# CS 328 - Week 2 Lab Exercise - 2026-01-29

## Deadline

Due by the end of lab.

## Purpose

To practice creating **and validating** a strict-style HTML document, and to practice with some HTML elements.

## How to submit

Submit your files for this lab using **`~st10/328submit`** on nrs-projects, each time entering a lab number of **82**.

## Requirements

- You are required to work in **pairs** for this lab exercise, in the BSS 317 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 **82**.
  - (Suggestions for creating the navigator's copy of these files are included later in this handout.)

## Problem 1

Now, as a pair:

- Copy the contents of the posted **`html-template.html`** into a file named **`lab2.html`** somewhere in the driver'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 this command on nrs-projects, **assuming** you are in your desired work directory for today's lab:

    **`cp ~st10/html-template.html lab2.html`**

### *update the opening comment*

- 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 **`lab2.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!)

## *edit the `title` element*

- Consider the following information about the **HTML `title` element**:
  - It *can* have content.
  - It should appear *within* the `head` element (it should be a child element of the `head` element).
  - from https://html.spec.whatwg.org/multipage/semantics.html#the-title-element :

    "The `title` element represents the document's title or name. .... The document's title is often different from its first heading, since the first heading does not have to stand alone when taken out of context.

    There must be no more than one `title` element per document."
  - Its content is document metadata, data about the data -- it is not part of the document's content itself. However, many browsers will use its content in the **tab** displaying the document.
- Knowing the above, now **EDIT** the current contents of the **`title`** element in your document `lab2.html` to have contents appropriate for this document.

## *add an `h1` (level-1 heading) element*

- Consider the following information about the **HTML `h1` (level-1 heading, top-level heading) element**:
  - It can have content.
  - It is a block element, and it should appear *within* the `body` element.
  - It represents a top-level heading for content.
- Knowing the above, now ADD an appropriate **`h1`** element to your document `lab2.html`.

## *add a `p` (paragraph) element*

- Consider the following information about the **HTML `p` (paragraph) element**:
  - It can have content.
  - It is a block element, and it should appear *within* the `body` element.
  - It represents a paragraph in one's content.
- Knowing the above, now ADD an appropriate **`p`** element to your document `lab2.html` that happens to include **BOTH of your names**.

## *add a `ul` (unordered list) or `ol` (ordered list) element*

- Consider the following information about the **HTML `ul` (unordered list)** and **`ol` (ordered list) elements**:
  - These have content, but that content is limited to instances of the **HTML `li` (list item) element**!
  - These are block elements, and these should appear *within* the `body` element.
  - **`ul`** should be used for an unordered list (for example, a bulleted list) in one's content, and

    **`ol`** should be used for an ordered list (for example, a numbered list) in one's content.
  - The **`li`** element should be used for a list item *within* either `ul` or `ol` elements, and the `li` element can have content.
  - You can have sublists, but the syntax is just a little persnickety, so ask me if you are interested.
- Knowing the above, decide on a topic and **ADD** an appropriate **`ul`** *or* **`ol`** element containing **at least three** appropriate **`li`** elements related to that topic to your document `lab2.html`

## *add an `a` (anchor) element*

- Consider the following information about the **HTML `a` (anchor) element**:
  - It can have content.
  - It is an inline element; it should be within an appropriate "containing" element, for example `p` or `li`.
  - This element is used to indicate text that is hyperlinked to other text or to another document.
  - It has an **attribute `href`** - hypertext reference - whose value is typically a URL (although some other options are also possible).
    - This URL can be **absolute**, starting with a protocol such as **`https://`**,

      ...or it can be **relative**, the path to the desired other document relative to THIS HTML document (in this lab's case, relative to **`lab2.html`**).
      - NOTE: AVOID starting a relative URL with a forward slash! Linux systems such as nrs-projects will treat those differently than you want.
  - Its content should **describe** the document being linked to.
- Knowing the above, decide on something you would like to link to (for example, a web site you like, or the CS 328 public course web page, or a document on the web, etc.) whose absolute URL is not too long, and:
  - Decide where it makes sense to put this -- it can be in your document's current `p` element, in a list element in your document's current list, or in a new `p` or list element in a new list that you add.
  - Add an appropriate **`a`** element in the location you have chosen, linking to your chosen site or file and including appropriate descriptive content.

