created for older browser versions.

Also, notice that in HTML5 the DOCTYPE element is kept very minimal and simply includes the root element name. Normally, when entering a DOCTYPE element in an XHTML/XML document, you would need to specify several other pieces of information, including a URL pointing to the DTD (see example below). However, since the DOCTYPE element is not being used to reference a DTD, we can omit this additional information.

e.g.,

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

Note: The DOCTYPE element is case insensitive. You could have written the following instead:

```
<!DOCTYPE HTML>

<!doctype html>
```

__5. Define the document's character encoding inside the head element as shown below.

```
<head>
   <meta charset="utf-8"/>
   <title>HTML 5 Template</title>
</head>
```

Note: The character encoding declaration is shorter than the one you're accustomed to writing in HTML4:

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

Note: HTML5 is not XML. Hence, element names are not case sensitive, elements don't require both a start and end tag, and attribute names don't need to be enclosed in quotes. Hence, you could have expressed the meta element in any of the ways listed below:

```
<META CHARSET="utf-8"/>

<META CHARSET=utf-8/>

<meta charset="utf-8">

<meta charset=utf-8/>

<Meta CharSet=Utf-8>
```

Having said that, we will use XML syntax going forward.

__6. Finally, add the **body** element and the **html** close tag to the document:

```
<body>
   Contents goes here...
</body>

</html>
```

Next, you will save the HTML file to the Apache Web Server document root.

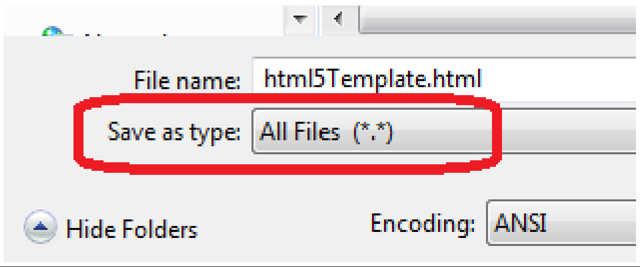
__7. Open Windows Explorer.

__8. Create a folder called **layout** under the following directory (the Apache folder may be installed in another location, check with your instructor):

```
C:\Apache2.2\htdocs
```

__9. Save the HTML file as **html5Template.html** within the layout folder.

Note: If you're using Notepad to save the file, make sure you choose **All Files (*.*)** in the **Save as type** field. Otherwise, a **txt** extension will automatically be added to the file name.



File name: html5Template.html

Save as type: All Files (*.*)

Encoding: ANSI

Part 3 - Test

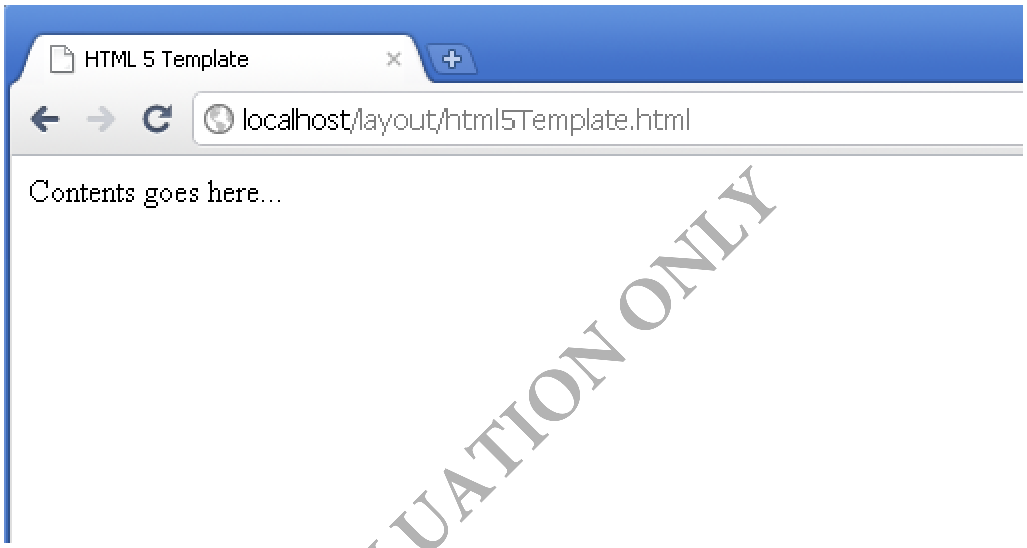
In order to test the HTML5 document, we'll use Google Chrome, since it provides the best support for HTML5.

