EECS 1012: LAB #8 – more on JavaScript (March 4– 10, 2019)

1. Read the lab instructions in this document and take the pre-lab quiz for Lab #8.

NOTE: Try to complete the tasks given in this write up before coming to your lab session, the lab must be completed and verified by the TA before the end of the lab session.

2. GOALS & OUTCOMES FOR THIS LAB

- To practice more concepts in programming, including variables, arrays, functions, and program control statements
- To use more JS objects, such as document, Math, and Date

3. LAB 8 – SIX (6) TASK

1) TASK 1: Simple "YES" or "NO" button output with an if-statement.
2) TASK 2: Passing variables to functions.
3) TASK 3: Passing variables and for-loop.
4) TASK 4: Random + string concatenation + if-statement.
5) TASK 5: Date object + array + string concatenation.
6) TASK 6: Global variable and if-statement

INCLUDED WITH ZIP FILE: The zip file contains an example (in the example folder) that uses random, arrays, functions, and an if-statement.

This zip file also contains two videos
(1) lab8_no_audio.mp4 - shows all the tasks as they should appear when working.
(2) finding_JS_errors_no_audio.mp4 – shows how to use the Browser's console to help debug JS errors.

4. SUBMISSIONS

1) [Manual verification by a TA]
As with previous labs, when you have completed all tasks, ask the TA to come and verify your code and output. You must sign the verification sheet to get marks. If you attended the entire lab and still cannot complete the lab before the end of the session, you will receive 50% credit for the lab.

2) Moodle submission
You will see an assignment submission link on Moodle.

1) Create a **folder** named "**Lab8**" and copy **all** of your HTML and JS files.

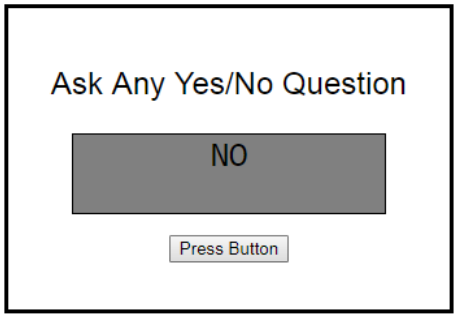
Task 1: Edit task1.js (you do not need to edit the HTML file).

For this task, we have already declared the JavaScript function `myFunction()` for you.

Your function should do the following.

Each time the button is clicked, your `myFunction()` code should generate a random number. If the random number is greater than 0.5, then have the `innerHTML` of the paragraph variable set to "YES", otherwise set it to "NO".

See below for example outputs.



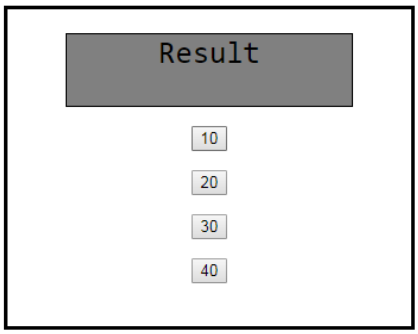
Task 2. Edit task2.html and task2.js

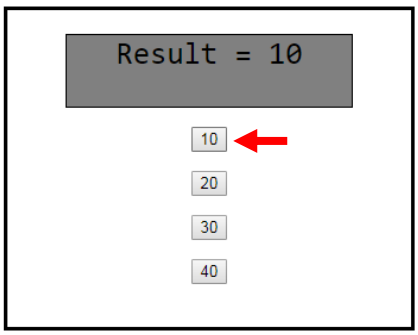
(1) Link your task2.js to your HTML code.

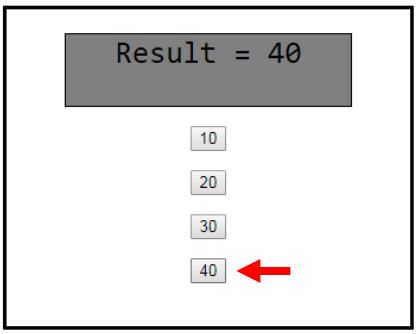
(2) Have the text in the paragraph "mydata" start with **Result** (see below).

