Assignment 6

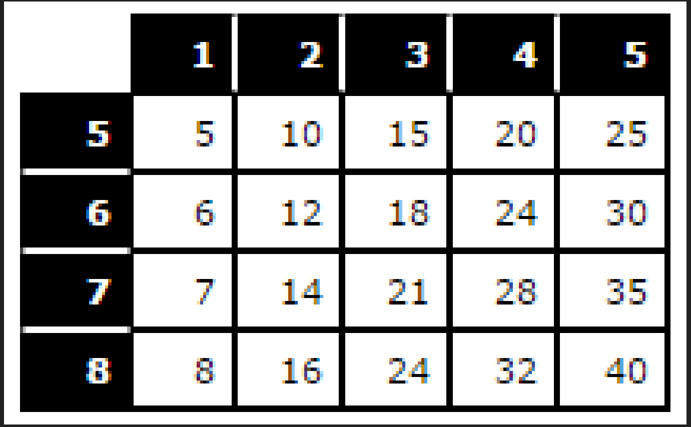
Assignment 6 - A Dynamic JavaScript Table

For this assignment, you will create a table dynamically based on parameters entered in an HTML form.

First, you must create a form that accepts the starting and ending numbers for both the horizontal (multiplier) and vertical (multiplicand) axes of a multiplication table.

Second, you will use the numbers entered into the form to create a multiplication table completely dynamically.

Ex: Inputs Multiplier: 1, 5 ; Multiplicand: 5, 8



	1	2	3	4	5
5	5	10	15	20	25
6	6	12	18	24	30
7	7	14	21	28	35
8	8	16	24	32	40

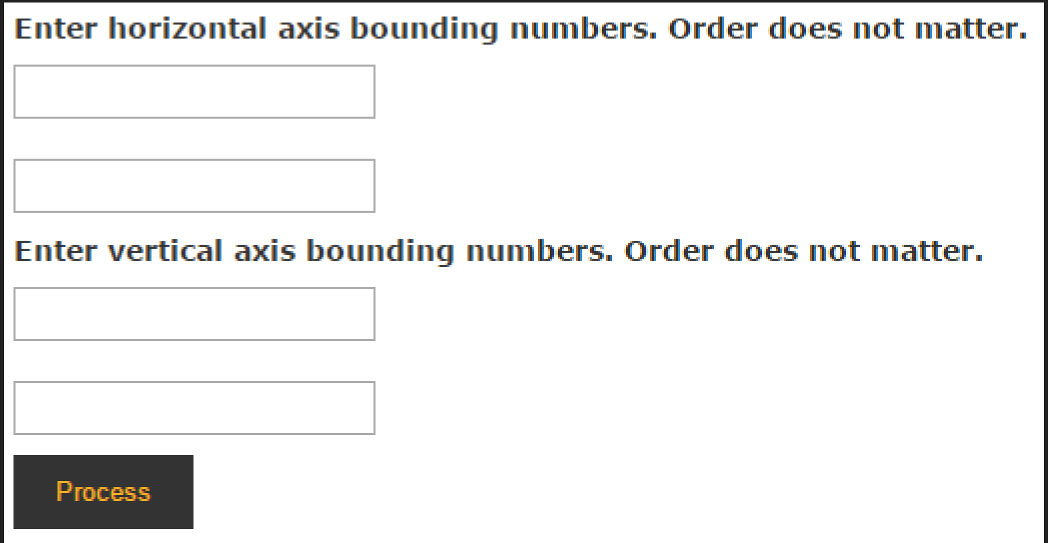
Assignment 6 - Tips

- When clicked, the submit button should kick off a JavaScript function, not submit the form.
- No table is initially present before submit.
- A table appears after valid submit.
- A table is replaced if new valid values are submitted.
- Encapsulate your JavaScript in functions with parameters so that the code you write for this assignment can be reused.

Assignment 6 - Validation

- The input elements need to be validated.
- When validating inputs, display a message/ indicate/ highlight invalid input to a user.
- Make sure a user only enters numbers, or be prepared to handle otherwise.
- Make sure the starting number is less than or equal to the end number, or be prepared to handle the case.
- Validate your page and test many parameters to make sure you hit all use cases.

