

## CS 328 - Week 3 Lab Exercise - 2025-02-06/07

### Deadline

Due by the end of lab.

### Purpose

To practice creating and validating a strict-style HTML document that includes practice with a **table** and a **form** element, including a selection of form widgets.

### How to submit

Submit your files for this lab using `~st10/328submit` on nrs-projects, each time entering a lab number of **83**.

### Requirements

- You are required to work in **pairs** for this lab exercise, in the BSS 313 lab.
  - 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 **83**.
- For this lab exercise's problems, the **only** CSS permitted is **EITHER**:
  - **Just** the external CSS `normalize.css` included in `html-template.html`, **OR**
  - The external CSS `normalize.css` included in `html-template.html`, **after** which you **also** put:

```
<link href="https://nrs-projects.humboldt.edu/~st10/styles/lab3-table.css" type="text/css" rel="stylesheet" />
```

(and this would be right **BEFORE** the `head` elements closing tag)  
...which adds a *little* mostly-table-related formatting.

### Lab Exercise set-up

As a pair:

- Copy the contents of the posted `html-template.html` into an `.html` file whose name includes **lab3** somewhere within it, somewhere in the navigator's `public_html` directory.
  - (That is, it might be in their `public_html` directory, or it might be in a world-executable subdirectory of their `public_html` directory.)
  - You could copy and paste the contents from one of the "syntax-colored" display versions of `html-template.html` posted on the course public site's home page or linked along with this

handout on the course Canvas site.

- You could also use a command such as this on nrs-projects, **assuming** you are in your desired work directory for today's lab (and varying the name as desired, as long as it includes **lab3** within it):

```
cp ~st10/html-template.html lab3.html
```

As you work through the following, ensure that this file contains **strict-style HTML that meets class coding and style standards** as well as meets the **requirements** below.

## Problem 0

To your **\*lab3\*.html** file for today's lab exercise, add the following:

- In the **comment** after the **html** element's start tag and right before the **head** element's start tag:
  - After **by:**, put **your names** (first *and* last names)
  - After **last modified:**, put **today's date**
  - On the line after **you can run this using the URL:**, put the (**absolute**) **URL** that can be used to view this **\*lab3\*.html** from a browser on the web
    - Check that this URL works! (successfully displays your document when pasted in a browser)
    - (You *will* lose credit if this URL does not work when pasted into a browser during grading!)
- Within the **head** element, edit the **title** element to have descriptive content appropriate for this document.
- Within the **body** element, include an appropriate **h1** element indicating this document is for the **CS 328 Week 3 Lab Exercise**
- Within the **footer** element, add a **p** element whose content includes **your last names**.

## Problem 1

To your **\*lab3\*.html** file for today's lab exercise, within its **body** element and before its **footer** element, add the following **after** the **h1** element you added in Problem 0:

- Add an appropriate **h2** element indicating the parts following are for **Problem 1**
- Within one or more **p** elements, include:
  - an **img** element for a *small* image of your choice
    - If your image's URL does not validate as strict-style HTML when used as your **img** element's **src** attribute,  
make a copy of the image in your nrs-projects account (if you can legally do so) or use a service such as such as tinyurl to avoid problematic characters, and use that URL instead.
  - a visible **special character** of your choice (something starting with **&** and ending with **;**, such as **&amp;**; for an ampersand character or **&lt;**; for a less-than character or **&#0223**; for a Greek beta)
- **code** is an **inline** element intended to represent a short fragment of computer code (see <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/code>).
- **pre** is a **block** element that represents preformatted text, which should be used for content in which newlines, blanks, etc. are considered semantically important (see

<https://developer.mozilla.org/en-US/docs/Web/HTML/Element/pre>).

And, it is considered good style to wrap your **code** element in a **pre** element if it is a fragment of computer code representing *more* than one line, in which the spacing and line breaks are to be preserved.

So: to practice with this:

- Decide on a programming language of your choice,
- and include a **pre** element whose content is a **code** element whose content is **at least 3 lines** of a code fragment of your choice using your chosen programming language.
- (You may also include other strict-style HTML elements in this part of your document if you wish.)
- End this problem's part with an **hr** element.

### ***validate your document-so-far***

Before you go on...

