LAB 7b

BEGINNING WITH TAILWIND CSS

What You Will Learn

- How to do some base styling with Tailwind CSS.
- How to use the Tailwind CLI tool

Approximate Time

The exercises in this lab should take approximately 30 minutes to complete.

Fundamentals of Web Development, 3rd Ed

Randy Connolly and Ricardo Hoar

Textbook by Pearson
http://www.funwebdev.com

Date Last Revised: January 9, 2024

PREPARING DIRECTORIES

The starting lab07b folder has been provided for you (within the zip file/folder downloaded from Gumroad).

1 If you haven't done so already, create a folder in your personal drive for all the labs for this book.

2 Copy the folder titled lab07b from the zip file/folder to your course folder.

Note: these labs use the convention of blue background *text to indicate filenames or folder names and* **bold red** *for content to be typed in by the student.*

(VERY) QUICK TOUR OF TAILWIND CSS

The Tailwind CSS has become perhaps the most popular CSS Framework used by practicing front-end developers. It is especially popular with developers and designers working with component-based JavaScript environments such as React.

The normal Tailwind workflow involves using the `tailwindcss` CLI (Command Line Interface) build tool, which requires using the `npm` CLI, which in turn requires installing `node`. You will eventually learn how to use such tools, but at this early stage, we will initially simply add in what is called the Play CDN to our HTML files so that we can try Tailwind without the need of a build step.

Exercise 7b.1 — USING TAILWIND CLASSES

1 Test lab07b-ex01.html in browser.

2 Examine lab07b-ex01.css in your code editor.

In this exercise, you will be replacing the styles in the CSS file with Tailwind equivalents in order to create a similar style result.

2 Replace the `<link>` element with the following (some markup omitted).

```
<head>
  ...
  <title>Lab07b<title>
  <script src="https://cdn.tailwindcss.com"></script>
</head>
```

This may be the first time you have seen the `<script>` tag. It is used to include a JavaScript file. Why do we need a JavaScript file here? Normally, you will use the tailwindcss build program which generates a minimized CSS file based on the Tailwind classes used in your HTML file; this generated CSS file can then be added using the <link>

element like with any other external css file. Initially, we are not using this build program, so this included JavaScript file here will allow us to use tailwind without the build step.

3 Test in browser.

Notice that including the Tailwind script has removed some of the default browser styling, such as fonts and margins.

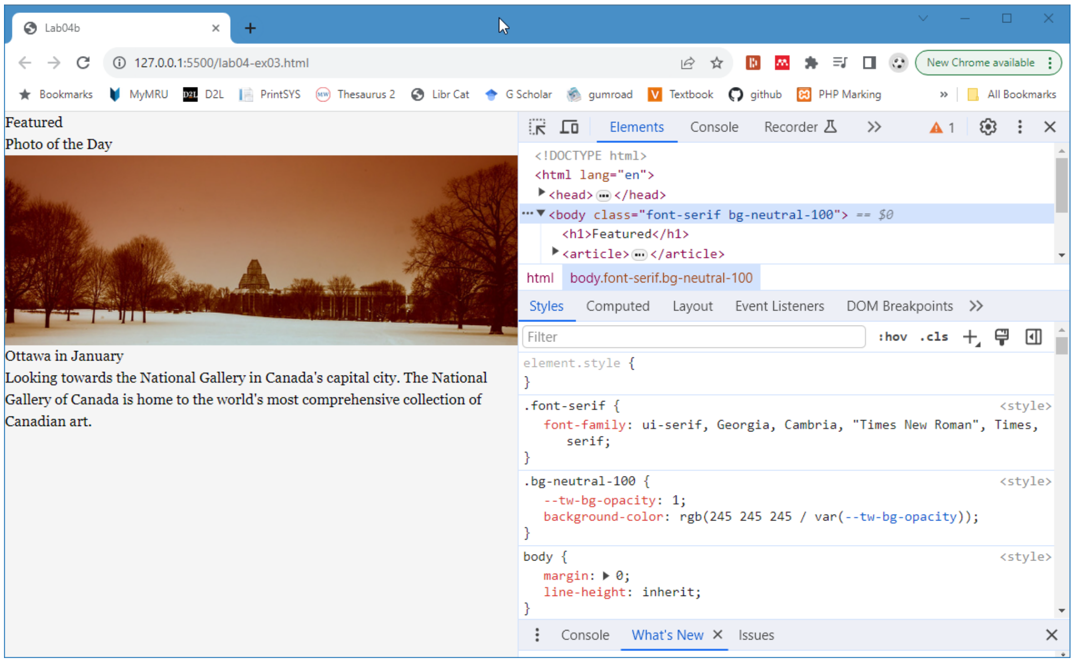
4 Add the following to the `<body>` element and test.

```
<body class="font-serif bg-neutral-100">
```

You style web elements in Tailwind using utility or atomic classes that map to a single CSS property.

