CS 328 - Week 6 Lab Exercise - 2026-02-26

Deadline

Due by the end of lab.

Purpose

To write a CSS external stylesheet that contains CSS rules to style an HTML document, adding a few additional kinds of selectors and properties and property values to those discussed in class so far.

How to submit

Submit your files for this lab using **`~st10/328submit`** on nrs-projects, each time entering a lab number of **86**.

Requirements

- You are required to work in **pairs** for this lab exercise.
  - This means **two** people working at **ONE** computer, one typing ("driving"), one saying what to type ("navigating"),

    while **BOTH** are looking at the **shared** computer screen and **discussing** concepts/issues along the way.
- Make sure **BOTH** of your names appear in each file submitted.
- When you are done, before you leave lab, **BOTH** of you should submit appropriate versions of these files using **`~st10/328submit`** on nrs-projects, with a lab number of **86**.

Task: Lab set-up

- Copy over the following "starter" HTML file into your desired `public_html` subdirectory, that you will then add to as part of today's lab exercise:

  `cp ~st10/lab6-start.html lab6.html`
- Fill in the opening comment block as specified, putting in your **names**, the last modified **date**, and the **URL** that can be used to run your document.
  - (You *will* lose some credit if this URL does not work when I or the grader paste it into a browser!)
- **Within** the **`head`** element, *after* the `link` element specifying that this document is being styled by `normalize.css`, add a *second* **`link`** element following course style standards specifying that this document will then be further styled using a file **`lab6.css`** that you are about to create for this lab exercise.
  - **Do not include any inline or internal CSS style sheets in your `lab6.html`.**
- Within the **`footer`** element near the end of the **`body`** element, in the **`p`** element whose content includes **`YOUR NAMES`**, replace **`YOUR NAMES`** with your names.
- Read through your **`lab6.html`** and notice what HTML elements are included in its **`body`** element.

Note 1: Available list of CSS properties

In the posted CS 328 CSS References, you can find a link to an alphabetical hyperlinked index of CSS properties that is part of the MDN CSS Web Docs:

https://developer.mozilla.org/en-US/docs/Web/CSS/Reference

Note 2: Some of the ways to represent colors in CSS

You will be reading more about CS color options in the zyBooks course text, in (soon-to-be-assigned) Chapter 3 - Section 3.4.

In the meantime, are several of the ways to specify colors in CSS:

- **predefined** color name - for example, `red`
- **hexadecimal** color code - for example, `#ffa500`
- **red-green-blue** color code - for example, `rgb(255, 165, 0)`
- **hue-saturation-luminance** code - for example, `hsl(128, 128, 64)`
- **semi-transparent red-green-blue-alpha** color code - for example, `rgba(255, 165, 0, 0.5)`

Note, also, that some of the references in the posted CS 328 CSS References include useful color-related tools.

Task: Create a CSS external style sheet `lab6.css`

Create an external CSS file **`lab6.css`** in the same directory as your **`lab6.html`** to style your **`lab6.html`**:

- Start this with an opening comment block containing your **names** and the last modified **date**.
- Now write CSS style rules -- each of which has a **visible, noticeable** effect in your displayed **`lab6.html`** -- meeting the following requirements as well as the posted **"CS 328 CSS Coding Standards so far"**.
  - You can **mix and match** and **combine** properties in your rules such that the following requirements are all met in your resulting **`lab6.css`**.

At least one rule should have multiple declarations

- The definition of a CSS **declaration** was discussed in Week 6 Lecture 1, and is included in its projected notes posted in the CS 328 In-class Examples.

At least one rule should have an element selector

- The definition of a CSS **element** selector was discussed in Week 6 Lecture 1, and is included in its projected notes posted in the CS 328 In-class Examples.

At least one rule's selector part should include grouping selectors

- What is meant by **grouping** selectors in a CSS rule's selector part was discussed in Week 6 Lecture 1, and is included in its projected notes posted in the CS 328 In-class Examples.
- That is, at least one rule's selector part should be a comma-separated list of selectors.

At least one rule should have a class selector

- A **class** selector is one that starts with a period, followed by a `class` attribute value -- for example:

  `.moo`

  - This selects all elements that include the attribute **`class="moo"`**.

