


Fundamentals of Web Design

CHAPTER 16 – CSS Layout with Flexbox and Grid

CLASSWORK INSTRUCTIONS part 1- B

In this exercise you will apply Flexbox properties to an online menu for the Bistro Items to Go section of the web site.

1. Create a new folder named  **namelastname_chapter16b**
2. Download the homework data files, decompress, and save them inside the new folder.
3. Open the *flex-menu.html* and the *flex-menu-style.css* in your text editor.
4. Open *flex-menu.html* in your favorite browser. You can see the product listing showing in one long column from top to bottom. You will change that in this exercise by using Flexbox.
5. Return to your text editor, add your name and last name inside the comment section of the *flex-menu.html* document.
6. Move to the *flex-menu-style.css* in your text editor.
7. Turn the #menu div container into a flexbox by setting its **display** to **flex** (add this declaration to the existing #menu style rule **by yourself**).
8. Save and refresh the browser. All the products in the menu show up in a row now. Depending on how wide your browser window is set, you may see content extending beyond the container and getting clipped off.
9. Return to the text editor.
10. Add the **flex-direction** property with the value: **row-reverse** to the existing #menu style rule.
11. Save the style sheet and refresh the browser. The product listing starts on the right side of the window.
12. Change the **flex-direction** property back to **row** and save again.
13. Set the **align-items** to **stretch** to the existing #menu style rule.
14. Save the style sheet and refresh the browser. The online menu needs more adjustments.
15. Return to style sheet. Instead of having all the items on one line and getting cropped by the edge of the browser, make them **wrap** onto multiple lines by using the **flex-wrap** property with **wrap** as the value. Add this declaration to the existing #menu container.
16. Save the style sheet and refresh the browser. Resize the browser window and watch the lines rewrap. Notice that each flex item is as tall as the tallest item in that row, but rows may have different heights based on item content.
17. If you'd like, you can replace the **flex-direction** and **flex-wrap** declarations with a single **flex-flow** declaration like so: **flex-flow: row wrap;**
18. If you save the style sheet and refresh the browser, you will notice no changes at all, it looks the same as before.
19. Change the main axis alignment of items inside the main container (#menu) with **justify-content: center**
20. The last modification to the online products listing menu consists of making the price buttons line up at the bottom of each menu item. Make **section** a container for the Flexbox. In the existing **section** style rule add a declaration to set its **display** to **flex** and specifying the **flex-direction** to **column** so they continue to stack up vertically. Add those 2 declarations **by yourself**.

21. Save the style sheet and refresh the browser. It looks about the same as when the sections were made of block elements. The subtle difference is that now the neighboring margins between elements stack up and do not collapse.
22. The last step will push the paragraphs containing the prices to the bottom by using the `margin-top: auto;` declaration. Add that declaration to the existing `.price` class rule **by yourself**.
23. Save the style sheet and refresh the browser. The online menu page shows the wrapping flex items and aligned prices.
24. You will use these files in the next exercise.