# Lab 4

Tables and Forms

# Overview

This lab will exercise your knowledge of the content in the [HTML Tables](https://developer.mozilla.org/en-US/docs/Learn/HTML/Tables) and [Web Forms](https://developer.mozilla.org/en-US/docs/Learn/Forms) chapters of MDN’s [Learn web development](https://developer.mozilla.org/en-US/docs/Learn) tutorial series, specifically the following sections:

* [HTML table basics](https://developer.mozilla.org/en-US/docs/Learn/HTML/Tables/Basics)
* [HTML Table advanced features and accessibility](https://developer.mozilla.org/en-US/docs/Learn/HTML/Tables/Advanced)
* [Your first form](https://developer.mozilla.org/en-US/docs/Learn/Forms/Your_first_form)
* [How to structure a web form](https://developer.mozilla.org/en-US/docs/Learn/Forms/How_to_structure_a_web_form)
* [Basic native form controls](https://developer.mozilla.org/en-US/docs/Learn/Forms/Basic_native_form_controls)
* [The HTML5 input types](https://developer.mozilla.org/en-US/docs/Learn/Forms/HTML5_input_types)
* [Other form controls](https://developer.mozilla.org/en-US/docs/Learn/Forms/Other_form_controls)
* [Sending form data](https://developer.mozilla.org/en-US/docs/Learn/Forms/Sending_and_retrieving_form_data)

**NOTE**: The above list does not include all the sections of the Web Forms chapter. You are NOT responsible at this point for any section not in the list above. Specifically, we will cover the styling of tables and forms later in the course.

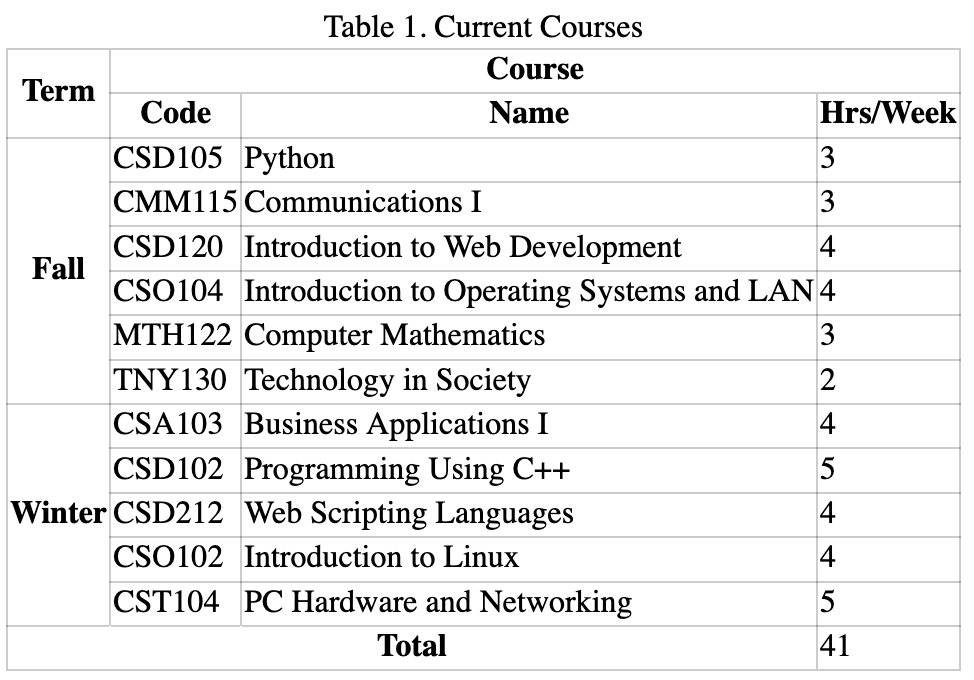
**IMPORTANT**: Before you attempt this lab, you are expected to have read through the above sections and completed the hands-on exercises and self-assessments throughout the readings to gain more hands-on experience.

