LESSON 5: Feb 23 due March 2

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.

HW5.0A Required video to watch -- 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 partner and me a sentence or two on something interesting you gleaned from this 20 minute video with Jeff.

Jo Gifford	Team 1	1st grade & Tech to 7&8 -- The Branch school	TX
Sophie Mortner	Team 1	2nd grade -- Rodeph Sholom School	NY
Brian DeCicco	Team 2	6th & 8th Science -- Edmund Burke School	DC
Jeff Funk	Team 2	Fifth Grade core subjects -- Gig Harbor Academy	WA
Amy Cataldo	Team 3	7th Grade Life Science -- Edmund Burke School	DC
Kent Vienot	Team 3	'6-9 -- Shore Country Day	MA
Jan Walker	Team 4	K-5 Computers -- Emerson School	MI
Stella Liberman	Team 4	3rd grade -- Rodeph Sholom School	NY
Barret Stump	Team 5	K-12 All -- Lafayette School	VA
Kathy Washington	Team 5	K-12 Technology -- Paideia School	GA
Jennifer Garvey	Team 6	K-4 technology -- The Peck School	NJ
Renee Durrant	Team 6	Technology k-8 -- Emerson School	MI
Cam McNall	Team 7	pre-K through 9 -- Shore Country Day School	MA
David Ritzman	Team 7	K-6 Tech. Director -- Rio Grande School	NM

Next

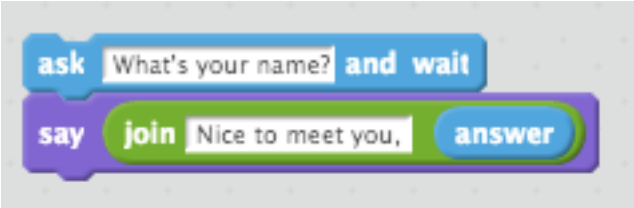
Let us push ahead with several commands that some of you have worked through in our individual conversations but I have not formally taught to you.

The JOIN command

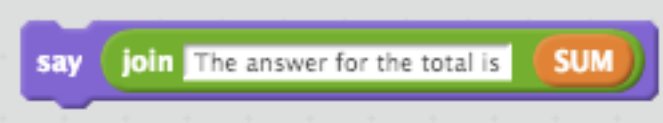
lets you combine words or phrases into one expression. You will find it in the light green OPERATORS section.

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.

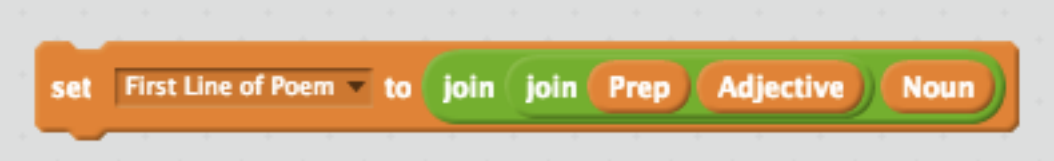


Or more elegant in your answer at the end of computation this way.



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, let us learn about BROADCASTING a MESSAGE -- another topic we have not discussed extensively

If you look under EVENTS you see 3 commands that involve the word BROADCAST.

This idea is that this command gives us a method for Sprite #1 to interact with Sprite #2. For example, I could take my Tom Brady program and change it so that when Tom goes to one of the corners, he calls one of the players and "broadcasts" a message. This way that player and ONLY that player chases him.

When you click on the triangle for any of the above commands, you can create a NEW MESSAGE of your choice.

The essence of the BROADCAST command is that we have a different way for one of the programs under one of the Sprites to begin. In the old days (e.g. a few weeks ago), every program under every Sprite would begin when the GREEN FLAG was clicked or perhaps when a certain variable or flag changed a certain value. Now a Script can begin when another Sprite "yells at it." So if I had Sprites named Larry, Moe and Curly, I can have the script for Larry begin when we click the green flag, but the scripts for Moe and Curly don't begin until Larry yells "Moe" or yells "Curly." Or you can have Larry broadcast "MoeCurly" and then those two scripts begin simultaneously.

Next: The LENGTH command and the LETTER command.

