Full Stack Development Lab Instructions
ENSF381: Lab03

CSS, Forms, Table Layout

Created by: Mahdi Jaberzadeh Ansari (He/Him)
Schulich School of Engineering
University of Calgary
Calgary, Canada
mahdi.ansari1@ucalgary.ca

Week 4, January 29/31, 2024

Contents

1 Introduction

1.1 Objectives

Lab 3 focuses on several key aspects of web development. You will learn to manage Git repositories using SourceTree, understand classic HTML layouts using tables, create web forms with HTML, and style web pages using CSS3.

1.2 Prerequisites

1. Basic understanding of computer operations.
2. A text editor (such as Notepad++, Visual Studio Code, or Sublime Text).
3. A web browser (Chrome, Firefox, Safari, etc.) to view your HTML file.
4. GitHub account and an installed Git client (SourceTree).

1.3 Forming groups

- In this lab session, you **MUST** work with a partner (groups of three or more are not allowed).
- The main goal of working in a group is to learn:
  - how to do teamwork
  - how to not be a single player
  - how to not be bossing
  - how to play for team
  - how to tolerate others
  - how to behave with colleagues
  - how to form a winner team
  - and ...
- Working with a partner usually gives you the opportunity to discuss some of the details of the topic and learn from each other. Also, it will give you the opportunity to practice one of the popular methods of program development called pair programming. In this method, which is normally associated with the "Agile Software Development" technique, two programmers normally work together on the same workstation (you may consider a Zoom session for this purpose). While one partner, the driver, writes the code, the other partner, acting as an observer, looks over his or her shoulder, making sure the syntax and solution logic are correct. Partners should switch roles frequently in such a way that both have equivalent opportunities to practice both roles. **Please note that you MUST switch roles for each exercise.**
- When you have to work with a partner:
  - Choose a partner that can either increase your knowledge or transfer your knowledge. (i.e., do not find a person with the same programming skill level!)
  - Please submit only one lab report with both names. Submitting two lab reports with the same content will be considered copying and plagiarism.

1.4 Before submission

- For most of the labs, you will receive a DOCX file that you need to fill out the gaps. Make sure you have this file to fill out.
- All your work should be submitted as a single file in PDF format. For instructions about how to provide your lab reports, study the posted document on the D2L called How to Hand in Your Lab assignment.
- Please note that if it is group work, only one team member must submit the solution in D2L. For ease of transferring your marks, please mention the group member's name and UCID in the description window of the submission form.
- If you have been asked to write code (HTML, CSS, JS, etc.), make sure the following information appears at the top of your code:
  - File Name
  - Assignment and exercise number
  - Your names in alphabetic order
  - Submission Date:

  Here is an example for CSS and JS files:

```
/*
=============================================================
Name         : lab3_exe_C.css
Assignment   : Lab 3 Exercise C
Author(s)    : Mahdi Ansari, William Arthur Philip Louis
Submission   : May 21, 2030
Description  : A CSS file for decorating X form
=============================================================
*/
```

- Some exercises in this lab and future labs will not be marked. Please do not skip them, because these exercises are as important as the others in learning the course material.
- In courses like ENSF381, some students skip directly to the exercises that involve writing code, skipping sections such as "Read This First," or postponing the diagram-drawing until later. That's a bad idea for several reasons:
  - "Read This First" sections normally explain some technical or syntax details that may help you solve the problem or may provide you with some hints.
  - Some lab exercises may ask you to draw a diagram, and most of the students prefer to hand-draw them. In these cases, you need to scan your diagram with a scanner or an appropriate device, such as your mobile phone, and insert the scanned picture of your diagram into your PDF file (the lab report). A possible mobile app to scan your documents is Microsoft Lens, which you can install on your mobile device for free. Please make sure your diagram is clear and readable; otherwise, you may either lose marks or it could be impossible for TAs to mark it at all.
  - Also, it is better to use the Draw.io tool if you need to draw any diagram.
  - Drawing diagrams is an important part of learning how to visualize data transfer between modules and so on. If you do diagram-drawing exercises at the last minute, you won't learn the material very well. If you do the diagrams first, you may find it easier to understand the code-writing exercises, so you may be able to finish them more quickly.
- **Due Dates:**

- You must submit your solution until 11:59 p.m. on the same day that you have the lab session.
- Submissions until 24 hours after the due date get a maximum of 50% of the mark, and after 24 hours, they will not be evaluated and get 0.