Before Validation

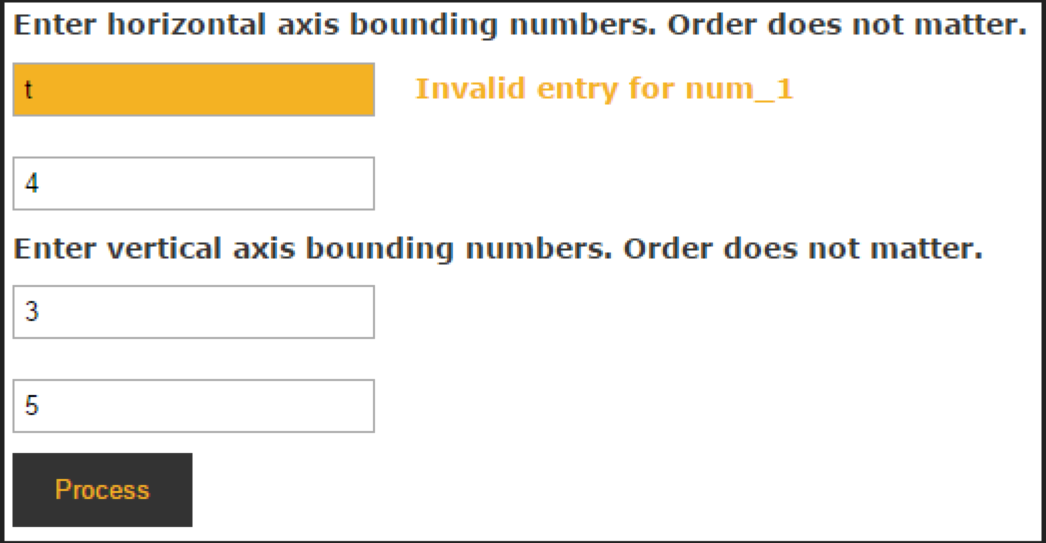


Enter horizontal axis bounding numbers. Order does not matter.

Enter vertical axis bounding numbers. Order does not matter.

What could be improved?

Validated With Errors



Enter horizontal axis bounding numbers. Order does not matter.

t

Invalid entry for num_1

4

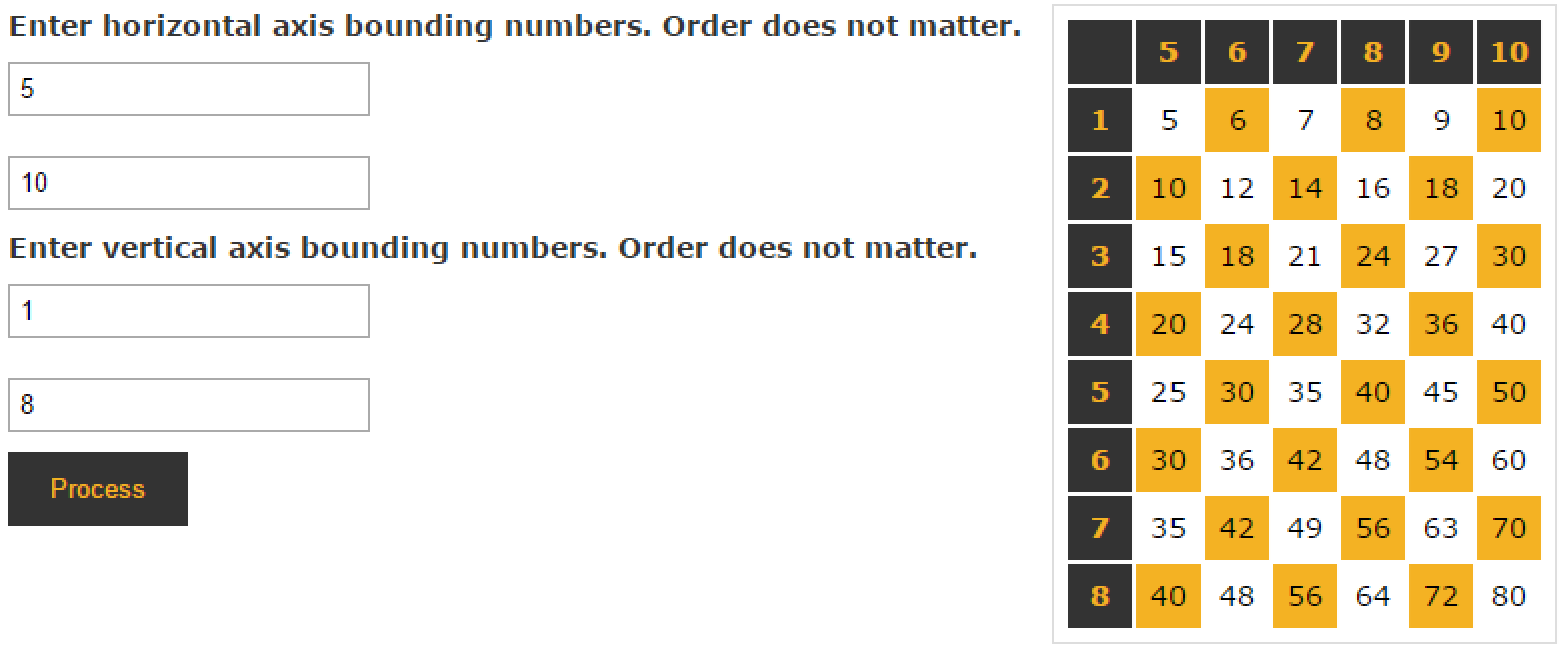
Enter vertical axis bounding numbers. Order does not matter.