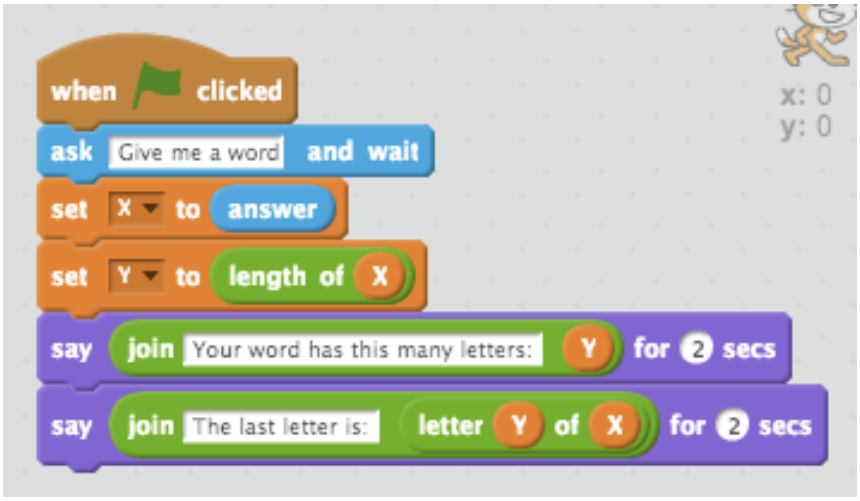
Another two commands that work on words are

and

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 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

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

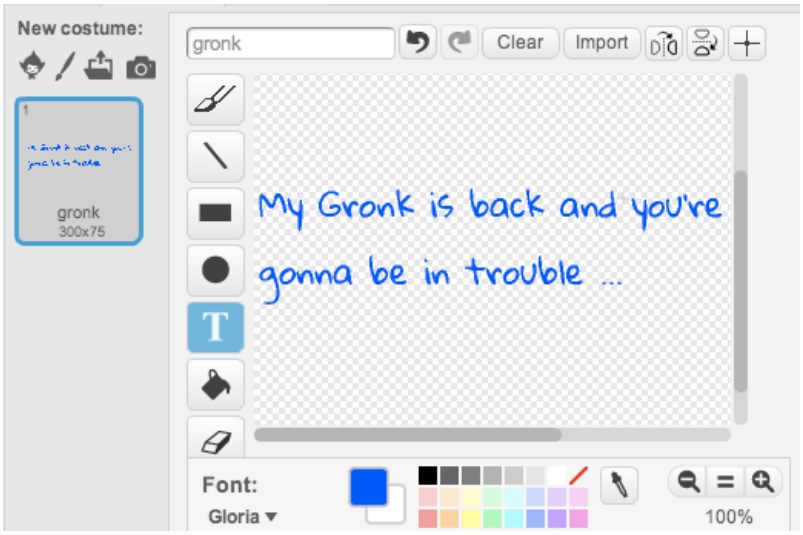
SOLUTION that is INTENTIONALLY UPSIDE DOWN

Next Trick: Adding a horizontal scrolling message as you might have seen in many of the Lesson 4.1 Sample Creations by others.

HW 5.2A REQUIRED Video at http://youtu.be/OBxX9mfyWwI 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

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 to change what I call the ZERO POINT.

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.

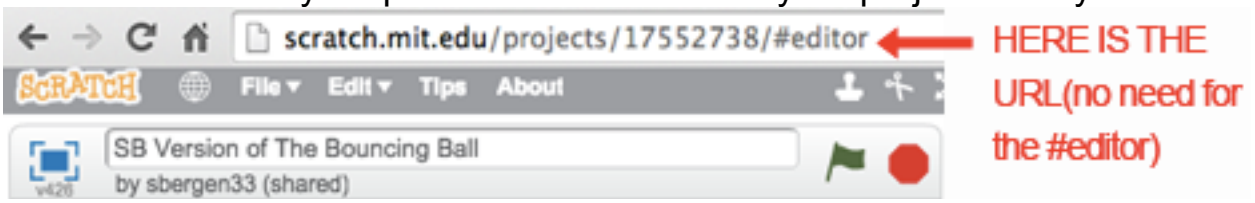
Notice that there is a new variable called GRONK that is just the X coordinate.

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 5.2B 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.