__1. Open the **Chrome** browser.

__2. Enter the following URL into the address bar:

```
http://localhost/layout/html5Template.html
```

__3. You should see the following screen:



Contents goes here...

__4. Close the all.

Part 4 - Review

In this lab, you created a template for your HTML5 files. In the process, you learned how to declare DOCTYPE and meta elements.

Lab 2 - New HTML5 Semantic Elements

In this lab, you will learn how to use some of the new semantic elements that were introduced in HTML5: header, footer, section and article. *Semantic elements* have no specific visual styling, but are used to provide structure and meaning to the content.

Prior to HTML5, the only semantic elements were div (block level) and span (inline), which were supposed to serve as generic containers. Many developers used div elements to structure the page into divisions or sections and identified them as header, footer, etc.

e.g.,

```
<div id="header">
   <h1>Welcome to My Site</h1>
</div>
...
<div id="footer">
   <p>Copyright 2011 WebAge Solutions Inc.</p>
</div>
```

With the introduction of the new semantic elements in HTML5 (e.g., header, footer, section, and article), it is no longer necessary to use div elements to specify different regions of the page. For example, if you use the header element, the browser will unambiguously know this corresponds with the header section of the page. Keep in mind that it is still possible and even desirable to style the new elements, since they don't have any default visual appearance.

Part 1 - Create a Layout using HTML5 Semantic Elements

In this part, you will create a new HTML5 file that makes use of the header, footer, section, and article elements.

__1. Run the following program to run a specialized editor. You do not need to use anything but a regular text editor for the labs but this program has some tools that can make it easier to work with the web pages you will be editing.

```
C:\Software\NotepadPlus\notepad++.exe
```

__2. If you are asked to install plug-ins simply click **Cancel**.

__3. If there is any opened file, close it.

__4. Open the template file you created in the previous lab, **html5Template.html**, and save it as **pageLayout.html** within the same folder (C:\Apache2.2\htdocs\layout).

__5. Change the **title** content to:

```
<title>HTML 5 Layout</title>
```

__6. Remove the content inside the body element.

Now, we will introduce our new semantic elements.

First we will create a header using the *header* element. The header element will be the first element on the page and normally contains the title of the website, the organization's logo, a link back to the home page, etc. Our header will simply contain the title of the website.

__7. Add the **header** element just after **<body>**:

```
<header>
   <h1>Welcome to My Site</h1>
</header>
```

Next we will create a footer using the *footer* element. The footer element should be the last element on the page and normally contains copyright information and links to other information, such as the company's privacy policy, accessibility policy, and about us page. Our footer will simply contain the copyright information.

__8. Add the **footer** element after the **header** element:

```
<footer>
    <p>Copyright 2013 WebAge Solutions Inc.</p>
</footer>
```

Next we will focus on the content section of the page. HTML5 does not have a content element. Instead, we use an *article* element to define a chunk of content that can live on its own. Essentially, the content is supposed to be reusable outside of your web site. Article does not literally mean a news article. It could also be a blog posting, a forum posting, product information, etc. We can potentially have several article elements inside the content section of our page.

A *section* element corresponds with a collection of text elements and typically includes a heading. Unlike an article, a section is not meant to live on its own (i.e., not reusable). It's used mainly to apply group styling or to hide/show/collapse content. It's common for an article to be broken up into several sections using the section element. It can also be used for identifying regions of a page (e.g., introduction, news items, contact information).