5 Using the browser's Inspect facility (or DevTools), examine the styles for the body element. You should see something similar to that shown in Figure 7b.1.

Most of the Tailwind classes map to a single CSS property and value pair.



Featured
Photo of the Day

Ottawa in January
Looking towards the National Gallery in Canada's capital city. The National Gallery of Canada is home to the world's most comprehensive collection of Canadian art.

```
<!DOCTYPE html>
<html lang="en">
 <head> … </head>
 <body class="font-serif bg-neutral-100"> == $0
   <h1>Featured</h1>
  <article> … </article>
```

```
element.style {
}
.font-serif {                                                        <style>
   font-family: ui-serif, Georgia, Cambria, "Times New Roman", Times,
      serif;
}
.bg-neutral-100 {                                                    <style>
   --tw-bg-opacity: 1;
   background-color: rgb(245 245 245 / var(--tw-bg-opacity));
}
body {                                                               <style>
   margin: ▸ 0;
   line-height: inherit;
}
```

Figure 7b.1 – Tailwind classes generally map to a single CSS property and value pair.

6 Add the following to the `<body>` element and test.

```
<body class="font-serif bg-neutral-100 p-4">
```

This adds padding…but how much?

7 Visit the official documentation at `https://tailwindcss.com/docs/padding` and examine the different padding classes and their values.

Notice that there are classes for padding values in a variety of different em sizes. Using these values makes it more likely that you will make use of standardized spacing values. As well, the tailwind CLI tool will only include the Tailwind classes you actually use in your project.

8 Add the following classes to the `<h1>` element and test.

```
<h1 class="font-serif font-extrabold text-neutral-500 text-2xl
uppercase tracking-widest">
```

Because this is just HTML we are modifying, it makes no difference what order you add these classes (remember, there is no CSS file that we are using: the one that was provided is there only for you to compare).

At around this point, most students will begin to feel a certain revulsion at this way of styling web pages. It does feel much like inline CSS styling. Its advantage over inline CSS is that by choosing classes from Tailwind, you are making use of a design system (for instance, pre-defined colors), which makes it easier to maintain visual consistency (in comparison to inline CSS). As well, Tailwind's classes can be made responsive, unlike inline CSS.

9 Visit one of the several online Tailwind cheat sheets, such as `https://tailwindcomponents.com/cheatsheet` or `https://nerdcave.com/tailwind-cheat-sheet`. Try experimenting with different values for font size, font weight, letter spacing, and text color.

10 Add the Tailwind class `ml-3` to the `<h1>` element and test. Can you guess what it does? Use the official documentation or one of the cheatsheet sites to find the answer.

11 Copy the class list from the `<h1>` element and add it to the `<h2>` element. Change the margin to `ml-2`, the letter spacing to `tracking-wider`, and the font size to `text-xl`. Test.

12 Examine the styles defined for `<article>` in `lab07b-ex01.css`. See if you can use the Tailwind documentation to achieve a similar (but not exact) result using the Tailwind classes.

Hint: the rounded corners and the shadow created by the CSS property `box-shadow` is split into two Tailwind classes: `shadow` and `rounded`. Here is the solution I came up with:

```
<article class="m4 p-2.5 w-3/4 shadow rounded bg-white">
```

13 Do the same for the remaining styles (`figure`, `figcaption`, and `p`) defined in `lab07b-ex01.css`.

Congratulations. You now know how to use Tailwind CSS!

Exercise 7b.2 — Tailwind Layout Classes

1 Test `lab07b-ex02.html` in browser.

2 Examine `lab07b-ex02.css` in your code editor.

In this exercise, you will be replacing the styles in the CSS file with Tailwind equivalents in order to create a similar style result. You can use this version to see the non-Tailwind version.

3 In `lab07b-ex02.html` replace the `<link>` element with the following

```
<script src="https://cdn.tailwindcss.com"></script>
```

4 Add the following styles to the `<body>` element.

```
<body class="font-serif font-bold m-6 bg-gray-100">
```

5 Add the following styles to the `<header>` element and test.

```
<header class="text-gray-500 mb-2">
   <h1 class="font-sans font-black text-3xl">Layout Exercise</h1>
   <p class="italic">Uses grid layout, auto margins, and
       inline-block</p>
</header>
```

6 Add the following styles to the `<section>` element.

```
<section class="w-3/4 grid grid-cols-3 gap-10">
```

This adds a variety of relevant grid layout classes. Compare to the CSS definitions for this element in `lab07b-ex02.css`.

