CW-03-04 CSS Positioning
Today's Activities CSS Layouts Part Deuces
Due Feb 8, 2022 11:59 PM
Objective: Obtain effiency for styling pages specifically for various layout designs:
Reminder: practice using your browser's page inspection resources!

How to Use FireFox or Chrome Web Inspectors

FireFox Developer Tools is a Firefox add-on that lets you dynamically examine or modify the content and styling of web pages. It comes pre-installed with the Fire-Fox browser. To open it, right-click an element and choose Inspect Element. The Chrome Web Inspector tool behaves the same way, and comes conveniently built-in with Chrome. You can launch both by hitting **F12** while in your browser.

Here is a link to Check out the more on the **Chrome Dev-Tools Browser** .

1. Arrange Your Page into Sections
2. Spacing with Padding and Margins, Backgrounds
3. Float, Alignment and Clear
4. Flexbox

Exercise 1: Layout a skeleton

Download this >>> Skeleton HTML & **Support image** Files and open the webpage in your text editor (Sublime, Atom, Notepad++, etc.). Recall that you can download an HTML file by either right-clicking the link and selecting "Save As..." to save in a local folder of your choice, and you can download an image by right-clicking the image and selecting "Save As..." in the same manner (as you'll see in example.html, the path to the support image assumes it is in the same directory as the HTML file).

Task : Follow the instructions in the HTML page to add page sections, and change the layout.

To start out, you'll need to create and link in a CSS file. From there, the HTML elements explain what to do.

My Sample Demo on floats click here

See the sample output below.

Sample Output (click on image below to view in full size of demo)

Please Note: The browser window size will definitely affect how the page is rendered. The sample image was generated on a wide screen, so it is possible that your page will flow differently.

About Flexbox

1. Flexbox is a set of CSS properties useful for aligning block level content into various layouts.

2. Flexbox can help manage how the elements should be sized relative to each other.

3. Like the position property, you have to think about your content in terms of the elements themselves, and the containers they are in.

4. The most complicated thing about flexbox is that the content can "flex" in either direction: horizontally into rows, or vertically into columns.

Basic properties for the flex container

1. Display: flex;
2. Makes an element a "flex container", items inside automatically become "flex items"
3. Justify-content: flex-end; (flex-start, space-around,...)
4. Indicates how to space the items inside the container along the main axis
5. Align-items: flex-end; (flex-start, center, baseline,...)
6. Indicates how to space the items inside the container along the cross axis
7. Flex-direction: row; (column)
8. indicates whether the container flows horizontally or vertically.

Basic properties for the flex items

flex-basis: 20%; (3em, 50px,..etc.)
indicates the default size of an element before the extra space is distributed among the items
align-self: flex-end; (flex-start, center, stretch,...)
indicates where to place this specific item along the cross axis

Flexbox Froggy Just Practice with this tool (Just for Fun)

There are great tutorials out in the world to help learn flexbox. This CSS-Tricks does a in-depth job of explaining the flex-box properties and has some great examples.

For the remainder of the Activity, you are to work on Flexbox Froggy, which is CSS game for learning the basics of Flexbox.