HTML5 Introduction

Unit: Programming Languages

Problem Area: HTML

Lesson: HTML5 Introduction

Student Learning Objectives.

Instruction in this lesson should result in students achieving the following objectives:

1. **Summarize new HTML5 elements.**
2. **Create sites using HTML5.**
3. **Integrate scalable vector graphics (SVG) with HTML5.**

Resources.

The following resources may be useful in teaching this lesson:

E-unit(s) corresponding to this lesson plan. CAERT, Inc. http://www.mycaert.com.

"Atom: A Hackable Text Editor for the 21st Century," *GitHub*. Accessed Sept. 9, 2016. https://atom.io/.

Bensound. Accessed Sept. 9, 2016. http://www.bensound.com/.

"Free Stock Footage and Video Loops," *Partners in Rhyme*. Accessed Sept. 9, 2016. http://www.partnersinrhyme.com/video/video/7599.html.

Index of /SVG/tools/svgweb/samples/svg-files. Accessed Sept. 9, 2016. https://dev.w3.org/SVG/tools/svgweb/samples/svg-files/.

Jankov, Jacob. "SVG Tutorial," *Jenkov.com*. Accessed Sept. 9, 2016. http://tutorials.jenkov.com/svg/index.html.

"Lorem Ipsum," *Lipsum.com*. Accessed Sept. 9, 2016. http://www.lipsum.com/.

"Online XML Editor," *TutorialsPoint*. Accessed Sept. 9, 2016. http://www.tutorialspoint.com/online_xml_editor.htm.

Copyright © by CAERT, Inc. | Reproduction by subscription only. | L780011

"SVG Editor/Viewer Online," *Free Code Format*. Accessed Sept. 9, 2016. http://www.freecodeformat.com/svg-editor.php.

"Underwater Scene (2) – Free Stock Video," *OrangeHD*. Accessed Sept. 9, 2016. http://orangehd.com/blog/underwater-scene-2-free-stock-video/.

Equipment, Tools, Supplies, and Facilities

- ✓ Overhead or PowerPoint projector
- ✓ Visual(s) from accompanying master(s)
- ✓ Copies of sample test, lab sheet(s), and/or other items designed for duplication
- ✓ Materials listed on duplicated items
- ✓ Computers with printers and Internet access
- ✓ Classroom resource and reference materials

Key Terms.

The following terms are presented in this lesson (shown in bold italics):

- audio element
- canvas element
- extensible markup language (XML)
- float property
- hypertext markup language (HTML)
- HTML5
- living standards
- root element
- semantics
- semantic elements
- scalable vector graphics (SVG)
- source element
- SVG element
- vector graphics
- video element

Interest Approach.

Use an interest approach that will prepare the students for the lesson. Teachers often develop approaches for their unique class and student situations. A possible approach is included here.

> *HTML5 is the latest version of HTML. It introduces many exciting new features that can be incorporated to create a data-rich website.*

Copyright © by CAERT, Inc. | Reproduction by subscription only. | L780011

SUMMARY OF CONTENT AND TEACHING STRATEGIES

Objective 1: Summarize new HTML5 elements.

Anticipated Problem: What new elements were introduced in HTML5?

I. New HTML5 elements

A. HTML4 versus HTML5

1. ***Hypertext markup language (HTML)*** is the language used to create webpages. It was originally designed to display text data and links in a webpage. When a link was clicked, the browser window moved to another webpage. Since the original days of the Web, the type of content webpages are expected to render has proliferated. Webpages now contain video, audio, and games. Since these types of data were not part of the original HTML standard, these were incorporated into webpages by using plug-ins (add-ons for a program to increase functionality). As plug-in versions changed, it became a challenge to keep the plug-in versions updated. For example:
   a. Webpages were initially designed for viewing on a PC monitor. Increasingly, the computer of choice has become a mobile device whose display is smaller than that of the PC. This led to webpages that did not render properly on smaller devices.
   b. Additionally, including geographical information in a webpage was not easy in HTML4. Eventually, the HTML5 recommendation was released (in 2014) to address these shortcomings.
2. ***HTML5*** is the fifth and newest major standard for HTML. HTML5 supports all the features of HTML4 pages and incorporates new ones. In addition, HTML5 contains specifications called ***living standards***—features continuously being developed and tested by browser vendors and the user community. When these new features become stable and widely adopted, they are incorporated into the HTML5 standards. This process fosters an environment of ongoing innovation and ensures that standards do not lag behind current developments.
   a. See VM–A to view differences between HTML4 and HTML5. The graphic depicts some of the important differences between HTML4 and HTML5. The new elements shown in the graphic are geared toward incorporating multimedia resources.
   b. The form element hosts many new input elements. The value attribute of the input element can now take in additional values, allowing for different types of input data. The third column shows elements deprecated in HTML5. [See VM–B.]

Copyright © by CAERT, Inc. | Reproduction by subscription only. | L780011

B. New HTML5 multimedia elements

1. The ***audio element*** is a feature used to play acoustic files. Audio files with the extensions MP3, OGG, and WAV are permitted in this element. The audio element may be written in one of two formats. [See VM–C.]
   a. With the source element: The audio element may be written as a two-sided element, with a start and an end tag. Nested inside the element is a one-sided element called "source." The ***source element*** is a feature that specifies the name of the audio file to be played (the src attribute specifies the location, or URL, of the audio file). Path considerations should be taken into account with the value placed in the src attribute. For instance, in the code shown in VM–C:
      (1) The MP3 file is placed in the same folder as the HTML file that contains the audio element to play it.
      (2) The text between the audio element's start and end tag is displayed when the page is rendered in a non-HTML5 compliant browser.
   b. Without the source element: The audio element may be written without the source element. In this format, only one tag is used, and the name of the file is specified using the src attribute of the audio element.
2. The ***video element*** is a feature used to play film/movie files. The file formats supported in the video element are MP4, WebM, and OGG. The .MOV format is not officially supported, but it currently works in Chrome. The video element works in the same fashion as the audio element and may be specified in one of two formats. Path considerations should be taken into account with the value placed in the src attribute. [See VM–D.]
3. The ***canvas element*** is a feature used to draw shapes and lines in an HTML file. Attributes of the canvas element can be used to render the border and color of the element. To draw shapes and lines inside the element, JavaScript code is needed. [See VM–E. This visual displays code to create a canvas element with shapes and lines inside it. JavaScript code is placed inside a script element and is shown highlighted in yellow. This lesson will not discuss JavaScript.]

C. New HTML5 form input element attributes

1. HTML5 introduced new form elements. These are identified using new values for the input attribute. [See VM–F. This visual displays the code and browser view of the new elements that hold color, date, email, and URL. The color element provides a color picker dialog where a specific color may be selected. The date element provides a month view where month, year, and date can be selected. It does not allow the user to input invalid month, date, or year. The email and URL fields accept strings that follow the pattern for email IDs and web URLs. The number attribute displays a textbox that only accepts numbers within the range specified by the max and min attributes.]
2. Validation of these new form elements is performed when the "submit" button is pressed. When errors are present in all three fields, only the first error message is displayed. [See VM–G.]

Copyright © by CAERT, Inc. | Reproduction by subscription only. | L780011

3. The input element supports a new attribute called "placeholder." [See VM–H. This visual displays examples of the placeholder attribute for email, URL, and text input elements. The placeholder text appears in gray lettering. When the mouse is clicked in a control, the text disappears. When text in one of these controls is deleted, the placeholder text reappears.]
4. Other new input element types are "datalist," "time," and "search." These provide additional functionality for forms.
5. Not all new HTML5 form elements work in all browsers. Code in this document was tested in Google Chrome Version 49.0.2623.112. [NOTE: To see browser compliance with HTML5 standards, open a browser window and navigate to the URL https://html5test.com/. It provides a numeric score regarding the HTML5 compliance of the browser product.]

Teaching Strategy: *Many techniques can be used to help students master this objective. Use VM–A through VM–H. Assign LS–A.*

Objective 2: Create sites using HTML5.

Anticipated Problem: What are HTML5 principles used to build a website?

II. HTML5 website creation

A. Semantic elements

1. HTML5 introduced semantic elements. ***Semantics*** are the inferred (indirect) meanings of a subject, such as a word or sentence to help interpret the content. ***Semantic elements*** are nuanced (somewhat fuzzy and open to interpretation) subjects that help the developer determine the purpose of the content in an HTML tag while being clear enough for a machine to interpret the tag in the same way. In HTML4, the div element was used to create sections in an HTML document, with each div element holding other elements inside it. HTML5 uses semantic elements, along with the div element.
2. See VM–I. This visual displays some of the new semantic elements introduced in HTML5 and the suggested visualization of a webpage with these elements. The "nav" element may appear across the top of the page or on the left side. The aside element is optional and does not need to be part of the webpage.
3. See VM–J. This visual displays the use of semantic elements and their rendering in a browser window.
4. Using the semantic elements by themselves produces a display as shown in VM–I. Instead, each of the semantic elements is treated as a block-level element and rendered one beneath the other. CSS styles are required to place the semantic elements in the page as needed.

B. The float property

1. The ***float property*** is a feature that alters the default placement of elements. It modifies the horizontal placement of elements and not the vertical place-

Copyright © by CAERT, Inc. | Reproduction by subscription only. | L780011

ment. [See VM–K. This visual displays two div elements with borders and width properties set. Since the div element is a block-level element, the second div is placed in a new line after the first div is rendered on screen. The width properties are provided in % units, so it can be set based on the size of the screen.]

2. See VM–L. This visual displays a modified version of the file used in VM–H. In VM–L, the div elements have settings for the float property. The first div element has a float property value of "left." This places it toward the left of the screen. The second div element has a float property value of "right," and this places it at the right edge of the screen. When the float property is set, the width property must be specified.
3. See VM–M. This visual contains three div elements. The one to the left and right of the screen are unchanged from the previous example. The div in the center was added to this example. Unfortunately, the float property does not have a "center" setting. Instead, it too, uses a float setting of "left." This causes the second div to be placed to the left of the area allocated to it, which is the portion of the screen available after the first div took over the left section of the screen.

C. Using CSS with semantic elements

1. CSS styles, such as "float" and "width," can be used to format semantic elements in a way consistent with their semantic meaning. [See VM–N. It displays code with semantic elements and the graphic from VM–I that shows the logical layout of semantic elements.]
2. The "section" element contains a style property called "overflow" that is set to "scroll." This causes the placement of scroll bars when content in the element is larger than the area on screen.
3. The "footer" element makes use of the "clear" style property and sets it to "both." This clears the float setting for this element, so it is not laid out relative to the positioning of the element before it, the "aside" element. The footer is rendered across the entire screen.
4. See VM–O. This visual shows the browser display of the file whose code is shown in VM–K. The website contains no design elements. This was done to focus on the semantic elements and their positioning within the page.
5. Even though the float property does not support a "center" value, other techniques can be used to center an element. When the margin property of an element is set to "auto," content inside the element is centered.
   a. See VM–P. In this visual image, the margin property of the body element is set to "auto." This causes the other elements in the document to appear centered.
   b. See VM–Q. In this visual image, the elements of the page appear centered. The width of the page is set to 800px, and the three elements are centered inside the specified width. This does not guarantee centering all page elements.

