Lab 9

Start Assignment

Due Sunday by 11:59pm   **Points** 100   **Submitting** a file upload

---

CS-546 Lab 9

Sort Arrays

For this lab, you will be using HTML, CSS, and JavaScript on the user's browser to make a simple array sort!

You will create an express server with a single page at the location `/` that will provide the user with a web page to allow them to enter a variable number of arrays (separated by comma) with numbers and it will return an array with the sorted values from each array. **The entire checking operation will be done using client-side JavaScript**. **Major points will be deducted if you perform the processing server-side.**

YOU MUST use the directory and file structure in the code stub or points will be deducted. You can download the starter template here: **Lab9_stub.zip (https://sit.instructure.com/courses/61549/files/10503571?wrap=1)** ⬇ **(https://sit.instructure.com/courses/61549/files/10503571/download?download_frd=1)**

PLEASE NOTE: THE STUB DOES NOT INCLUDE THE PACKAGE.JSON FILE. YOU WILL NEED TO CREATE IT! DO NOT ADD ANY OTHER FILE OR FOLDER APART FROM PACKAGE.JSON FILE.

The Server

Your server this week should not be doing any of the processing! Your server only exists to allow someone to get to the HTML Page and download the associated assets to run the array sort page.

`/` The Whole Array Sort Application

Your page should have a few basic user interface elements:

- A header tag, with an h1 naming your site, with a title for your page
- A footer with your name, student ID, and any other info about yourself you wish to include
- A single unordered list with an id of `results`. All the arrays you have checked so far (until you refresh the page) will appear in this list as list items. Each item in the list will alternate the text colors starting with green, then red, then green, then red etc..
- An example that would be displayed in LI of the list:
  - [0,0,1,2,2,3,6,7,7,9]
  - [1,1,1,5,6,8, 10, 11]
  - [5,5,6,7,8,8,9,20,25,40]
  - [4,4,4,4,6,9, 11, 15]

Your page will have a form with the following:

- A label with a `for` attribute referencing your input
- A input with a `name` and `type` of `text`
- A button to submit the form

Using JavaScript in your browser only, you will listen for the form's `submit` event; when the form is submitted, you will:

- Get the value of the input text element.
  - You should be expecting a variable number of arrays typed into the input separated by commas: For example: `[3,0,1,2,4], [1,2,8,15], [6,3,10,25,29]`
  - All array elements should be whole numbers (negative and 0 are allowed), no decimals.
  - Each array should have at least one element that is a whole number (negative and 0 are allowed), no decimals.
  - You can ignore any extra commas for example, inputting: `[3,0,1,2,4], [1,2,8,15], [6,3,10,25,29],`
  - There should be at least one array inputted.
- You will then return a single array that has all the values from the arrays inputted sorted from lowest to highest number. For example: If our input was: `[3,0,1,2,4], [1,2,8,15], [6,3,10,25,29]` You would return: `[0,1,1,2,2,3,3,4,6,8,10,15,25,29]`
- Add a list item to the `#results` list of result of the sort you have just completed. You will alternate the class for each list item using the classes `is-green` and `is-red` (described below), starting with `is-green` first.

If the user does not have a value for the input when they submit, you should not continue processing and instead should inform them of the error on the page. If the user enters bad data, you should not continue processing and instead inform them of the error on the page.

The style

You will style your page using at least 10 CSS selectors for general CSS styling. You will place the CSS in its own file.

You *must* style the `is-green` class to make text have a `color` of `#00E676` and `is-red` class to make text have a color of `#FF3D00`.

References and Packages

Basic CSS info can easily be referenced in the **MDN CSS tutorial (https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Getting_started)** .

Requirements

1. All previous requirements still apply.
2. You **must remember** to update your package.json file to set `app.js` as your starting script!
3. **Your HTML must be valid (https://validator.w3.org/#validate_by_input)** or you will lose points on the assignment.
4. Your HTML must make semantic sense; usage of tags for the purpose of simply changing the style of elements (such as `i`, `b`, `font`, `center`, etc) will result in points being deducted; think in terms of content first, then style with your CSS.
5. **You can be as creative as you'd like to fulfill front-end requirements**; if an implementation is not explicitly stated, however you go about it is fine (provided the HTML is valid and semantical). Design is not a factor in this course.
6. **Your client side JavaScript must be in its own file and referenced from the HTML accordingly**.
7. All inputs must be properly labeled!