# Assignment08 – Hot dogs and buns (conditional statements, functions)

THE PROBLEM: Assume hot dogs come in packages of 10, and hot dog buns come in packages of 8. (That’s the only way you can buy them!) Write a program that calculates the number of packages of hot dogs and the number of packages of hot dog buns needed for a cookout. You also need to calculate the hot dogs and buns that will be left over.

The program should ask the user for the number of people attending the cookout. Assume each person will eat one hotdog.

The program should calculate:

* The minimum number of packages of hot dogs required
* The minimum number of packages of hot dog buns required
* The number of hot dogs that will be left over
* The number of hot dog buns that will be left over

All the information should be reported in a detailed message that looks like this:

We will need X1 packages of hot dogs and will have X2 leftover.

We will need Y1 packages of buns and will have Y2 leftover.

So... if you had 11 guests the message would look like this:

We will need 2 packages of hot dogs and will have 9 left over.

We will need 2 packages of buns and will have 5 left over.

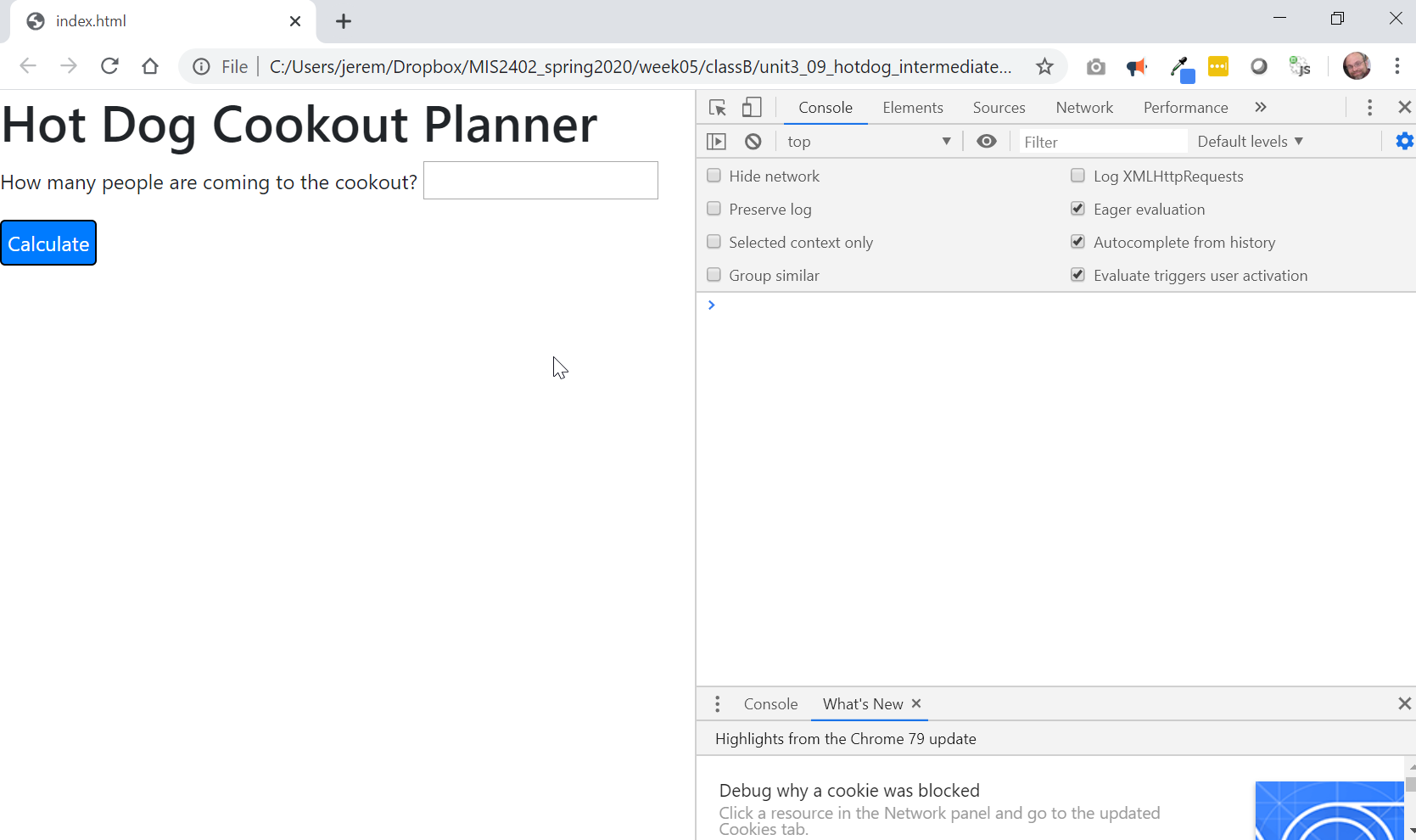
Finally, if the user enters anything that is not a natural number, the program should report a message that looks like this:

Bad data. Try again.