Copyright © by CAERT, Inc. | Reproduction by subscription only. | L780011

D. Alternate semantic layouts

1. Rather than the navigation element being placed at the left of the page, it may be laid above the section element, just beneath the header element. [See VM–R. This visual shows the navigation bar beneath the header element and spans the width of the page. The section element and the aside element are laid out underneath the nav element.]
2. See VM–S. This visual displays the code for the page. The code for the nav element is highlighted. It no longer contains a setting for the float property. Additionally, the style settings for the ul element and the li elements inside the nav element have been changed. The ul property of list-style-type is set to none, and this prevents the bullet from displaying at the start of the li element. By default, li elements are block elements, so each li element starts on a new line. In the code shown in VM–S, the display property of the li element is set to inline, which indicates that li elements are not required to start on separate lines.

E. CSS differences in browsers

1. While all browsers support CSS styles, each vendor has created enhancements to the standards. These enhancements are not supported across all browsers. The features are accessed using browser prefixes. [See VM–T. The browser prefixes used are shown in this visual.] In theory, some of these experimental features will eventually get incorporated into the W3C standard and will be accessible without using prefixes.
2. One feature that needs browser prefixes is the ability to create HTML elements with gradient colors instead of solid colors. [See VM–U. This visual displays a div element with a gradient background and the code to create the gradient.] Gradient backgrounds are now part of the W3C standards, but prefixes are still used for compatibility with older browsers.

Teaching Strategy: *Many techniques can be used to help students master this objective. Use VM–I through VM–U. Assign LS–B.*

Objective 3: Integrate scalable vector graphics (SVG) with HTML5.

Anticipated Problem: How is SVG integrated with HTML5?

III. SVG technology and HTML interaction

A. XML

1. ***Extensible markup language (XML)*** is a form of communication used as a way to describe data. The XML standard is the primary way to share structured information among applications. Both HTML and XML contain markup symbols to describe the contents of a file or a page. While HTML is useful in displaying content on the Web, it has its limitations. There is no way to add to the number of tags that are part of HTML. For example, h1 elements always render

Copyright © by CAERT, Inc. | Reproduction by subscription only. | L780011

larger than h2. The ul element always creates a list. In essence, HTML documents contain a mixture of content and style. For example:

   a. Suppose there is a need to display data in the form of a hierarchy with elements represented as branches and leaves. It is not possible to do so with HTML since no elements represent that layout semantically.
   b. While a combination of CSS and JavaScript can be used to produce the layout, it is not part of HTML specifications.

2. XML was developed as a way to create web content in a more extensible, customizable manner. Additionally, content is separated from style or display considerations.
3. XML was developed to help alleviate perceived problems of HTML. An XML file only maintains data, with no information about how it should be presented. The structure of an XML document is strictly enforced, and computer applications can read XML documents and validate them. Again, XML is the standard for sharing information among applications on the Web.

B. Creating XML documents

1. XML documents are ASCII documents. Creating an XML document is similar to creating an HTML document. They can be created in Notepad or NotePad++. XML, like HTML, is a tag-driven language. However, some basic differences exist between HTML and XML tags. For example, in XML:
   a. All tags are case sensitive.
   b. All open tags must have a close.
   c. All tags must be properly nested.
2. XML documents are stored in files with the extension ".xml." [See VM–V. This visual displays the code for an XML document called courses.xml. The first line in an XML document is a declaration statement that indicates the document is an XML document. Declaration statements start and end with a question mark as: <? and ?>. The first XML element in the file is "courses." This is referred to as the root element of the document.]
   a. All XML documents must contain one ***root element*** or the parent of all other elements in the document.
   b. In an XML document, all elements must be properly closed. [NOTE: Therefore, the last statement in the document is the end tag for the course's root element in VM–V.]
3. Multiple "course" elements are present in the VM–V document. Each has a corresponding end tag, and they are all nested properly inside the root element. The course elements contain attributes specified across the start tag of the course element. [NOTE: See VM–W. This visual depicts the XML document as an inverted tree, where each child course element is a branch of the course's root element.]

C. SVG

1. ***Scalable vector graphics (SVG)*** is a type of XML language that defines specific XML elements that can be used to render 2D graphics. The graphics produced by SVG are ***vector graphics*** (images made up of paths defined by a

Copyright © by CAERT, Inc. | Reproduction by subscription only. | L780011

start and an end point). In contrast, JPEG, GIF, and BMP images are made up of grids of pixels rather than paths. Vector graphic images are maintained as a series of mathematical equations of how a graphic should be drawn. For example:

a. SVG graphics may be resized without loss of precision.
b. SVG uses XML principles.
c. SVG elements are created as XML elements.

2. SVG files are a specific type of XML file. XML dictates that all elements must be written with start and end tags. The start tag for an SVG element is <svg>, and the end tag is </svg>.
   a. See VM–X. This visual displays the code for a simple SVG file and the file rendered in a browser window. The file is not displayed as a graphic because the browser cannot directly draw SVG graphics. This is in line with XML philosophy that content is kept separate from style information. The SVG file contains content, and additional information is needed to style the content maintained in the file.
   b. See VM–Y. This visual displays the same code as VM–X, with additional information. The browser version of the file is also shown. In the SVG start element, an attribute called "xmlns" (XML namespace) is specified with a URL address. This attribute states that the SVG element and its child tags belong to the namespace with the identifier "http://www.w3.org/2000/svg." This namespace defines the specifics of the SVG language, which is a type of XML language.

D. SVG elements and attributes
   1. SVG technology renders shapes in a 2D space. Therefore, it is important to understand the coordinate system used in SVG. As in geometry, the x-coordinate refers to placement on the horizontal axis, and the y-coordinate refers to placement on the vertical axis. [See VM–Z. The coordinate used in SVG graphics is shown in this visual. The top left-hand side corner holds the coordinates x=0 and y=0. Moving from the top left-hand side to the right (horizontally) increases the x-coordinate. Starting from x=0, y=0 downwards (vertically) increases the y-coordinate.]
   2. SVG provides a rich language set to render graphics. [See VM–AA. It lists commonly used SVG elements.] SVG elements by themselves cannot be used in a meaningful fashion without attribute settings for the elements. [See VM–BB. It lists attributes for SVG elements shown in VM–AA.]

E. Simple SVG graphic creation
   1. SVG elements, along with their attributes, can be used to render graphics on screen. The SVG Editor/Viewer Online at http://www.freecodeformat.com/svg-editor.php URL provides an easy mechanism for testing SVG code. The SVG code can be entered in the text box. When the "draw" button is clicked, it is shown on the same page in a view box.
   2. See VM–CC. This visual displays a simple SVG command and the shape it creates. The SVG element has a start and an end tag. Nested inside the SVG ele-

Copyright © by CAERT, Inc. | Reproduction by subscription only. | L780011

ment is a <circle> element, which is coded with only the tag that combines the start and end tag, much like the img element. The attributes of the circle element specify the position, size, fill, and stroke of the rectangle to be rendered.

3. See VM–DD. This visual displays examples of SVG files with code to create various shapes and lines, along with their rendering in a browser window.

F. Working with SVG text

1. SVG supports an element called "text." It can be used to render text in an SVG document. [See VM–EE. This visual displays the attributes for the text element.] Unlike the SVG elements used to render shapes, the text element is a two-sided element. The text to be displayed is placed between the start and end element tags.
2. Note that some of the style properties mimic those that CSS supports, such as font-family and font-size. The font-family property supports the same values as the font-family property in CSS.

G. Incorporating SVG files with HTML

1. SVG files can be displayed in an img element, and the browser treats them in the same manner as image files. [See VM–FF. This visual displays the code for an SVG file and its usage in an img tag via the src attribute.]
2. It is rendered at the same position that the img tag is situated in the HTML document. Since the src attribute value does not contain any path information, the SVG file must be placed in the same folder as the HTML file that uses it in an img element.

H. The SVG element

1. HTML5 supports an element called the SVG element. The ***SVG element*** is a two-sided element placed between the start tag and end tag of this element. SVG elements are rendered using SVG principles. [See VM–GG. This file contains SVG code to render a circle and is the same code used earlier in a SVG file to create a circle.]
2. See VM–HH. This file incorporates a SVG element to create an h1 element with text that exhibits properties, such as stroke and fill.

Teaching Strategy: *Many techniques can be used to help students master this objective. Use VM–V through VM–HH. Assign LS–C.*

- **Review/Summary.** Use the student learning objectives to summarize the lesson. Have students explain the content associated with each objective. Student responses can be used in determining which objectives need to be reviewed or taught from a different angle. If a textbook is being used, questions at the ends of chapters may be included in the Review/Summary.

- **Application.** Use the included visual master(s) and lab sheet(s) to apply the information presented in the lesson.

Copyright © by CAERT, Inc. | Reproduction by subscription only. | L780011

Evaluation.

Evaluation should focus on student achievement of the objectives for the lesson. Various techniques can be used, such as student performance on the application activities. A sample written test is provided.

Answers to Sample Test:

Part One: Completion

1. living standards
2. video element
3. date attribute
4. nav element
5. circle element
6. img elements

Part Two: True/False

1. T
2. F
3. T
4. F
5. T
6. F

Part Three: Short Answer

1. The statement to set up a form element that requests the user to enter a valid email address is:
   <input type="email" name="inpemail"/>
2. The code to set up a circle using SVG elements is:
   <circle cx="40" cy="40" r="24" style="stroke:red; fill:green" />
3. The code to render text using SVG elements is:
   <text x = "36" y = "36"
   font-size:36;" >
   Hello!
   </text>

Copyright © by CAERT, Inc. | Reproduction by subscription only. | L780011

Sample Test

Name _____

HTML5 Introduction

Part One: Completion

Instructions: **Provide the word or words to complete the following statements.**

1. The specifications that are in use but have not yet been incorporated into official standards are called _____.
2. The element that can be used to play films or tapes in an HTML document is called a/an _____.
3. The attribute on the input element that allows the user to enter a valid date is the _____.
4. The semantic element that is supposed to hold navigation links in a website is called a/an _____.
5. The SVG element that can be used to render a circle on a page is a/an _____.
6. The elements used to view SVG files in an HTML5 document are a/an _____.