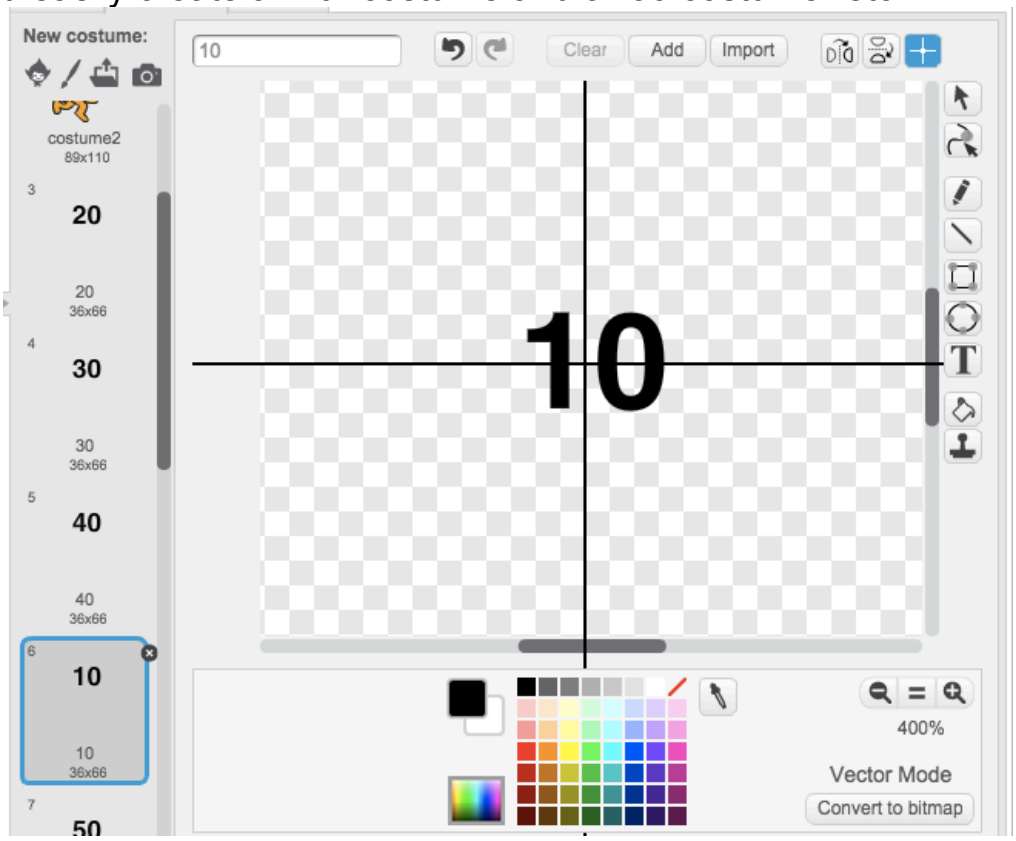
Here is an example of the 4.1 project by Jen from NJ that I enhanced with a scrolling text message. The zero point has been set at the start of the message.
http://scratch.mit.edu/projects/49030474

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

NEXT TOPIC An Intro to STAMP

The STAMP command found under the PEN category takes the CURRENT COSTUME and stamps it on the screen like a rubber stamper on a piece of paper. I was showing this last Thursday afternoon to Kathy from Georgia in the context of the

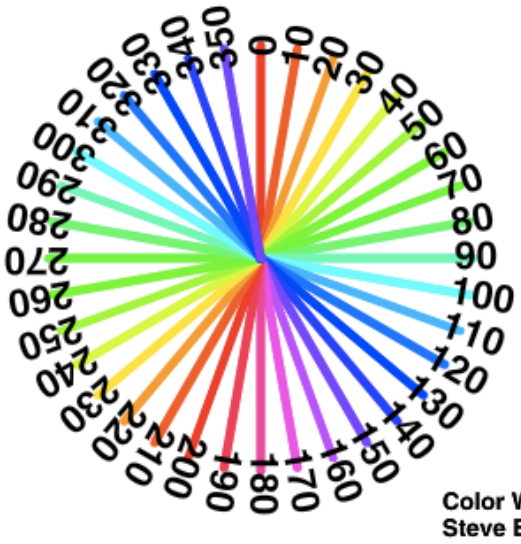
color wheel. I created a NEW COSTUME called "10" and typed the text on it to be just a 10. I made sure that the ZERO POINT was situated directly between the 1 and 0. I did this in VECTOR MODE (not bitmapped) so that I could then DUPLICATE the costume and easily create a "20" costume and a "30 costume" etc.



I woke up Friday morning and decided to finish what Kathy and I had started. Yes I made a new costume for each angle from 0 degrees to 350 degrees using the DUPLICATE feature for each subsequent costume and then via vector mode just changing one digit. Notice that the costume for "10" is named "10" using the blank at the upper left.

All that remained is to have 36 IF statements (also done efficiently with duplicate) and I had my Friday morning fantasy! Yes coding and programming is about being compulsive at times, having a fantasy and then working to implement it sometimes declining to eat until it was done ;-)

Here is a snippet of the program and then the finished result on the right. And yes, I used the STAMP command to put my name and the title at the bottom right of the finished product.



You can find this program at http://scratch.mit.edu/projects/48813722 if you want. No I did not eat breakfast Friday morning until it was done in the spirit of Joseph Weizenbaum's observations in the 1970s about "compulsive programmers."