__9. Add an article element before the footer tag:

```
<article>
      <h1>Acme A180 Wood Baseball Bat</h1>

</article>
```

__10. Insert the following **section** elements before **</article>**. If you'd prefer not to type all the code, feel free to copy and paste it from C:\LabFiles\pageLayout.html.

```
<section>
   <h2>Product Description</h2>
   <p>Price: $99.99</p>
   <p>Specifications:</p>
   <ul>
      <li>Northern white ash wood</li>
      <li>Natural finish</li>
      <li>32, 33, or 34 inches long</li>
   </ul>
</section>
<section>
   <h2>Product Comparison</h2>
   <p>This bat is comparable to the Batt Bros. GX300 and King Bats
TR18, but costs less, offers a better sweet spot, and a better feel due
to its long, thin tapered handle.</p>
</section>
<section>
   <h2>User Review</h2>
   <p>Posted by Bob on September 29, 2010</p>
   <p>I love this wooden baseball bat!  It's reasonably priced, very
light weight, and has yet to break.</p>
</section>
```

__11. Save and close the file.

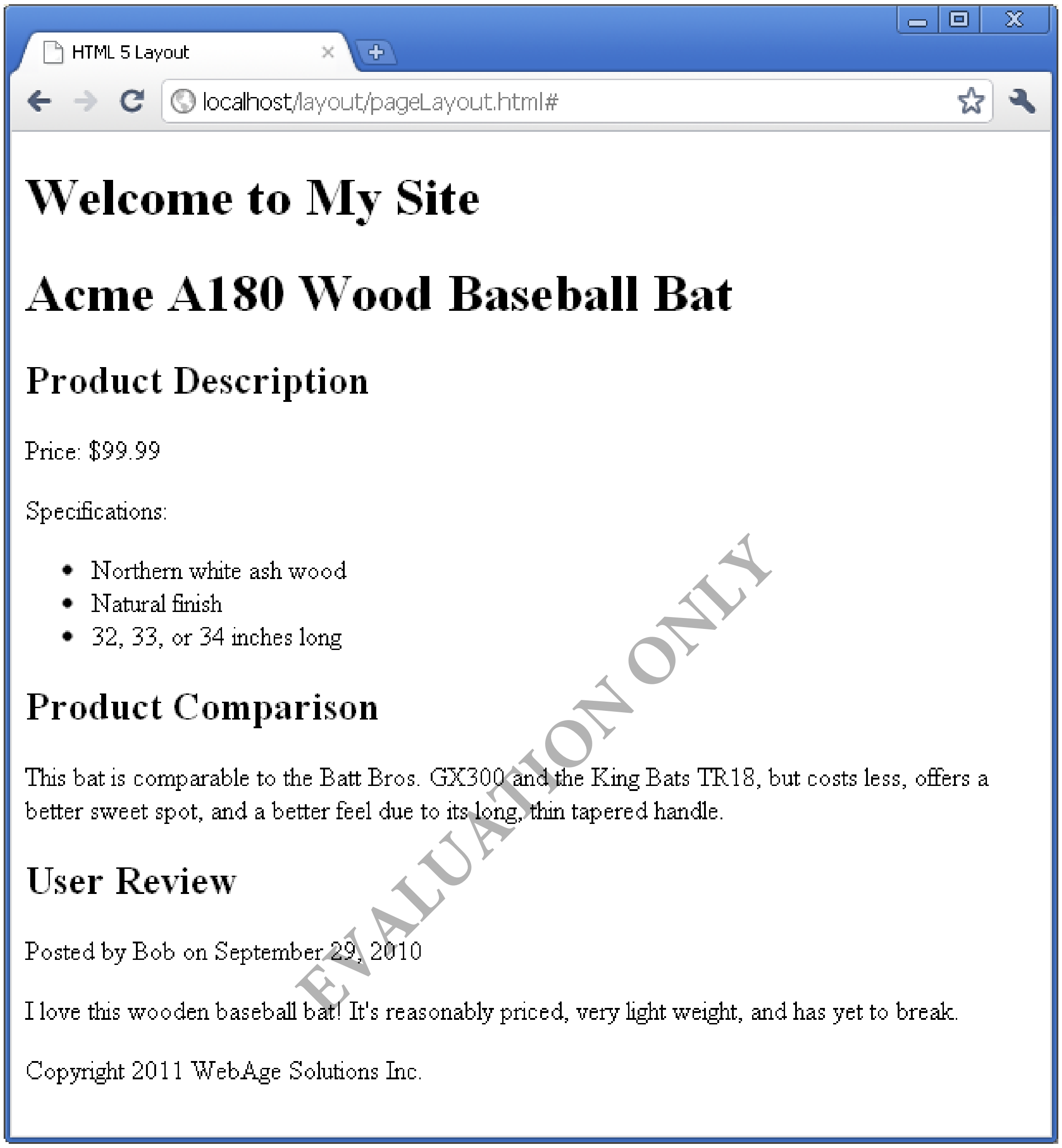
Part 2 - Test

__1. Open Google Chrome.

__2. Navigate to the following URL:

```
http://localhost/layout/pageLayout.html
```

__3. You should see the following screen:



Welcome to My Site

Acme A180 Wood Baseball Bat

Product Description

Price: $99.99

Specifications:

- Northern white ash wood
- Natural finish
- 32, 33, or 34 inches long

Product Comparison

This bat is comparable to the Batt Bros. GX300 and the King Bats TR18, but costs less, offers a better sweet spot, and a better feel due to its long, thin tapered handle.

User Review

Posted by Bob on September 29, 2010

I love this wooden baseball bat! It's reasonably priced, very light weight, and has yet to break.

Copyright 2011 WebAge Solutions Inc.

As expected, there's no visual styling of the header, footer, section, and article elements. The elements simply specify the page structure to the browser. We will address styling in the next section.

Part 3 - Visually Style Semantic Elements

In this part, you will create a CSS file and link it to the HTML page.