Part Two: True/False

Instructions: **Write *T* for true or *F* for false.**

_____1. There is an input element that allows the user to select a color.
_____2. Semantic elements are automatically placed in appropriate locations by the system.
_____3. Semantic elements can be used in conjunction with CSS.
_____4. The float property can be used to center an element in a document.
_____5. SVG files are considered XML files.
_____6. SVG files are used to create 3D models.

Copyright © by CAERT, Inc. | Reproduction by subscription only. | L780011

Part Three: Short Answer

Instructions: **Answer the following.**

1. Write the statement to set up a form element that requests the user to enter a valid email address.

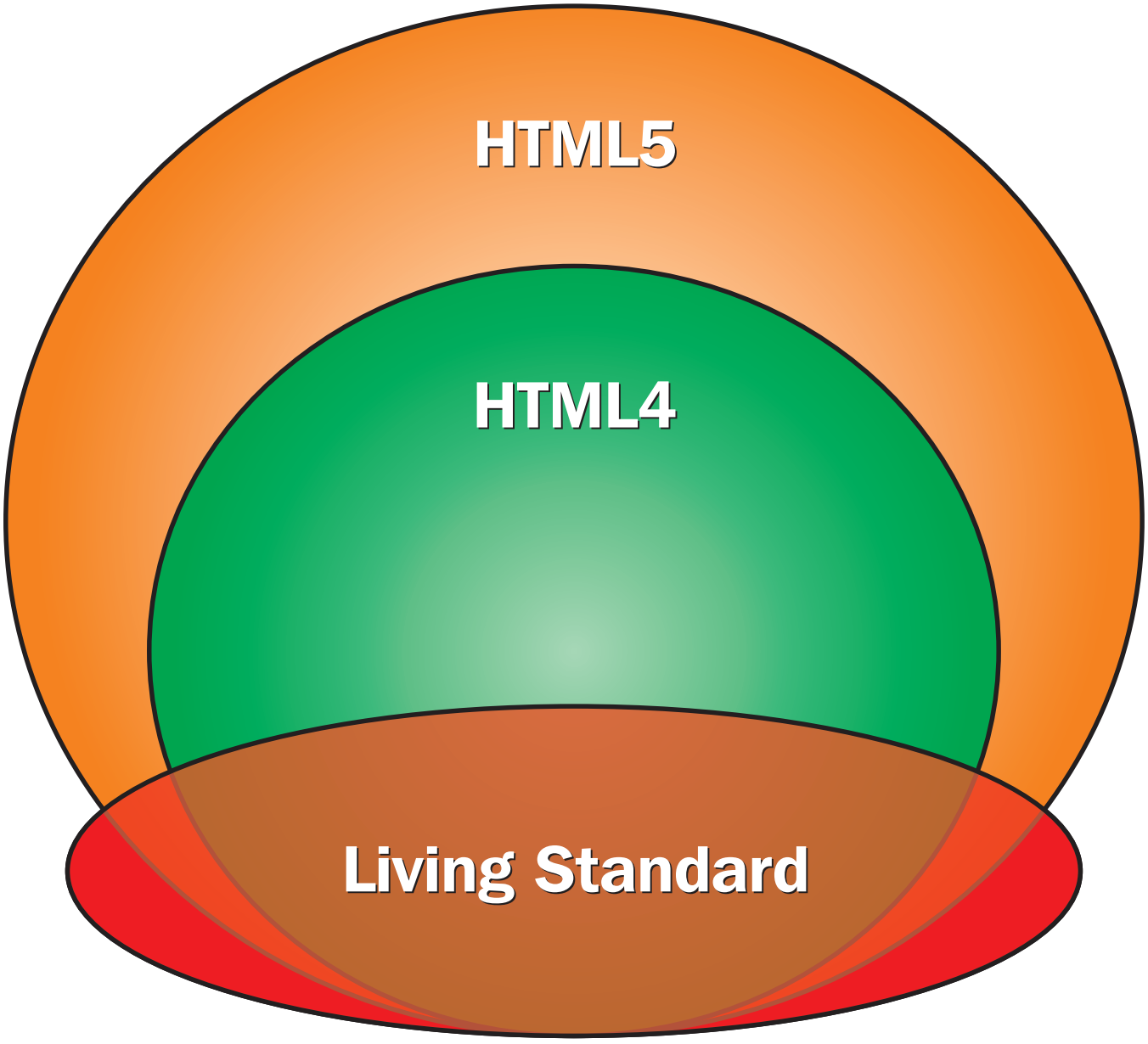
2. Write the code to set up a circle using SVG elements.

3. Write the code to render text using SVG elements.

Copyright © by CAERT, Inc. | Reproduction by subscription only. | L780011

VM–A

HTML5 STANDARDS



Copyright © by CAERT, Inc. | Reproduction by subscription only. | L780011

HTML4 AND HTML5 DIFFERENCES

New multimedia elements in HTML5	New HTML5 form input types	Elements dropped in HTML5
• audio • canvas • semantic based elements • svg • video	• color • date • email • number • range • time • url	• applet • center • font • frame related elements

Copyright © by CAERT, Inc. | Reproduction by subscription only. | L780011

HTML5 AUDIO ELEMENT

Element Name	Code	Browser Display

Format #1	`<audio controls>`
	`  <source`
	`  src="bensound-popdance.mp3"`
	`  type="audio/mpeg">`
	`*audio element not supported*`
	`</audio>`
		

Format #2	`<audio  src="bensound-popdance.mp3"` `type="audio/mpeg" />`	

Copyright © by CAERT, Inc. | Reproduction by subscription only. | L780011

HTML5 VIDEO ELEMENT

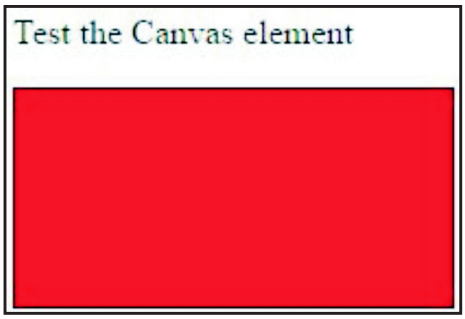
Element Name	Code	Browser Display
Format #1	<video width="400" controls> <source src="Underwater-3.mp4" > Your browser does not support HTML5 video. </video>	
Format #2	<video controls width="400" src="Underwater-3.mp4" type="video/mp4"/>	

Copyright © by CAERT, Inc. | Reproduction by subscription only. | L780011

HTML5 CANVAS ELEMENT

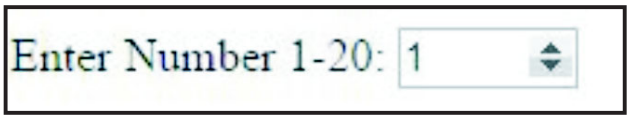
Code	Browser Display

```
<!DOCTYPE html>
<head>
    <meta charset= "utf-8" />
    <title>Myfirst HTML5 page</title>
</head>
<body>
    <p>Test the Canvas element</p>
    <canvas id="myCanvas"
       width="200"
          height="100"
       style="border:1px solid black;" />

    <script>
       var c = document.getElementById("myCanvas");
       var ctx = c.getContext("2d");
       ctx.fillStyle = "red";
       ctx.fillRect(0,0,200,100);
    </script>

</body>
</html>
```



Copyright © by CAERT, Inc. | Reproduction by subscription only. | L780011

NEW HTML5 FORM INPUT ELEMENTS

Code	Browser Display
Enter Color: <input type="color" name="inpColor"/> 	
Enter Date: <input type="date" name="inpdate"/> 	
Enter email: <input type="email" name="inpemail"/> 	
Enter URL: <input type="url" name="inpurl"/> 	
Enter Number 1-20: <input type="number" min="1" max="20" value = 0 /> 	

Copyright © by CAERT, Inc. | Reproduction by subscription only. | L780011

VM–G

NEW HTML5 FORM ELEMENT INPUT VALIDATION ERRORS

Copyright © by CAERT, Inc. | Reproduction by subscription only. | L780011

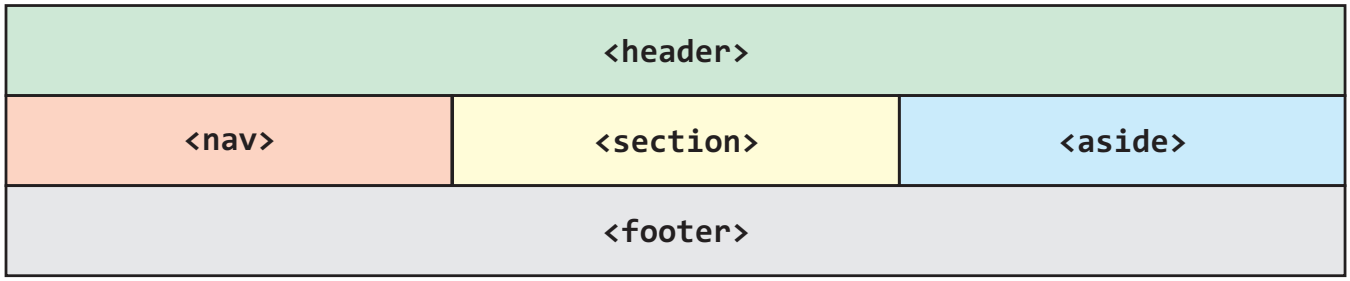
NEW HTML5 FORM INPUT ELEMENT ATTRIBUTES

Code	Browser View
Enter email: <input type="email" name="inpemail" placeholder = "____@___.___"/> Enter URL: <input type="url" name="inpurl" placeholder = "http://www.google.com"/> Enter Name: <input type="text" name="inpname" placeholder = "___ ____"/> 	

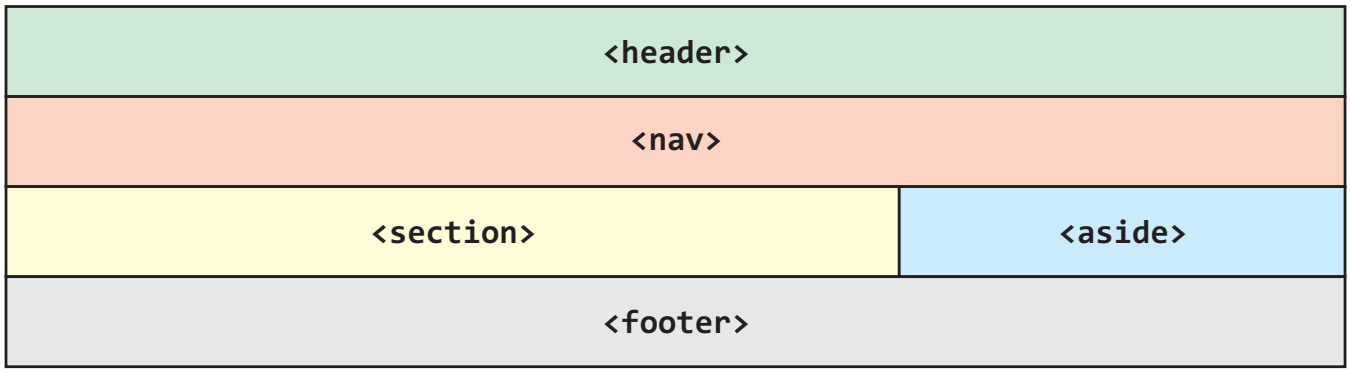
Copyright © by CAERT, Inc. | Reproduction by subscription only. | L780011

