HTML and CSS Level 1

**Skill Check 5: Add Blocking Elements and Link Styles**

Given the results of Skill Check 4 and a text editor, complete the following tasks:

**PART ONE: BLOCKING ELEMENTS**

1. Copy your *html-1-skill-check-4* folder, and save it as *html-1-skill-check-5.*
2. Open your HTML and CSS files in Notepad. Display the HTML files in a web browser.
3. Add the Remy Sharp shim code to your site, to add browser support for older versions of Internet Explorer that don’t understand the new HTML5 blocking elements.
   1. *Gmail’s filters won’t allow you to send a .js file, so you’ll need to use the Cloudfare hosted script method:* [*http://ravengildea.net/html-css-1/sections.php#remy*](http://ravengildea.net/html-css-1/sections.php#remy)
   2. Use a conditional comment to add the <script> tag to the <head> tags of each of your HTML documents. Set the conditional comment to be read by versions of IE that are older than IE9.
4. Use semantic blocking elements to create the following page sections in each HTML file:
5. nav *#nav-main*
6. header *#header*
7. main *#content*
8. section
9. aside *#secondary*
10. footer *#footer*
    * + Note that a site’s main nav may or may not be enclosed in the header element. *In this case, keep the nav outside the header.*
      + Note that the Hours, Location, and Contact information might be considered either aside content or footer content. *In this case, we’ll make it an aside.*
      + Be sure to **add a *unique id*** to each blocking element that needs one (everything except the <section> tags).
      + When you are finished, check to be sure your pages have no unblocked content.
11. Use a generic blocking element to create a container element for layout purposes.
12. Enclose the <header> and <main> elements in the container. *In this particular case, we will leave the main nav, aside, and footer out of the container.*
13. Give this element a unique id of *container*.
14. Validate your HTML.

**PART TWO: STYLING LINKS**

1. In the About portion of *index.html* and *about.html*, link the words “Viennese coffee house” to this external web page: <https://en.wikipedia.org/wiki/Viennese_coffee_house>
   1. Make this link open in a new browser tab.
2. In your *style.css* file, write *global* link styles*.*

*Note that these will be the styles you want for inline text links within the main content, but your CSS selectors should be global, and apply to* ***all*** *links on the page. We will override them as needed with more specific selectors.*

1. Give unvisited and visited links the same style: a *text-decoration: underline* and a color of #903A13 (rust).
   * + Linked text should look different from unlinked text: links must be obvious.
     + Most browsers underline links by default, but setting the rule ensures we’ll get the same styles in all browsers.
     + Note that the styles we set on links in their unvisited/visited states will be inherited by their hover/focus states.
2. Use *text-decoration: none* to make the underline disappear when links are hovered or have focus.
   * + Hovered/focused links should be visually distinct from unhovered/unfocused links.
3. Be sure to write your link styles in LVHFA order. (It is not necessary to give your links a distinct visual style for their active state.)

screenshot-global-link.tiff

1. **Use the unique id** on the nav-main to style the main navigation.

*Note that we’ll use the nav’s unique id as our selector, because most web pages will eventually have more than one <nav> element. We want to be prepared to add new <nav> elements as needed in the future, without having them follow the styles of our main navigation.*

1. Set the <nav> element’s background-color to our previously-used rust color.   
   The comments you’ve left in your code should make this color easy to find. Be sure to include a comment here as well, to make the color easy to identify.
2. Use ***descendent selectors*** to set the anchor tags inside the main navigation to our previously-used cream color.   
   The comments you’ve left in your code should make this color easy to find. Be sure to include a comment here as well.
3. Give the nav-main links a *text-decoration: none;* in their unvisited and visited states.
4. Give the nav-main links a *text-decoration: underline* in their hover and focus states.
5. Be sure to write complete descendent selectors whenever you have two or more selectors separated by a comma.
6. Be sure to list your pseudo-class selectors in LVHFA order.

screenshot-nav-main.tiff

1. **Use the unique id** on the <aside> element to override our global styles with different styles for the aside and its contents.
2. Set the <aside> element’s background-color to our previously-used rust color.
3. Set the text within the aside to our previously-used cream color.
4. Use descendent selectors to set the headlines within the aside to our cream color.
5. Use descendent selectors to set the anchor tags inside the aside to our cream color.
   * + Be sure to *write complete descendent selectors* whenever you have two or more selectors separated by a comma.
     + Note that we only need to set the color on the visited and unvisited link states: the hover and focus states will inherit this color.
     + Note that our aside links will inherit our global text-decoration styles.

screenshot-aside.tiff

1. Use the unique id on the footer element to give the footer and footer links the same styles as the <aside> element.

screenshot-footer.tiff

1. Use a descendent selector to override the global link styles and uniquely style the site id link.
2. Set the h1 link (our site id, the word “VIENNA”) to the same rich brown color as our body text.
3. Give this link a *text-decoration: none* in its unvisited and visited states.

screenshot-site-id.tiff

1. Change this link’s color to rust in its hover and focus states.  
   *Note that the text-decoration style we set for unvisited/visited links will be inherited by hover/focus links.*

screenshot-site-id-hover.tiff

1. Validate your CSS and HTML.