1.5 Academic Misconduct/Plagiarism

- Ask for help, but don't copy.
  - You can get help from lab instructor(s), TAs, or even your classmates as long as you do not copy other people's work.
  - If we realize that even a small portion of your work is a copy from a classmate, both parties (the donor and the receiver of the work) will be subject to academic misconduct (plagiarism). More importantly, if you give exercise solutions to a friend, you are not doing him or her a favor, as he or she will never learn that topic and will pay off for this mistake during the exam or quiz. So, please do not put yourself and your friend in a position of academic misconduct.
  - You can use ChatGPT, but please note that it may provide similar answers for others too, or even the wrong answers. For example, it has been shown that AI can hallucinate, proposing the use of libraries that do not actually exist [1]. So, we recommend that you imagine ChatGPT as an advanced search engine, not a solution provider.
  - In order to find out who is abusing these kinds of tools, we will eventually push you toward the incorrect responses that ChatGPT might produce. In that case, you might have failed for the final mark and be reported to administration.
  - If we ask you to investigate something, don't forget to mention the source of your information. Reporting without reference can lead to a zero mark even by providing a correct answer.

1.6 Marking Scheme

- You should not submit anything for the exercises that are not marked.
- In Table 1, you can find the marking scheme for each exercise.

Table 1: Marking scheme

Exercise	Marks
A	5 marks
B	15 marks
C	15 marks
D	15 marks
Total	50 marks

1.7 Complains

- Your grades will be posted one week following the submission date, which means they will be accessible at the subsequent lab meeting.
- Normally, the grades for individual labs are assessed by a distinct TA for each lab and section. Kindly refrain from contacting all TAs. If you have any concerns regarding your grades, please direct an email to the TA responsible for that specific lab.

[1] https://perma.cc/UQS5-3BBP

2 Exercise A (Creating a repository)

2.1 Initialize the Project

Like last time, we are going to clone an existing project from a remote repository. Cloning is particularly useful when you want to create a local copy of a remote repository to work on. Follow these steps:

- Create a new repository on GitHub named `Lab3`. Remember to:
  - Check the box for `Add a README file`. This file serves as an introduction and overview of your project.
  - Select `MIT License` for your project's license. This is a permissive license that allows others considerable freedom with your code while still crediting you.
- After creating the repository, it should contain two files: a `README.md` and a `LICENSE` file. The README.md is where you can write about your project, and the LICENSE file contains the license text.

2.2 Using SourceTree for Git Operations

While previously we focused on using Git commands directly, this section will introduce **SourceTree**, a free graphical interface for Git, offering a more user-friendly way to manage repositories in Windows and MacOS.

1. **Downloading and Installing SourceTree:**
   - Visit the SourceTree website and download the software based on your OS.
   - Follow the installation instructions to set up SourceTree on your system.
2. **Cloning a Repository using SourceTree:**
   - Open SourceTree and connect it to your GitHub account. Please note that if you have activated 2FA on your GitHub account, you need to use a personal access token (PAT) instead of your password. To learn more, visit this link.
   - After connecting successfully to your GitHub account, use the 'Clone' feature to clone your `Lab3` repository from GitHub to your local machine. It is better that you select the address `~/ENSF381/Lab3` as the destination on your local machine.
   - The process eliminates the need for command-line operations, providing a graphical interface for all actions. If you need more detailed instructions on how to do it, visit this link.
   - After a successful cloning, you should be able to see the `README.md` and `LICENSE` files, along with any other files that were in the remote repository.