REQUIRED HW 5.2C: Enhance HW 4.1 yet again. Make a new sprite and new costume with your name and then write a script for this sprite to HIDE then GO TO a corner of your stage and then STAMP your name onto the canvas after a few seconds of your program running. Here is an example of 5.2C I did for you using the 4.1 project by Jo from TX that I enhanced with a STAMP message. The zero point has been set at the start of the message -- http://scratch.mit.edu/projects/48560712

NEXT TOPIC An Intro to LISTS

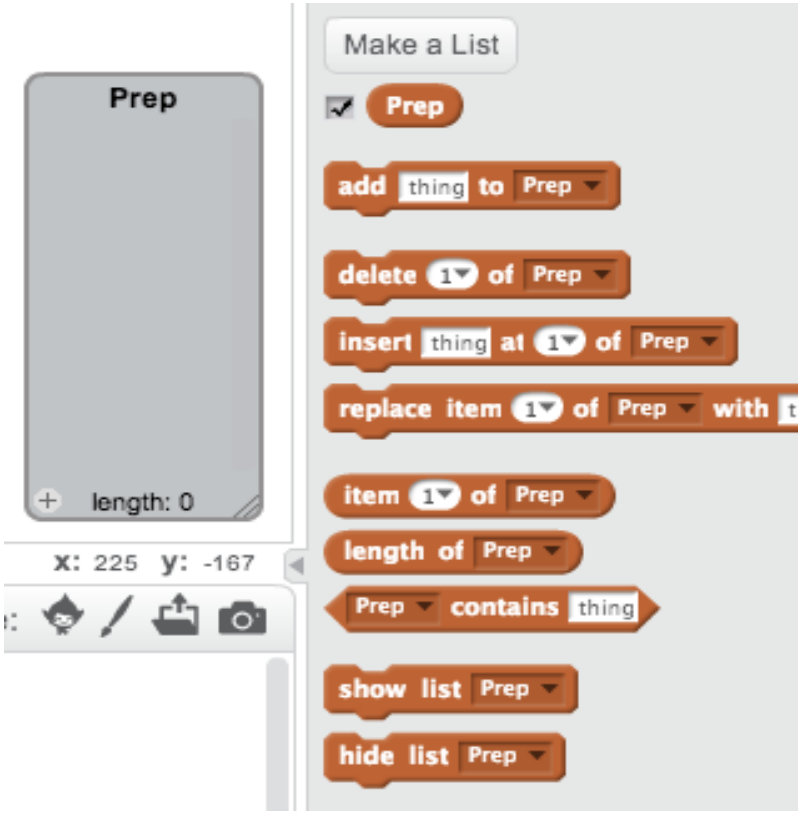
We are going to learn about LISTS this week and then use Scratch to write poetry! I am very excited to teach you this lesson -- based on something I did in 1973-74 when I was young and started teaching programming at an independent school in Charlottesville, Virginia. Let us continue to work with words 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.

Jo from TX worked on the 3.6 optional program called GUESS MY NUMBER and wanted the student to see all the guesses, so she added a LIST to the program. You can see her revised program at http://scratch.mit.edu/projects/48866192 and see a great example of how Jo used a list to remember all the guesses of the number.
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 an important and real 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)

HW 5.3A Here is the 7 minute required starter video for LISTS.
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

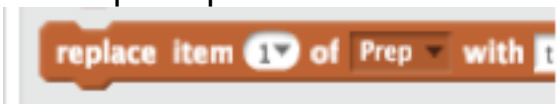


Let's make sense of each of them above.

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.

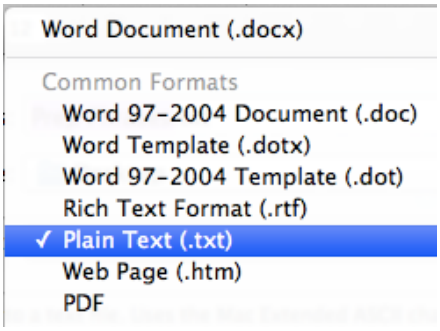
Plan A which is the easy way: You click the + sign at the bottom left of the Prep list. Now you can type each word into the list. Nothing easier than direct typing or editing!

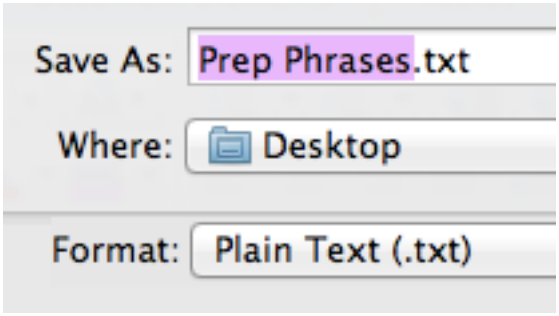
Or Plan B: You can right mouse click on the gray Prep List box and import a text file that was saved from a word processor. Clearly, this is useful if you have a class list already created on a WP or SS document. Here are steps for Plan B:

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

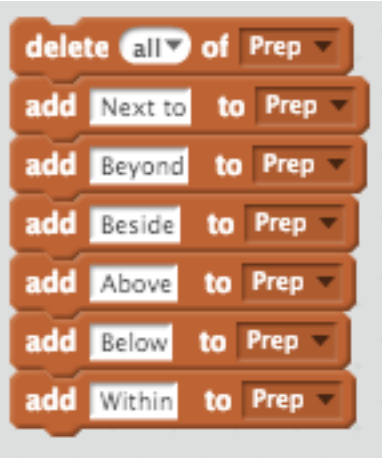
Step 2: you can then choose SAVE AS and change the FILE FORMAT to be PLAIN TEXT





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 that 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

But you can do better since RANDOM is built-in to the ITEM command

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

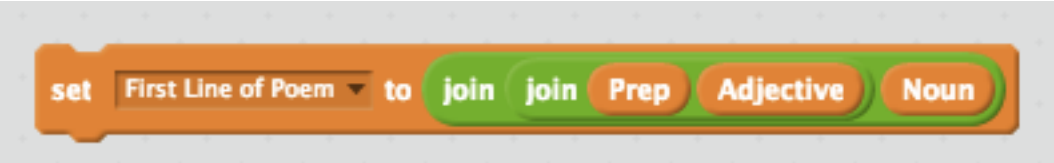
Next to misty stars
Forests of visions unfold
Next to the white fields

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 5.3B REQUIRED Your assignment is to enhance 4.1 and have the computer write ONE LINE of random "madlib type" 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.

REQUIRED HW 5.3B: Make a new sprite with a 1 line poem to enhance 4.1 so that the poem appears at the end of the program. Here is an example of the 4.1 project by Kent from MA that I enhanced with a STAMP message. The variable uses the LARGE READOUT option that you get by right-mouse clicking PC (or ctrl clicking Mac) on the variable: http://scratch.mit.edu/projects/48988862

the MATHEMATICS STUDENT
VOLUME 22 / NUMBER 3
FEBRUARY 1975

An Experiment in Computer Poetry

貫壽財

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

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 geishas 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

(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 x 10 x 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 4.11×10^{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	of 35 possibilities 2 syllable verbs	46 possibilities 2 syllable verbs
LINE III (four words)		
19 possibilities 2 syllable prepositions	the 37 possibilities 1 syllable adjectives	73 possibilities 1 syllable nouns

$19 \times 59 \times 73 \times 35 = 1 \times 35 \times 46 \times 19 \times 1 \times 37 \times 73 = 4.0977818 \times 10^{14}$

The possibility of reversing the words in Line 1 yields 4.1155956×10^{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 stars
Forests of visions unfold
Next to the white fields

Since nearly all these haiku conform to e paradigm, it is understandable that there ll be 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?

Francois Lamy
St. Anne's – Belfield School
Charlottesville, Va.

Here is the pattern Francois used

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 (needed to be two syllables): Next to or Beyond or Beside or Above

ADJ2 (two syllable adjective): misty or paper or immense or ghostly

ADJ1 (one syllable adjective): white or bright or true or cold

NOUN (plural and one syllable): stars or walls or seas or moons

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

LB Doll

HW 5.4A 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 (*ignore reference to 6.3 in the video*)

HW 5.4B OPTIONAL AND 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.

HW 5.5A REQUIRED Watch this 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 script is organized on the next page. Read it and study it while you watch the video or after you watch the video