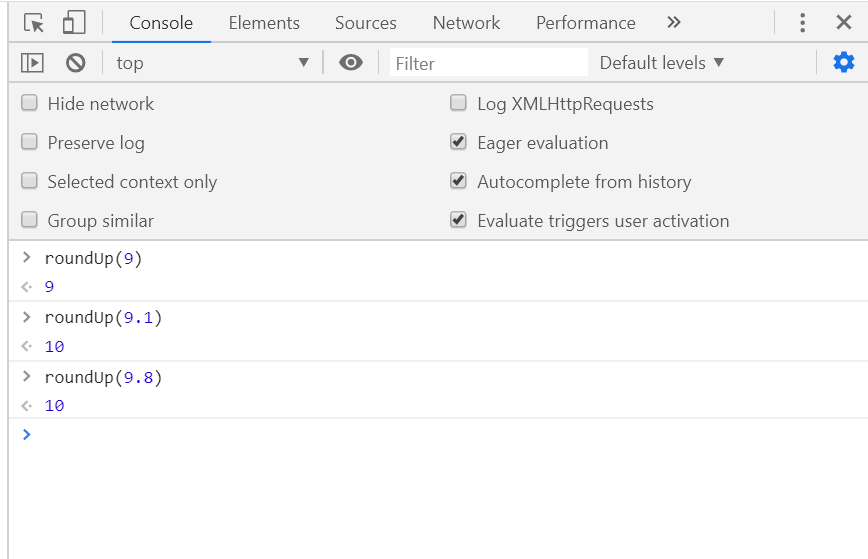
## Getting started (Together as a class)

1. Retrieve assignment08\_hotdog.zip provided by your instructor.
2. Extract the code into your mis2402workspace and open the index.html file in Visual Studio Code.
3. Review the code in the file. Notice that the problem has been broken down into a collection of smaller problems. Each one of the “small” problems is represented by a function!
4. There is one function has been written for you. It’s called roundUp(). It takes a number with decimal precision and rounds up to the next highest integer.

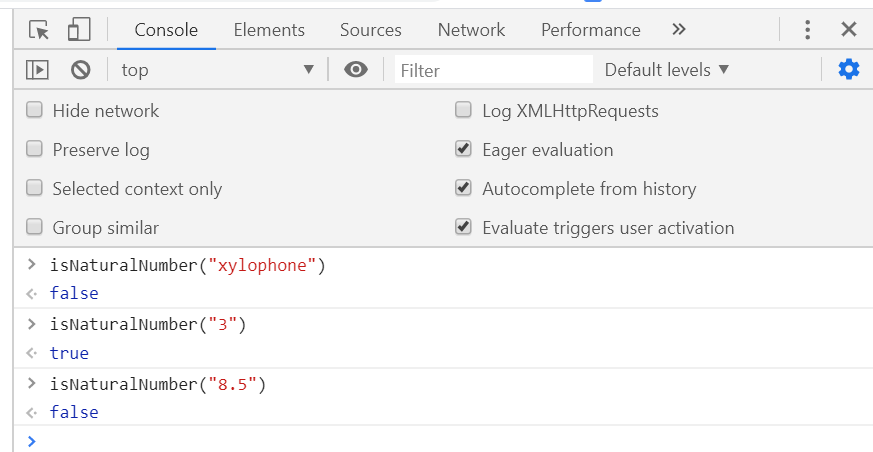
Take a moment and test that out by opening index.html in Google Chrome. Then press Ctrl-Shift-I to open the Chrome Developer tools.



1. FANCY STUFF! The Chrome Web Developer tools have a lot of features that can help with various programming tasks. One such feature is the Web Developer Console. It allows you to type single lines of JavaScript and see their immediate output. This includes functions. Use the Web Developer Console to test the roundup() function with the following examples:
   1. roundUp(9)
   2. roundUp(9.1)
   3. roundUp(9.8)



1. Now recreate the isNaturalNumber() function you created earlier in the semester. When you are done, test it using the console.



