

**LESSON 5: Oct 27 due Nov 3**

In the spirit of creating a culture with your Scratch students where we "learn from others" I begin Lesson 5 by asking you to listen to a tech teacher from Houston Texas -- Jeff Diedrich -- whom I interviewed back in 2013 on Skype.

**Required video HW5.0a-- 20 min -- <http://youtu.be/pgc0-y-ZXJ0>**

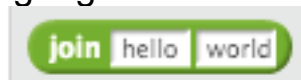
I learned from one of his students how to make a ball bounce and you will see an optional video on this at the very end of Lesson 5.

**HW5.0b** Please share via email with your two partners and me a sentence or two on something interesting you gleaned from this 20 minute video.

## Next

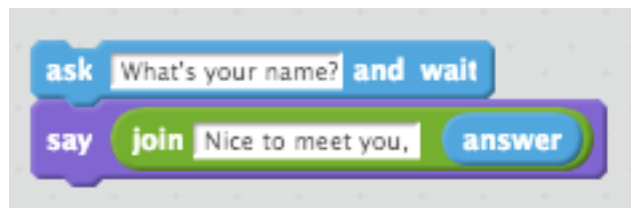
Let us push ahead with several commands that some of you have worked through in our individual conversations but I have not formally taught to you.

**The JOIN command** lets you combine words or phrases into one expression. You will find it in the light green OPERATORS section.

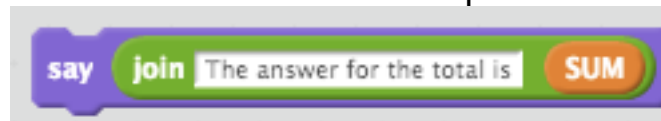


You can type any text into the white rectangles where it says "hello" and "world" including variables or even another JOIN command.

So you can be more elegant in your dialogue this way.



Or more elegant in your answer at the end of computation this way.

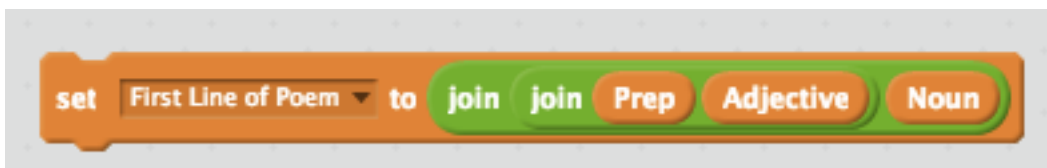




By putting two JOINS into a third JOIN you can have this one 4 part combo!



Or imagine if we were to pick a random prepositional phrase plus a random verb plus a random noun, we could have this first line of a poem.



## Next: The LENGTH command and the LETTER command.

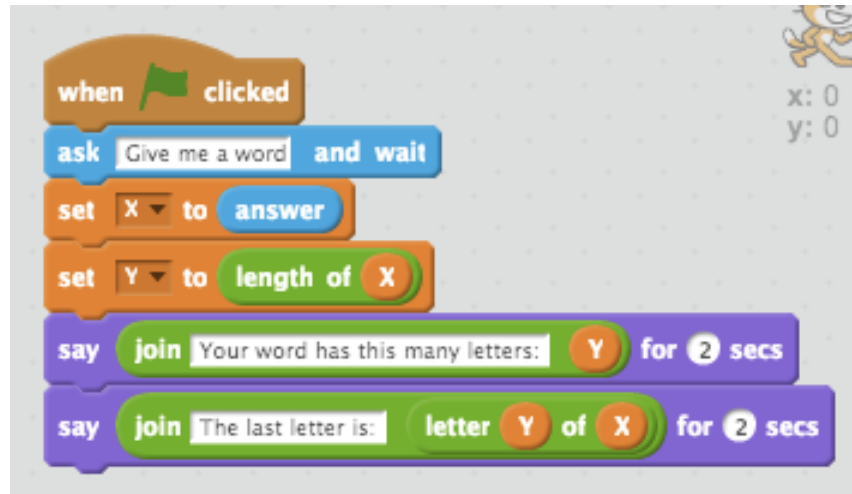
Another two commands that work on words are



Length finds "how many bytes" or "how many characters" long a word is or an alpha variable is. The Letter command identifies any specific character (or byte) of a word.