7 Add the following styles to the `<figure>` and `<img>` elements and test.

```
<figure class="m-0 p-4 bg-white shadow-md rounded-lg">
   <img src="images/museum.jpg" alt="British Museum"
      class="mx-auto p-2 border border-solid border-gray-300">
```

8 Go to one of the Tailwind documentation sites (from the previous exercise) and search for "`mx-auto`".

It sets the margin-left and margin-right to auto. Prior to flex box layout, this was the most common way to center a block element with another block element.

9 Set up the text styling as follows and test.

```
<figcaption class="m-3">
   <h2 class="text-center font-light text-2xl mt-4">British
Museum</h2>
   <p class="text-gray-900 font-light mt-2 italic text-justify">
The library in the British Museum in London. The British Museum
Reading Room, situated in the centre of the Great Court of the
British Museum, used to be the main reading room of the British
Library. </p>
   <p class="text-gray-900 font-light mt-2 text-justify">
In 1997, this function moved to the new British Library building at
St Pancras, London, but the Reading Room remains in its original form
at the British Museum.</p>
   <p class="text-center mt-4">
      <a href="#">Visit</a>
   </p>
</figcaption>
```

10 Set up the link styling as follows and test.

```
<p class="text-center mt-4">
  <a href="#"
      class="inline-block w-2/5 rounded bg-indigo-900
      hover:bg-indigo-500 p-4 no-underline uppercase
      font-light text-gray-100"
  >
    Visit
  </a>
</p>
```

This changes the link element from inline to inline-block, which allows you to specify widths, widths, padding, or any other block-level styling.

Notice also that it specifies hover behavior (in this case, a color change). Many classes in Tailwind can be applied conditionally by adding a modifier to the beginning of the class name that describes the condition. These include pseudo classes (such as hover and focus) and media queries.

11 Now reproduce these same classes for the other two figures.

This should make you hate Tailwind a little bit: imagine if there were 50 figures!

Tailwind is not really ideal (in fact, it is border-line awful) for repetitive HTML styling. *Where it really shines is in situations where HTML elements are being dynamically generated in vanilla client-side JavaScript, React JavaScript, or even with server-based element generation using something like PHP or Node.*

Also, there is a way to construct new classes out of existing Tailwind classes using the `@apply` directive, which would make it easier to apply the same styling to multiple elements. However, this does require using the tailwind CLI tool.

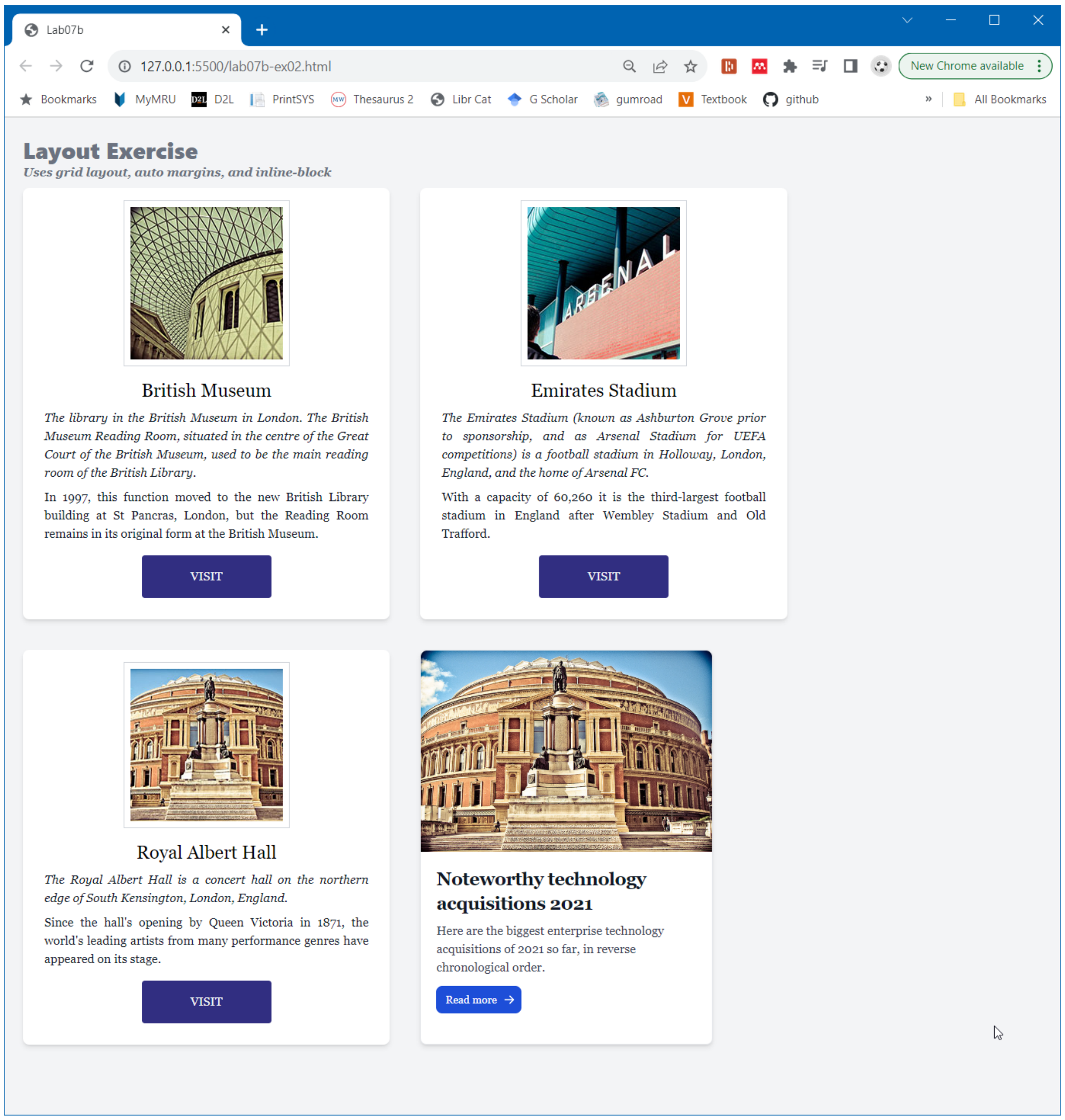
12 Visit the following URL: `https://tailwindcomponents.com/component/tailwind-css-card` and click on the Show Code button, which will display the markup along with the Tailwind classes used in the example.