HTML5 SEMANTIC ELEMENT FIELDS

Layout #1



Layout #2

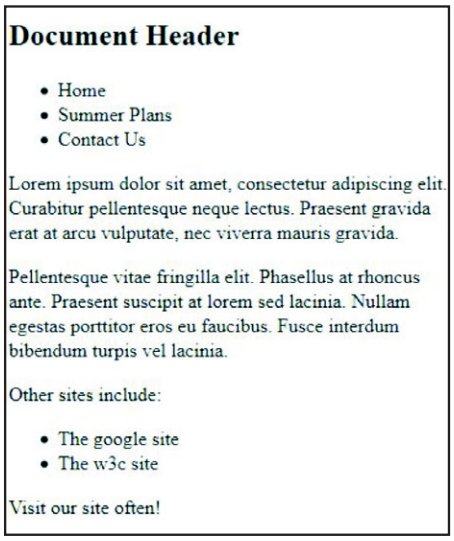


Copyright © by CAERT, Inc. | Reproduction by subscription only. | L780011

HTML SEMANTIC ELEMENTS PROPERTY

Code	Browser Display

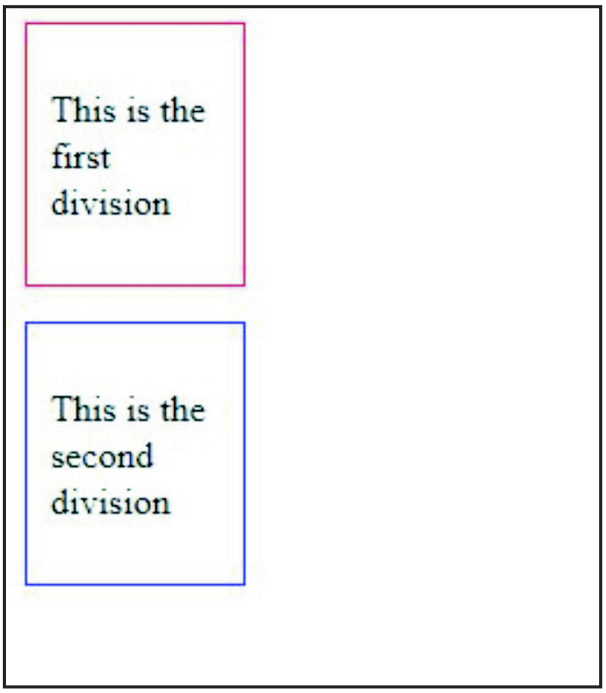
```
<!DOCTYPE html>
<head>
   <meta charset= "utf-8" />
   <title>Myfirst HTML5 page</title>
</head>
<body>
   <header>
      <h2>Document Header</h2>
   </header>
   <nav>
      <ul>
         <li>Home</li>
         <li>Summer Plans</li>
         <li>Contact Us</li>
      </ul>
   </nav>
   <section>
      <p>Lorem ipsum dolor sit amet, consectetur
         adipiscing elit. Curabitur pellentesque
         neque lectus. Praesent gravida erat
         at arcu vulputate, nec viverra mauris
         gravida.

         <p>Pellentesque vitae fringilla elit.
            Phasellus at rhoncus ante. Praesent
            suscipit at lorem sed lacinia.
            Nullam egestas porttitor eros eu
            faucibus. Fusce interdum
            bibendum turpis vel lacinia.
         </p>
      </section>
      <aside>
         <p>Other sites include:</p>
         <ul>
            <li>The google site</li>
            <li>The w3c site</li>
         </ul>
      </aside>
      <footer>
         <p>Visit our site often!</p>
      </footer>
   </body>
</html>
```



Copyright © by CAERT, Inc. | Reproduction by subscription only. | L780011

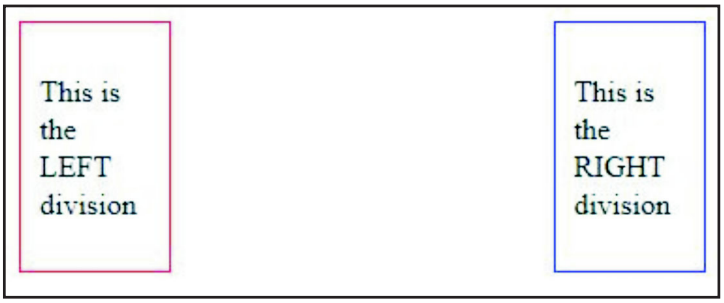
VM–K

CSS FOR FILE WITH MULTIPLE DIV ELEMENTS

Code	Browser Display
<!DOCTYPE html> <head> <meta charset= "utf-8" /> <title>Myfirst HTML5 page</title> </head> <body> <div style="border:1px solid red;margin:15px; padding:10px; width:15%"> This is the first division <div style="border:1px solid blue; margin:15px; padding:10px width:15%"> This is the second division </body> </html>	

Copyright © by CAERT, Inc. | Reproduction by subscription only. | L780011

CSS FLOAT PROPERTY 1

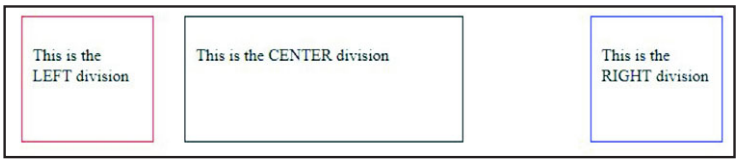
Code	Display in Browser
<!DOCTYPE html> <head> <meta charset= "utf-8" /> <title>Myfirst HTML5 page</title> </head> <body> <div style="border:1px solid red; float:left; margin:15px; padding:10px; width:15%"> This is the first division <div style="border:1px solid blue; float:right; margin:15px; padding:10px width:15%"> This is the second division </body> </html>	

Copyright © by CAERT, Inc. | Reproduction by subscription only. | L780011

CSS FLOAT PROPERTY 2

Code	Browser Display

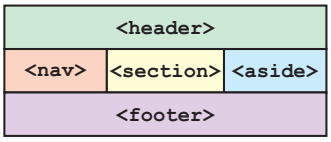
```
<!DOCTYPE html>
<head>
    <meta charset= "utf-8" />
    <title>Myfirst HTML5 page</title>
</head>
<body>
  <body>
    <div style="border:1px solid red;
        float:left;
        height:100px;
        margin:15px;
        padding:10px;
        width:15%;">
        <p>This is the LEFT division</p>
    </div>

    <div style="border:1px solid black;
        float:left;
        height:100px;
        margin:15px;
        padding:10px;
        width:35%;">
        <p>This is the CENTER division</p>
    </div>

    <div style="border:1px solid blue;
        float:right;
        height:100px;
        margin:15px;
        padding:10px;
        width: 15%;">
        <p>This is the RIGHT division</p>
    </div>

  </body>
</html>
```



Copyright © by CAERT, Inc. | Reproduction by subscription only. | L780011

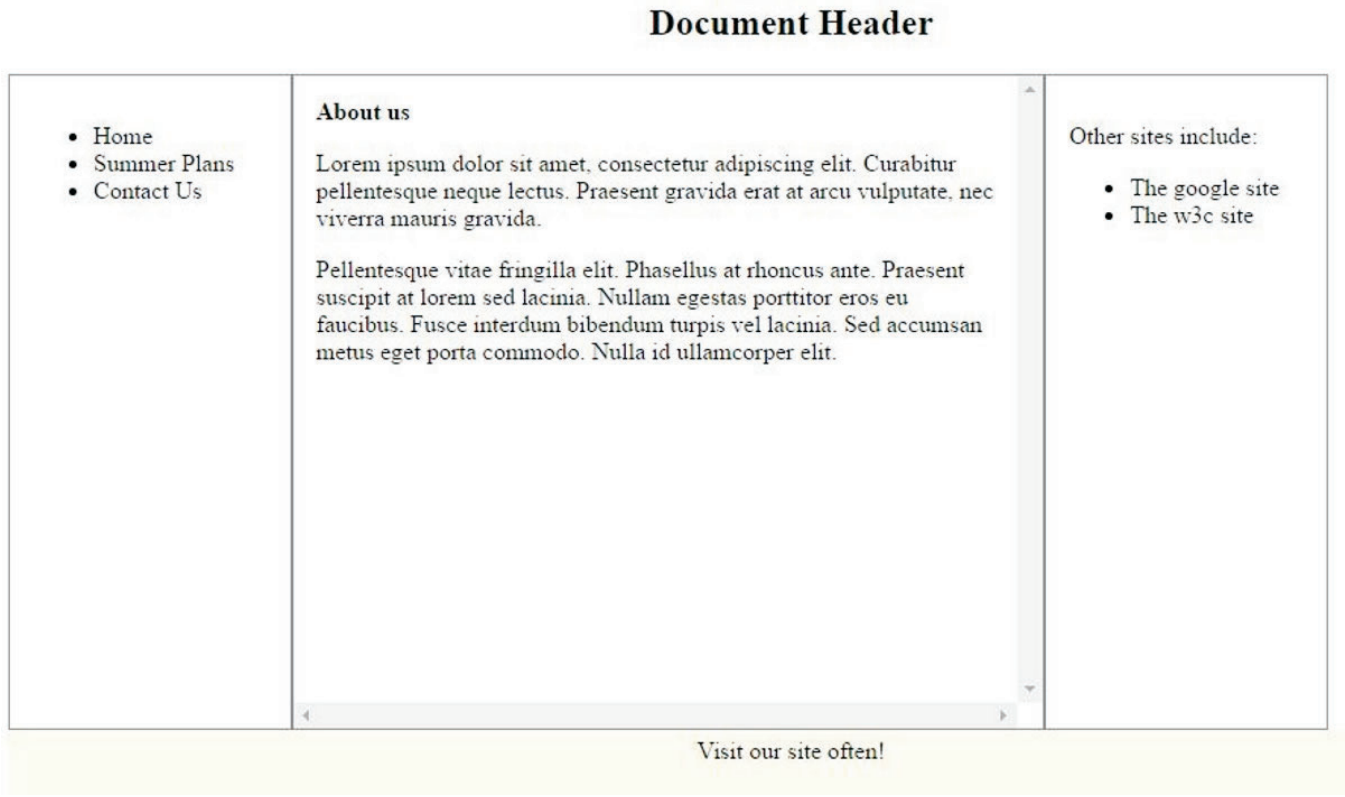
HTML5 SEMANTIC ELEMENTS WITH CSS: CODE VIEW

```
<!DOCTYPE html>
<head>
   <meta charset= "utf-8" />
   <title>Myfirst HTML5 page</title>
</head>
<body>
   <header style="text-align:center">
      <h2>Document Header</h2>
   </header>
   <nav style="border:1px solid gray;
         width:15%;
         float:left;
         padding:15px;
         height:400px">
      <ul>
         <li>Home</li>
         <li>Summer Plans</li>
         <li>Contact Us</li>
      </ul>
   </nav>
   <section style="border:1px solid gray;
         width:45%;float:left;
         padding:15px;height:400px;
         overflow:scroll">
      <b>About us</b><br/>
      <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
      Curabitur pellentesque neque lectus. Praesent gravida erat at
      arcu vulputate, nec viverra mauris gravida. </p>

      <p>Pellentesque vitae fringilla elit. Phasellus at
      rhoncus ante. Praesent suscipit at lorem sed lacinia.
      Nullam egestas porttitor eros eu faucibus. Fusce interdum
      bibendum turpis vel lacinia. Sed accumsan metus eget porta
      commodo. Nulla id ullamcorper elit.
      </p>
   </section>

   <aside style="border:1px solid gray;
            width:15%;float:left;
            height:400px;padding:15px">
      <p>Other sites include:</p>
      <ul>
         <li>The google site</li>
         <li>The w3c site</li>
      </ul>
   </aside>

   <footer style="text-align:center;clear:both;
      background-color:ivory;padding:5px;"
         <p>Visit our site often!</p>
      </footer>
   </body>
</html>
```