Note that you could have a variable above where it says "world."

Here is a simple program that uses both of them.



In step 4, we have the new command LENGTH so that the variable Y is assigned the length of X which is the word that the human entered. Most of you know the word "byte" but just in case, I will remind you that a byte is a keystroke or a character. So the word "larry" has 5 bytes and the text string "larry bird" has 10 bytes.

in step 5 you see the use of the JOIN command again. Once you start using JOIN, it will become one of your favorites!

In step 6, the final step, you see the use of the LETTER command which is picking a specific byte of a certain variable.

Hope this is all clear, but here is a HW problem that will get you to use these commands which as we all know is the only way to make it your own.

**HW 5.1 REQUIRED** Allow the user to enter a word. The computer then takes the word apart, stating each letter one at a time. So if the user types SCRATCH, the computer shows an "S" then a second later a "C", and so on.

**HW 5.1B OPTIONAL CHALLENGE:** Have computer count # of vowels in the word.

**HW 5.1C OPTIONAL AND FAIRLY HARD CHALLENGE:** Have computer share the individual letters in reverse order.

### **OPTIONAL VIDEOS and SOLUTION for 5.1C**

A 1 minute video demonstration of 5.1C (not the solution). Ignore any confusion that might say 6.2 which was a mistake on my part -- [http://youtu.be/NGsYVv\\_KMvU](http://youtu.be/NGsYVv_KMvU)



Allow the user to enter a word. The computer then takes the word apart, stating each letter one at a time **in reverse order**. So if the user types SCRATCH, the computer shows an "H" then a second later a "C", then a second later an "T" and so on.

A 3 minute video of hints without giving it away

<http://youtu.be/3RYrnBvK5pk>

• Spoiler Alert, a 6 minute solution and instruction video <http://youtu.be/aiJTJVvfiBA>

SOLUTION that is INTENTIONALLY UPSIDE DOWN



## Next Trick: Adding a horizontal scrolling

**message** as you might have seen in many of the Lesson 4.1 Sample Creations by others.

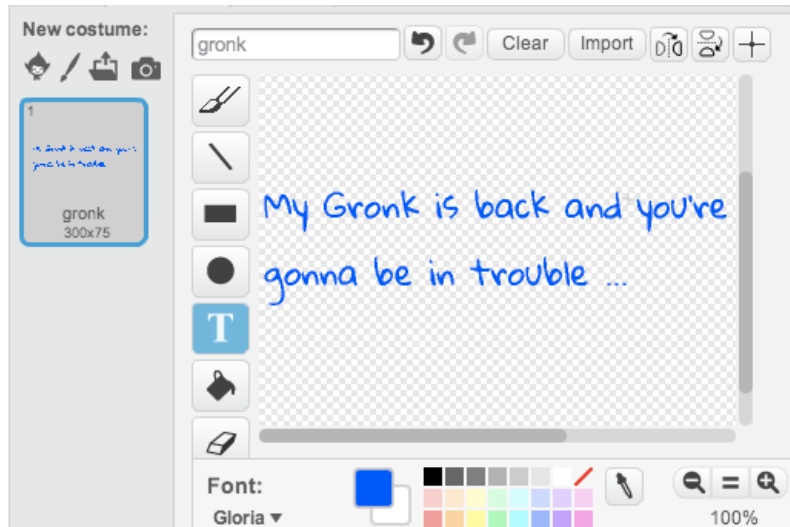
Video at <http://youtu.be/OBxX9mfyWwl> is 3 minutes.


Here is the idea. Animation is simply showing a repeated still picture at a different position every tenth (or fraction) of a second. So why not create a new sprite on a blank screen and use the text tool to type a word for this sprite. So that is just what I



## summercore

did, making a new Sprite called Gronk, clearing the screen and using the - to give me 100% view and more room to type. So your NEW SPRITE is TEXT in 1 or 2 lines that you type onto the screen in a desired color and font!



Reminder that the  sign is on the graphic screen if you magnify and it is the PRECISE LOCATION that is used when you say GO TO (x,y) coordinates.



You can change the location of the + sign in the graphic screen by using the + icon at the top right of the graphics screen.



Or you can use this same icon to "make the +" if you cannot find it.



