EECS 1012. LAB 7: More on JavaScript (Nov 2–4, 2020)

A. REMINDERS

1) Each lab including the pre-lab mini quiz is about 2.0 % of your overall grade.
2) You must attend your own lab session (the one you are enrolled in). If you need to change your lab enrollment, you should go to the department. Instructors or TAs cannot change your enrollment. TAs are available via Zoom to help you. The attendance is optional, but is highly recommended. You can also have your work verified and graded during the lab sessions. Feel free to signal a TA for help if you stuck on any of the steps below. Yet, note that TAs would need to help other students too. In case you run out of time, the submission you make over eClass will be marked by the TAs after the lab ends (possibly not by the same TAs who assisted you during the lab session).
3) You can submit your lab work anytime before the deadline. We do not accept late submissions.
4) You must complete the pre-lab quiz posted on eClass no later than the first 15 minutes of your lab time.

B. LEARNING KIT

Complete your My Learning Kit Project (that you started from Lab03) with 30 problem definitions, flowcharts, and JavaScript Solution. You may not want yet to include anything for the "*another solution*" panel.

At the beginning of Lab 7 this week, as soon as you are done with the pre-lab quiz, you may be asked to show your Learning Kit project to your TA for credit. Obviously, TAs may reach some students immediately after the pre-lab quiz, but it will probably take longer to see all. Hence, you should make sure your Learning Kit project is ready before you go to the lab. As you might need this project for Lab 8 and/or Lab 9, not completing it by Lab 7 can impact your grade for those labs too. This part of the lab should be uploaded to eClass not later than the first 20 minutes of your lab session.

C. GOALS/OUTCOMES FOR LAB

To practice more concepts in programming, including variables, arrays, functions, and program control statements

To use JS objects, such as document, Math, and Date

D. TASKS

TASK 1: Simple button generating random output using an if-statement.
TASK 2: Passing string variables to functions.
TASK 3: Passing numeric variables to functions.
TASK 4: Random + string concatenation + global variables + for-loop.
TASK 5: Date object + array + string concatenation.
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 a video *finding_JS_errors_no_audio.mp4* – which shows how to use the Browser's console to help debug JS errors. Note that when using Firefox, the equivalent functionality is available via Web Console (Ctrl-Shift-K in Windows). Try to find an error in the included example.js file (in the example folder).

E. SUBMISSIONS

1) Manual verification by a TA (optional)

   You may have one of the TAs verify your lab before submission. The TA will look at your various files in their progression. The TA may also ask you to make minor modifications to the lab to demonstrate your knowledge of the materials. The TA can then record your grade in the system.

2) eClass submission (lab)

   Create a **folder** named "**Lab7**" and copy all of your HTML and JS files there. This folder should be compressed into .zip (or tar.gz), and the compressed file submitted.

3) eClass submission (kit)
   Create a **folder** named "**LearningKit**" and copy all of your kit files there. This folder should be compressed into .zip (or tar.gz), and the compressed file submitted.

In case you work with a partner, **only one** person should submit the files. In such a case, an extra file **group.txt** should be included in the submission, containing the full names and the student numbers of the team members. The student not submitting the other files, should only submit the **group.txt** file. Both students will then receive the same grade. Note that the pre-lab quizzes are still expected to be done by each student separately.