**HOT TIP – You can use the isNaturalNumber() function later in the assignment. It will save you from having to write all those conditional statements over again a second time.**

1. Now look at the rest of the file, starting at the bottom.

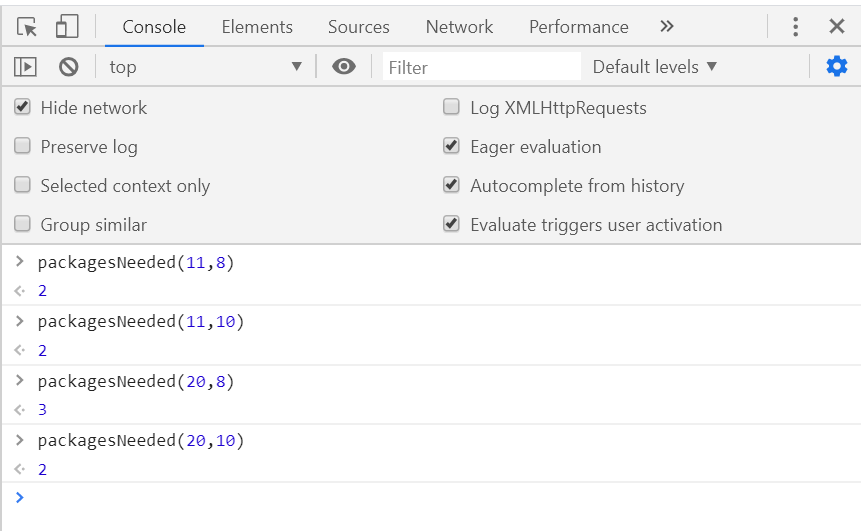
The function getMessage() takes one input, the number of people attending the cookout. When it’s done, it should return a string containing the output message.

The function getMessage() depends on two other functions, getMessageBuns() and getMessageHotDogs(). Wow... if we had two functions to compute the number of leftovers and the packages needed, we would be pretty much set. So, let’s work on those.

1. Write the function packagesNeeded() - This function should take a number of people, divide it by a number of items per package and round up. Notice that the parameter itemsPerPack lets you use this one function for both hot dogs (which come 10 in a pack) and hot dog rolls (that come in packs of 8)

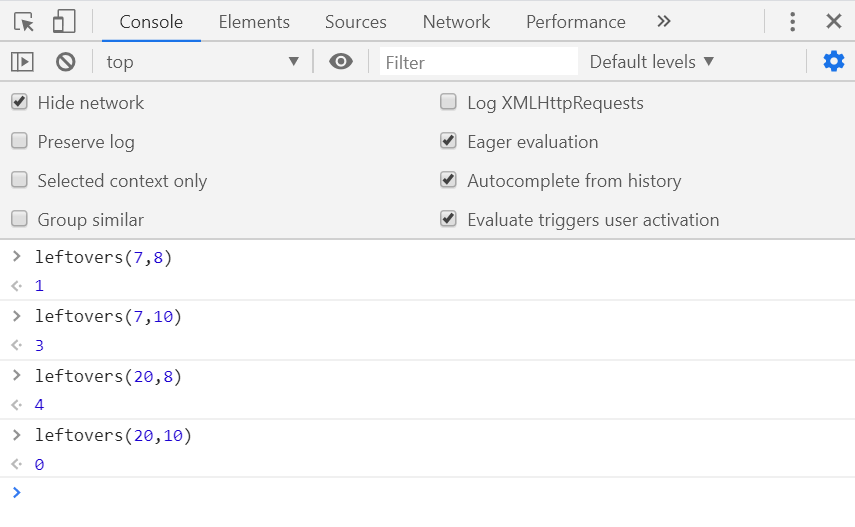
Discuss – what is the formula we need here. How do we write it?

Test this function when you are done.



1. Almost there! Now we need a function that will determine the number of **individual** items left over. For example, if I have an 8 pack of rolls, and 7 guests, I will have one **individual** roll left over.

Again, discuss -- what is the formula we need here. How do we write it?



## On your own

1. Complete the functions getMessage(), getMessageHotDogs() and getMessageBuns().
2. Be sure to test your work. Does your solution have the error trapping feature described in the problem statement? Figuring out where that goes and how to write it is an important part of the assignment.
3. Upload your work. Be sure that you can find your work on the class server by typing in its URL in the browser. Test your work \*again\* on the class server.  
     
   For example:   
   http://misdemo.temple.edu/tux99999/assignment08\_hotdog

How will this assignment be graded?

This assignment will be evaluated by an automated process.

* If your work is not found at the expected location on misdemo, you will get a score of **zero**.
* If your work generates **all output** correctly, you will get a score of 100%.
* If your work generates **almost all output** correctly (**only one** bad output), you will get a score of 80%
* If your work generates **some output** correctly (some right output, some wrong output), you will get a score of 60%
* If your work generates **only one output** correctly, you will get a score of 40%
* If your work does not generate any correct output, you will get a score of **zero**.