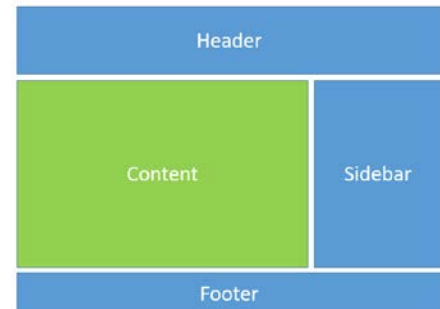# CSC 170 Lab 05: CSS for Layout

Step 1: Make a copy of Lab 4

- *After* you've completed Lab 4 and got an acceptable grade, make a copy of all the files in it to a new folder named **lab05**

Step 2: Add a layout technique to your webpage

- ❑ Using the **Inline-block**, **Table-cell**, or **Flex** technique, position major blocks of content in your webpage side-by-side.
    - ○ A columnar layout requires you to *do more* than take two small elements and position them side-by-side. The idea is to intentionally "lay out", i.e. place content in a way that makes sense, based on the content you have on your web page. For example: the classic "c-clamp" layout will probably work well enough for this lab assignment.
    - ○ The most common candidates to position side-by-side are your **<article>** element and **<aside>** element
- ❑ As much as possible, do not alter your original HTML document.
    - ○ The intent of this lab is that you *not* re-structure your HTML document, but if you need to move things around (a little), then do what you must.
    - ○ You may add elements (create new DIVs or SECTION elements) to nest content and add classes as necessary.
- ❑ In addition to side-by-side content, your document must use a "container" to constrain its content in some way. You don't have to constrain *all* your content (although that's okay if you do), but you must demonstrate good use of the trick…
    - In HTML…
    ```
    <body>
    <div class="container">

            <!-- all your content would go in here -->

    </div><!--.container-->
    </body>
    ```
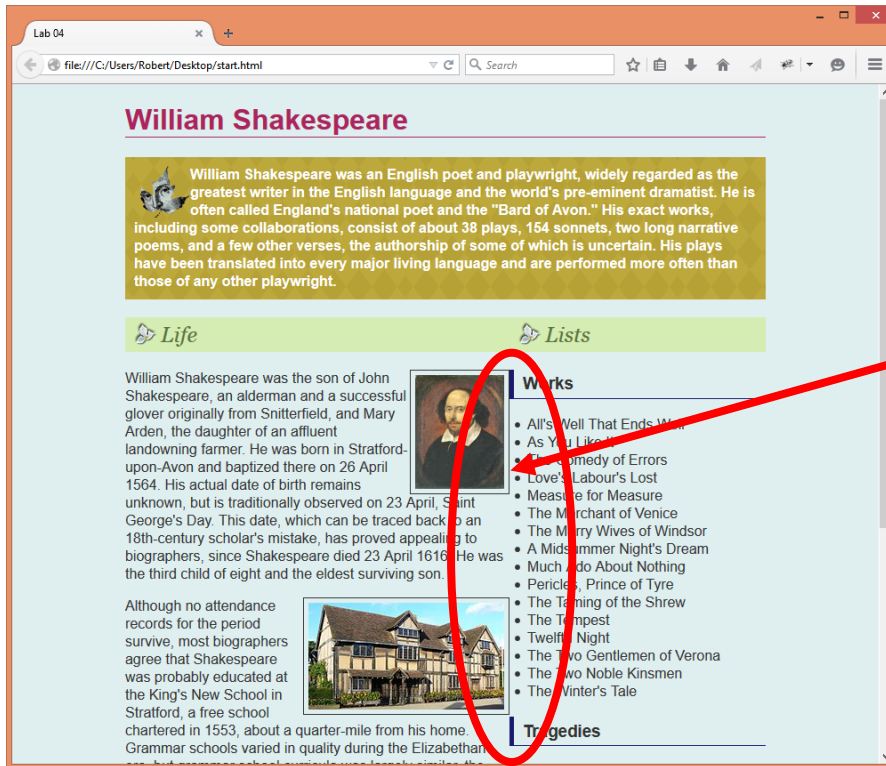
    - In CSS…
    ```
    .container {
        margin: 0 auto;
        width: 80%; //or whatever
        max-width: 960px; //or whatever
    }
    ```

❑ When you position the content chances are you'll need to adjust the box model of the elements that are side-by-side to keep the content readable and generally usable. For example, if the ARTICLE and ASIDE elements were full of text, and then you lay them out side-by-side, the content smashes against the sides like this:

The content needs to be separated by making alterations to the box model of the ARTICLE and/or ASIDE

Alter the box model of the elements (in this case, that'll probably be the ARTICLE and/or ASIDE) by adding margin and/or padding to them.[1]

Your Layout Must Work on All Normal Desktop Sizes

❑ Part of the grading process will include shrinking and stretching the browser width to see what happens to your design. Your layout must "hold together" (make sense) at any width greater than 720px.

- NO horizontal scroll bars should appear at any browser width greater than 720px.
- Your layout does not have to be "responsive" or "adaptive" (mobile friendly), and it doesn't have to do anything different at widths less than 720px, but it can't suddenly fall apart just because the person viewing your webpage happens to be looking at it on a different screen size or resolution than yours.

---

[1] The exact way to fix the layout depends on which layout strategy you used plus the particulars of the content. Good luck!

Step 3: Upload your work

When you are done with your webpage, use an FTP tool (like *WinSCP*) to access your account on
_____.**rochestercs.org** to create a folder: **lab05** and upload all your files into it.

❑   In a web browser (any), go to this address to check your handiwork:
   accountname.**rochestercs.org /lab05/start.html**
   (where "*accountname*" is your account name)

Remember: All files and folder names must always be all lowercase and no spaces.

Report your work

Remember: Use the W3C HTML validator (**validator.w3.org**) AND the W3C CSS validator
(**http://jigsaw.w3.org/css-validator**) to check your work before you report it!

•   In our Blackboard section, in Lab 05, post a link to your webpage to receive credit for this Lab.