F. INSTRUCTIONS

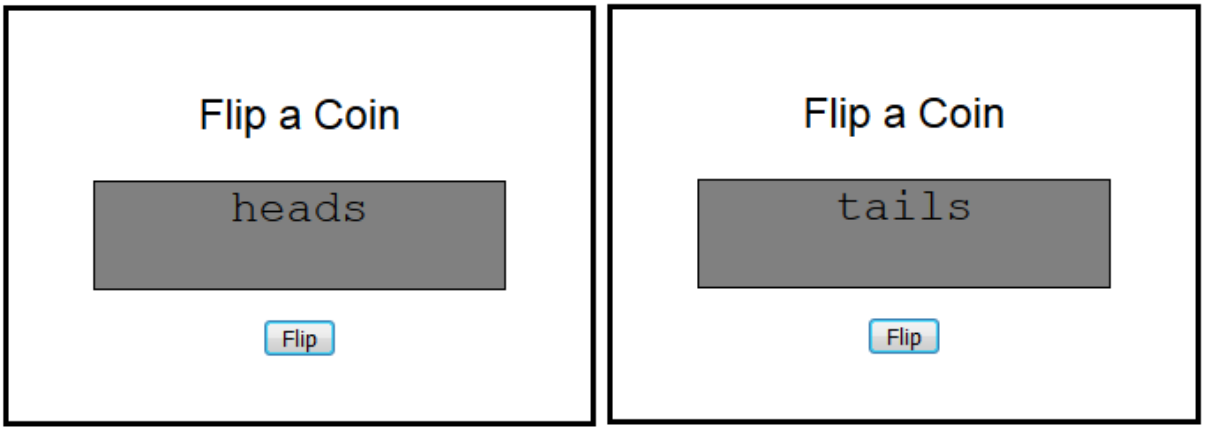
Task 1: Edit task1.js (you do not need to edit the HTML file).

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 or equal to 0.5, then have the `innerHTML` of the paragraph variable set to "tails", otherwise set it to "heads".

See below for example outputs.



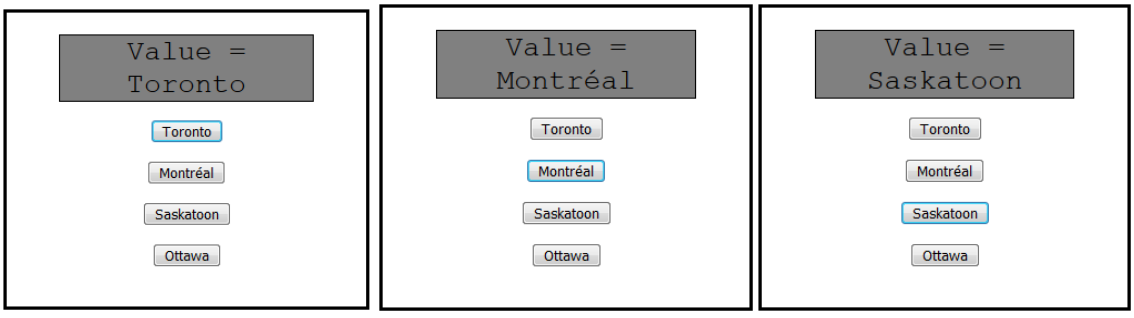
Task 2. Edit task2.html and task2.js

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

(2) Have the text in the paragraph "mydata" start with **Value** (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., "Toronto", "Montréal", …). Your function should change the innerHTML of the paragraph to the passed value as "Result = **VALUE**". See example below.