2.3 Copy/Paste initial materials

1. Download the lab content from D2L and unzip the materials.
2. Copy and paste materials related to this lab into the `Lab3` folder. Please note that you must only copy and paste web-related materials (e.g., *.html or *.css file; **no *.docx file**). Also, if there is an `images` folder, copy and paste it with all its contents.
3. Commit and push the initial contents to GitHub, using SourceTree with this comment: 'Add initial contents'.

2.4 Inviting Collaborators to Your GitHub Repository

Collaborating with others on a project is a key aspect of software development. GitHub allows you to add collaborators to your repository. Here's how you can invite your colleagues to the repository you created in the previous steps:

- First, log in to your GitHub account and navigate to the repository you want to share. In this case, it's the `Lab3` repository.
- On your repository's page, locate the `Settings` tab near the top of the page and click on it. This will take you to the repository settings.
- In the settings menu, on the left side of the page, you will find a section called `Manage access`. Click on it.
- On the `Manage access` page, you will see a button labeled `Invite a collaborator`. Click on this button.
- GitHub will prompt you to enter the GitHub username or email address of the person you want to invite. Enter the username or email of your colleague.
- Once you enter the username or email, GitHub will suggest the matching account. Click on the account to select it.
- After selecting the collaborator, you can set the role for them. For most cases, the `Write` access level is sufficient. This allows them to push changes to the repository but doesn't allow actions like deleting the repository.
- Click on `Add collaborator` to send the invitation. Your colleague will receive an email from GitHub with a link to accept the invitation.
- Inform your collaborator to check their email and accept the invitation to collaborate on your repository.

Once your colleague accepts the invitation, they will have access to the repository and can start collaborating with you on the project. This process is crucial for teamwork in software development, allowing multiple people to work together seamlessly on the same codebase. Ask your colleges to clone the repository on their local machines with SourceTree.

Please note: *In the subsequent sections, each exercise should be completed by a single individual. It is imperative to rotate the role of the developer. This means that if Person X performs the steps in Exercise B, then Exercise C must be undertaken by Person Y, and this pattern should continue accordingly. We will check the commits on GitHub to give you marks.*

2.5 Deliverable

1. Add the address of your GitHub repository to your answer sheet file. Please make sure the repository is public and do not delete it until the end of the semester.
2. Make a screenshot from the History tab of your SourceTree application. It must at least show two commits to your GitHub in this step. Add the screenshot to your answer sheet file.

3 Exercise B (Understanding Old-School HTML Layouts)

3.1 Introduction

Historically, tables in HTML were widely used for webpage layout. This approach involved arranging content in rows and columns, providing a straightforward method to structure pages. However, it had limitations:

Pros:

- Easy to understand and implement.
- Consistent rendering across early web browsers.

Cons:

- Rigid layout structure.
- Poor accessibility and semantic meaning.
- Difficulty in maintenance and updates.
- Non-responsive design, leading to issues on different devices.

3.2 Lab Activities

1. Open the ~/ENSF381/Lab3 folder in Visual Studio Code (VSC).
2. Run the `old-school.html` file using "Open with Live Server".
3. You must probably see an output like what has been shown in Figure 1.
4. We expected to have an article with 100% width on top as the large item and 30% small and 70% medium items side by side in the second row of the table. Let's modify the CSS classes `.large`, `.medium`, and `.small` in the `old-school.html` file by uncommenting them.

```
.large { width: 100%; }
.medium { width: 70%; }
.small { width: 30%; }
```

5. Observe the changes and discuss why these modifications may not be effective.
6. Locate the following line in the `old-school.html` file and activate the line. This line will show some borders around the table's cells. Experiment with the width percentages in the CSS classes `.large`, `.medium`, and `.small`. Analyze the layout issues and write down your understanding of the reason for the problem in the answer sheet file.

```
td { border: 1px dashed gray;}
```

7. Resolve the layout problem by creating a new table in the second row to achieve a 3%-70% width split between the `.small` and `.medium` sections. Figure 2 shows the expectation.
8. Change my name in the header of the medium item to your name. Commit and push to GitHub.
9. Utilize Google Chrome's Developer Tools to assess your webpage's mobile responsiveness. Open the tools and switch to the mobile device simulator. Examine how your page appears in various mobile sizes. Is the layout and content appropriately adjusted for a smaller screen? Note any discrepancies or layout issues that may affect the mobile user experience.