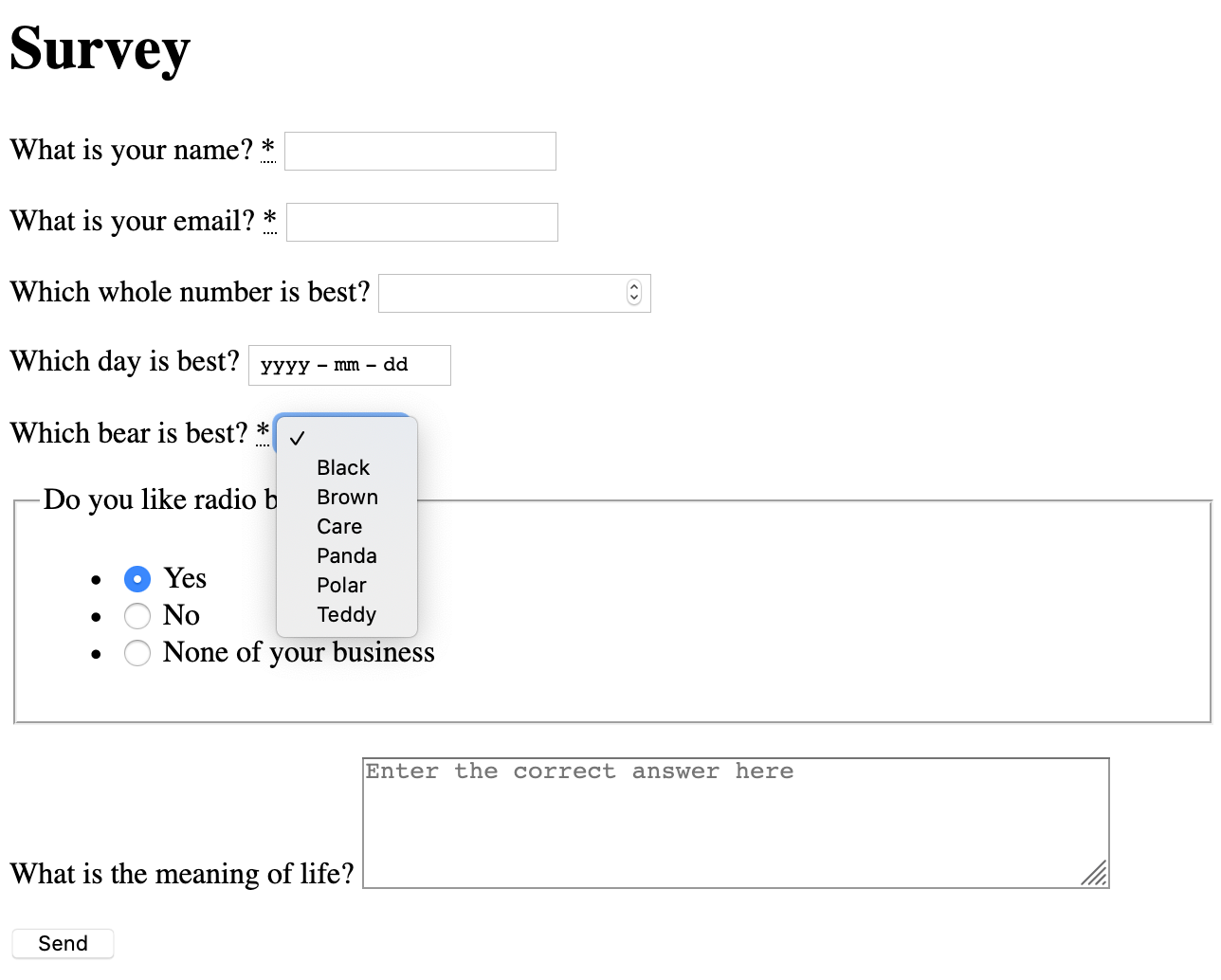
# Lab Deliverables

* Do all your work for this lab in the CSD120Labs folder you created in Lab 1
* Commit AND PUSH your completed lab work to the GitHub repository you configured in Lab 1
* After completing this lab, your CSD120Labs folder should contain the following file structure:
  + CSD120Labs
    - images
      * logo.png
      * ...
    - index.html
    - links.html
    - glossary.html