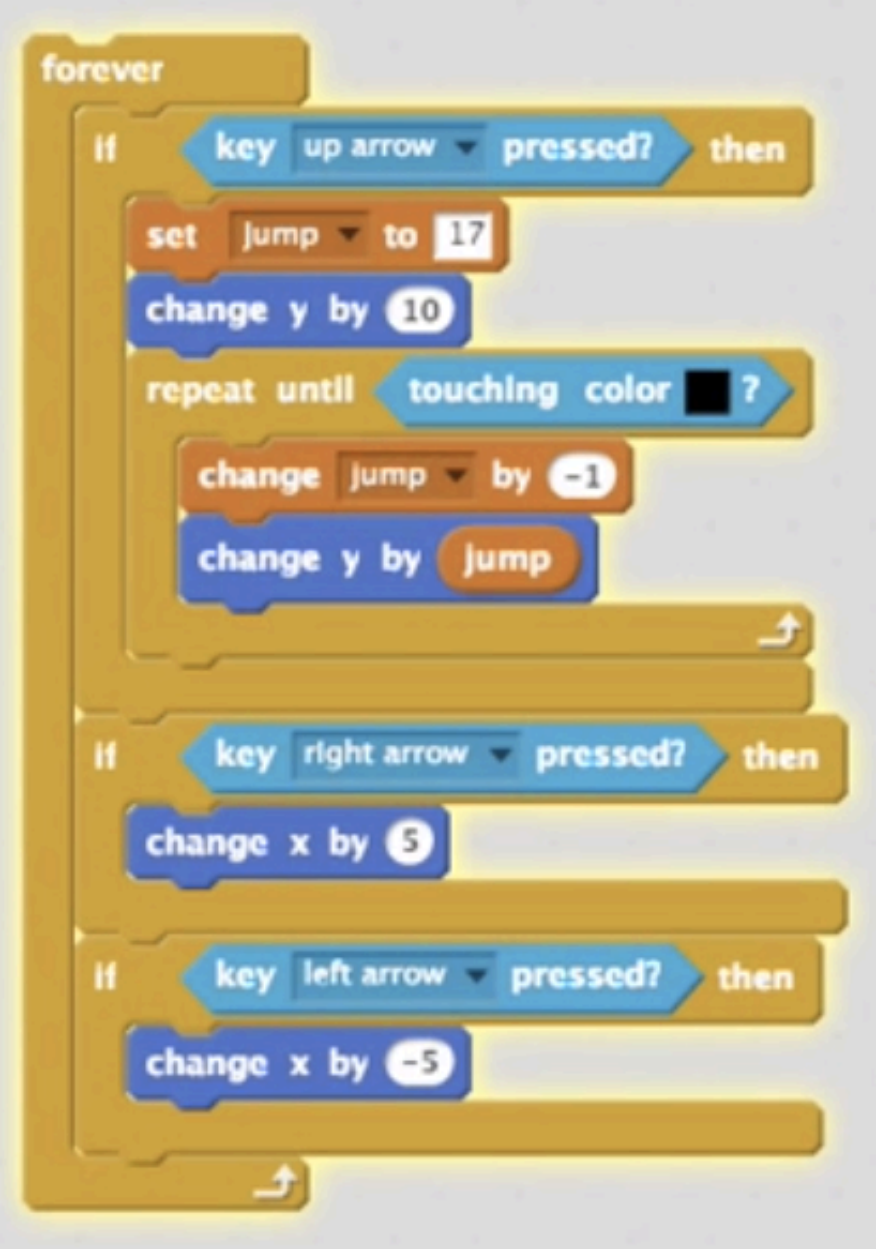
HW 5.5B REQUIRED Send an email to your partner and to me in which you either
a) ask a question about the bouncing ball
or
b) provide an insight about the bouncing ball script

HW 5.5C OPTIONAL and DIFFICULT CHALLENGE: get the code on the bouncing ball from my library at http://scratch.mit.edu/projects/49073398/ and then implement it in your 4.1 program so that some object falls like the bouncing ball. It does not need to be a ball, but it is important you decide on the color of the object that it hits that makes it bounce.

Beware! This is probably one of the most complicated challenges I have ever given you. Why?
You need to transport three programs via your backpack.
You need to change backdrops and make your other sprites disappear.
You need to deal with SHOW when the program begins and HIDE when the backdrop changes.

Here is a sample of the implementation that I did using the 4.1 Pokemon program by Amy from Wash DC. After the charizard catches the main character, the backdrop changes to a soccer goal and the charizard becomes the goalie. You can move the ball with the arrows and try to score but the charizard is intent on eating the ball and may succeed. See http://scratch.mit.edu/projects/49070838/

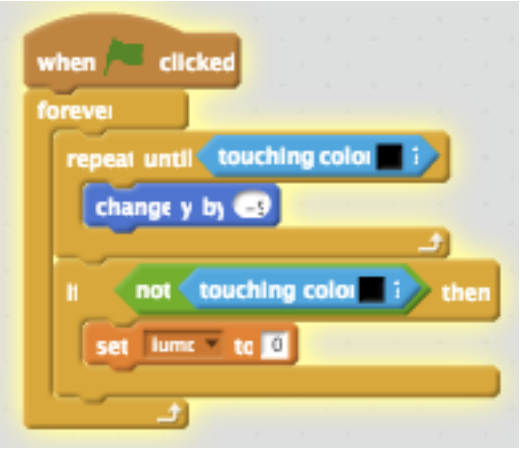
On the next page are 2 of the 3 programs in the bouncing ball script.



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.

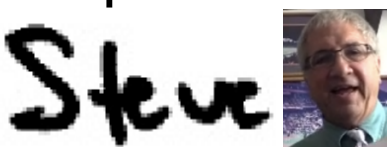


THIS SECOND SCRIPT MAKES THE Y COORDINATE DECREASE BY 5 AS LONG AS THE BALL DOES NOT TOUCH ANYTHING BLACK.

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.

Have a good week! Everything below is optional.