(2) Add four buttons to your Task2.html as shown below.

(3) Write a function in javascript that has one parameter. When a button is pressed, it should pass the value shown in the button (e.g., 10, 20, 30, or 40). Your function should change the innerHTML of the paragraph to the passed value as "Result = **VALUE**". See example below.







Task 3. Edit task3.html and task3.js

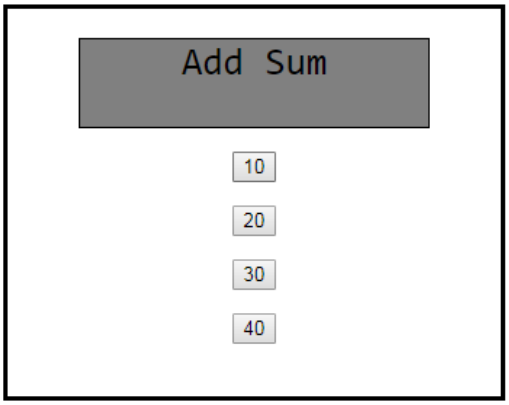
(1) Link your task3.js to your HTML code.

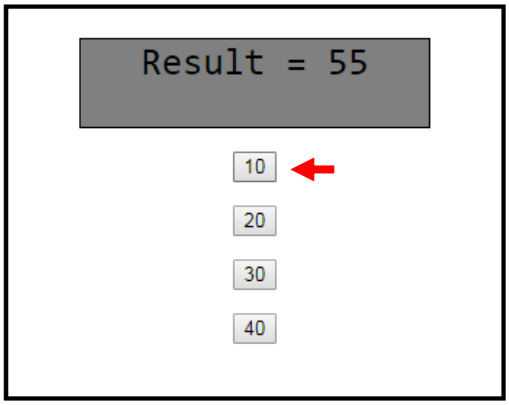
(2) Have the text in the paragraph "mydata" start with **Add Sum** (see below).

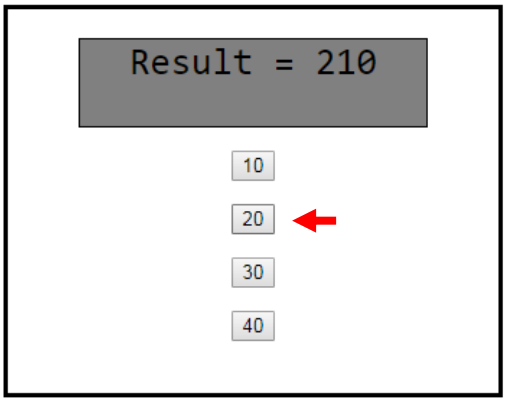
(2) Add four buttons to your Task3.html as shown below (you can also copy over task2.html).

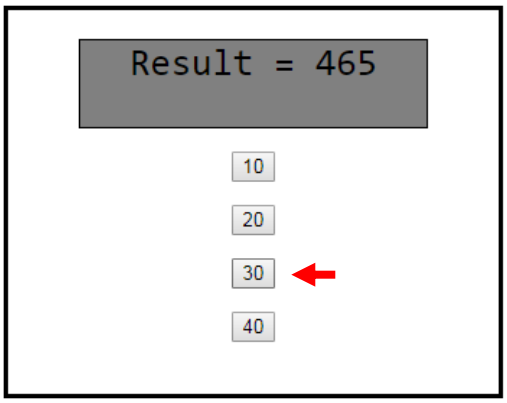
(3) Write a function in javascript that has one parameter. When each button is pressed, it should pass the *value* shown in the button (e.g., 10, 20, 30, or 40). Use a for-loop to compute the sum of 0 to the *value*.