- REMEMBER that it does not check for all of the CS 328 required style standards -- but you can use an `.xhtml` copy of your document with <https://validator.w3.org/nu> or <https://html5.validator.nu> to assist you in validating and checking over your document.
- SAVE a copy of your document-so-far with your Problem 1 elements as follows -- at the nrs-projects prompt, type the Linux command:
 

```
cp your-1ab3-file-name.html your-1ab3-file-name.xhtml
```
- Go to either <https://validator.w3.org/nu> or <https://html5.validator.nu> and paste in the absolute URL of the `your-1ab3-file-name.xhtml` version of your file and click the "Validate" button.
  - SAD-BUT-TRUE: these validators seem to get overwhelmed frequently -- if you get an error "Bad Gateway 502", that's likely the culprit.
  - Wait a little bit, then try again. If necessary, let me know, submit what you have by the end of lab, and if you try later and find there was something you needed to fix, resubmit your fixed file and let me know, so that version will be graded instead.
  - Correct any issues that are shown before going on to Problem 2.

### **Problem 2 - add a table element**

Decide on some tabular data you'd like to include in your document, consisting of **at least 3 rows** and **at least 3 columns**.

- (It can be as simple as showing data for three of the departments from `set-up-ex-tbls.sql`'s dept table, or it can be data about a topic you choose.)

To your HTML document for today's lab exercise, add the following **after** the **hr** element you added at the end of Problem 1's parts above:

- Add an appropriate **h2** element indicating the parts following are for **Problem 2**.
- Add a **table** element structuring and displaying your tabular data that also meets the following requirements:

- It should include an appropriate **caption** element.
- It should include appropriate **tr** elements that contain appropriate **td** and/or **th** elements.
- It should include at least one **th** element, and each **th** element should include an appropriate **scope** attribute.
- End this problem's part with an **hr** element.

### **validate your document-so-far**

Before you go on...

- Again SAVE a copy of your document-so-far, now also including your Problem 2 elements:  
`cp your-lab3-file-name.html your-lab3-file-name.xhtml`
- Go to either <https://validator.w3.org/nu> or <https://html5.validator.nu> and paste in the absolute URL of the `your-lab3-file-name.xhtml` version of your file and click the "Validate" button.
  - Correct any issues that are shown before going on to Problem 3.

### **Problem 3 - add a form element**

To your HTML document for today's lab exercise, add the following **after** the **hr** element you added at the end of Problem 2's parts above:

- Add an appropriate **h2** element indicating the parts following are for **Problem 3**.
- Add a **form** element that meets the following requirements:
  - It should have an **action** attribute whose value is a "real" URL of your choice (because we haven't gotten to writing an actual application program to handle this form yet).
  - It should have a **method** attribute whose value is "get".
  - It should contain **at least two input** elements with **type="text"** with appropriate **name** and **id** attributes,  
and appropriate logically-related **label** elements for each of those **input** elements.
  - It should include an **input** element with **type="submit"**
- Remember that, for this lab exercise, you may **not** use CSS to format or layout this form, although you **can** tastefully use **br** void elements, **p** elements, or **div** elements if you wish.
- Try entering some info into your form, click submit, and **look over** the name=value pairs that appear at the end of your form's action URL as a result.
  - **Think about:** can you explain why those are the name=value pairs submitted?

### **validate your document-so-far**

Before you go on...

- Again SAVE a copy of your document-so-far, now also including your Problem 3 elements:  
`cp your-lab3-file-name.html your-lab3-file-name.xhtml`

- Go to either <https://validator.w3.org/nu> or <https://html5.validator.nu> and paste in the absolute URL of the *your-1ab3-file-name.xhtml* version of your file and click the "Validate" button.
  - Correct any issues that are shown before going on to Problem 4.

## Problem 4 - add a `fieldset` element

### *fieldset* element

- Used to visually group logically-related form widgets within a form
- IMPORTANT TO KNOW:
  - A `fieldset` element is a block element that can have content.
  - Its intended content is logically-related form widget elements (and their associated elements, such as `label` elements, etc.)
  - Browsers typically format this by placing a **thin solid border** around its contents.
  - If you include a `legend` element within a `fieldset`'s content, it "labels" that border.

To your HTML document for today's lab exercise:

- **Modify** your `h2` element for **Problem 3** so that it now indicates that the parts following are for **Problems 3, 4**.
- Add a `fieldset` element that contains an appropriate `legend` element as well as your `input` elements and their `label` elements that you added in Problem 3.
  - (that is, place this `fieldset` around your existing `input` elements, also including a `legend` element)

### *validate your document-so-far*

Before you go on...

- Again SAVE a copy of your document-so-far, now also including your Problem 3 elements:  
`cp your-1ab3-file-name.html your-1ab3-file-name.xhtml`
- Go to either <https://validator.w3.org/nu> or <https://html5.validator.nu> and paste in the absolute URL of the *your-1ab3-file-name.xhtml* version of your file and click the "Validate" button.
  - Correct any issues that are shown before going on to Problem 5.