Copyright © by CAERT, Inc. | Reproduction by subscription only. | L780011

HTML5 SEMANTIC ELEMENTS WITH CSS: BROWSER VIEW



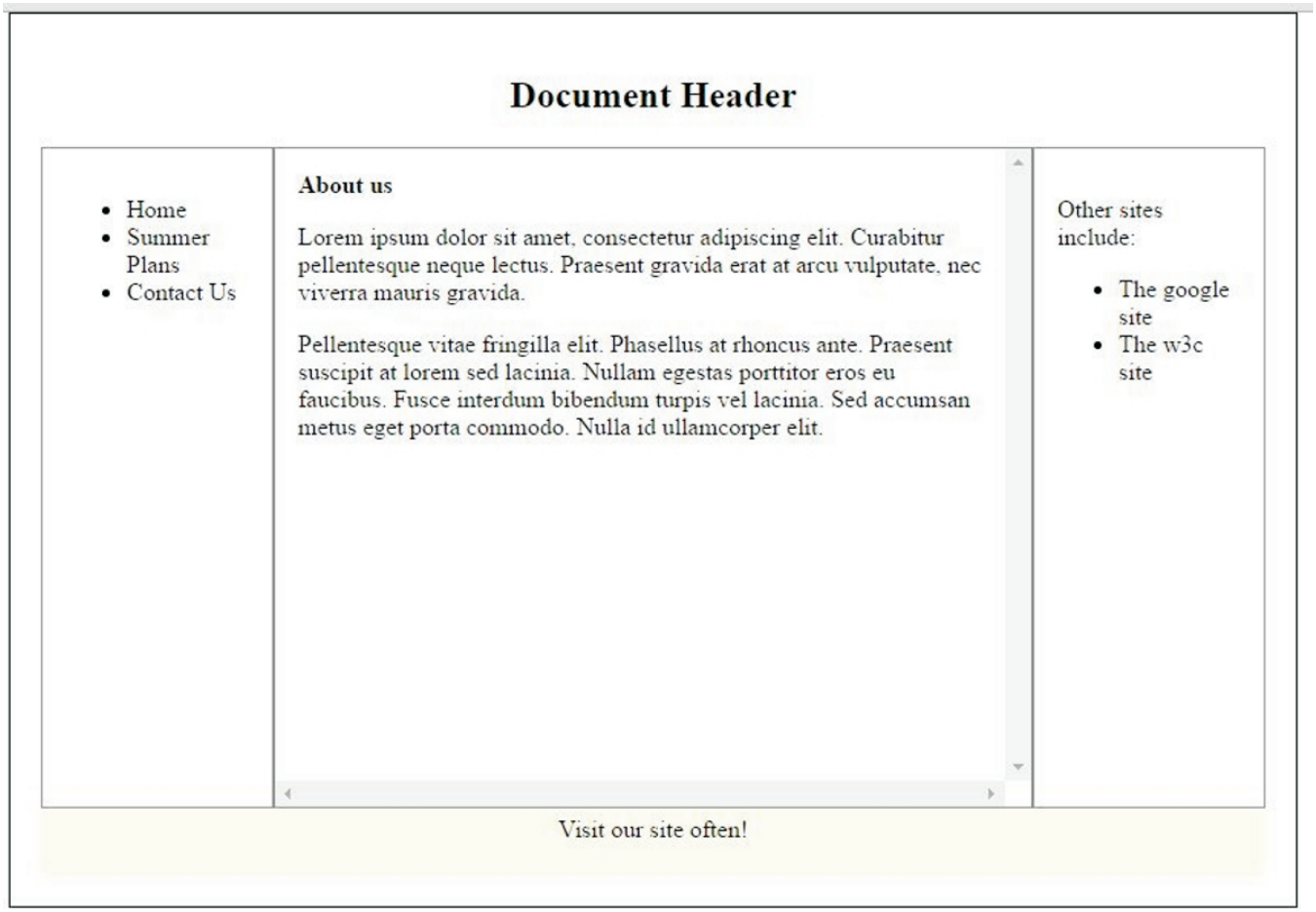
Copyright © by CAERT, Inc. | Reproduction by subscription only. | L780011

USING CSS TO CENTER ELEMENTS

```
<!DOCTYPE html>
<head>
   <meta charset= "utf-8" />
   <title>Myfirst HTML5 page</title>
</head>

<body style="width: 800px;
          margin: auto ;
          padding: 20px;
          border: 1px solid black;">

   <header style="text-align:center">
      <h2>Document Header</h2>
   </header>
   <nav style="border:1px solid gray;
         width:15%;
         float:left;
         padding:15px;
         height:400px">
      <ul>
         <li>Home</li>
         <li>Summer Plans</li>
         <li>Contact Us</li>
      </ul>
   </nav>
   <section style="border:1px solid gray;
         width:58%;float:left;
         padding:15px;height:400px;
         overflow:scroll">

      <b>About us</b><br/>
      <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur
pellentesque neque lectus. Praesent gravida erat at arcu vulputate, nec viverra
mauris gravida. </p>

      <p>Pellentesque vitae fringilla elit. Phasellus at rhoncus ante. Praesent
suscipit at lorem sed lacinia. Nullam egestas porttitor eros eu faucibus. Fusce
interdum bibendum turpis vel lacinia. Sed accumsan metus eget porta commodo. Nulla
id ullamcorper elit.
      </p>
   </section>


   <aside style="border:1px solid gray;
            width:15%;float:right;
            height:400px;padding:15px">
      <p>Other sites include:</p>
      <ul>
         <li>The google site</li>
         <li>The w3c site</li>
      </ul>
   </aside>

   <footer style="text-align:center;clear:both;
      background-color:ivory;padding:5px;"
         <p>Visit our site often!</p>
      </footer>
   </body>
</html>
```


Copyright © by CAERT, Inc. | Reproduction by subscription only. | L780011

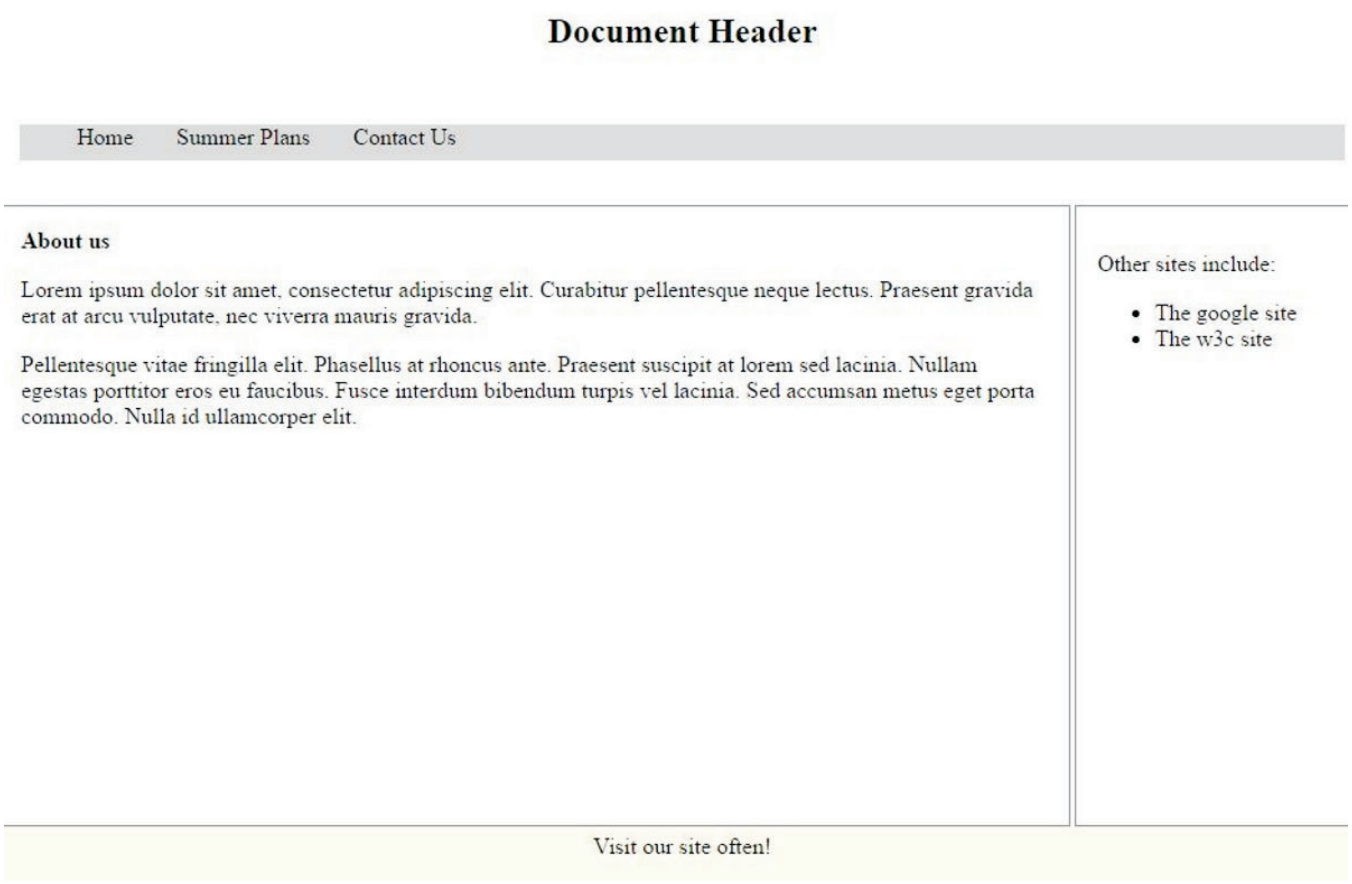
RENDERING OF HTML5 SEMANTIC ELEMENTS WITH CSS



