

CS 328 - Small-Homework-Bonus JavaScript Problem

Deadline

11:59 pm on Friday, May 8, 2026

Purpose

To provide an optional opportunity to practice a bit more with basic unobtrusive-style client-side JavaScript.

Requirements

- To receive **any** bonus credit, all of your JavaScript must be in **either** the external JavaScript file described **or** at the end of the HTML/PHP document's **head** element -- there must be neither JavaScript nor **script** elements in its **body** element.
 - This is because this problem's purpose is to practice with unobtrusive-style client-side JavaScript in particular.
- To receive full credit, be sure to modify the JavaScript DOM **after** the document is loaded -- that is, set the DOM **window** object's **onload** attribute to a function to make any changes desired to the document.
- To receive full credit, write your JavaScript so strict mode is enforced -- that is, put:
"use strict";
at the very top of your external JavaScript file. (*If you also have a non-empty **script** element in your document's **head** element, start it with this, also.*)
- This assignment will appear in Canvas as a 0-point assignment, so that its up-to-20-points will then be treated as 20 additional homework points.
 - (That is, if you receive an 80 on one of the homeworks, and you receive 20 points on this problem, the result would be like you received a 100 on that homework whose score is 80.)
- Please ask me if you have any questions about the above.

How to submit

Each time you wish to submit your work-so-far, submit your files using **~st10/328submit** on nrs-projects, with a homework number of **12**.

Bonus Problem - a little client-side JavaScript - 20 points

Purpose: to get additional practice with unobtrusive-style client-side JavaScript.

FUN FACTS (that might be of use in this problem):

- JavaScript function **parseFloat** expects a string that is reasonable to parse into a floating-point number and returns such a floating-point number if it can -- but may return the special value **NaN**

(not a number) if given an argument such as "Jimmy"

- As noted in zyBooks Section 7.2, there is also a JavaScript function **isNaN** -- it expects one argument and returns **true** if the argument is not a number, and returns **false** otherwise.
- There is a JavaScript function **parseInt** as well, that expects a string that is reasonable to parse into an integer number and returns such an integer number if it can -- but may return the special value **NaN** (not a number) if given an argument such as "Jimmy".
 - What happens if you give **parseInt** a non-integer number? Looks like it returns an integer version -- and looks like it just truncates any fractional part!

Bonus part a

Task: Determine at least one type of numeric computation you would like to perform. (It can be as simple as addition, or as involved as you would like.)

Then, create a **PHP or HTML** document (your choice) in a file whose name includes **number-fun** that meets the following requirements:

- Include your name and last modified date in its opening comment, **AND** the URL this can be run from
 - (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, include an appropriate **title** element.
- Include an appropriate **h1** element
- Include **at least two** number fields (**input** elements with **type="number"**) (so the user can enter the needed numbers for the computation you chose).
 - Fun fact: if you give a number field a **step** attribute whose value is not an integer, then that number field can accept non-integer numbers!
 - Reference: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input/number>
- Include at least one **button** element (so the user can indicate that they would like for a computation to now be done)
- (you may also include additional elements as you would like)
- Within its **footer** element near the end of the **body** element, add a **p** element whose content includes your name.
- How will you show the result? You get to determine this. (You could display it in a textfield, for example, or within a paragraph, or within a textarea, etc.)

Bonus part b

Task: Using **unobtrusive-style client-side JavaScript**, write an **external JavaScript** in a file whose name *also* includes **number-fun** to now:

- perform the numeric computation(s) you decided upon, using ***number-fun***. {**php** or **html**}'s number fields' contents when its **button** element is clicked,

- making sure to somehow show the computation's results to the user.

Then, add the appropriate **script** element to the end of ***number-fun* . {php or html}**'s **head** element so that it uses this external JavaScript ***number-fun* .js**.

Remember to also do what is needed so that the DOM is not modified until the document is loaded.

- This may be within your external JavaScript **or** within a **script** element in your document's **head** element, your choice!

Submit your resulting files:

- ***number-fun* . {php or html}**
- ***number-fun* .js**