

CS 328 - Week 9 Lab Exercise - 2026-03-26

Deadline

Due by the end of lab.

Purpose

To practice creating a PHP postback-style document that can create a form and (hopefully-safely) obtain data from that form when it is submitted (and also practice a little with a server-side include along the way).

How to submit

Submit your files using `~st10/328submit` on nrs-projects, each time entering a lab number of **89**.

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.
- Make a `public_html` **sub-directory** for today's lab exercise with a **not-too-obvious name** of your choice, and be sure to **chmod** it to permissions **711**.
- And, your **.php** and **.css** and **.txt** files for this lab exercise should all have permissions **644**.
- 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 **89**.
- You are expected to follow CS 328 style standards, including those from the posted "CS 328 PHP Coding Standards so far" (at <https://nrs-projects.humboldt.edu/~st10/s26cs328/328-php-coding-standards.pdf>).

Set up your files for this lab exercise

- Use the course HTML template to start up a PHP document `328lab09.php`:

```
cp ~st10/html-template.html 328lab09.php
```
- Copy this file into the **same** directory as your file `328lab09.php`:

```
cp ~st10/328lab09-form.txt . # DON'T FORGET the SPACE and PERIOD!!
```
- Copy this external CSS file into the **same** directory as your file `328lab09.php`:

```
cp ~st10/328lab09.css . # DON'T FORGET the SPACE and PERIOD!!
```

Requirements for 328lab09.php

In your file `328lab09.php`:

- Fill in the opening comment block, 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**, edit the **`title`** element, giving it appropriate content.
- **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 the file `3281ab09.css`.
 - **Do not include any inline or internal CSS rules in your `3281ab09.php`.**
- Within the **`body`** element, include an **`h1`** element with appropriate content.
- **Within the `footer` element** near the end of the **`body`** element, add a **`p`** element whose content includes your names.

You are now going to add to this so that `3281ab09.php` will be a **postback** document, so that, depending on the method used to request it, it **either** creates a form **or** crafts a response to that form when it is submitted (as demonstrated in the posted Week 9 Lecture 2 example `3281lect09-2.php`).

In an `if` statement's `if`-action in the body of `3281ab09.php`: What form?

Here, you are going to try out one of PHP's functions that allow for so-called **server-side includes**.

In the **`body`** element of `3281ab09.php`, *after* its **`h1`** element and *before* its **`footer`** element, you should start a regular PHP tag that has the beginning of an **`if`** statement.

- This **`if`** statement's **`if`**-part should, if the `$_SERVER` array's value for the key `"REQUEST_METHOD"` is `"GET"`, create a form by including the contents of the file `3281ab09-form.txt` using the statement:


```
require_once("3281ab09-form.txt");
```

 - This can go in a regular PHP tag, and has the side-effect of including the contents of the specified file in the resulting PHP response.
 - (Fun facts: this is supposed to throw a fatal error if it cannot find the file, and if you accidentally were to try to use this to include the same file more than once within a PHP, this is supposed to make sure that it would only include its contents once.)
 - (Why did I give this file a suffix `.txt`? To help stress that it is not an entire HTML document -- this file contains JUST a `form` element!)

In the `if` statement's `else`-action in `3281ab09.php`: What response?

If you have not already done so yet, **read** the **`form`** element in `3281ab09-form.txt`. When this form is submitted, your PHP will be grabbing and sanitizing the name=value pairs from its submission.

That is,

- ...the **first** time a browser requests your `3281ab09.php`, the method will be `"GET"`, and your PHP's response will include the form in `3281ab09-form.txt`.
- But, when the user fills out the form and then clicks the submit button in this form, the resulting submission will request `3281ab09.php` with a method of `"POST"`, and will include the name=value pairs from this submitted form.

So, your PHP **if** statement's **else**-part should:

- Craft a response that *cautiously* displays, within a **ul** element, the values submitted from that form.
- **REMEMBER!**
 - You can **CLOSE** a regular PHP tag, put desired HTML, and include PHP expression tags within that HTML as desired to include the values of desired PHP expressions in the executed result.
 - Start another regular PHP tag to then include the ending **}** for your **else** part. (See the posted **328lab09-2.php** if you are not sure what I mean by this.)
 - Do **not** include any PHP `echo` or `print` statement in your `328lab09.php`.
- **REMEMBER! Never trust user data!** Appropriately use **htmlspecialchars** and/or **strip_tags** so that you don't unwittingly execute user-injected inappropriate code or other badness.
- **IN ADDITION:** function **trim** expects a string, and removes any leading or trailing blanks from that string, and returns the resulting string.
 - This is also useful for more-pragmatic sanitizing!
 - For example, you can use **trim** to remove any accidental leading or trailing blanks that users probably-accidentally entered into, for example, input fields and text areas.
- **FUN FACT:** I don't *think* you'll need it for handling this form, but you can see IF an array KEY exists in an associative array using:


```
array_key_exists($desired_key, $desired_array)
```
- **FUN FACT:** I don't *think* you'll need it for handling this form, but you can see IF a variable or array reference has a value set for it using:


```
isset($var_or_array_ref)
```

How can one strict-validate the HTML resulting from a PHP?

Beware -- if you put the URL of this PHP directly into the validator at <https://html5.validator.nu/>, it looks like it does regular-HTML validation (not strict-validation) of just the response containing its form. This is not a bad start, but it is not strict-validation, nor does it check your PHP's response with a submitted form's values.

So, do the following to strict-validate your PHP's resulting HTML:

- Put your **328lab09.php**'s URL in a browser and view its source, copy and paste that source into a file named **328lab09-1.xhtml**, and put the URL of your **328lab09-1.xhtml** into the validator.
- Put your **328lab09.php**'s URL in a browser **and fill out and submit its form**, *then* view that *response's* source, and copy and paste that *response's* source into a file named **328lab09-2.xhtml**, and put the URL of your **328lab09-2.xhtml** into the validator.

Quick note about 328lab09.css

In the interests of time -- and to give another example of using CSS flexbox and CSS grid layout to lay out the parts of a form -- an external CSS file **328lab09.css** is being provided as part of the lab exercise.

HOWEVER, if you have the time and interest, you may **ADD rules** to and/or **ADD declarations** to rules in your copy of **328lab09.css**.

IF you do this:

- **ADD** an "**adapted by:** " followed by your names to `328lab09.css`' opening comment block
- If you change any of the default foreground and background colors, make sure the result is **at least WCAG 2 AA Compliant** based on the tester at: https://snook.ca/technical/colour_contrast/colour.html
- Make sure your resulting `328lab09.css` still validates as valid CSS.

BEFORE you leave lab:

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

- `328lab09.php`
- `328lab09.css`
- `328lab09-form.txt`
- `328lab09-1.xhtml`, `328lab09-2.xhtml`

...and you BOTH submit these using `~st10/328submit` on nrs-projects, with a lab number of **89**.