Copyright © by CAERT, Inc. | Reproduction by subscription only. | L780011

VM–R

HTML5 ALTERNATE SEMANTIC ELEMENTS LAYOUT VIEW



Copyright © by CAERT, Inc. | Reproduction by subscription only. | L780011

VM–S

HTML5 ALTERNATE SEMANTIC ELEMENTS LAYOUT VIEW: CODE VIEW

```
<!DOCTYPE html>
<head>
   <meta charset= "utf-8" />
   <title>Myfirst HTML5 page</title>
</head>

<body>
   <header style="text-align:center">
      <h2>Document Header</h2>
   </header>

   <nav style="padding:15px;">
      <ul style="list-style-type: none; background-color:lightgray;height:25px;">
         <li style = "display:inline;padding-right:30px">Home</li>
         <li style =" display:inline;padding-right:30px">Summer Plans</li>
         <li style =" display:inline;padding-right:30px">Contact Us</li>
      </ul>
   </nav>

   <section style="border:1px solid gray;
            float:left;
            height:400px;
            padding:15px;
            width:75%;">

      <b>About us</b><br/>
      <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur pellentesque
      neque lectus. Praesent gravida erat at arcu vulputate,
      nec viverra mauris gravida. </p>

      <p>Pellentesque vitae fringilla elit. Phasellus at rhoncus ante.
      Praesent suscipit at lorem sed lacinia. Nullam egestas porttitor eros eu
      faucibus. Fusce interdum bibendum turpis vel lacinia. Sed accumsan
      metus eget porta commodo. Nulla id ullamcorper elit.
      </p>
   </section>


   <aside style="border:1px solid gray;
            float:right;
            height:400px;
            padding:15px;
            width:18%;">
      <p>Other sites include:</p>
      <ul>
         <li>The google site</li>
         <li>The w3c site</li>
      </ul>
   </aside>

   <footer style="text-align:center;clear:both;
      background-color:ivory;padding:5px;"
         <p>Visit our site often!</p>
      </footer>
   </body>
</html>
```

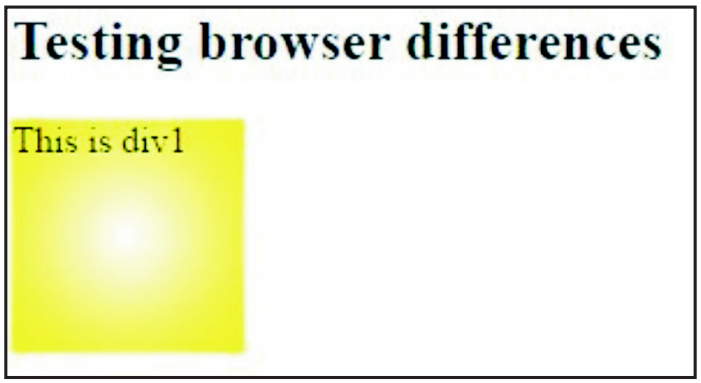

Copyright © by CAERT, Inc. | Reproduction by subscription only. | L780011

BROWSER PREFIXES

Browser	Prefix
Chrome, newer versions of Opera	**-webkit-**
Firefox	**-moz-**
Old versions of Opera	**-o-**
Internet Explorer	**-ms-**

Copyright © by CAERT, Inc. | Reproduction by subscription only. | L780011

BROWSER PREFIXES: TESTING BROWSER DIFFERENCES



HTML Code

```
<!DOCTYPE html>
<head>
    <meta charset= "utf-8" />
    <title>Myfirst HTML5 page</title>
</head>

<body>
<h2>Testing browser differences</h2>
<div
    style= "
    background: red; /* For browsers that do not support gradients */
    background: -webkit-radial-gradient(white, yellow); /* For Safari 5.1 to 6.0 */
    background: -o-radial-gradient(white, yellow); /* For Opera 11.1 to 12.0 */
    background: -moz-radial-gradient(white, yellow); /* For Firefox 3.6 to 15 */
    background: radial-gradient(white, yellow); /* Standard syntax */
    height: 100px;
    width: 100px; ">
<p>This is div1</p>

</div>
</body>
</html>
```


Copyright © by CAERT, Inc. | Reproduction by subscription only. | L780011

SAMPLE XML FILE: CODE VIEW

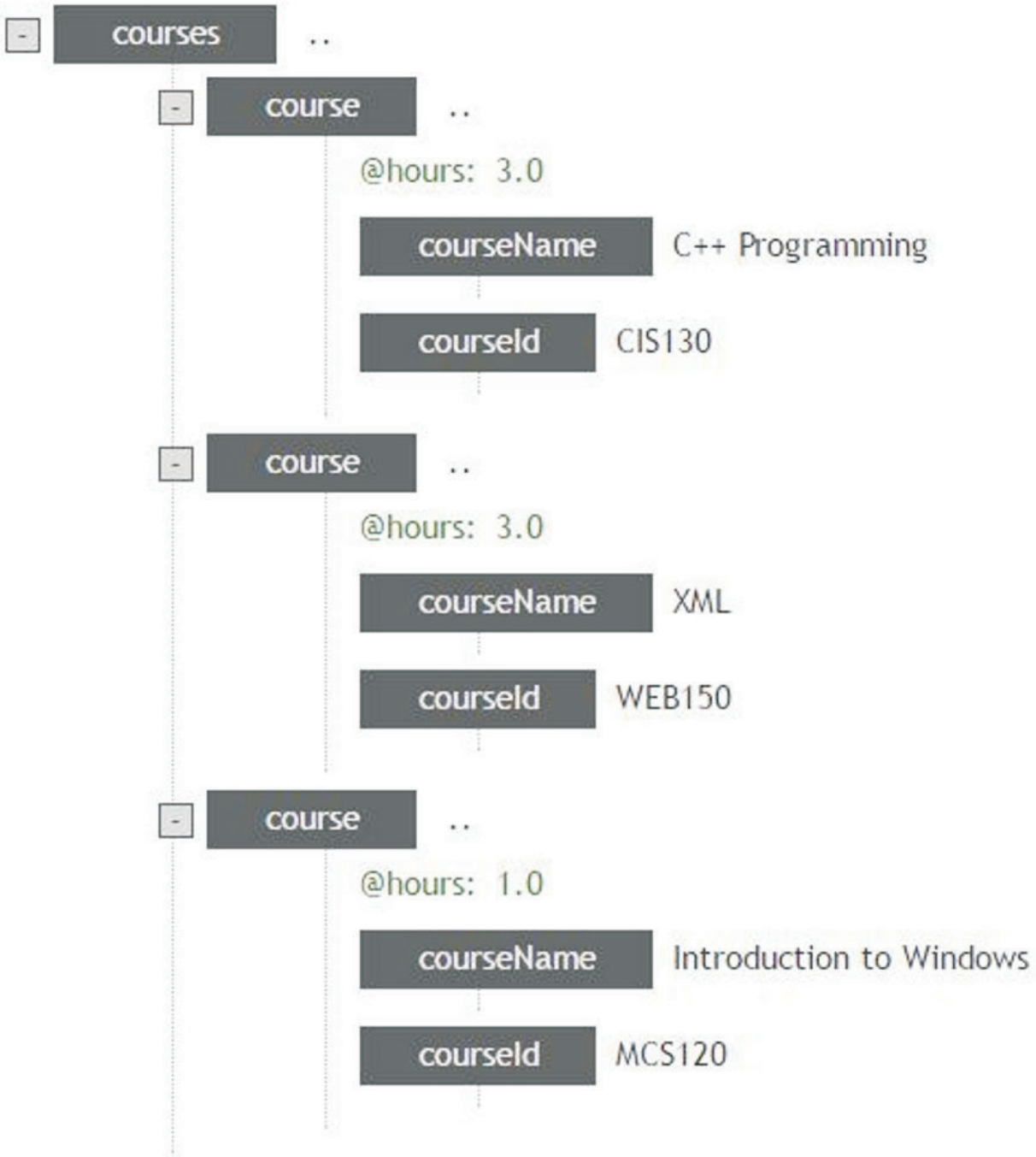
Code for courses.xml

```
<?xml version ="1.0"?>
<courses>
   <course hours = "3.0">
      <courseName>C++ Programming </courseName>
      <courseId>CIS130</courseId>
   </course>

   <course hours = "3.0">
      <courseName>XML </courseName>
      <courseId>WEB150</courseId>
   </course>

   <course hours = "1.0">
      <courseName>Introduction to Windows </courseName>
      <courseId>MCS120</courseId>
   </course>

</courses>
```

"Courses" is the root element in the document.

