**How to Use the Code Editor**

1. Select the "Run Code" button to execute the program.
2. Select the "Calculate Grade" button to generate a score based on the completed tasks.
3. Continue to modify, run, and calculate your code until you are happy with the grade.
4. Select the "Submit" button to turn in the assignment to your instructor. Complete the coding activity. Your instructor will grade this activity manually.

**How to Use the Code Editor**

1. Select on the "Run Code" button to execute the program.
2. Make any necessary modifications to the code.
3. Continue to modify and run your code until you are happy with the output.
4. Select the "Submit" button to turn in the assignment to your instructor.

**Summary**

Marjorie meets with you to discuss the redesign of the blog page showing parenting tips. As with the other pages you’ve worked on, she wants this page to be compatible with mobile devices, tablet and desktop devices, and printers. Marjorie has already written the page content and has done much of the initial design work. She needs you to complete the project by writing media queries for the different display options. *Figure 5–61* shows a preview of the mobile design and the desktop design.

 *Figure 5-61*

You’ll use several flexboxes to create the layout for these two designs so that the page content automatically rescales as the screen width changes.

**Document Setup**

Open the *tf\_tips.html*, *tf\_styles4.css*, and *tf\_print2.css* files and enter your **name** and the **date** in the comment section of each file. Next, go to the *tf\_tips.html* file and 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**.

Create links to the following style sheets:

1. The *tf\_base.css* file to be used with all devices.
2. The *tf\_styles4.css* file to be used with screen devices.
3. The *tf\_print2.css* file to be used for printed output.

Open the *tf\_styles4.css* file. 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.

**General Flex Styles**

Go to the "General Flex Styles" section. Within this section, you’ll create a flexible layout that varies in response to changing screen widths.

In the "General Flex Styles" section create a style rule for the body element that:

* displays the page body as a flexbox flowing in the row direction, and
* wraps content to a new line as needed.

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 flex growth and shrink rates to **1** and **8** respectively, and
* sets its flex basis size to **130** pixels.

The right section requires more screen width. Create a style rule for the right section that:

* sets its flex growth and shrink values to **8** and **1**, and
* sets its flex basis size to **351** pixels.

Next, set the display of the section element with class ID of tips as a flexbox. Have the content of the flexbox flow in the row direction with row wrapping enabled.

Create a style rule for the article element that:

* lays it out with a flex growth value of **2**,
* sets the flex shrink value of **1**, and
* sets a flex basis size of **351** pixels.

The biographical asides within each tips section need to occupy less screen space. Create a style rule for the aside element that:

* lays it out with a flex growth value of **1**,
* sets the flex shrink value of **2**, and
* sets a flex basis size of **250** pixels.

Finally, the horizontal navigation list at the top of the page will also be treated as a flexbox. Create a style rule for the nav.horizontal ul selector that displays it as a flexbox in column orientation with wrapping.

**Mobile Devices**

Go to the "Mobile Devices" section and create a media query for screen devices with a maximum width of **480** pixels.

For mobile devices, the vertical list of links to archived parenting tips should be displayed in several columns at the bottom of the page. Within the media query you created in the last step, add the following style rules:

1. For the nav.vertical ul selector, create a style rule that displays it as a flexbox in column orientation with wrapping. Set the height of the element to **240** pixels.
2. To give the section element with an ID of left a flex order value of **99** to place it near the bottom of the page.
3. To give the body > footer selector an order value of **100** to put it at the page bottom.

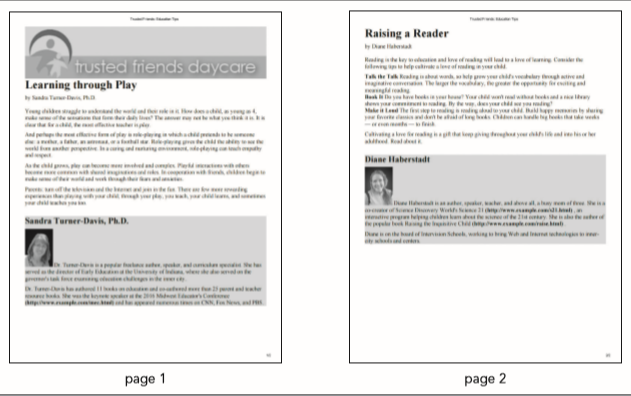
Marjorie wants to hide the navigation list at the top of the page when viewed on a mobile device unless the user hovers (or taps) a navicon. Using the technique shown in this tutorial, add the following style rules to set the behavior of the navicon within the media query for mobile devices:

1. Display the navicon by creating a style rule for the a#navicon selector to display it as a block.
2. Set the display property of the nav.horizontal ul selector to **none**.
3. Display the navigation list contents in response to a hover or touch by creating a style rule for the a#navicon:hover+ul, nav.horizontal ul:hover selector that sets its display value to **block**.

**Tablets and Desktop Devices**

Go to the "Tablets and Desktop Devices" section. Create a media query for screen devices with a width of at least **481** pixels. Under the wider screens, the contents of the horizontal navigation list at the top of the page should be displayed in several columns. In order to have the list items wrap to a new column, add a style rule to the media query that sets the height of the ul element within the horizontal navigation list to **160** pixels.

Verify that as you change the screen width the layout of the page automatically changes to match the layout designs shown in *Figure 5–61*. Next, you’ll create the print styles for the Parenting Tips page. *Figure 5–62* shows a preview of the output on a black and white printer.

 *Figure 5-62*

**Hidden Objects**

Go to the *tf\_print2.css* file and go to the "Hidden Objects" section and hide the display of the following page elements: all navigation lists, the h1 heading in the body header, the left section element, and the body footer.

**Page Box Styles**

Go to the "Page Box Styles" section and set the page size to **8.5** inches by **11** inches with a margin of **0.5** inches.

**Header Styles**

Go the "Header Styles" section and add a style rule that displays the logo image as a block with a width of **100%**.

**Typography Styles**

Go to the "Typography Styles" section and add the following style rules for the text in the printed pages:

1. For headers within the article element, set the bottom margin to **0.2** inches.
2. For h1 headings within the article element, set the font size to **24** points and the line height to **26** points.
3. For the aside element, set the background color to **rgb(211, 211, 211)** and add a top margin of **0.3** inches.
4. For h1 headings in aside elements, set the font size to **18** points and the line height to **20** points.
5. For images within aside elements, set the width to **0.8** inches.
6. For paragraphs, set the font size to **12** points with a top and bottom margin of **0.1** inches.

**Hypertext Styles**

Go to the "Hypertext Styles" section and add style rules to display all hypertext links in **black** with no underline. Also, insert a style rule that adds the text of the URL after the hypertext link in bold with the word-wrap property set to break-word.

**Page Break Styles**

Go to the "Page Break Styles" section and add the following style rules to

1. Insert page breaks after every aside element.
2. Never allow a page break within an ol, ul, or img element.
3. Set the size of widows and orphans within paragraphs to **3** lines each.

**Task 1**: When you have completed the assignment, check this task, open the task list and submit your work to your instructor.

**Task #01:** When you have completed the assignment, check this task and submit your work to your instructor.

RUN CODE

SUBMIT