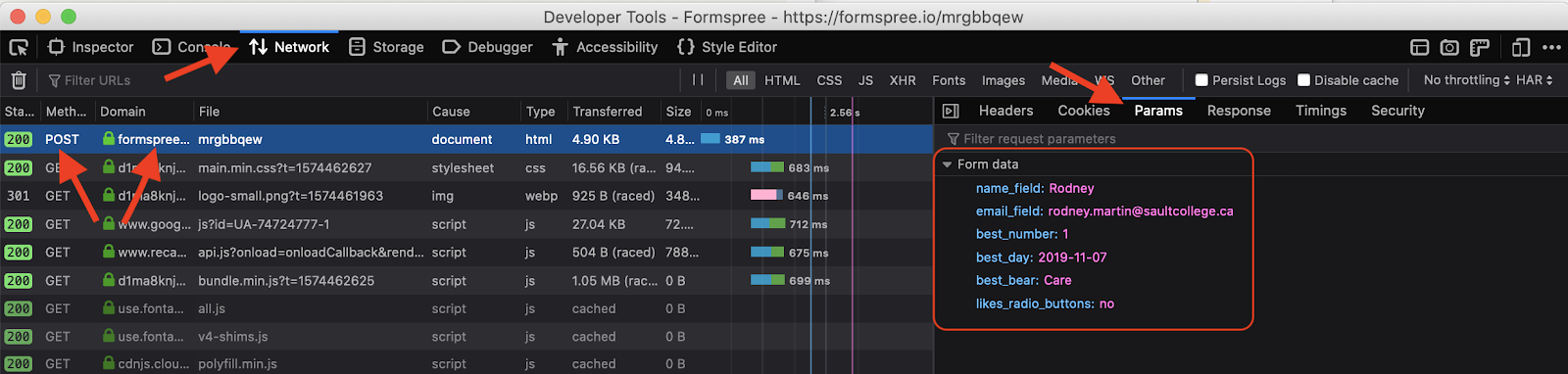
**NOTE**: It is a good idea to commit your work to Git periodically as you complete significant portions of the lab. For example, you might commit your work after finishing all the tasks in each major section below. You do not have to push each time you commit, (but you may). You only need to push when you really do want your GitHub remote to be up to date (for example, when you are ready to submit your lab to LMS, or when you want to view your updates on your GitHub Pages site). When you do so, ALL the commits that are not yet in your remote will be pushed up automatically.

# Instructions

1. Make a table
   1. Just before your “My Favourite Things” heading in index.html, add a **table** element
   2. Give the table a **caption** with the text “Table 1. Current Courses”
   3. Use appropriate HTML to produce a table like the one below, containing a list of the courses you are taking  
        
      **NOTE**: Your table should NOT have lines. (Yes, it will be ugly for now. You will style this table in a later lab.) The lines included in this image are to help you understand the table layout. If you would like to see these lines in your own web page for debugging purposes, add the following to your <head> (but be sure **remove it before submitting**!):  
        
      <style>  
       table { border-collapse: collapse; }  
       th,td { border: 1px solid rgba(0,0,0,0.2); }  
      </style>  
        
      In your table, you must...
      1. Indicate table **heading** cells using an appropriate element
      2. Use attributes to cause the cells indicated below to **span** multiple rows and/or columns
      3. Use appropriate **scope** attributes on column and row headings
      4. Distinguish the **head, body, and foot** of the table using appropriate elements. (All the rows containing course information are part of the table body. Rows above that are head, rows below are foot.)  
           