*Because developing with Tailwind results in markup styled with Tailwind's utility classes, sites using Tailwind can be examined easily in the browser. They can also be easily replicated! These markup + Tailwind classes are typically called **Tailwind Components**, and if you search for those two words, you will find dozens of sites offering free and purchasable examples.*

13 Copy the markup for this example card (but not the first wrapper `<div>`), and paste it into your `lab07b-ex02.html` example after the last `<figure>` but within the `<section>` container.

The result should look similar to that shown in Figure 7b2.

This type of copying and pasting of styling examples would rarely if ever work easily with "normal" CSS examples, but works easily with Tailwind.



Layout Exercise

Uses grid layout, auto margins, and inline-block

British Museum

The library in the British Museum in London. The British Museum Reading Room, situated in the centre of the Great Court of the British Museum, used to be the main reading room of the British Library.

In 1997, this function moved to the new British Library building at St Pancras, London, but the Reading Room remains in its original form at the British Museum.

VISIT

Emirates Stadium

The Emirates Stadium (known as Ashburton Grove prior to sponsorship, and as Arsenal Stadium for UEFA competitions) is a football stadium in Holloway, London, England, and the home of Arsenal FC.

With a capacity of 60,260 it is the third-largest football stadium in England after Wembley Stadium and Old Trafford.

VISIT

Royal Albert Hall

The Royal Albert Hall is a concert hall on the northern edge of South Kensington, London, England.

Since the hall's opening by Queen Victoria in 1871, the world's leading artists from many performance genres have appeared on its stage.

VISIT

Noteworthy technology acquisitions 2021

Here are the biggest enterprise technology acquisitions of 2021 so far, in reverse chronological order.

Read more →

Figure 7b.2 – Completed Exercise 7b.1

Exercise 7b.3 — Tailwind Responsive Approach

1 Test `lab07b-ex03.html` in the editor and then in the browser.

2 Experiment by changing the browser window width: in particular, examine what happens when the window width is made very small or very wide.

Because the container element (the `<section>`) uses a width class that is a percentage of the window width (`w-3/4`), the cards within the container also grow or shrink. While this might be okay, it doesn't work very well at the very wide or the very small (i.e., mobile) sizes.

3 Visit the following URL: `https://tailwindcss.com/docs/responsive-design`.

Notice that you can add responsive prefixes to ANY Tailwind class.

4 Make the following change to `lab07b-ex03.html` and test (by changing the browser width).

```
<section class="w-full lg:w-4/5 grid grid-cols-2 gap-10">
```

5 Make the following change and test.

```
<section class="w-full xl:w-11/12 grid grid-cols-1 md:grid-cols-2
xl:grid-cols-3 gap-10">
```

Tailwind takes a mobile-first approach to responsive utilities. This means you begin by specifying the default design for mobile, and then add responsive utilities to alter it for larger device sizes.