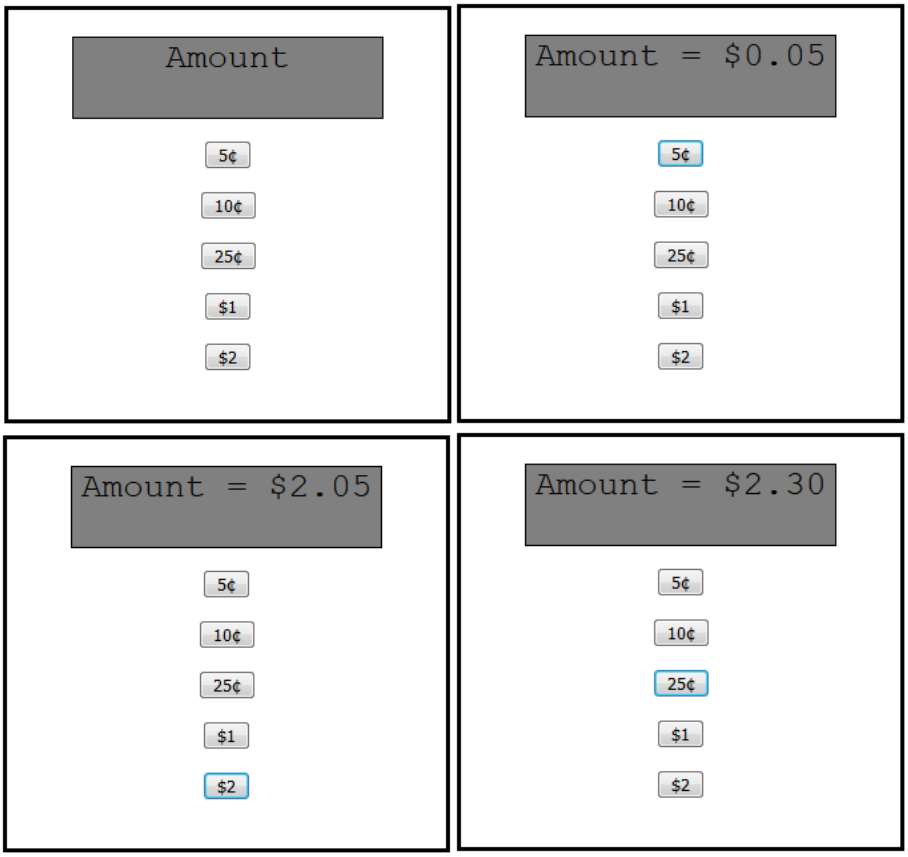
Task 3. Edit task3.html and task3.js

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

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

(2) Add four buttons to your Task3.html as shown below.

(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., 0.05, 0.10, 0.25, …). The function should add the value to the existing one – as illustrated below. For that, you might want to use a *global* variable[1].



[1] Global variables are variables that are accessible by *any* function inside a JS file. They are placed outside any functions you might have there. More on global variables in Task 6.

Task 4. Modify task4.html and task4.js

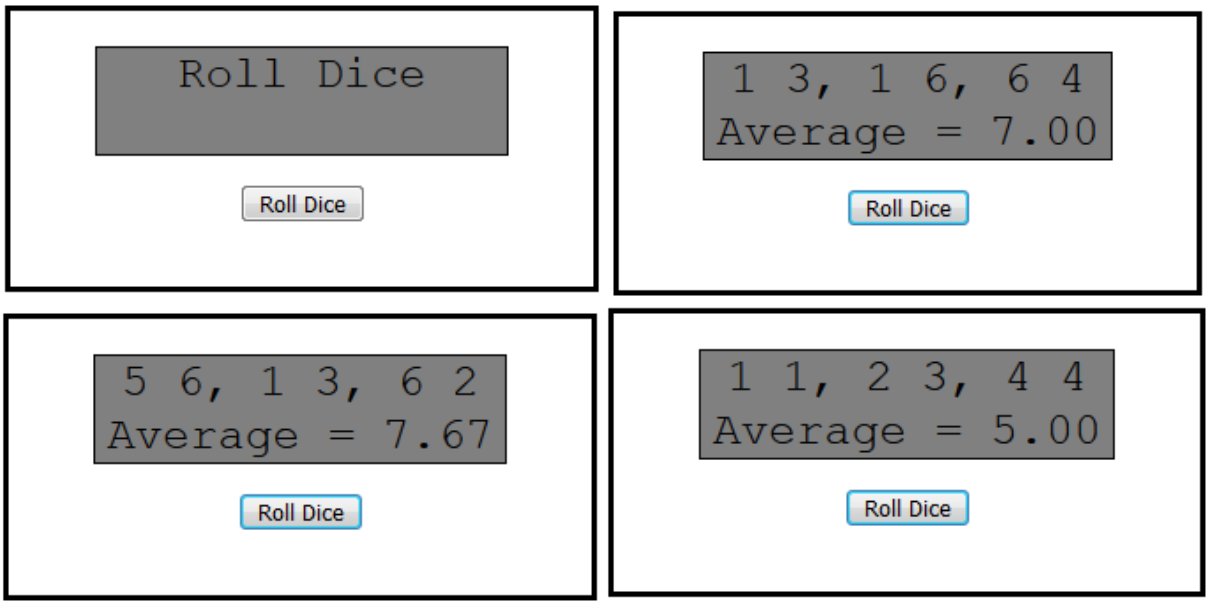
(1) Link your JavaScript file to your HTML file.

(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 onclick for your button link to your JavaScript function. The function does not have parameters.