## *add an `em` (emphasis) or `strong` (strong emphasis) element*

- Consider the following information about the **HTML `em` (emphasis) element**:
  - It can have content.

  - It is an inline element; it should be within an appropriate "containing" element, for example `p` or `li`.
  - This element is used to indicate content that is to be emphasized.
  - Most browsers' default is to italicize emphasized content (although this can be modified using CSS).
- Consider the following information about the **HTML `strong` (strong emphasis) element**:
  - It can have content.
  - It is an inline element; it should be within an appropriate "containing" element, for example `p` or `li`.
  - This element is used to indicate content that is to be strongly emphasized.
  - Most browsers' default is to display strongly-emphasized content using boldface (although this can be modified using CSS).
- Knowing the above, decide on something you would like to emphasize or strongly emphasize. (This can be content currently in your `lab2.html`, or new content that you add.)
  - Make sure it is within an appropriate "containing" element (for example, in your document's current `p` element, in a list element in your document's current list, or in a new `p` or list element in a new list that you add).
  - Add an appropriate **`em`** element or **`strong`** element in the location you have chosen, emphasizing or strongly emphasizing your desired content.

## *optional additions*

- You may include additional elements to your `lab2.html` if you would like, but note that strict-style HTML is a CS 328 style requirement for full credit.

## *before you go on*

- Make sure your `lab2.html` works! (That is, make sure it displays when you paste its absolute URL from its opening comment into a browser!)

## *validate your document*

- SAVE a copy of your document as follows -- at the nrs-projects prompt, type the Linux command:

  **`cp lab2.html lab2.xhtml`**
- Go to either https://validator.w3.org/nu or https://html5.validator.nu

  and paste in the absolute URL of the **`lab2.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.
  - If you get this error, try the other validator.
  - If you get this in both validators:
    - 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.

- Once it validates, the driver's `lab2.html` and `lab2.xhtml` are ready to be submitted after you complete Problem 2;

  but if it does not validate, you should work together to correct this `lab2.html` and repeat the above until its `lab2.xhtml` copy successfully validates.

## Problem 2

Now, make a version of **`lab2.html`** in the *navigator's* `public_html` directory on nrs-projects. (This is to double-check yet again that everyone's nrs-projects account are appropriately set up for web pages!)

- 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 username `dr12`, and the navigator has username `na89`. (Replace these with your actual usernames when you actually do this.)
  - Assume the driver has, in their `public_html` directory, a sub-directory `328lab02` containing `lab2.html`. (Adapt the following accordingly based on your driver's actual `lab2.html` location.)
  - The NAVIGATOR `na89` can now:
    - log in to THEIR (`na89`'s) nrs-projects account, and run these commands:

      ```
      cd public_html
      mkdir 328lab02   # or other name they choose
      chmod 711 328lab02
      cd 328lab02
      cp ~dr12/public_html/328lab02/lab2.html . # note the space & period!
      ```

    - Now the navigator `na89` should have their own copy of `lab2.html`.
- You only need to make **ONE** change to the navigator's version of `lab2.html` -- modify its opening comment so that it contains the (absolute) URL one can use to view the *navigator's* version of `lab2.html`, the version in the *navigator's* nrs-projects account.
  - (for example, changing the `~dr12` to `~na89`, and possibly changing the subdirectory)
  - Again, double-check that the modified URL in this comment indeed works!
- In the *navigator's* nrs-projects account, make an `.xhtml` copy of the modified `lab2.html`:

  **`cp lab2.html lab2.xhtml`**
- Just to make sure, validate the *navigator's* modified file:
  - Go to either https://validator.w3.org/nu or https://html5.validator.nu

    and paste in the absolute URL of the **`lab2.xhtml`** version of this file

    and click the "**Validate**" button.
- Once it validates, now both the driver's and the navigator's files `lab2.html` and `lab2.xhtml` are ready to be submitted;

  but if it does not validate, you should work together to correct this `lab2.html` and repeat the above

until its `lab2.xhtml` copy successfully validates.

## Before you leave lab:

Make sure that you *each* have files **`lab2.html`** and **`lab2.xhtml`** with the correct URL in the opening comment for successfully executing *your* copy from *your* nrs-projects account.

And, **EACH** of you should submit your versions of **`lab2.html`** and **`lab2.xhtml`** using **`~st10/328submit`** with a homework number of **82**.