__1. Create a new file called **styles.css** in the **C:\Apache2.2\htdocs\layout** folder.

__2. Open this file using Notepad++.

__3. Add the following style rules to the style sheet:

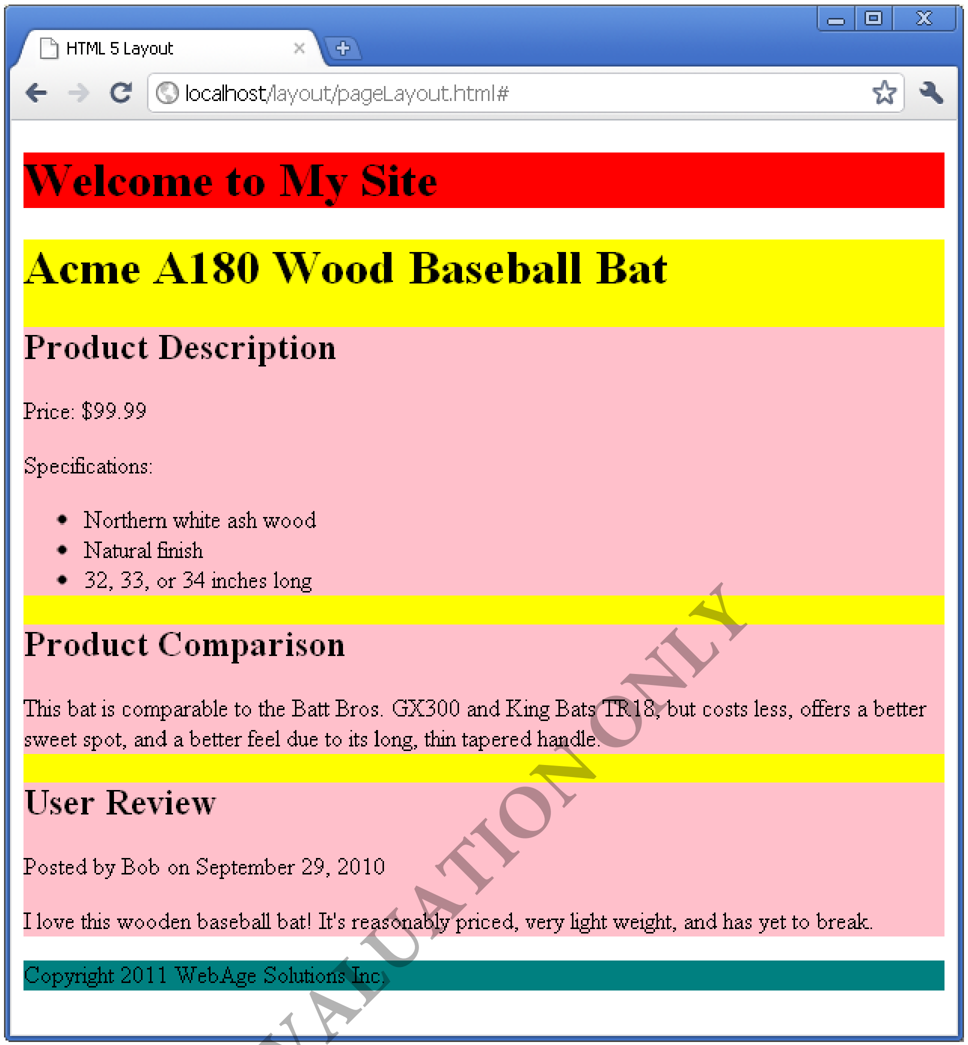
```
header {background-color:red}
footer {background-color:teal}
article {background-color:yellow}
section {background-color:pink}
```

__4. Save and close the file.

__5. Open the **pageLayout.html** file and link the HTML page to the stylesheet by adding the following **link** element before **</head>**:

```
<link rel="stylesheet" type="text/css" href="styles.css">
```

__6. Save and then refresh the page in the browser. You should see the following screen:



Welcome to My Site

Acme A180 Wood Baseball Bat

Product Description

Price: $99.99

Specifications:

- Northern white ash wood
- Natural finish
- 32, 33, or 34 inches long

Product Comparison

This bat is comparable to the Batt Bros. GX300 and King Bats TR18, but costs less, offers a better sweet spot, and a better feel due to its long, thin tapered handle.

User Review

Posted by Bob on September 29, 2010

I love this wooden baseball bat! It's reasonably priced, very light weight, and has yet to break.

Copyright 2011 WebAge Solutions Inc.

__7. Close all open files and browsers.

Part 4 - Review

In this lab, you learned how to use some of the new semantic elements (i.e., header, footer, article, and section), in place of the old div and span elements, to specify the structure of a page.

Lab 3 - Add More Semantic Value to Your Page with HTML5

In this lab, you will learn how to use some of the other semantic elements that were introduced in HTML5:

nav – defines a section of the page that contains links to other pages or to other parts within a page (e.g., page anchors). nav elements are often used to create navigation bars. When linking to other pages, the HTML5 spec recommends to nest only major navigational links inside a nav element, not every group of links. For example, search result links should not be nested inside a nav element.

aside – contains content that is tangentially related to the surrounding content and can be considered separate from it. aside elements are often used to create sidebars in an article, which usually contain either "pull quotes" (quotes displayed in a distinctive font) or additional content not covered in the main article. A nav element can also be nested inside an aside element, since navigational links are tangential to the content.

time – used to encode a date, time, or both, in a machine readable fashion, while still displaying some value to the user. For example, you may encode the date 2011-12-25 and display the value "Christmas Day". Since the date/time information is encoded, the time element can be used by a browser to offer to add a reminder of some event (e.g., anniversary) to a user's calendar.