Copyright © by CAERT, Inc. | Reproduction by subscription only. | L780011

SAMPLE GRAPHICAL XML FILE: TREE VIEW



Copyright © by CAERT, Inc. | Reproduction by subscription only. | L780011

VM–X

SAMPLE SVG FILE WITH BROWSER RENDERING

```
<?xml version ="1.0"?>

    <svg >
       <circle cx="40" cy="40" r="24"
          style="stroke:red; fill:green" />

</svg>
```

Rendering in Browser

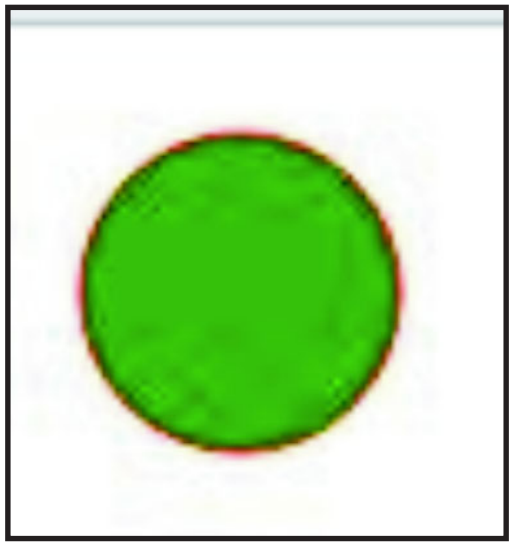
```
▼<svg>
   <circle cx="40" cy="40" r="24" style="stroke:red; fill:green"/>
 </svg>
```


Copyright © by CAERT, Inc. | Reproduction by subscription only. | L780011

SAMPLE SVG FILE WITH NAMESPACE

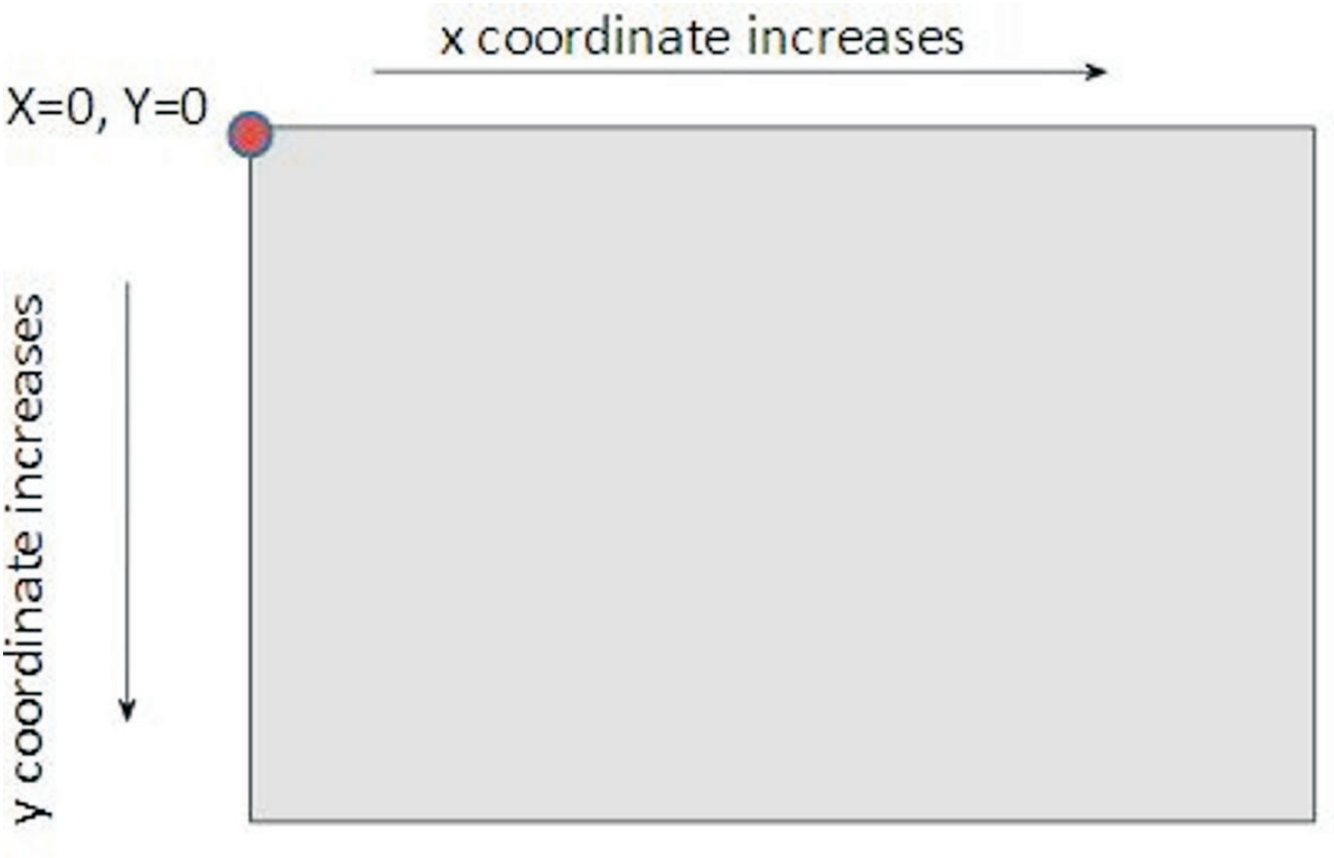
```
<?xml version ="1.0"?>

    <svg xmlns="http://www.w3.org/2000/svg">

    <circle cx="40" cy="40" r="24"
        style="stroke:red; fill:green" />

</svg>
```

Rendering in Browser



Copyright © by CAERT, Inc. | Reproduction by subscription only. | L780011

SVG COORDINATE SYSTEM



Copyright © by CAERT, Inc. | Reproduction by subscription only. | L780011

SVG ELEMENTS

Element Name	Description
svg	Root element of a svg document
rect	Renders a rectangle
circle	Renders a circle
line	Renders a line
text	Writes text

Copyright © by CAERT, Inc. | Reproduction by subscription only. | L780011

COMMON SVG ELEMENTS AND ATTRIBUTES

The rect element	
Attribute Name	**Description**
`x`	x-coordinate of top left hand side
`y`	y-coordinate of top left hand side
`style`	Sets color of outline and fill color via these properties: • fill • stroke • stroke-width
`height`	Height of rectangle in pixels
`width`	Width of rectangle in pixels

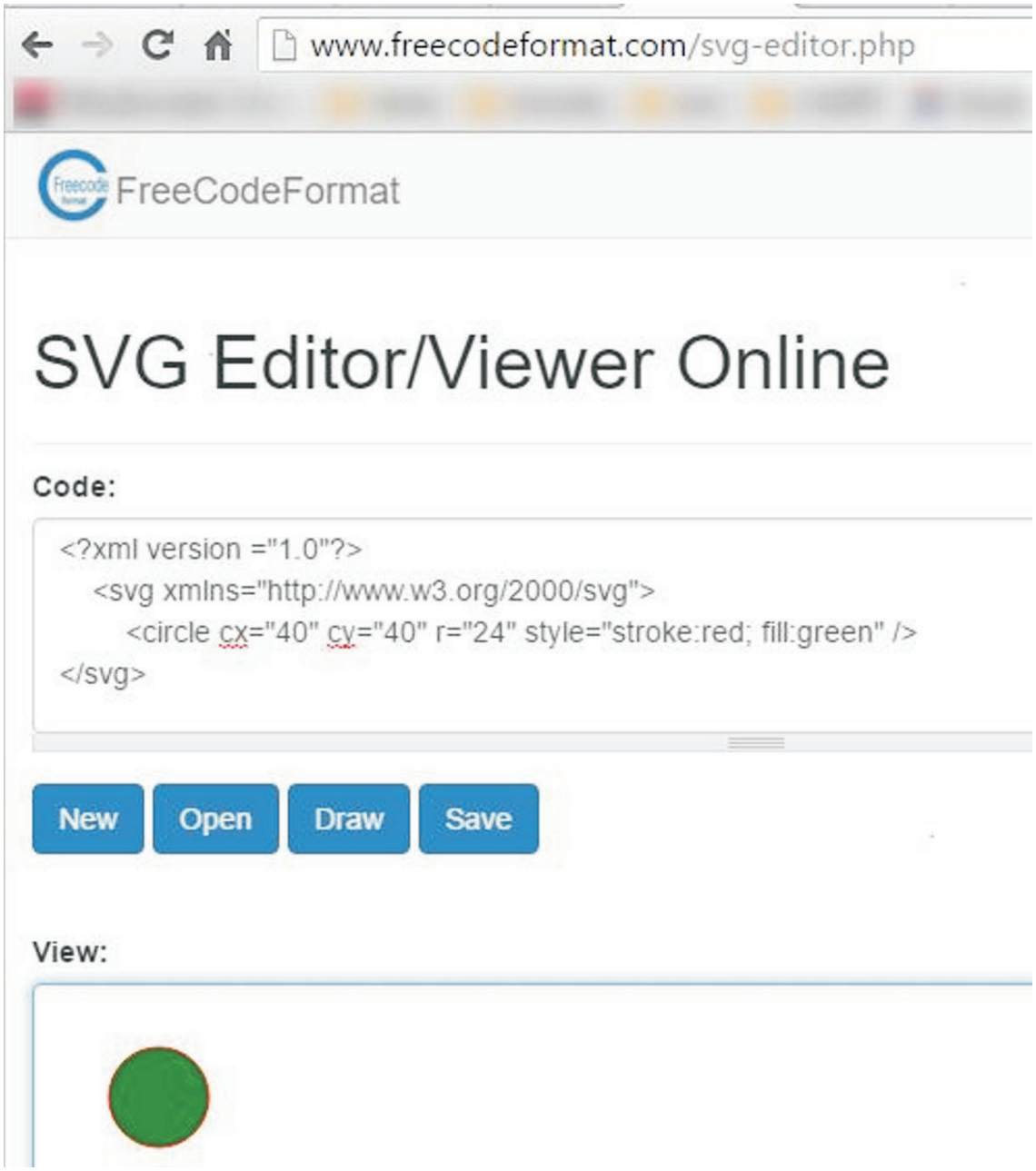
The circle element	
Attribute Name	**Description**
`cx`	x-coordinate of circle center
`yy`	y-coordinate of circle center
`style`	Sets color of outline and fill color via these properties: • fill • stroke • stroke-width
`r`	Radius of rectangle in pixels

The line element	
Attribute Name	**Description**
`x1`	x-coordinate of line begin point
`y1`	y-coordinate of line begin point
`x2`	x-coordinate of line end point
`y2`	y-coordinate of line end point
`style`	Sets color of outline and fill color via these properties: • stroke • stroke-width

Copyright © by CAERT, Inc. | Reproduction by subscription only. | L780011

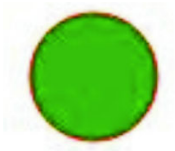
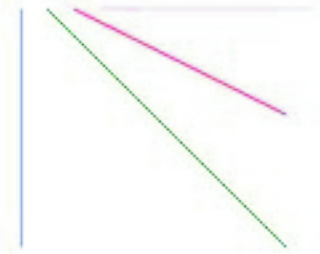
TESTING SVG CODE ONLINE

Test SVG Code using FreeCodeFormat at http://www.freecodeformat.com/svg-editor.php.