3

5

What could be improved?

Assignment 6 - Sample Output



Enter horizontal axis bounding numbers. Order does not matter.

5

10

Enter vertical axis bounding numbers. Order does not matter.

1

8

	5	6	7	8	9	10
1	5	6	7	8	9	10
2	10	12	14	16	18	20
3	15	18	21	24	27	30
4	20	24	28	32	36	40
5	25	30	35	40	45	50
6	30	36	42	48	54	60
7	35	42	49	56	63	70
8	40	48	56	64	72	80

Assignment 6 - CSS

- Use CSS to style the table in an appropriate way.
- Be creative, but be mindful of usability.
- The `:nth-child()` and other CSS3 pseudo-classes will be very helpful.
- css-tricks.com/useful-nth-child-recipies
- reference.sitepoint.com/css/css3psuedoclasses

Assignment 6 - Pseudo Selectors

Single Colon

:active
:after
:before
:checked
:default
:disabled
:empty
:enabled
:first-child
:first-letter
:first-line
:first-of-type
:focus
:hover
:in-range
:indeterminate
:invalid
:lang
:last-child
:last-of-type
:not
:nth-child
:nth-last-child
:nth-last-of-type
:nth-of-type
:only-child
:only-of-type
:optional
:out-of-range
:read-only
:read-write
:required
:root
:target
:valid
:visited

Double Colon

::after
::before
::first-letter
::first-line
::link
::selection

Experimental

:dir ::backdrop
:fullscreen :scope
::placeholder

smashingmagazine.com/2016/05/an-ultimate-guide-to-css-pseudo-classes-and-pseudo-elements/

Assignment 6 - Useful Pseudo Selectors

:first-child

the first child of its parent element.

:last-child

the last child of its parent element.

:nth-child(param) * ↓

:nth-last-child(param) * ↑

targets one or more elements depending on order.

:focus

Styles elements that have come into focus.

:required, :optional, :invalid, :valid

Styles elements marked required through the corresponding attribute.

```
<input type=”text” required/>
  <input type=”text” valid/>
 <input type=”text” invalid/>
```

```
<ul>
  <li>1</li>
  <li>2</li>
  <li>3</li>
</ul>
```

* Param can be the values odd, even, or a function() in the form of { A n [OP OFFSET] : OP is +- } so 2n+1 is valid

https://www.smashingmagazine.com/2016/05/an-ultimate-guide-to-css-pseudo-classes-and-pseudo-elements/#nth-child

Github Submission

Node/ Automation/ Testing

JavaScript Server
nodejs.org/en/

Task Runner
npmjs.com/

Package Manager
npmjs.com/package/gulp

Transpiler
babeljs.io/

Test Framework
mochajs.org/

Setting Up An Assignment

Create your directories where # is the assignment number

```
    $ mkdir a#
$ cd a#
    $ mkdir dep app
$ mkdir app/css app/js app/img
```

Commit

```
$ git add -A
$ git commit -m "Setting up a#"
$ git push -u origin master
```

Create your files

Commit Again

Application 6 - Submission

Once you have everything working on your local system, upload your assignment to GitHub.

Submit your assignment by submitting the GitHub URL where your application resides. By following the instructions you will be able to submit with a URL like this one

`username.github.io/username/a#/index.html`

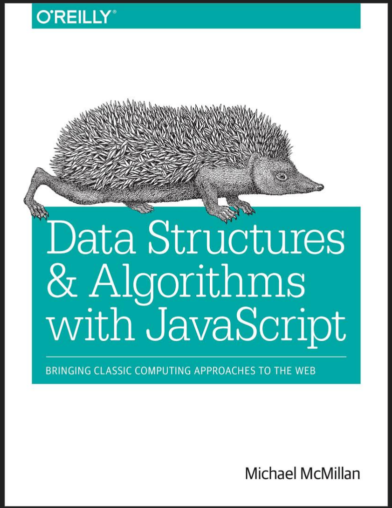
Where

`username` is your Github user name

`#` is the assignment number

JavaScript Resources

Recommended Reading



Discusses what parts of the JavaScript language should be used and how to create error free and effective code.

How to implement algorithms and data structures specifically in the JavaScript language.