3.3 Deliverable

1. Capture a screenshot of your final webpage layout. Add this image to your answer sheet file, explaining the changes you made.
2. Don't forget to add your observations and analyses from the previous section.

Figure 1: Output of the old-school.html file

Header

Google

UofC

Revolutionary New Computer Changes the Tech World

The Dawn of a New Era

In a groundbreaking development, TechCorp has unveiled a computer with the ability to learn and adapt. This new technology, which the company has been developing in secret for years, promises to revolutionize the way we interact with machines.

Unprecedented Capabilities

Experts are astounded by the computer's capabilities, which include advanced artificial intelligence, unparalleled processing speed, and an intuitive user interface. "It's unlike anything we've ever seen," said a leading tech analyst.

The Future is Here

With this release, TechCorp not only sets a new standard in computing but also opens up a world of possibilities for future technology. The company plans to make this technology available to the public by the end of the year.

Calgary's Natural Beauty Captivates Tourists

Mahdi Ansari Takes a Historic Spacewalk Near the ISS

Fictional News Article for Educational Use at UofC

Figure 2: Output of the final old-school.html file

Header

Google
UofC

Revolutionary New Computer Changes the Tech World

The Dawn of a New Era

In a groundbreaking development, TechCorp has unveiled a computer with the ability to learn and adapt. This new technology, which the company has been developing in secret for years, promises to revolutionize the way we interact with machines.

Unprecedented Capabilities

Experts are astounded by the computer's capabilities, which include advanced artificial intelligence, unparalleled processing speed, and an intuitive user interface. "It's unlike anything we've ever seen," said a leading tech analyst.

The Future is Here

With this release, TechCorp not only sets a new standard in computing but also opens up a world of possibilities for future technology. The company plans to make this technology available to the public by the end of the year.

Calgary's Natural Beauty Captivates Tourists

Emerald Oasis in the Heart of the City

Calgary, known for its vibrant city life, is now gaining fame for its lush green spaces and rich biodiversity. The city's parks and riversides have become a haven for nature lovers.

Urban Wildlife Flourishes

Recent reports indicate a rise in urban wildlife sightings, adding to the city's allure. Deer, coyotes, and various bird species are now common sights in Calgary's green spaces.

A Green Transformation

The transformation of the city into a green paradise has been praised by environmentalists, calling it a perfect blend of urban development and nature conservation.

Your Name Takes a Historic Spacewalk Near the ISS

A Historic Moment in Space Exploration

In an unprecedented feat of human courage and technological prowess, astronaut Mahdi Ansari completed a spacewalk near the International Space Station (ISS), marking a new milestone in space exploration.

A Breathtaking Journey

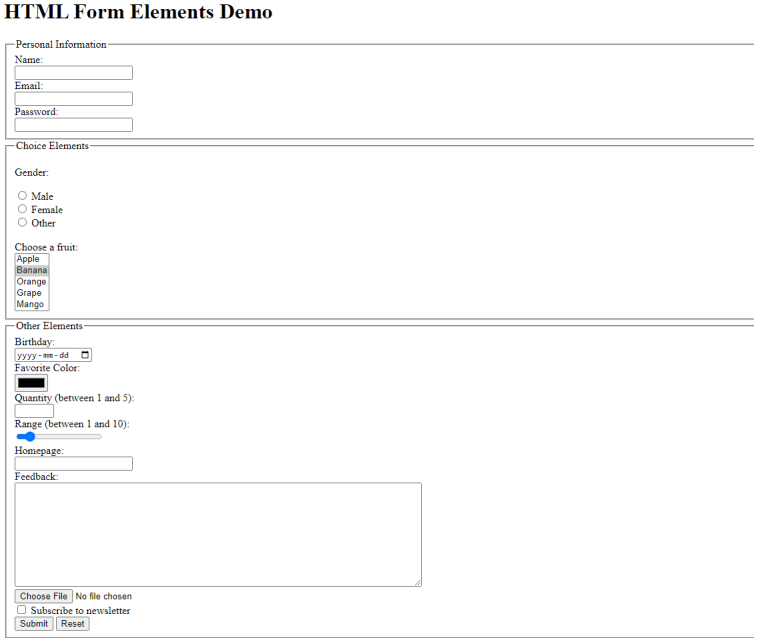
Fictional News Article for Educational Use at UofC