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

Optional 5.6: Here are several articles and videos on coding and Scratch

Article: Tips to get your Kids Excited about Coding
http://thejournal.com/articles/2013/10/23/6-tips-to-get-your-kids-excited-about-coding.aspx?=THE21

Video: Code, The New Literacy
http://www.youtube.com/watch?v=MwLXrN0Yguk

Video: What Most Schools Don't Teach
http://www.youtube.com/watch?v=nKIu9yen5nc

Learn Scratch Website http://learnscratch.org/

Scratch Curriculum Cards you can print out
http://scratch.mit.edu/help/cards/

Textbook
http://www.nostarch.com/learnscratch
Someone told me about this book which I have NOT seen yet -- Steve
TABLE OF CONTENTS -- http://www.nostarch.com/download/samples/LTPWS_DTOC.pdf

Chapter 1: Getting Started
Chapter 2: Motion and Drawing
Chapter 3: Looks and Sound
Chapter 4: Procedures
Chapter 5: Variables
Chapter 6: Making Decisions
Chapter 7: Repetition: Definite and Indefinite Loops
Chapter 8: String Processing
Chapter 9: Lists

Learn to Program with Scratch

A Visual Introduction to Programming with Games, Art, Science, and Math
by Majed Marji

February 2014, 288 pp.
ISBN: 978-1-59327-543-3
Full Color

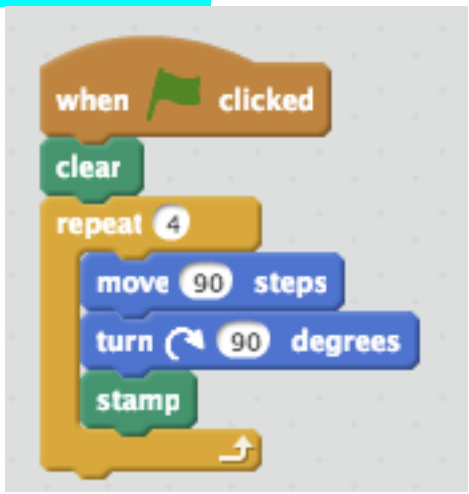
$34.95 Print Book and **FREE** Ebook
$27.95 Ebook (PDF, Mobi, and ePub)

Add to cart

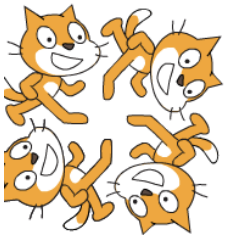
Optional 5.6: More tricks with the fascinating STAMP command

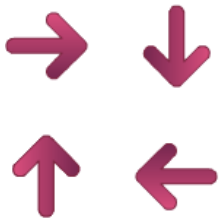
Let us talk about the STAMP command that a few of you have seen but up until now it has not served any real purpose. What STAMP does is to take the current costume at that moment and paints it onto screen at that position.

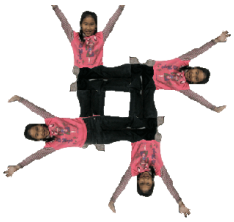
So consider this program called **Using the Stamp Command**
http://scratch.mit.edu/projects/14349753/



What does it do? Yes, it draws a square but it "stamps the costume" once at each corner.

If the Sprite is a cat, then I get this: 

While if the Sprite is an arrow, then I get this: 

Or if the cursor is a person, then I get this: 

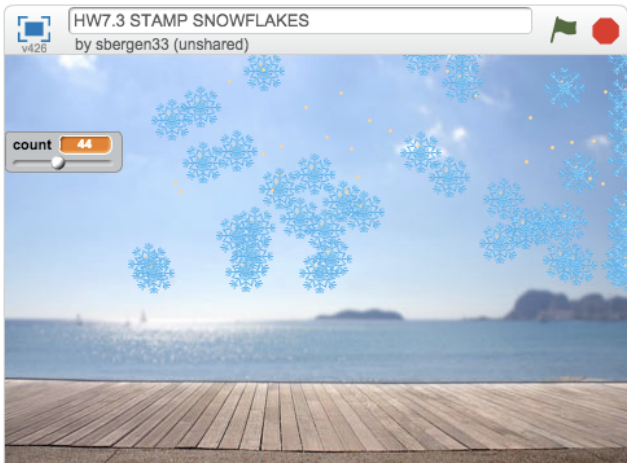
Got it?

just takes the current costume and "stamps it" on the screen. Note that STAMP is a PEN command (dark green).