6 Make the following changes and test.

```
<figure class="m-0 p-4 bg-white shadow-md rounded-lg">
   <img src="images/museum.jpg" alt="British Museum"
     class="mx-auto w-3/5 lg:w-min p-2 border border-solid border-
gray-300 ">
   <figcaption class="m-3">
      <h2 class="text-center font-light text-4xl lg:text-2xl mt-
4">British Museum</h2>
      <p class="text-xl lg:text-base text-gray-900 font-light mt-2
italic text-justify">The library in the British Museum in London.
The British Museum Reading Room, situated in the centre of the Great
Court of the British Museum, used to be the main reading room of the
British Library. </p>
      <p class="text-xl lg:text-base text-gray-900 font-light mt-2
text-justify">In 1997, this function moved to the new British
Library building at St Pancras, London, but the Reading Room remains
in its original form at the British Museum.</p>
      <p class="text-center mt-4">
         <a href="#" class="text-xl lg:text-base w-4/5 lg:w-2/5
inline-block rounded bg-indigo-900 hover:bg-indigo-500 p-4 no-
underline uppercase font-light text-gray-100">Visit</a>
      </p>
   </figcaption>
</figure>
```

Here we are changing the text sizes and image/button widths for different screen sizes.

7 If you wish, make the same changes for the other `<figure>` elements.

Tool-Based Tailwind

In the exercises so far, you have used a Tailwind script to get Tailwind working. This approach is less than ideal from a performance perspective. The ideal approach is to use the Tailwind CLI which will generate a minimal CSS file based on the classes actually used. As well, the CLI will allow you to use more advanced features to customize Tailwind, for instance, to use different fonts or different colors, to alter the responsive breakpoints, create custom classes, or make use of other custom components.

Warning: The next exercise uses npm, which requires installing Node. This is detailed in Labs 11 and 13, so if you don't feel ready for this yet, then just ignore this section.

Exercise 7b.4 — Tailwind CLI

1 Assuming you have already installed Node, in the same folder you have used so far in this lab, run the following command via the terminal/command window/GIT Bash window.

```
npm install -D tailwindcss
```

You will now have a node_modules folder and a package.json file that specifies Tailwind as a dependency.

2 Run the following command:

```
npx tailwindcss init
```

This will create the Tailwind config file.

3 Examine the just-created `tailwind.config.js` file in the editor.

4 Create a folder named `src` inside your current working folder.

5 Move the provided `lab07b-ex04.html` file into the `src` folder.

6 Make a copy of the `images` folder and paste it into the `src` folder.

6 Examine the `lab07b-ex04.html` file. Notice in the `<head>` section it includes two new fonts from Google Fonts.

In order for Tailwind to use these fonts, we will have to make a change to our configuration file and to the CSS class names.

7 Add the following to the `tailwind.config.js` file.

```
module.exports = {
  content: ["./src/**/*.{html,js}"],
  theme: {
    extend: {
      fontFamily: {
        dosis: ['Dosis', 'sans-serif'],
        merri: ['Merriweather', 'serif'],
      }
    },
  },
  plugins: [],
}
```

Here we are telling the Tailwind CLI to process all HTML and JavaScript files in the src folder; we are also adding two new Tailwind CSS classes for these two new Google fonts.

8 In `lab07b-ex04.html` in the `src` folder, make the following changes.

```
<h1 class="font-merri font-black text-3xl">
...
<h2 class="font-merri text-center font-light text-4xl lg:text-2xl
mt-4">British Museum</h2>
<p class="text-xl lg:text-base text-gray-900 font-dosis mt-2
italic text-justify">The library in the British Museum ...
<p class="text-xl lg:text-base text-gray-900 font-dosis mt-2
text-justify">In 1997 ...
```

Here we are using the new class names defined in step 7.

9 Now run the following command in the terminal.

```
npx tailwindcss -o ./src/css/output.min.css --minify -watch
```

This command generates a CSS output file in minified format in the specified folder. It also will regenerate this css file anytime any content (location specified in `tailwind.config.js`) changes.

Notice that after you run this command, you now have a output.min.css file in the src/css folder.

10 In `lab07b-ex04.html` in the `src` folder, add the following to the `<head>` and then test in browser.

```
<link href="css/output.min.css" rel="stylesheet">
```

11 Add the same font changes to the other figures. Because we specified the watch flag, when you save your source file, the output CSS is generated automatically.

There is still much more to learn with Tailwind, but hopefully this lab has helped you begin using Tailwind!