## Problem 5 - add some checkboxes

### *checkbox*

- A `checkbox` is an `input` element with `type="checkbox"`
- Used when a user is to select, or not select, some specific option.

- A collection of checkboxes can be a good choice for allowing a user to make **ZERO OR MORE** choices from a set of specific options.
- **IMPORTANT TO KNOW:**
  - An `input` element with `type="checkbox"` is a checkbox.
  - Reminder: the `input` element is a void element, so it has no content.
  - Browsers typically format a checkbox as a small square -- clicking it checks or unchecks this checkbox.
  - It should have a `name` attribute.
    - If this checkbox is checked when its form is submitted, this name attribute's value will be sent as the name in a name=value pair.
    - By default, the value for the name=value pair submitted for a checked checkbox will be `on` (although if there is a `value` attribute, its value will be the value instead.)
    - What if a checkbox is not checked when its form is submitted? In that case, **NO** name=value pair will be submitted for this checkbox!
  - Include attribute `checked="checked"` for EACH checkbox you'd like to be **INITIALLY** selected when the form is created
    - **USE THIS FOR GOOD, NOT EVIL!** ...consider: what is a **REASONABLE** default? considering user convenience...!
  - Add a logically-related `label` element along with each checkbox instance to logically associate that checkbox's label text with that checkbox.

To your HTML document for today's lab exercise:

- Modify your `h2` element for **Problems 3, 4** so that it now indicates that the parts following are for **Problems 3, 4, 5**.
- Add another `fieldset` element that contains an appropriate `legend` element and also contains:
  - **at least three** checkboxes, **at least one** of which is initially shown as checked
  - an appropriate logically-related `label` element for each checkbox
- Try checking and unchecking your checkboxes, and:
  - Notice what happens when you click on the label for a checkbox.
  - Look over the name=value pairs that appear at the end of your form's action URL as a result when you submit your form.
  - **Think about:** can you explain why those are the name=value pairs submitted?

### ***validate your resulting document***

Before you go on...

- Again **SAVE** a copy of your document-so-far, now also including your Problem 5 elements:  
`cp your-lab3-file-name.html your-lab3-file-name.xhtml`

- Go to either <https://validator.w3.org/nu> or <https://html5.validator.nu> and paste in the absolute URL of the *your-1ab3-file-name.xhtml* version of your file and click the "Validate" button.
  - Correct any issues that are shown before submitting your *\*1ab3\*.html* and *\*1ab3\*.xhtml* files.
- If necessary, let me know, submit what you have **by the end of lab**, and if you try later and find there was something you needed to fix, resubmit your fixed file and let me know, so that version will be graded instead.

### BEFORE you leave lab:

Make sure that you *each* have copies of the files *\*1ab3\*.html* and *\*1ab3\*.xhtml* on your nrs-projects accounts.

And, **EACH** of you should submit your versions of *\*1ab3\*.html* and *\*1ab3\*.xhtml* using `~st10/328submit` with a homework number of **83**.

- I will leave it up to the navigator to decide if they would like to UPDATE their *your-1ab3-file-name.html* and *your-1ab3-file-name.xhtml* so its opening comment includes the URL to *their* copy, or if they want to leave the URL for the driver's copy.
  - HOWEVER: NOTE that you *will* lose some credit if the URL in your *your-1ab3-file-name.html* file does not work when I or the grader paste your submitted file's URL into a browser, in either case.
  - (So, if the driver were to MOVE or delete their copy of *your-1ab3-file-name.html* before this lab exercise is graded, the navigator would also lose some credit if they included the driver's URL here.)

### Reminder: how the navigator can get copies of these files

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

- Because this is a file the nrs-projects web server has to be able to reach, the navigator should be able to get a copy of this file from the driver using an approach like this:
  - Assume the driver has, in their `public_html` directory, a sub-directory `3281ab03` containing *your-1ab3-file-name.html*. (Adapt the following accordingly based on your driver's actual *your-1ab3-file-name.html* location.)
  - The NAVIGATOR `na89` can now:
    - log in to THEIR (`na89`'s) nrs-projects account, and run these commands:
 

```
cd public_html
mkdir 3281ab03 # or other name they choose
chmod 711 3281ab03
cd 3281ab03
cp ~dr12/public_html/3281ab03/your-1ab3-file-name.html . # note space & period!
```
    - Now the navigator `na89` should have their own copy of *your-1ab3-file-name.html*

- Either repeat for *your-1ab3-file-name.xhtml*,  
or (if desired) change the URL within the navigator's *your-1ab3-file-name.html* to refer to the navigator's copy, and then make the *.xhtml* copy (and double-check that it still validates):  
**cp *your-1ab3-file-name.html* *your-1ab3-file-name.xhtml***