Part 1 - Create Page Layout with other Semantic Elements

In this part, you will add a nav, aside, and time element to the HTML page you created in the previous lab.

__1. Open the layout file that you created in the previous lab called **pageLayout.html**

First we will create an aside element that contains a nav element, which will be used in conjunction with one another to create a navigation bar.

__2. Add the **aside** and nested **nav** element after **</header>**:

```
<aside>
   <nav>
      <a href="#">Home</a><br/>
      <a href="#">About Us</a><br/>
      <a href="#">Contact</a><br/>
   </nav>
</aside>
```

Note: Currently the links are dummied up and don't take us anywhere.

Next, we will use a time element to indicate when the article was posted.

__3. Change the paragraph element indicating when the user review was posted to the following:

```
<p>Posted by Bob on
   <time datetime="2010-09-29T15:35Z">September 29, 2010 at 3:35
PM</time>
</p>
```

Notice the following:

- The information that's displayed is different than what's encoded inside the element.
- The date is expressed in the format: yyyy-mm-dd.
- The time is expressed in the format: hh:mm. You can also specify seconds and milliseconds: hh:mm:ss.sss (e.g., 15:35:05.313).
- The time is expressed in military time (24 hr format).
- The "T" is a literal which is used to separate the date from the time.
- The "Z" is a literal which indicates the time is being expressed using Coordinated Universal Time (UTC), which is a replacement for Greenwich Mean Time (GMT).

Note: You can omit the datetime attribute, provided you don't mind displaying the machine readable format to the user.

