day101 Scratch Reaction Time
Due Thursday 5/25/23

Your task today is to make a program that measures your reaction time. It will be a lot like all the other reaction time programs we've done (micro:bit, Arduino, etc.)

When the user presses the green flag, a sprite should explain the process to the user, then the reaction timing should begin. You could have something appear and need to be clicked on, or you can have someone press a certain key, or the mouse when something appears. The main thing is that the trigger needs to have some element of randomness. After each round, report the current reaction time, along with the average reaction time, and display both values on the screen. After each round ask the user if they want to go again. Give a way to reset all values.

You will need multiple variables: one to keep track of the current reaction time, and then a list variable to keep track of all the times (you need that one in order to be able to find the average each time), plus a variable to use with the loop when you are adding up the values to find the average.

Remember, to find the average of a list of number you add them all up and then divide by how many numbers there were.

Use this checklist to plan your program:

- The program has a sprite which introduces the game and tells you how it is going to work. The current time and average time fields show the value of 0.
- The user's reaction time is measured somehow in a randomly timed scenario (a sprite appears and needs to be clicked on, a sprite changes color and then you click on it/type a key, etc...)
- After each round the current time and the overall average time is displayed in the visible variables, and also said in a speech bubble by the main sprite.
- The program repeats forever.
- There is some way given to reset all the values and start fresh. It can't just be that the user secretly knows that you type the spacebar to reset, someone has to tell the user how to reset.
- The average gets calculated after each turn.
- The only variables showing are the current and average. There is a fun background showing.

Test your program. Do a quick reaction time, then a longer one, make sure the average is about in the middle between those two values. Do another quick reaction time, and verify that the average gets smaller. In other words, test your program and make sure it seems to be working correctly.

When it's working and you think you are done, turn in a share link on the Google Classroom. Get a classmate to try it out.