- This can be combined with an element selector. That is, if you put an element name before the period, it only selects elements of that type that have the specified value for the `class` attribute -- for example:

  `fieldset.oink`

  - This selects only **`fieldset`** elements that include the attribute **`class="oink".`**

- Several of the elements in **`lab6.html`** include a `class` attribute. Include at least one rule that uses a class selector that visibly styles at least one of these elements.

At least one rule should have an id selector

- An **id** selector is one that starts with a `#`, followed by an `id` attribute value -- for example:

  `#moo`

  - This selects the at-most-one element that includes the attribute **`id="moo"`**.

- This also can be combined with an element selector. That is, if you put an element name before the period, it only selects the at-most-one element of that type that has the specified value for the `id` attribute -- for example:

  `fieldset#oink`

  - This selects only the at-most-one **`fieldset`** element that includes the attribute **`id="oink".`**

- Two of the elements in **`lab6.html`** include an `id` attribute. Include at least one rule that uses an id selector that visibly styles at least one of these elements.

At least two rules should use the `color` property...

- ...to visibly style **at least two** different kinds of elements,
- ...using **at least two** different colors.

At least two rules should use the `background-color` property...

- ...to visibly style **at least two** different kinds of elements,
- ...using **at least two** different colors.

Within your rules' declarations, represent colors using at least three of the five ways...

- ...discussed in the **Note 2** section earlier in this lab exercise handout.

Within your rules' declarations, use at least two different properties that are NOT color-related...

- ...to visibly style **at least two** different kinds of elements.

- Because we did not get to very many properties in Week 6 Lecture 1, here are some examples of such properties:
  - **`text-align`** (can be used for inline elements, with values such as **`left`**, **`right`**, **`center`**)
  - **`text-decoration`** (with a value such as **`underline`**)
  - **`font-weight`** (with a value such as **`bold`**)
  - **`font-style`** (with a value such as **`italic`**)
- Note: **if** you decide to include a property whose value is a numeric size, **NOTE** the following:
  - When a property value has a unit, there can be NO blank between the value and that unit. (That is, **`3em`** would work, but `3 em` would not!)
  - CS 328 class style is to use so-called ***relative*** units for sizes instead of ***absolute*** units. We'll be discussing this further, but note that this means **`em`** (the width of a capital M in the current font) and **`%`** are fine, but `pt` and `px` are **NOT**.)
- (We will also be discussing the **CSS Box Model**, including what is meant by padding, border, and margin.
  - But in the meantime, if you would like some space between your content and the left-hand-side of the window, a rule setting the **`body`** element's property **`margin-left`** or **`margin`** to a small value -- for example, **`1em`** -- will do this.)

Task: Validate your files:

Make sure an **`.xhtml`** copy of your **`lab6.html`** validates as **strict-style HTML**,

AND that your **`lab6.css`** validates as **valid CSS** at https://jigsaw.w3.org/css-validator/.

Task: BEFORE you leave lab:

Make sure that you **both** have copies of the files:

- **`lab6.html`**, **`lab6.xhtml`**, and **`lab6.css`**

...and you BOTH submit these using **`~st10/328submit`** on nrs-projects, with a lab number of **86**.

How the navigator can get `lab6.html` and `lab6.css`:

(for a driver with username *dr12*, and a navigator with username *na89* - replace these with your *actual* usernames when you actually do this)

- Because these are files the nrs-projects web server has to be able to reach, the navigator should be able to get a copy of them from the driver using an approach like this:
  - Assume the driver has, in their `public_html` directory, a sub-directory `328lab06` containing `lab6.html` and `lab6.css`. (Adapt the following accordingly based on your driver's actual `lab6.html` and `lab6.css` location.)
  - The NAVIGATOR *na89* can now:
    - log in to THEIR (*na89*'s) nrs-projects account, and run these commands:

```
cd public_html
mkdir 328lab06   # or other name they choose
```

```
chmod 711 328lab06
cd 328lab06
cp ~dr12/public_html/328lab06/lab6.html .  # note space & period!
cp ~dr12/public_html/328lab06/lab6.css .  # note space & period!
```

  - Now the navigator *na89* should have their own copies of **`lab6.html`** and **`lab6.css`**.

- Either repeat for **`lab6.xhtml`**,

  or (if desired) change the URL within the navigator's **`lab6.html`** to refer to the navigator's copy, and then make the **`.xhtml`** copy (and double-check that it still validates):

  ```
  cp lab6.html lab6.xhtml
  ```