Snap*!* Lab 5 Day 3

Operations on variables, guess my number game

Click the links below to jump to a task.

- 8. Operations on variables
  - RAISE TO THE POWER block
- 9. Number Guessing Game

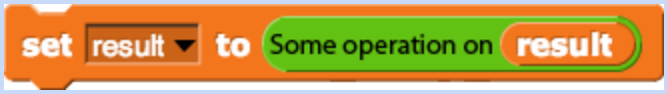
Today you will be working alone. Open Snap*!* at snap.berkeley.edu/run. Your job is to use what you have learned so far to make a game in Snap*!*.

Save this project as **Lab5 - guess my number**.

Remember, anything in a blue box is what we discussed in class.

8. Operations on variables

You will sometimes find it useful to use some expression with the current value of a variable to reset the value of the variable. Here is how that might look in Snap:



The operation might be 

or 

or many others.

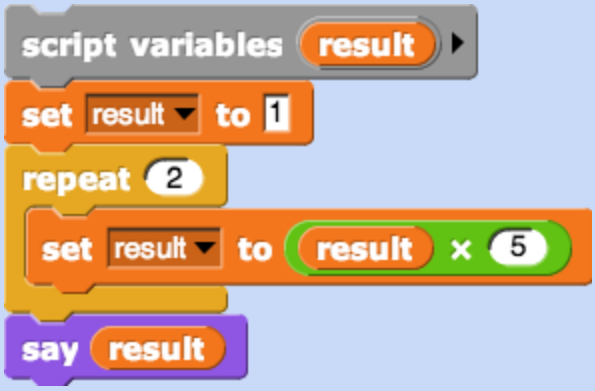
1. Here is one example:



   - Build it, click it, and figure out why 

says one thing in the first “say” block and a different thing in the second “say” block. You are encouraged to discuss this with a neighbor.
   - Predict what this script will do if you change the 5 to a 7, then run the script to see if you are correct.

2. Here's a more useful example:



   - Build the script, try it, and figure out how it does what it does.
   - Change the 5 to 3 and predict what it will do. (this number is the **base**) Then change the repeat 2 to repeat 3. (this number is the **exponent**)
   - Change the base to 2 and the exponent to 5. This code does the calculation: $base^{exponent}$, so if base = 2 and exponent = 5, the calculation is $2^5 = 32$.

3. Another place where script variables can be useful is in a program that interacts with the user.



   - Build this code. Note that this block of code has two script variables.

     Remember, if you click the right arrow 

, a second (third, fourth, …) orange variable oval will appear. Click the variable to rename it.

RAISE TO THE POWER block

4. Build a reporter block: 

.

Your block should work like this:









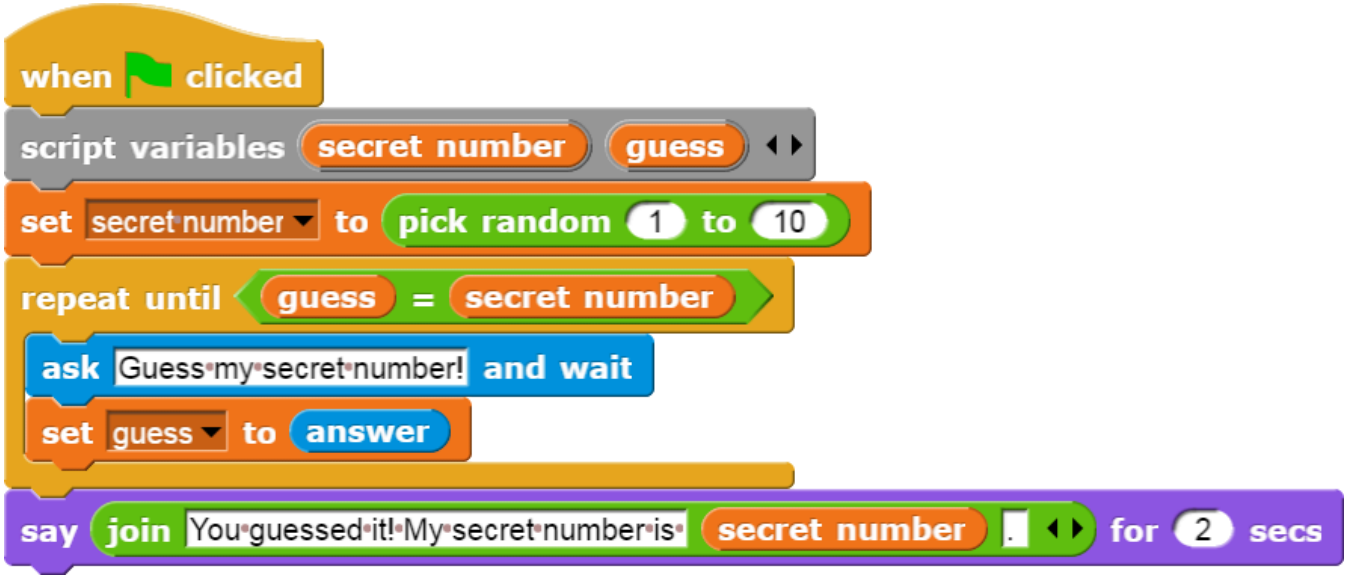
And make sure it works for this special “base case”:



Check lots of values (as always!) to make sure it works as you expect.

9. Number Guessing Game

This block is a first draft of a number-guessing game. The computer chooses a number and asks the player to guess it. You will improve the game to make it seem more like a person is playing with you.



Here are some ways to make the code act more like a human. Try each of these in order. When you are done, your finished code should be able to deal with all of them.

1. (20 pts) Have the sprite welcome the player and ask for their name before beginning.



2. (20 pts) Have the sprite tell the player whether each guess is correct or not, using the player's name. For example, "Nope, Maria, that's not my number. Try again." or "Wow, Maria, you got it!" Be creative!



3. (20 pts) Right now, the script always picks a number between 1 and 10. At the start of the game, ask the player what the maximum number should be, and set a variable named “maximum” to that number. Let your program choose a number between 1 and maximum.



4. (10 pts) Have the sprite tell the player whether the number the player guessed is too big or too small. "That's too big, David. Try again."



5. (10 pts) Keep track of how many guesses the player makes (maybe use a variable named “count”) before getting the right number. When the player guesses the secret number, say how many guesses it took and congratulate the player, using the player's name. Make sure “count” resets to 0 when the game starts.



(the “show variable” view 

doesn’t count!)

(5 pts each) Here are some more ideas of ways to make your game even better. Feel free to use any or all of these ideas, or come up with your own.

- If the player guesses something that is not a number, tell them they should guess a number.



- If the player guesses a number outside of the guessing range (larger than the maximum or smaller than the minimum), remind them of the range and ask them to guess again. You decide whether this counts as a guess or not.

- Let the user select a level before they start. Make sure the welcome message gives them good instructions, and give them an error message that asks them to try again if they type something that isn’t available. Some ideas to get you started:
  - Easy: secret number between 1-10
  - Difficult: secret number between 1-50 (or whatever)
  - Custom: allow the user to select the max number
  - Evil: make the secret number reset each time the player guesses (but you don’t have to tell them that 😈)

- Add fun graphics, costumes, and sounds to your game. Make it your own!

Lab 5: Guess My Number Game Name: _____ Period: ____

Peer grader: _____ *Mark each feature you observe in the game.*

Requirement	Complete?		Points
	peer	self	
a) welcome the player b) ask for their name	❑ a ❑ b	❑ a ❑ b	/ 20
c) tell the player whether each guess is correct or not d) use the player's name	❑ c ❑ d	❑ c ❑ d	/ 20
e) ask the player what the maximum number should be	❑ e	❑ e	/ 20
f) say whether the number they guessed is too big or too small	❑ f	❑ f	/ 10
g) when the player guesses the secret number, <u>say</u> how many guesses it took (displaying the number on the screen doesn't count)	❑ g	❑ g	/ 10
h) if the player guesses something that is not a number, tell them they should guess a number and to try again	❑ h	❑ h	/ 5
i) if the player guesses a number outside of the guessing range (larger than the maximum or smaller than the minimum), remind them of the range and ask them to guess again	❑ i	❑ i	/ 5
j) let the user select a level before they start k) make sure the welcome message gives them good instructions l) give them an error message that asks them to try again if they type something that isn't available	❑ j ❑ k ❑ l	❑ j ❑ k ❑ l	/ 5
m) add fun graphics, costumes, and sounds to your game	❑ m	❑ m	/ 5
n) an original game feature - describe:	❑ n	❑ n	/ 5

TOTAL: _____ / 100