Next I wrote this script for this Sprite that makes my TEXT be ANIMATED.



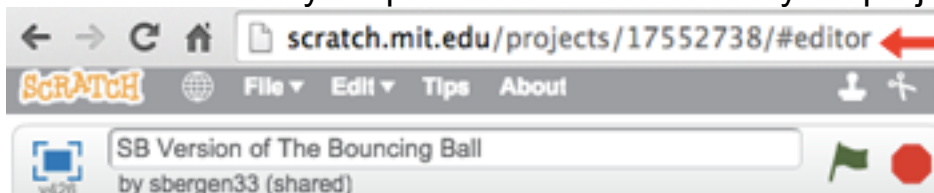
Notice that there is a new variable called GRONK which is just the X coordinate.

The variable starts at 450 so that in the FOREVER LOOP, the animation begins at location 450,100 on the screen which is off the screen to the right but that is because the + sign is between "and" and "you're" in my sentence.

Each time through the loop, the variable GRONK decreases by 1 and so that produces the animation, displaying at 449,100 then 448,100 and so on. When GRONK gets below -240, we are at the left side of the screen so the IF/THEN statement sets it back to 480.

Trivia Pursuit? Gronk is the tight end of the New England Patriots and the song is based on a song from 1963 by the Angels called "My Boyfriend's back ... "

**HW 5.2 REQUIRED** Enhance your program from 4.1 so that it has a scrolling message on the screen. The message content can be a welcome message or instructions on what keys the user should push or a description of the game. Send a quick email to me and your partner with the URL of your project once you are done.



HERE IS THE  
URL(no need for  
the #editor)

**HW 5.2B OPTIONAL:** Make the text animation be vertical instead of horizontal.

## NEXT TOPIC An Intro to LISTS

We are going to learn about LISTS this week and then use Scratch to write poetry! I am very excited to teach you this lesson -- based on something I did in 1973-74 when I was young and started teaching programming at a Virginia independent school. Let us continue to work with words and stay away from numbers and graphics this week. The programming structure called LISTS is an essential part of SCRATCH and other serious computer languages.



A LIST is a bunch of items in sequence, numbered as ITEM 1, ITEM 2, or ITEM 3, perhaps we can say "an array of variables" or a "list of variables."

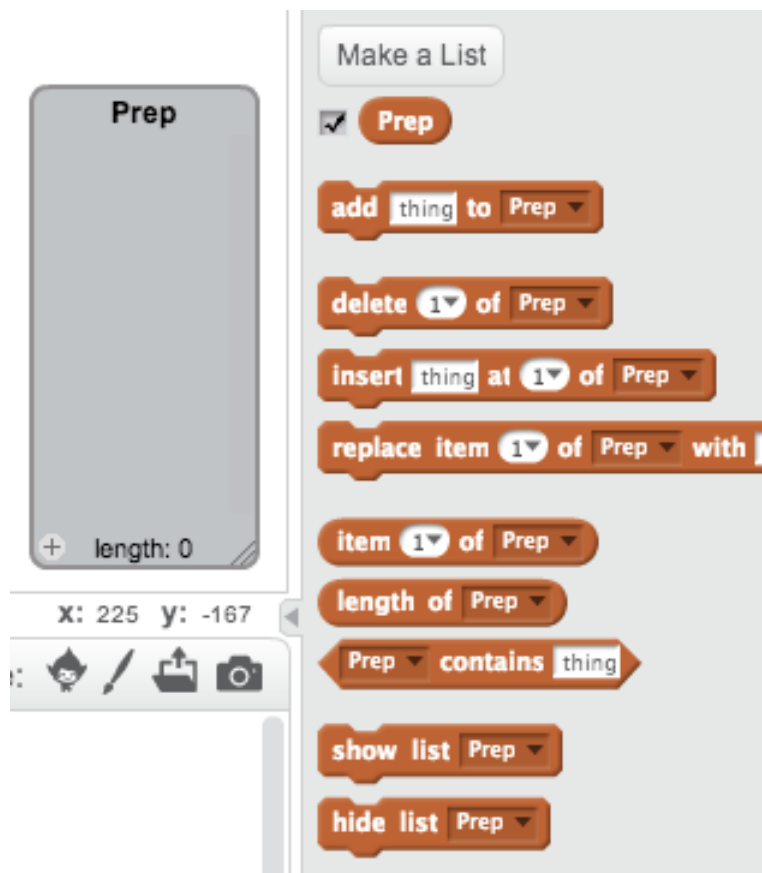
Yes, a LIST is a term from Computer Science. Here is a techie excerpt from Wikipedia:

In [computer science](#), a **list** or **sequence** is an [abstract data type](#) that implements a finite ordered collection of [values](#), where the same value may occur more than once. An instance of a list is a computer representation of the [mathematical](#) concept of a finite [sequence](#); the (potentially) infinite analog of a list is a [stream](#). Lists are a basic example of [containers](#), as they contain other values. Each instance of a value in the list is usually called an **item**, **entry**, or **element** of the list; if the same value occurs multiple times, each occurrence is considered a distinct item. Lists are distinguished from [arrays](#) in that lists only allow sequential access, while arrays allow [random access](#).  
SOURCE: [http://en.wikipedia.org/wiki/List\\_\(abstract\\_data\\_type\)](http://en.wikipedia.org/wiki/List_(abstract_data_type))

Here is the 7 minute starter video for LISTS. You can watch it before you read this PDF or after or both. By now, you have your style of learning and preference.

<http://www.youtube.com/watch?v=c4XbuJinwFk&feature=youtu.be>

When in Scratch, I go to ADD A LIST under DATA and make a list called PREP, I get all of these commands added



Let's make sense of each of them above.



lets you add a word to the LIST called PREP. If PREP has nothing then the word you have here (e.g. "thing") would become item 1 of PREP. If PREP has 5 items, then this word becomes the 6th item.



lets you remove any item from the list. If you click the triangle, you can also DELETE ALL.



lets you add an item to the list. But it doesn't go at the end (like the ADD command). Instead, it is inserted and everything from that item on is bumped up one.



should be self-explanatory. A word is replaced with another word. I wonder if we could use this command to alphabetize a list. What if go through the list and each time item X is out of alphabetical order with item X+1 then we switch them. Hmm, sounds like a challenge to me!



This is very important and we will use it frequently. You can use it to move this item into a standard variable. Or you can put the RANDOM command into the white box. Or you can use this command in IF/THEN logic.



This is most appreciated since we can set up a REPEAT loop with a counter (e.g. C) and have the computer go from C=1 to C=LENGTH. By length, we mean "how many items in the list."



This is also appreciated since it allows us to check if something is on the list. For example, let us say we have a LIST of the students in your class. You could ask the user for her/his name and instantly respond with verification that this is a member of the class or not.

**Okay, I hope you are ready for our exciting attempt to write computer poetry!**



To create a list of Prepositional Phrases, I have several choices for you.

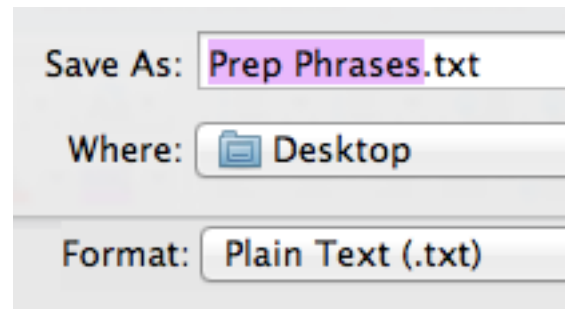
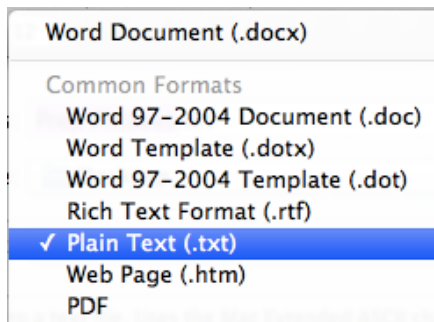
Plan A which is the easy way: You click the + sign at the bottom left of the Prep list. Now you can type each word into the list. Nothing easier than direct typing or editing!