4 Exercise C (Working with form tags)

In this section, we will explore the basic elements of creating forms. The anticipated outcome of these activities is depicted in Figure 3. This exercise will not only improve your HTML skills but also familiarize you with the process of interpreting and implementing code modifications based on inline documentation and comments.

HTML is well-documented online. Use resources such as W3Schools, MDN Web Docs, and HTML.com as some references for solving this exercise.

Figure 3: Final output of the 'form.html' file



4.1 Fixing Issues in an HTML File

To enhance your understanding of HTML and gain hands-on experience, follow these steps:

1. Remember to rotate the developer role for this exercise. Each member should get an opportunity to contribute.
2. First, pull the latest changes that your colleagues have made to the project.
3. We expected to have a `form.html` file in the repository by now. However, if it is not there, download the `form.html` file available on the D2L platform. Add the downloaded file to your project folder. Commit the addition to your Git repository with the message 'Add form file' and push the changes to GitHub. If the file has already been added, ignore this step.
4. Before making any modifications, open the `form.html` file in a web browser to view its current state.
5. Begin updating the file based on the instructions provided within the comments in the HTML code. There are nine specific comments you need to address. Each comment guides you on how to modify or enhance a particular part of the HTML file.
6. Commit and push the final changes to your GitHub repository.

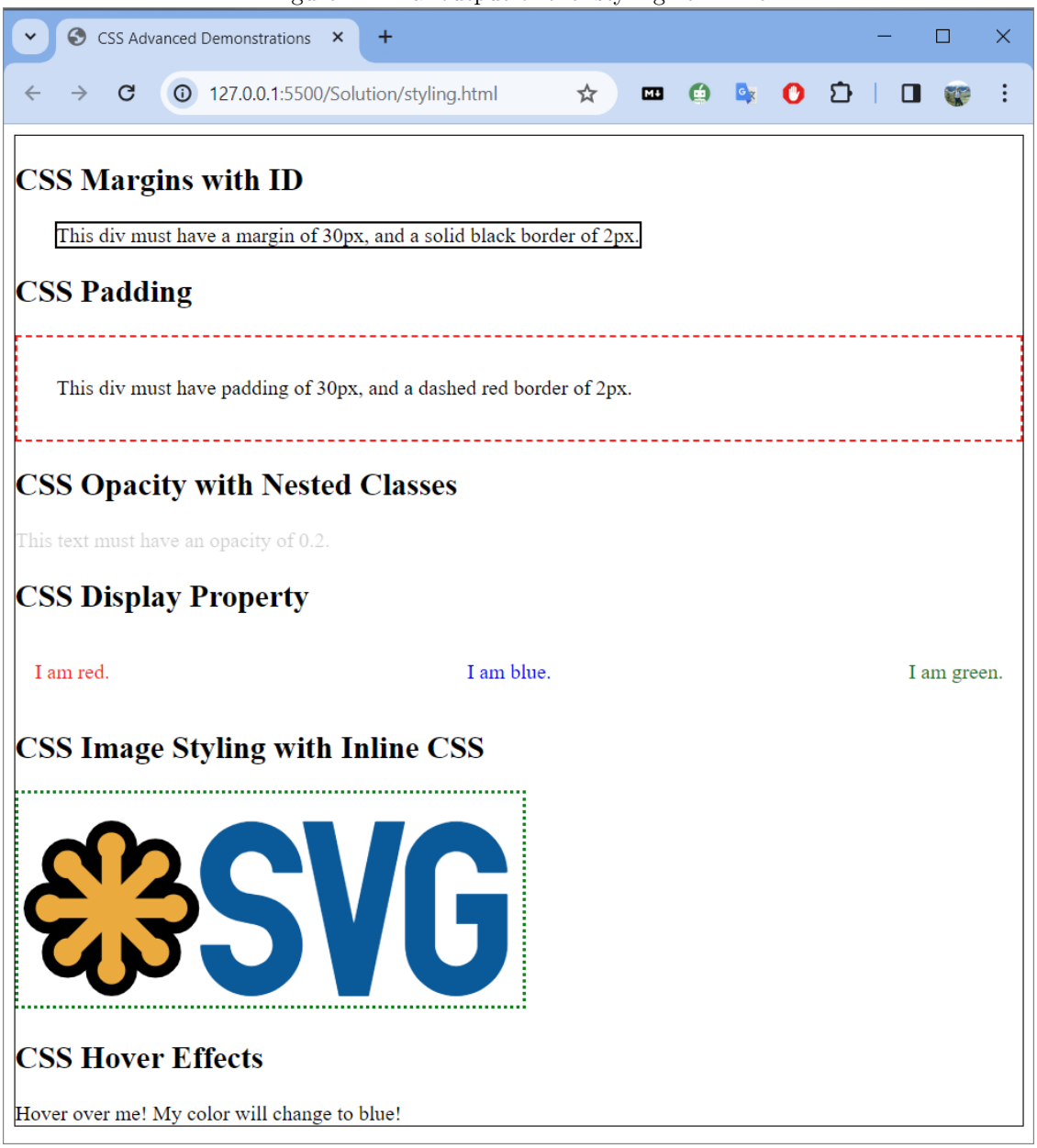
4.2 Deliverable

1. Make a screenshot from the <html> tag of the final file, as you did in the previous exercise. **Please note that the screenshot must be from the rendered page, not the code.** Just make the screenshot using your browser facilities. Include this screenshot in your answer sheet file.

5 Exercise D (Working with CSS)

This exercise focuses on enhancing your CSS skills, particularly in styling and layout techniques. The expected outcome has been shown in Figure 4.

Figure 4: Final output of the 'styling.html' file



5.1 Exercise Overview

