HTML and CSS Level 1

**Skill Check 6: Boxes and Layout**

Given the results of Skill Check 5, the *interior-2700.jpg* image file, and a text editor, complete the following tasks:

**PART ONE: BOXES**

1. Copy your *html-1-skill-check-5* folder, and save it as *html-1-skill-check-6.*
2. Open your HTML and CSS files in Notepad++. Display the HTML files in a web browser.
3. Add browser resets to your CSS file:
4. Reset margin and padding to zero
5. Set box-sizing to border-box
6. Remove borders from linked images
7. Adjust the vertical white space between your text elements:
8. Paragraphs
9. Lists (all three varieties)
10. List items and *dd* elements for definition lists
11. Check your <address> items at the bottom of the page. If you did not use <p> tags in your markup, you will need to add white space under the address element as well.
12. Adjust the h1 headline spacing by setting this element’s line-height to 1.2
13. Add list indents:
14. Ordered and unordered lists must be indented, so their bullets and numbers will display.
15. Whether you indent the *dd* tags in your definition lists is up to you.
16. While you’re there, set your *strong* and *dt* tags to display in the bold version of your body font.
17. Add margins, padding, and border to your blockquote:
18. Give it a 2px solid border on top and bottom, matching the color of the blockquote text.
19. Set 1em padding top and bottom inside the borders, and 2em padding to the sides.
20. Set 2em margins top and bottom, with no extra margins on the sides.
21. Give the images rounded corners, with a radius of 25px.
22. Validate your CSS.

**img-1.tif**

**PART TWO: LAYOUT**

Here’s a sketch of the layout we’ll be creating for the *interior* pages:



1. On each of the interior pages (not the index.html page), add a layout div to the middle portion of your page.
   1. Start this div under the main navigation, and end it above the footer element. Your container, header, main, and aside elements should be nested inside the new div.
   2. Give the new div an id of *middle*.
   3. Set the #middle div to a width of 80%, and center it within its parent element.
   4. Give the #middle div a min-width of 730px. This will keep your layout from collapsing on smaller screens. *(… sort of. We’ll learn more about this in the Responsive Design course.)*
   5. Be sure to add a comment to #middle’s closing </div> tag.
2. Use the container div to create a floated column on the left side of the page:
3. Change the body element’s background-color to the same rich brown color as your body text.
4. Set the background-color of the container div to rgba(246, 200, 131, .75)
   * + Note *that this is the same color as our previously-used cream color, #F6C883, but with a 75% opacity.*
     + Add a comment to help you identify the color.
5. Give the container a width of 500px and a padding of 1.25em.
6. Float the container to the left. *(aside and footer will rise up to sit under the nav.)*

img-3.tif

1. Use the <aside> element’s unique id to create a floated column on the right side of the page:
2. Set the aside’s width to 210px, and give it a padding of 1.25em.
3. Float the aside to the right.
4. Change the aside’s background-color to rich brown, and add a color comment.

img-4.tif

1. Use the footer’s unique id to style it:
2. Note that the footer rises up to sit under the nav element: this is because all of #middle’s contents are floated. Clear the floats by adding a *clear:both;* declaration to the footer.
3. Give the footer a padding of 1.5em top and bottom and 10% on each side.
4. Remove the horizontal rule from the footer in your HTML files.

img-5.tif

1. Use the nav’s unique id to style it:
2. Give the nav element padding to match the footer.
3. Use a descendent selector to remove the bullets in the nav list items (but not other list items on the page) with a declaration of *list-style: none;*
4. Override your global list item indents by setting the nav list items (but not other list items) to a margin-left of zero.
5. Use a descendent selector to float the nav list items to the left.
   * + Separate them from one another with a margin-right of 2.7em.
     + Add a *class="last"* attribute to the last list item (Contact).
     + Use the class on the last nav list item to set its margin-right to 0.

*(This step is not necessary on this particular design, but in many navbars the last item needs a margin-right of zero to fit without dropping to a new line.)*

1. Because the nav’s contents are now floated, add an *overflow:hidden;* declaration to the nav element, to make its background extend downward to enclose the floated contents.

img-6.tif

1. Validate your CSS and HTML.

***To pass Skill Check 6, you can stop here.***

*If you have time and energy to do some more cool stuff, you can keep going with the following extra items.*

**PART THREE (OPTIONAL)**

1. Add the image *interior-2700.jpg* to your images folder, and apply it to the body element with the following CSS:

background-image: url(images/interior-2700.jpg);

background-repeat: no-repeat;

background-size: cover;

background-attachment: fixed;

*(We will cover background-images in detail in HTML/CSS Level 2)*

screenshot-10.tiff

1. Give unique styles to your home page.
2. On *index.html,* add a *class="home"* to the body tag.   
   *This will allow you to write descendent selectors to style the index page differently from other pages.*
3. Create a new section in your CSS, labeled   
   /\* HOME PAGE STYLES \*/
4. Use descendent selectors to add the following styles to your ***index.html page, only:***
   * + Set the home page container and aside to *float: none;*
     + Set the aside’s padding to match the footer’s padding.
     + Set the aside’s background-color to rust and width to 100%.
     + Give the container a margin-left of 10%.
     + Remove the horizontal rule from the footer, and give the copyright paragraph a 2px solid cream-colored border-top.
     + Add 1.5em padding-top to the copyright paragraph.
5. Create three floated columns within the home page aside:
   * + Wrap each chunk of aside content (Hours, Location, Contact) in a div with a class of *"widget".   
       (You can call this class whatever you want, but "widget" is common usage in WordPress sites, so that’s what I use.)*
     + Use the class to set the widgets to a width of 30%, with a margin-right of 2%.
     + Float the widgets to the left.
     + Because the home page aside’s contents are all now floated, add an   
       *overflow: hidden;* declaration to the aside element. This will expand its background downward to enclose the floated elements.

screenshot-11.tiff

1. Validate your CSS and HTML.
2. Admire your work and pat yourself on the back.  
     
   **NOTE:** In real life, the index page of this site would not contain all the content that’s also on the interior pages. But here you have examples of two different ways of building a site: with multiple pages, all linked in the nav, or a one-page site, as shown on your index.html page. For a one-page site, the nav would be updated with jump links that take users down the page to each section.

*That’s all, folks!*