Or Plan B: You can right mouse click on the gray Prep List box and import a text file that was saved from a word processor. Clearly, this is useful if you have a class list already created on a WP or SS document. Here are steps for Plan B:

Step 1: you go to Microsoft Word or whatever word processor you use and you type these words into a blank document,

Next to  
Beyond  
Beside  
Above  
Below  
Within

Step 2: you can then choose SAVE AS and change the FILE FORMAT to be PLAIN TEXT



Step 3: you now can IMPORT this file into your LIST by right mouse clicking on the gray area of the list.





Plan C: You can create this program which puts the words into the LIST.



Whether by Plan A, B or C, we get the LIST that appears in the Scratch window that looks like this



This is a LIST. It has a LENGTH of 6. Please note that LENGTH is a formal Scratch word. Each word is numbered and so we can refer to ITEM 3 or ITEM 5 of the LIST called PREP. A list is also called an ARRAY in computer languages.

This random command would allow us to pick a random preposition.





Got it? So if we take a famous poem such as one of these by Matsuo Basho

Next to misty stars  
Forests of visions unfold  
Next to the white fields

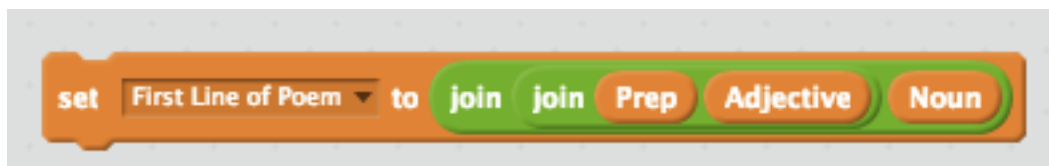
Beyond paper walls  
Voices of geishas whisper  
About the bright moon

Beside immense seas  
Valleys of absence wander  
Next to the true hills

Beyond ghostly moons  
Faces of silence whisper  
Within the cold storm

we can have the computer pick a CHOSEN preposition, then a random ADJECTIVE and then a random PLURAL NOUN

Using our JOIN command, we can have the computer generate the first line of a poem with this kind of command.



### HW 5.3 REQUIRED

Your assignment is to make the computer write ONE LINE of structured random poetry using LISTS, the JOIN command and the RANDOM command. Believe it or not, when I started teaching computer programming in the 1970s, I had one of my high school students who did a great job with this exact program then write an article for a national student magazine. Sorry for the quality of the graphic on the next page which was scanned many years ago, but I don't have the original anymore.

# the MATHEMATICS STUDENT

VOLUME 22 / NUMBER 3  
FEBRUARY 1975

## An Experiment in Computer Poetry

*Between moonlit waves  
Oceans of candles arise  
Next to the wild sky*

貴  
壽  
財

This is a haiku, a Japanese verse-form that has attained universal appeal. This haiku, like most others, consists of seventeen syllables divided into three lines, the first and last having five syllables, the middle seven, but this haiku, unlike most others, was written by a computer!

Having the proper number of syllables in each line is haiku's major requirement, for it involves no rhyme scheme or meter of any kind. This fact makes programming a computer to write poetry not quite so difficult as it might otherwise seem, for all that is necessary is to choose words with the proper number of syllables.

In order to compose haiku with a computer, I had to choose a haiku to serve as a model on which all those of the computer would be based. I therefore read several dozen haiku and finally settled upon one consisting of two two-syllable prepositions, two adjectives (one of one syllable, the other of two), a two-syllable verb, four nouns (two each of one and two syllables) and the words "the" and "of". The author of this haiku is one of the masters of this style, Matsuo Basho (1644-1694):

*Beyond paper walls  
Voices of gelsas whisper  
About the bright moon*

I chose this particular haiku as the paradigm (pattern) because nine of its eleven words can be varied in all haiku ("the" and "of" remain the same in each), thereby offering a great variety of possibilities.

The program was written for a Hewlett-Packard 2000C computer and includes 265 different words — "the" and "of" plus nineteen prepositions (all of two syllables), ninety-six adjectives (fifty-nine of two syllables, thirty-seven of one), forty verbs (all disyllabic) and one hundred and eight nouns (seventy-three of one syllable, thirty-five of two). The computer composes haiku in the following manner.