(4) Each time you click, have your function compute **three sets** of **two** random numbers from 1 to 6. These represent three sets of dice rolls. When rolling the dice display the resulting values, as shown below. Use the *innerHTML* to present the die values. Note, there is no comma after the last set of numbers

(5) At the end, display the average value of the three tosses, as shown below



Task 5. Modify task5.html and task5.js

(1) Link your JavaScript 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 onclick for your button link to your JavaScript function. It does not have parameters.

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

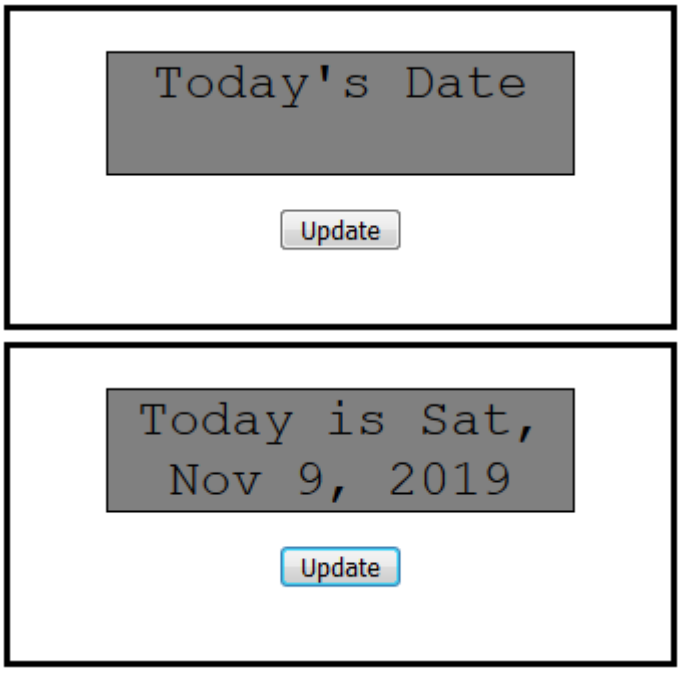
Get the following data from the Date object[2].

(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", "Tue", …).

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



[2] https://www.w3schools.com/js/js_date_methods.asp

Task 6. Modify task6.html and task6.js

(1) Link your JavaScript 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.

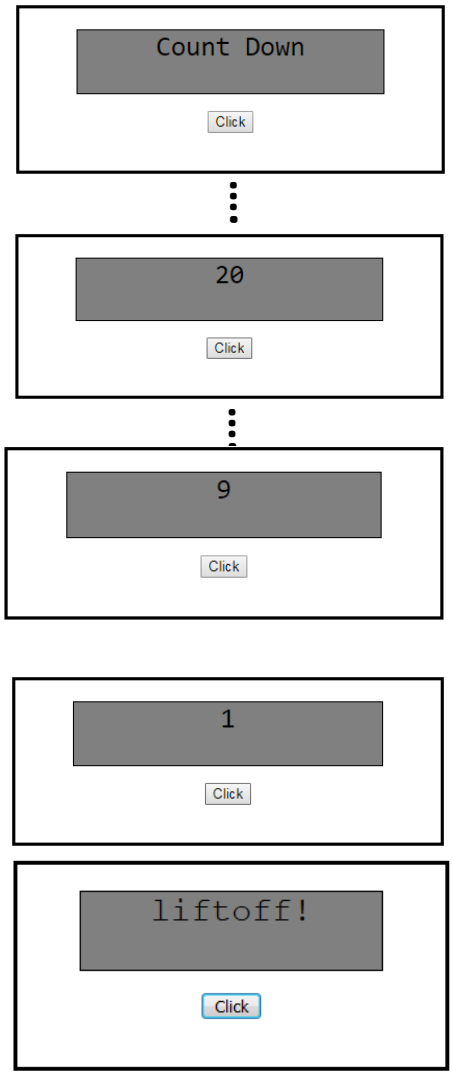
```
var i=20;

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 global variable.

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

See on side.



Please feel free to discuss any of these questions in the course forum or contact the TAs and/or your instructor for help.