1. Download the `styling.html` file from D2L and add it to the project if it is not there. Otherwise, ignore this step.
2. Open the `styling.html` file in the VSC editor.
3. Follow the inline comments in the file to add or modify CSS styles. These comments guide you through various CSS tasks. You can ignore the following list and stick to the comments inside the file. However, the comments are as follows:
   (a) **Margin and Border:** Modify `#margin-demo` to have a margin of 30 pixels and a solid black border of 2 pixels.
   (b) **Padding and Border Style:** Adjust `.padding-demo` to include 30 pixels padding and a dashed red border of 2 pixels.
   (c) **Opacity:** Change the opacity of `.nested-opacity` to 0.2.
   (d) **Text Alignment:** Ensure `.block-demo` centers its text.
   (e) **Color and Float:** Apply colors and float properties to the children of `.block-demo`. The first child should be red and float left, the second blue and float in the center, and the third green and float right.
   (f) **Image Styling:** Modify the `.image-demo` class to adjust the image's width and add a dotted green border inline.
   (g) **Hover Effect:** Create a hover effect for `.hover-demo` that changes its color to blue and cursor to grabbing.
4. Test your changes by viewing the file in a web browser.
5. Ensure each modification meets the requirements outlined in the comments.
6. Commit and push your changes to GitHub.
7. Reflect on how each CSS change impacts the layout and appearance of the webpage.

5.2 Deliverable

1. Capture a screenshot of your final webpage, showing the successful implementation of the CSS tasks. The screenshot must include the address bar of your browser as well. Include this screenshot in your answer sheet file.
2. Observe the black border around the <`body`> element. We have not explicitly set the visible margin around this border in the <`body`> tag. Research online to understand why this margin exists by default in web browsers. Ensure to include references in your explanation. Any response without proper citations will not be considered. Additionally, investigate how this default margin can be removed or overridden, and discuss the methods for achieving a margin-free body element in your webpage. This understanding is essential for web design and layout control.