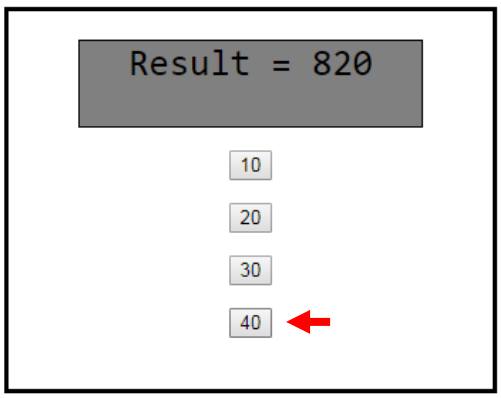
For example, if the value passed is 10, then compute 0+1+2+3+4+5+6+7+8+9+10=55. See below.











Task 4. Modify task4.html and task4.js

(1) Link your javascrpit file to your HTML file.

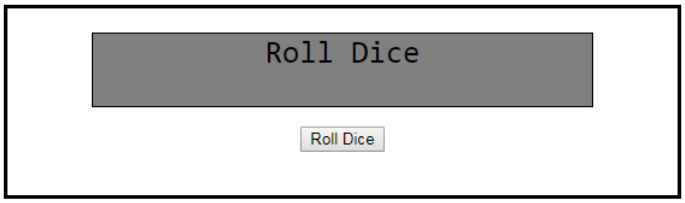
(2) Have the text in the paragraph “mydata” start with **Roll Dice**. Add a button “Roll Dice”. Have this button respond the click event.

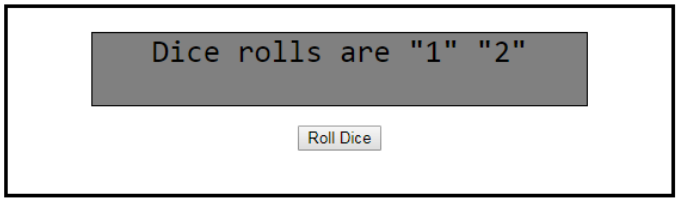
(3) Have the onlick for your button link to your javascript function. The function does not have parameters.

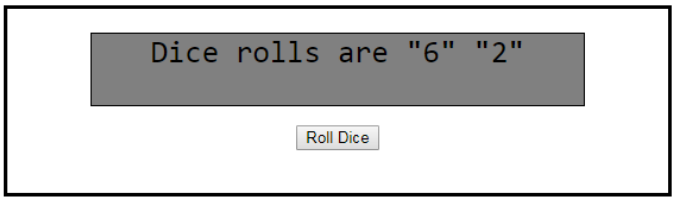
(4) Each time you click, have your function compute two random numbers from 1-6. These represent dice. Change the innerHTML to say Dice rolls are “value1” and “value2”, where value1 and 2 are the results of your random number.

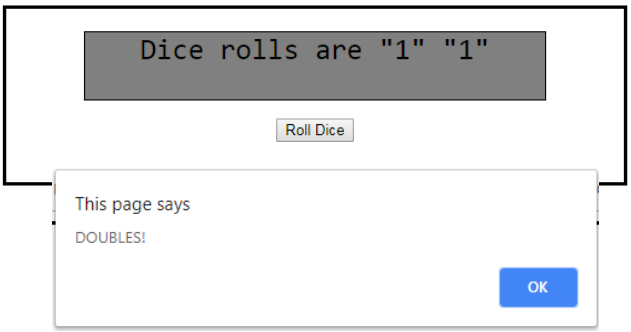
(5) If the two numbers are the equal, the create an alert that says “DOUBLES!”.

See examples below.









Task 5. Modify task5.html and task5.js

(1) Link your javascrpit file to your HTML file.

(2) Have the text in the paragraph “mydata” start with **Today’s Date**. Add a button “Click”. Have this button respond the click event.

(3) Have the onlick for your button link to your javascript function. It does not have parameters.

(4) When you click, your function should create a Date object.