(Continued next page)

## An Experiment in Computer Poetry

貴  
壽  
財

(Continued from Page One)

There are nine slots — one for each variable word. All the words assigned to the first slot are two-syllable prepositions, so as to conform to the first word in the paradigm, "beyond". All those assigned to the second are two-syllable adjectives, patterned after "paper". Those assigned to the third are one-syllable nouns, as is "walls", and so on. The computer thus selects one word to fill each slot according to the paradigm. It is actually a simple matter to alter the order of registration of the nine slots in any way one chooses (see third and fourth examples below), although one must keep the requisite number of syllables in each line.

The number of different possible haiku can be calculated by multiplying the total number of words assigned to each slot. Thus if the haiku had only three slots, with a ten-word vocabulary for each, there would be  $10 \times 10 \times 10$  or 1,000 different haiku. Because of the vastness of the vocabulary, the number of slots, and the possibility of reversal in line 1, however, there are upwards of  $4.11 \times 10^{14}$  haiku possible — over 411,000 billion! This is illustrated by the following chart:

LINE I (three words)		
19 possibilities 2 syllable prepositions	59 possibilities 2 syllable adjectives	73 possibilities 1 syllable nouns
LINE II (four words)		
35 possibilities 2 syllable nouns	35 possibilities of 2 syllable nouns	48 possibilities 2 syllable verbs
LINE III (four words)		
19 possibilities 2 syllable prepositions	37 possibilities the 1 syllable adjectives	73 possibilities 1 syllable nouns
$19 \times 59 \times 73 \times 35 \times 35 \times 48 \times 19 \times 37 \times 73 = 2.0577018 \times 10^{14}$		

The possibility of reversing the words in

Line I yields  $4.1155956 \times 10^{14}$  haiku. In other words, there would be over 100,000 different poems for every person on earth!

Obviously not every haiku will be good, or even fair. Most of them, in fact, are totally meaningless. Indeed only one out of fifteen or twenty composed is of poetic interest.

re are some of the best yet:

*Beside immense seas  
Valleys of absence wander  
Next to the true hills*

*Out of immense snow  
Voices of valleys are hushed  
Next to the deep moon*

*Profound depths before  
Forests of meadows reside  
Beyond the still seas*

*Vacant trees below  
Rivers of music are hushed  
Beyond the dark moons*

*Beyond ghostly moons  
Faces of silence whisper  
Within the cold storm*

*Next to misty start  
Forests of visions unfold  
Next to the white fields*

Since nearly all these haiku conform to the paradigm, it is understandable that there is some monotony owing to the repetition.

There are two ways to remedy this: to choose one — or even several — more paradigms, or to vastly increase the vocabulary, or both. When this will eventually be done, the haiku will lose their uniformity, and the results will no doubt be amazing.

That a computer can compose poetry makes one wonder if the machine is capable of creating thought by itself. I can certainly not claim that I wrote these haiku because the computer is responsible for the selection of the words printed. I did, however, choose the paradigm, restrict the computer to certain words, and then decide which poems were "good". But can I in spite of this take credit for these haiku?



Francis Lang  
St. Anne's —  
Belfield School  
Charlottesville, Va.

Here is the pattern to make it easier for you. If you want to use a different haiku as a template, that is also fine and would be quite exciting!

You can google "famous haiku" and get many hits and ideas. Reminder that a Haiku has a structure of 5-7-5 in terms of # of syllables in each of the three lines.

I am using ADJ2 for a two syllable adjective and ADJ1 for a one syllable adjective.

PREP ADJ2 NOUN

NOUN of NOUN VERB

PREP the ADJ1 NOUN



PREP (need to be two syllables)

Next to

Beyond

Beside

Above (you can make up more)

ADJ2 (two syllable adjective)

misty

paper

immense

ghostly (you can make up more)

ADJ1 (one syllable adjective)

white

bright

true

cold (you can make up more)

NOUN (needs to be plural and one syllable)

stars

walls

seas

moons

fields (you may want a SECOND NOUN or you can just use one)

Good luck -- I look forward to seeing your random poetry!

Steve

*Hey everyone -- shh ... Steve tricked you on page 11.  
He made you think that all four poems were  
written by Matsuo Basho but that is not true.  
Three of the four were written by François'  
computer program in 1974.*

LB Doll



**HW 5.4 REQUIRED** Create a list called STUDENTS that contain 5 or more kids that you teach. Write a program that will randomly pick one of the students.