<time>2010-09-29T15:35Z</time>

__4. Save and close the file.

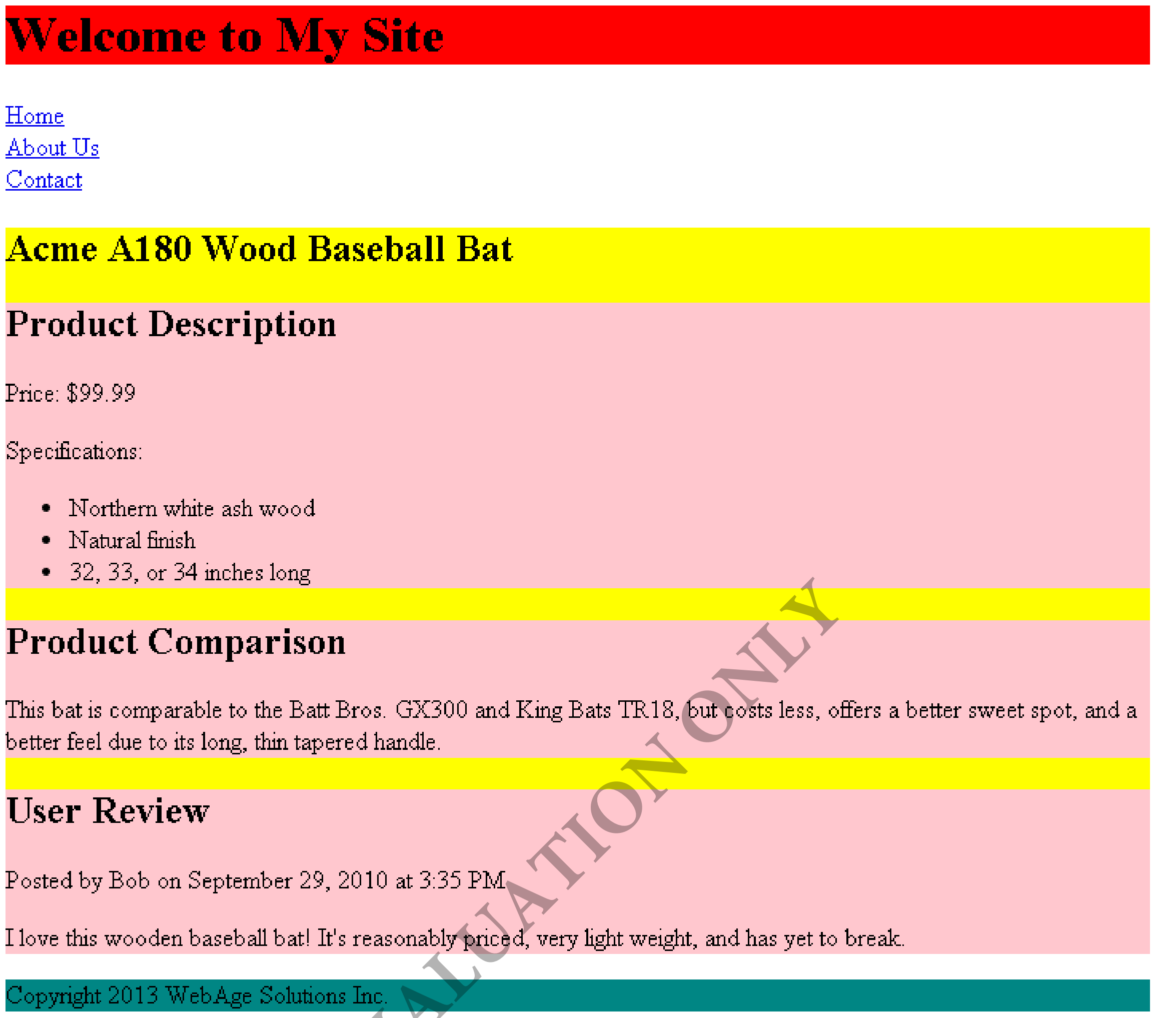
Part 2 - Test

__1. Open Google Chrome.

__2. Navigate to the following URL:

```
http://localhost/layout/pageLayout.html
```

__3. You should see the following screen:



Welcome to My Site

Home
About Us
Contact

Acme A180 Wood Baseball Bat

Product Description

Price: $99.99

Specifications:

- Northern white ash wood
- Natural finish
- 32, 33, or 34 inches long

Product Comparison

This bat is comparable to the Batt Bros. GX300 and King Bats TR18, but costs less, offers a better sweet spot, and a better feel due to its long, thin tapered handle.

User Review

Posted by Bob on September 29, 2010 at 3:35 PM

I love this wooden baseball bat! It's reasonably priced, very light weight, and has yet to break.

Copyright 2013 WebAge Solutions Inc.

Notice that the navigation links are displayed, as are both the date and time.

Let's modify the style sheet a little so that the navigation bar has a background color and is displayed on the left side of the page.

