

1. Use your HTML editor to open the **tf_tips_txt.html**, **tf_styles4_txt.css**, and **tf_print2_txt.css** files from the `html05 ▶ review` folder. Enter *your name* and *the date* in the comment section of each file, and save them as **tf_tips.html**, **tf_styles4.css**, and **tf_print2.css** respectively.

2. Go to the **tf_tips.html** file in your editor. Add a viewport `meta` tag to the document head to set the width of the layout viewport equal to the width of the device and set the initial scale of the viewport to 1.0.
3. Create links to the following style sheets: a) the **tf_base.css** file to be used with all devices, b) the **tf_styles4.css** file to be used with screen devices, and c) the **tf_print2.css** file to be used for printed output.
4. Take some time to study the contents and structure of the document, paying special attention to the IDs and class names of the elements, and then save your changes.
5. Go to the **tf_styles4.css** file in your editor. Note that Marjorie has placed all of her styles in the **tf_designs.css** file and imported them into this style sheet. You will not need to edit that style sheet file, but you might want to view it to become familiar with her style rules.
6. Go to the General Flex Styles section. Within this section, you'll create a flexible display layout that varies in response to changing screen widths. Note that when you use the different flex styles be sure you include the latest WebKit browser extension followed by the W3C specification.
7. In the General Flex Styles section create a style rule for the page body that displays the body as a flexbox oriented as a row, wrapping content to a new line as needed.
8. The page content is divided into two `section` elements with IDs of *left* and *right*. The left section does not need as much of the screen width. Create a style rule for the left section that sets its growth and shrink rates to 1 and 8 respectively and sets its basis size to 130 pixels.
9. The right section requires more screen width. Create a style rule for the right section that sets its growth and shrink values to 8 and 1 and sets its basis size to 351 pixels.
10. Next, you'll create a flexbox for the `section` element with class ID of *tips* that contains an article and a biographical aside, which will be displayed either in two columns or in a single column depending on the screen width. Add a style rule that displays the class of *tips* section elements as flexboxes in the row direction with wrapping.
11. The articles within each tips section need to occupy more of the screen width. Create a style rule for `article` elements that lays them out as flex items with a growth value of 2, shrink value of 1, and a basis size of 351 pixels.
12. The biographical asides within each tips section need to occupy less screen space. Create a style rule for `aside` elements that lays them out as flex items with a growth value of 1, shrink value of 2, and a basis size of 250 pixels.
13. Finally, the horizontal navigation list at the top of the page will also be treated as a flexbox. Create a style rule for the `ul` element within the horizontal navigation list displaying it as a flexbox in column orientation with wrapping. You do not have to define the sizes of the flex items because the width and height are set in the **tf_designs.css** style sheet.
14. Go to the Mobile Devices section and create a media query for screen devices with a maximum width of 480 pixels.