If you are feeling shaky about LISTS then **watch this video** where I do this program and teach you more. You can always try it yourself 24 hours later which is a good learning thing to do.

optional 8 minute video related to 5.4 that shows the solution and teaches you more about lists: [http://youtu.be/TA3L\\_YG8gLc](http://youtu.be/TA3L_YG8gLc) (*ignore reference to 6.3 in the video*)

**HW 5.4 OPTIONAL VERY DIFFICULT CHALLENGE** Modify the program above so that it results in a random sequence of the students each time you run the program as if the students are going to a museum and this is the sequence of who leaves first then second and so on. One quirky challenge is that your program should be able to run repeatedly i.e. you don't want the list to disappear or have to be re-entered each time the program is run. So pretend the kids are going to the museum every day and your program will be run each morning to generate the random sequence. If you are feeling shaky about LISTS and want more instruction rather than a challenge then watch this optional 15 minute video at <http://youtu.be/rtjqHL7veJ0> where I write this program and teach you more.

Have a good week!



781-953-9699 • skype name = stevebergen (no spaces) • [sbergen33@gmail.com](mailto:sbergen33@gmail.com)

p.s. we end this Lesson with an **OPTIONAL 5 minute VIDEO** at <http://youtu.be/Aq8WJGP2wQs> that shows you how Jeff's student created the bouncing ball. He got the script online from another scratcher by searching within the Scratch library of users.

The code is below **FOR THOSE OF YOU BRAVE ENOUGH** to want to study it and struggle with it! I would be glad to help you one-on-one with this when we have our conversation.

Optional exercise: get this code by remixing from my library at <http://scratch.mit.edu/projects/17552738/> and then implement it in your 4.1 program so that some object falls like the bouncing ball.

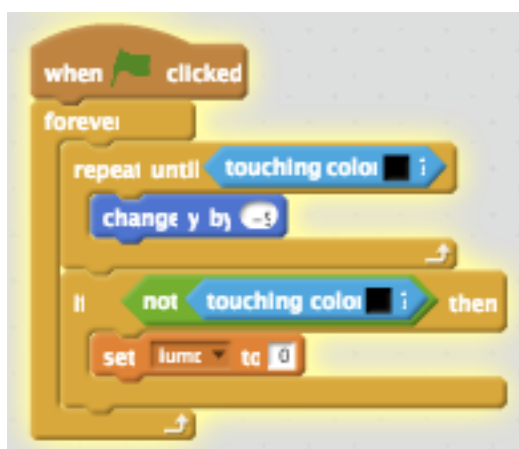




THIS IS THE SPECIAL SCRIPT THAT DOES SOMETHING AMAZING WHEN THE UP ARROW IS PRESSED. FIRST, THE VARIABLE JUMP IS CHANGED TO 17 SINCE IT WAS PROBABLY SET TO 0 WHEN THE BALL WAS SITTING ON BLACK. NEXT, THE Y COORDINATE OF THE BALL IS INCREASED BY 10 WHICH IS WHY THE BALL GOES UP. THIS HAPPENS REPEATEDLY SINCE WE ARE IN A FOREVER LOOP. SO WE GO UP, UP, UP.

HOWEVER, THE BALL IS NOT JUST GOING UP BY 10 EACH TIME THRU THE LOOP. IT IS ALSO GOING UP BY JUMP WHICH STARTS AT 17 BUT THEN BECOMES 16 THEN 15 THEN 14.

SO THIS PROGRAM MOVES THE BALL UP WHILE THE ONE BELOW CALLED SECOND SCRIPT MAKES THE BALL GO DOWN.



THIS SECOND SCRIPT MAKES THE Y COORDINATE DECREASE BY 5 AS LONG AS THE BALL DOES NOT TOUCH ANYTHING BLACK.

ONCE THE BALL TOUCHES BLACK THEN THERE IS NO MORE CHANGE TO Y. THAT IS WHY IT STOPS!

THINK OF THIS PROGRAM AS AN ELEVATOR THAT GOES DOWN 5 FLIGHTS UNTIL IT HITS BLACK WHEN THE VARIABLE JUMP CHANGES TO 0.