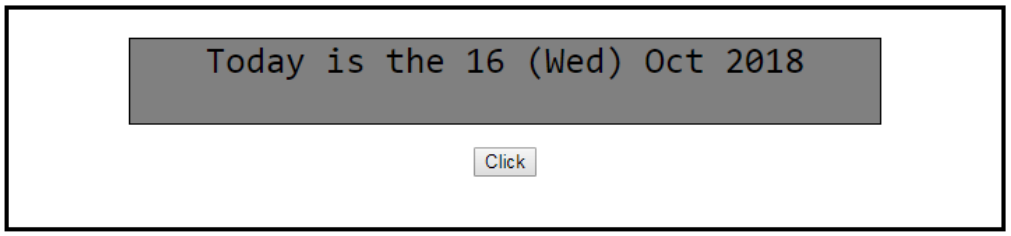
Get the following data from the Date obect.

(i) day of the month
(ii) day of the week
(iii) month
(iv) year

Using this data, change the innerHTML to output the string below.

Hint: You should use an array to store the three letter days of the week (“Sun”, “Mon”, “Tues”, …).

Hint: You should use an array to store the three letter abbreviation of the month (“Jan”, “Feb”, …).



Task 6. Modify task6.html and task6.js

(1) Link your javascrpit file to your HTML file.

(2) Have the text in the paragraph "mydata" start with **Count Down**. Add button "Click". Have this button respond the click event.

(3) Declare a global variable. This is a variable that is created outside your function. Inside your function, you do not need to declare it again. If you modify the variable, the modification will be remembered next time you access the function. See example code here.

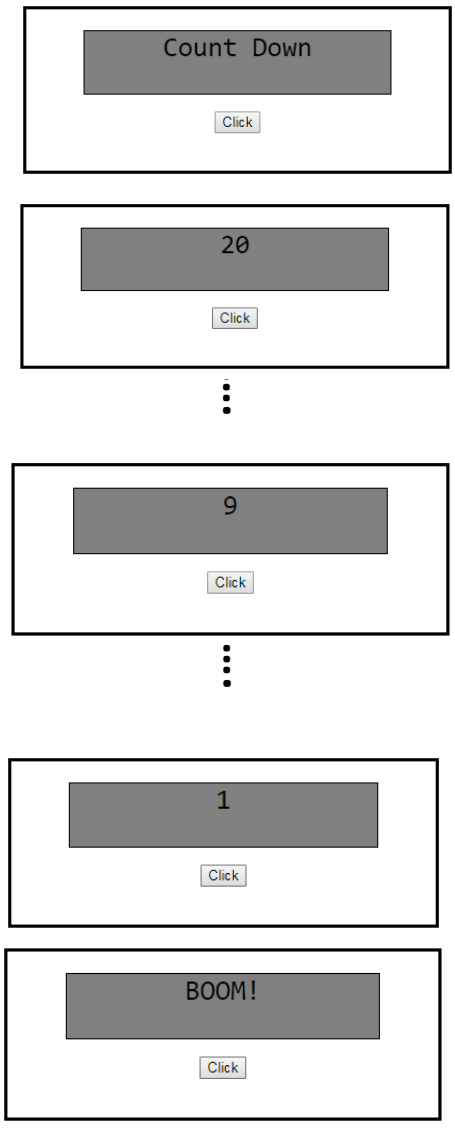
```
var i=20;

function myFunction()
{
 i = i--; // the value of i will be remembered next function call
}
```

(4) Each time your button is clicked, you should print out the global variable and reduce it by 1 (one).

Your innerHTML of the paragraph with id "mydata" should show the current value of the the global variable.

(4) When the variable gets to 0 or less, have the your innerHTML change to BOOM!

See below.



Bonus work for up coming labs:

1) Complete your Lab3 work with 40 problem. Recall. Lab3 was the lab in which you defined 10 buttons each for a computational thinking problem, and you showed the flowchart and JS solution. Now, you have solved many problems, you can complete your kit. Suggestion: instead of choosing ProblemX for the button caption, where X is a number; choose a more readable caption, something like pXX<description> where XX is a two-digit number and <description> is a one word description about that problem. Examples: p12factorial; p18prime; p32array; p35array2D, etc. Show your complete lab to your TA next week.

2) Use html, CSS, JS, (and your creativity)) to design the client side of the code-breaker game—of your taste. Soon, you would need to complete that work in the week after. It's better you start early.