         
2. Create a web form
   1. In your main project directory create a new file named survey.html
   2. Copy the contents of index.html into survey.html
   3. In survey.html
      1. **Delete** all the contents of the <main> element
      2. Nest a **level-1 heading** inside <main> with the content “Survey”
      3. Nest a **form** element inside <main>
3. Add the following controls to your survey.html form:  
     
   **IMPORTANT**: Unless otherwise stated...
   1. Each control below must be **paired with a label** using appropriate ‘for’ and ‘id’ attribute
   2. Each label-control pair must be **inside a paragraph** element
   3. The labels for **required fields** must contain an **accessible indicator** as discussed in class
   4. A single-line text field with the following characteristics
      1. Label text: What is your name?
      2. Name: name\_field
      3. It is a required field
   5. An email field with the following characteristics
      1. Label text: What is your email?
      2. Name: email\_field
      3. It is a required field
   6. A number field with the following characteristics
      1. Label text: Which whole number is best?
      2. Name: best\_whole\_number
      3. Only integers above 0 may be entered in this field
   7. A date field with the following characteristics
      1. Label text: Which day is best?
      2. Name: best\_day
   8. A dropdown list with the following characteristics
      1. Label text: Which bear is best?
      2. Name: best\_bear
      3. Options are: Black, Brown, Care, Panda, Polar, Teddy
      4. In addition, the first option is a blank option
      5. It is a required field
   9. A set of radio buttons with the following characteristics
      1. The whole set is surrounded with a fieldset instead of a paragraph element
      2. The legend text for the fieldset is “Do you like radio buttons?”
      3. Each label-button pair is a list item in an unordered list
      4. Options are: Yes, No, None of your business
      5. Respective values for the above options: yes, no, ?
      6. Default checked option is ‘None of your business’
      7. Only one option may be checked at a time
   10. A multi-line text field with the following characteristics
       1. Label: What is the meaning of life?
       2. Name: life\_meaning
       3. 5 rows and 50 columns of text as default
       4. Placeholder text: Enter the correct answer here
   11. A button with the following characteristics
       1. No label
       2. Text on button: Send
       3. Form is submitted when the button is clicked

If you have completed the above steps correctly, your survey should look something like the image below. (Yes, it’s ugly; again, you will be fixing this later in the course using CSS.)  
  


1. Connect your form to a back end  
     
   Now let’s hook this form up to a proper back end! You can have the form submit to any server endpoint using the action attribute. For this lab, we will use a pre-configured back end from FormSpree.   
   1. Start by creating FormSpree account at <https://formspree.io/register>
   2. Once you have created your FormSpree account, log in, then click the ‘Create Form’ button
   3. Fill in the fields (use any reasonable values) on the popup that appears then click the ‘Create Form’ button
   4. Select the ‘Integration’ tab in the page that appears after creating the form
   5. Copy the endpoint URL that FormSpree has made available for you, and use this URL as the ‘action’ of your form
   6. Set your form’s method to ‘post’
   7. Try using your form! After every successful submission of the form, you should be able to view the data that was sent to your back end in the ‘Submissions’ tab for the form you set up in FormSpree.
2. **OBSERVE**: Inspect your form submissions as they are sent by the browser
   1. Open your browser inspector and click on the ‘Network’ tab to observe the raw data being sent.
   2. When your form is submitted, the first request the browser makes (visible in the main panel of the inspector) should be a POST to formspree.io. Select this request.
   3. After selecting the POST request, you should see a panel at the right with several tabs that display various information about the request. Take a moment to peruse the information in the Header tab. Note the Request URL and Request method, which correspond to the ‘action’ and ‘method’ attributes of your form element.
   4. Now select the ‘Params’ tab and behold your form data in all its glory! Note that the keys correspond to your form control ‘name’ attributes, and the values are the values that were entered by the user.



# Submission

1. **Validate** ALL your HTML files using the [w3.org validator service](https://validator.w3.org/nu/). Submitting invalid HTML will result in a lower lab grade
2. **Format** ALL your code files using the “Format Document” command found in the Command Palette (View -> Command Palette -> Format Document). Submitting poorly formatted code will result in a lower lab grade.
3. **Save** all your work in VS Code.
4. **Commit** all your changes in Git.
5. **Push** all your commits to GitHub.
6. **Submit** the **latest commit ID** AND your **GitHub URL** to the submission folder for this assignment.