Copyright © by CAERT, Inc. | Reproduction by subscription only. | L780011

SVG SHAPES

SVG File Contents	Rendering
<?xml version ="1.0"?> <svg xmlns="http://www.w3.org/2000/svg"> <circle cx="40" cy="40" r="24" style="stroke:red; fill:green" /> </svg>	
<?xml version ="1.0"?> <svg xmlns="http://www.w3.org/2000/svg"> <rect x = "20" y = "30" height = "100" width = "200" style = "stroke:blue; fill: none" /> </svg>	
<?xml version ="1.0"?> <svg xmlns="http://www.w3.org/2000/svg"> <line x1="0" y1="10" x2="0" y2="100" style="stroke:blue;"/> <line x1="10" y1="10" x2="100" y2="100" style="stroke:green;"/> <line x1="20" y1="10" x2="100" y2="50" style="stroke:red;"/> <line x1="30" y1="10" x2="110" y2="10" style="stroke:pink;"/> </svg>	

Copyright © by CAERT, Inc. | Reproduction by subscription only. | L780011

SVG TEXT ELEMENT ATTRIBUTES

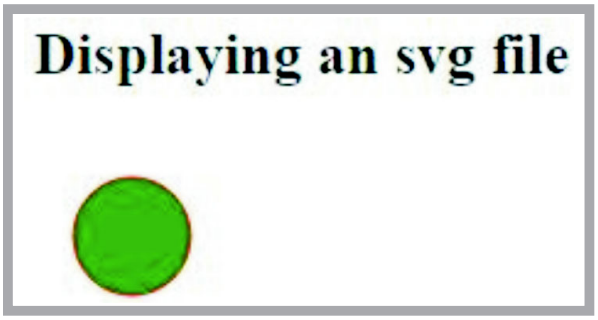
The text element	
Attribute Name	**Description**
x	x-coordinate of top left hand side
y	y-coordinate of top left hand side
style	Sets color of outline and fill color via these properties: • fill • font-family • font-size • stroke • stroke-width
height	Height of rectangle in pixels
width	Width of rectangle in pixels

Text Element Code

Code	Browser Display
<?xml version ="1.0"?> <svg xmlns="http://www.w3.org/2000/svg"> <text x = "50" y = "50" style = "fill: beige; font-family:fantasy; font-size:48; stroke:blue; stroke-width:4px;;" > Hello World! </text> </svg>	

Copyright © by CAERT, Inc. | Reproduction by subscription only. | L780011

HTML WITH SVG FILE

Code	Browser Display
<!DOCTYPE html> <head> <title>Myfirst HTML5 page</title> </head> <body> Displaying an svg file </body> </html>	

Code for Example VM–Y.svg

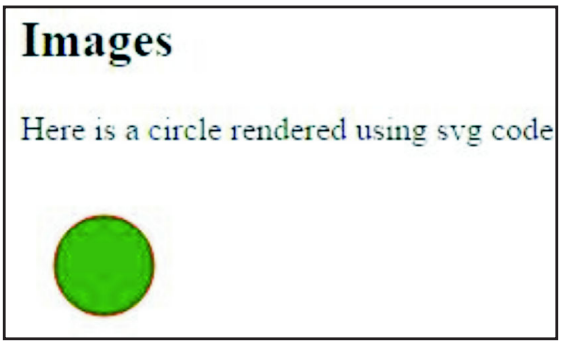
```
<?xml version ="1.0"?>

   <svg xmlns="http://www.w3.org/2000/svg">
      <circle cx="40" cy="40" r="24"
         style="stroke:red; fill:green" />
</svg>
```


Copyright © by CAERT, Inc. | Reproduction by subscription only. | L780011

HTML FILE WITH SVG ELEMENT

HTML Code	Browser View

```
<!DOCTYPE html>
<html>
   <head>
      <meta charset="UTF-8" />
      <title>Example 1</title>
   </head>
   <body>
      <h2>Images</h2>
         <p>Here is a circle
         rendered using svg code
         </p>

      <svg>
         <circle cx="40" cy="40" r="24"
            style="stroke:red; fill:green" />
      </svg>
   </body>
</html>
```



Copyright © by CAERT, Inc. | Reproduction by subscription only. | L780011

HTML FILE WITH SVG TEXT ELEMENT

HTML Code	Browser View

```
<!DOCTYPE html>
<head>
   <meta charset= "utf-8" />
   <title>Myfirst HTML5 page</title>
</head>

<body>
   <h2>
      <svg height="50px"
            width="250px">
         <text x = "36" y = "36"
         style =  "fill: beige;
         font-family:fantasy;
         font-size:36;
         stroke:blue;
         stroke-width:2px" >
      SVG Text
         </text>
      </svg>
   </h2>
   <p>Look! An svg file has been used
for the h2 element!
   </p>

</div>
</body>
</html>
```

Copyright © by CAERT, Inc. | Reproduction by subscription only. | L780011

LS–A

Name _____

Coding HTML5 Forms

Purpose

The purpose of this activity is to practice coding HTML5 forms.

Objectives

1. Set up a document
2. Create a form element to collect a user name and an email address.
3. Save the file.
4. Test the file in an HTML5 compliant browser.
5. Review your HTML5 file with your instructor. Make any needed revisions.

Materials

- lab sheet
- computer with a text editor
- flash drive

Procedure

1. Read the following instructions and code an HTML5 file. Save the file to your flash drive.
    a. Develop a HTML5 document and place a form element in it. Do not code any attributes for the form element.
    b. Place code to set-up an input element that requests a user name from the user.
    c. Place code to set-up an input element that requests a user email from the user. Use the new HTML5 input type that holds email address. Use placeholders for both the elements
    d. Code a submit button.
    e. Save the file and test it in a HTML5 compliant browser.
2. Review your HTML5 file with a colleague and with your instructor. Make any revisions.
3. Turn in your completed file and test to your instructor.

Copyright © by CAERT, Inc. | Reproduction by subscription only. | L780011

Coding HTML5 Forms

Following is a sample HTML5 form document solution. Student work will look different depending upon the properties they select.

```
<!DOCTYPE html>
<head>
  <meta charset= “utf-8" />
  <title>Lab A</title>
</head>
<body>
  <l2>Lab A</h2>
  <form action="" method = “post”>
     <p>Enter Name:
        <input type="text" name="txtName",
        placeholder="Jane Doe"/>
     </p>

     <p>Enter email:
        <input type="email" name="inpemail"
        placeholder = “____@___.___”/>
     </p>
     <p><input type="submit" /></p>
  </form>
</body>
</html>
```


Copyright © by CAERT, Inc. | Reproduction by subscription only. | L780011

LS–B

Name _____

Create a HTML5 Document with Semantic Elements and Format with CSS

Purpose

The purpose of this activity is to learn to code semantic elements and to format those elements.

Objectives

1. Create a document with HTML5 semantic elements.
2. Format the document using CSS.
3. Review your HTML5 file with a colleague and with your instructor. Make any revisions.

Materials

- lab sheet
- computer with multiple browser applications (e.g., Google Chrome, Firefox, Internet Explorer, and/or NotePad++)

Procedure

1. Read the following instructions. Then, write the HTML5 statement that fulfills each instruction.
    a. Create an HTML5 document.
    b. Create a header with your name.
    c. Create a horizontal navigation bar with links to two of your favorite websites.
    d. The main body of the document must contain code. Use the *Lorem Ipsum* generator at http://www.lipsum.com/ to generate text to populate section.
    e. Create a footer with the current calendar year.
2. Review your HTML5 file with a classmate and with your instructor. Make any revisions.
3. Turn in your updated file to your instructor.

Copyright © by CAERT, Inc. | Reproduction by subscription only. | L780011

Create a HTML5 Document with Semantic Elements and Format with CSS

Following is a sample HTML5 document solution. Student work will look different depending upon the properties they select.

```
<!DOCTYPE html>
<head>
  <meta charset= “utf-8" />
  <title>My page</title>
</head>

<body>
  <header style="text-align:center">
     <h2>Jane Doe</h2>
  </header>
  <nav style="padding:15px;text-align:center">
     <ul style="list-style-type: none; background-
color:lightgray;height:25px;">
        <li style="display:inline;padding-right:30px">
        <a href = “http://www.google.com”>
        Home</a></li>
        <li style="display:inline;padding-right:30px">
        <a href = “http://www.w3schools.com/html/html5_intro.asp”>
        HTML5 Introduction
        </a>
        </li>

     </ul>
  </nav>
  <section style="border:1px solid gray;
               height:400px;
          padding:15px;">

     <h2>About Me!</h2>
     <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Curabitur pellentesque
     neque lectus. Praesent gravida erat at arcu vulputate,
```


Copyright © by CAERT, Inc. | Reproduction by subscription only. | L780011

```
     nec viverra mauris gravida. </p>

     <p>Pellentesque vitae fringilla elit. Phasellus at rhoncus ante.
     Praesent suscipit at lorem sed lacinia. Nullam egestas porttitor
eros eu
     faucibus. Fusce interdum bibendum turpis vel lacinia. Sed accumsan
metus eget porta commodo. Nulla id ullamcorper elit.
     </p>
  </section>


  <footer style="text-align:center;clear:both;
     background-color:ivory;padding:5px;"
        <p>2016</p>
     </footer>
  </body>
</html>
```


Copyright © by CAERT, Inc. | Reproduction by subscription only. | L780011

Name _____

Create Simple SVG Graphics Rendered in a HTML5 Document

Purpose

The purpose of this activity is to create and view SVG graphics.

Objectives

1. Create SVG graphics in an HTML5 document.
2. View the SVG graphics in the browser
3. Review your HTML5 file with a classmate and with your instructor. Make any revisions.

Materials

- lab sheet
- computer with multiple browser applications (e.g., Google Chrome, Firefox, Internet Explorer, and/or NotePad++)

Procedure

1. Read the following instructions. Then write the HTML5 statement that fulfills each instruction.
   a. Create an HTML5 document, and place an SVG element in it.
   b. Create three concentric circles in the SVG element.
   c. View the document in the browser.
2. Review your HTML5 file with a classmate and with your instructor. Make any revisions.
3. Turn in your updated file to your instructor.

Copyright © by CAERT, Inc. | Reproduction by subscription only. | L780011

Create Simple SVG Graphics Rendered in a HTML5 Document

Following is a sample document solution. Student work will look different depending upon the properties they select.

```
<!DOCTYPE html>
<html>
  <head>
     <meta charset="UTF-8" />
     <title>Example 1</title>
  </head>
  <body>
     <h2>Images</h2>
     <p>Here is a circle rendered using svg code
     </p>

     <svg width = “300" height = ”300">
        <circle cx="150" cy="150" r="100"
           style="stroke:red; fill:green" />
        <circle cx="150" cy="150" r="75"
           style="stroke:black; fill:ivory" />
        <circle cx="150" cy="150" r="25"
           style="stroke:red; fill:red" />
     </svg>
  </body>
</html>
```


Copyright © by CAERT, Inc. | Reproduction by subscription only. | L780011