

# CS 328 - Week 9 Lab Exercise - 2024-03-22

## Deadline

Due by 11:59 pm on Friday, March 22.

## Purpose

To practice using PHP, including using it to obtain data from a submitted form.

## How to submit

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

## Requirements

- You may work individually OR using pair programming for this lab exercise, your choice!
  - If you do use pair programming, make sure BOTH of your names appear in each file submitted!
- Whether working individually or using pair programming, make sure you have a copy of your lab exercise files, and that you submit them using `~st10/328submit` on nrs-projects, with a lab number of **89**.

## Your Tasks

Create a PHP document named `328lab09.php` that **both** creates a form **and** crafts a response to that form when it is submitted (as demonstrated in the posted Week 9 Lecture 2 example `php-form-handling.php`).

- Fill in its opening comment block with the URL I can use to run your version, your name(s), and the last modified date.
  - It will be a penalty if your document does not display using this URL when I/the grader try it.

## What form?

The form your PHP is to create when it is initially called should be the `form` element in the provided file `328lab09-form.txt`. (This file contains JUST a form element -- it is NOT an entire HTML document!)

- Copy this file into an appropriate location on one of your nrs-projects accounts:

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

Note that you have several options for how this form element gets into your PHP document for this lab exercise -- you may paste it appropriately into the appropriate part of the `if` statement within your `328lab09.php`, or you may appropriately include its contents using `require_once`.

## What form response?

When its form is submitted using `method="POST"`, your PHP should craft a response that *cautiously* displays, within a `ul` element, the values submitted from that form.

- 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 leading or trailing blanks from that string, and returns the resulting string.
  - This is also useful for more pragmatic sanitizing -- removing accidental leading or trailing blanks that users probably-accidentally entered into, for example, input fields and textareas is a good practice, as well.
- 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.

Here is one approach I have successfully used to strict-validate a multi-document PHP such as `328lab09.php`:

- Put your `.php`'s URL in a browser and view its source, copy and paste that source into a file with suffix `.xhtml`, (for example, `328lab09-1.xhtml`) and put the URL of that `.xhtml` document into the validator.
- Put your `.php`'s URL in a browser and submit its form, then view that *response*'s source, and copy and paste that *response*'s source into a file with suffix `.xhtml`, (for example, `328lab09-2.xhtml`) and put the URL of that `.xhtml` document into the validator.

### ***Optional additions***

- You may add an external CSS formatting your PHP document if you would like; if you do so, also submit that `.css` file.

Submit your resulting `328lab09.php` (and all additional files it uses, if any) with a lab number of **89**.