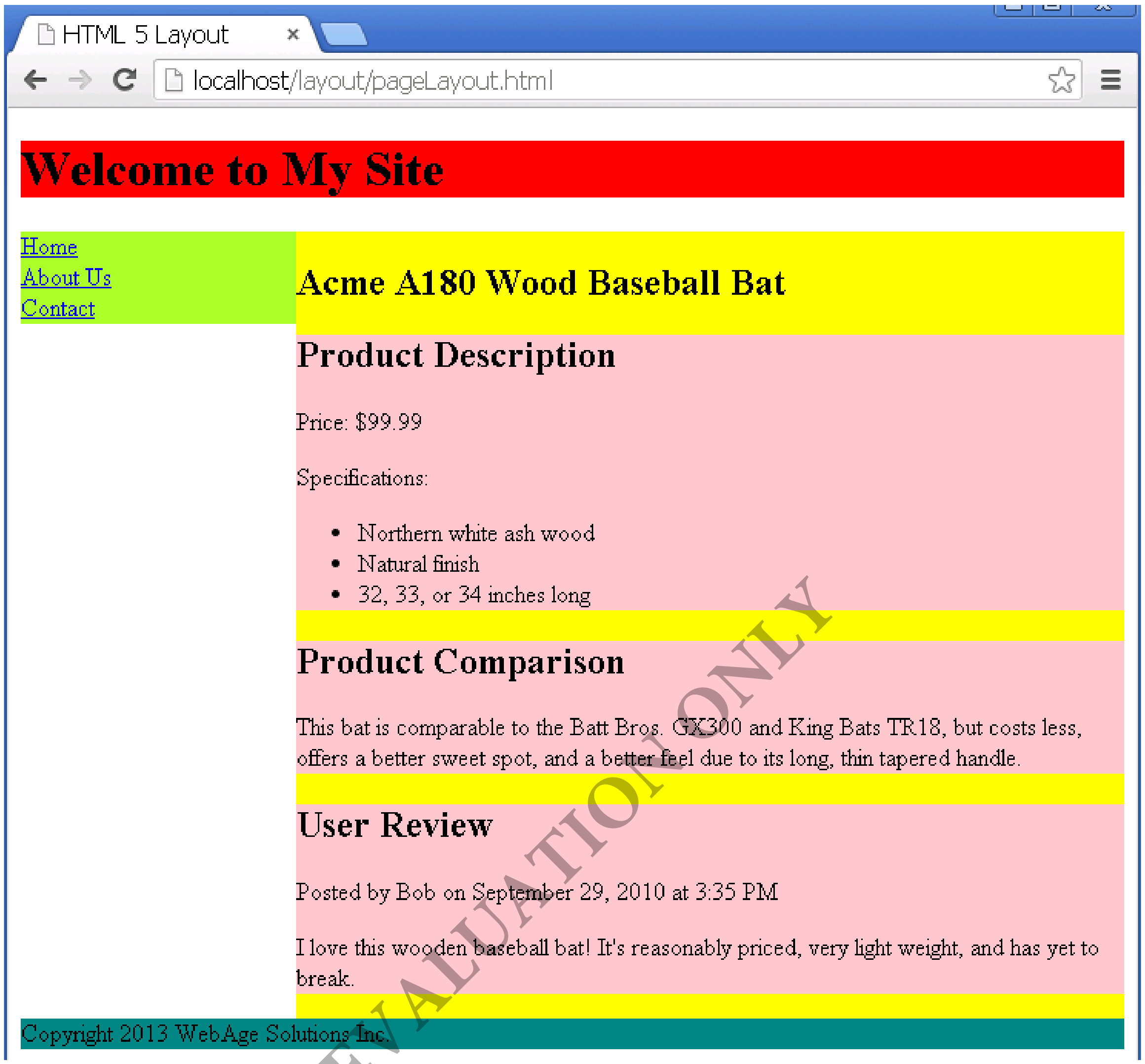
__4. Open the **styles.css** file that you created in the previous lab.

__5. Make the following changes/additions in bold:

```
header {background-color:red; clear:both}
footer {background-color:teal; clear:both}
article {background-color:yellow; float:right; width:75%}
section {background-color:pink}
nav {background-color:greenyellow; float:left; width:25%}
```

__6. Save and close the file.

__7. Refresh the page in the browser. You should see the following screen:



Welcome to My Site

Home
About Us
Contact

Acme A180 Wood Baseball Bat

Product Description

Price: $99.99

Specifications:

- Northern white ash wood
- Natural finish
- 32, 33, or 34 inches long

Product Comparison

This bat is comparable to the Batt Bros. GX300 and King Bats TR18, but costs less, offers a better sweet spot, and a better feel due to its long, thin tapered handle.

User Review

Posted by Bob on September 29, 2010 at 3:35 PM

I love this wooden baseball bat! It's reasonably priced, very light weight, and has yet to break.

Copyright 2013 WebAge Solutions Inc.

__8. Close all open files and browsers.

Part 3 - Review

In this lab, you learned how to use some additional semantic elements introduced in HTML: nav, aside and time. The nav and aside elements were used to add a navigation bar to the page you created in the previous lab. The time element was used to encode and display both the date and time.