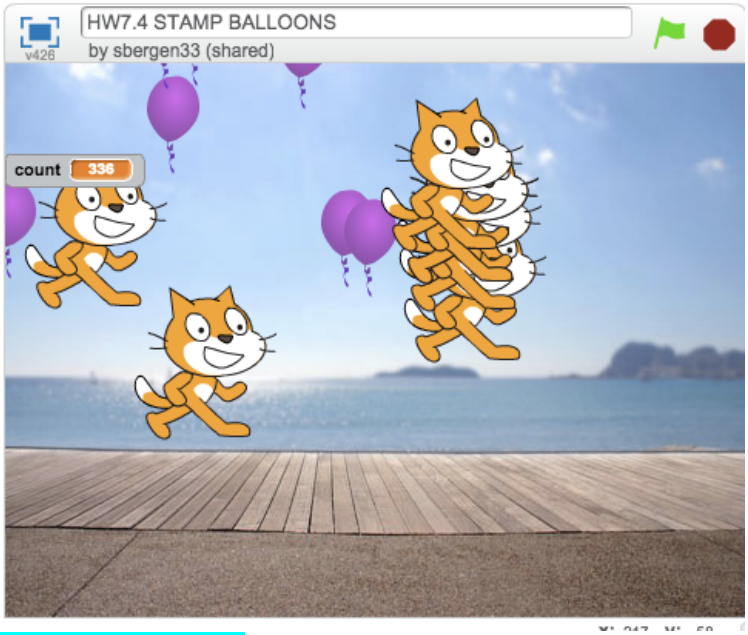
What you have to remember is that the STAMP command takes the current costume and then puts it onto the screen at that location as if it were a "stamper."

Optional HW 5.7 using the STAMP command

Use the snowflake costume found in the Holiday section of the standard Scratch library and write a program that randomly draws snowflakes on the screen using one of the backdrops from the Scratch library. Make sure you SHRINK the snowflake so it is smaller. You can shrink using the command or via vector mode in the graphics screen. Add a counter to keep track of how many snowflakes. When your program runs, it should look like this.



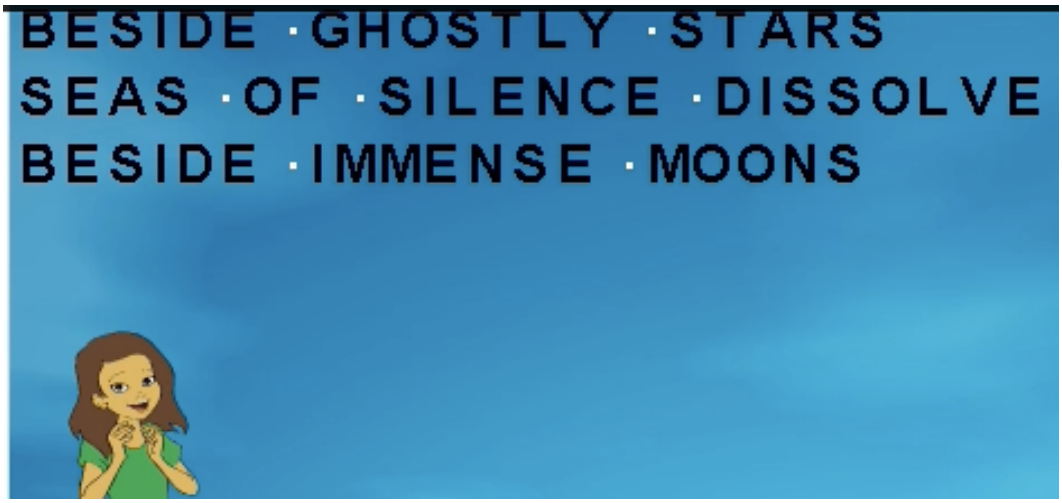
Use the balloon found in the Holiday section of the standard Scratch library and write a program that draws exactly 10 balloons on the screen. Now have your program switch back to the cat and allow the cat to "walk around the screen with the 4 direction arrows. Program the cat so that when it lands on a balloon (the color purple) it STAMPS itself over the balloon. When done, your program should look like this



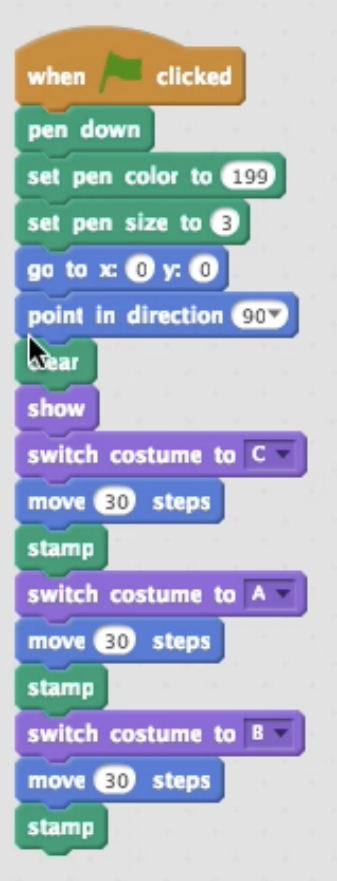
OPTIONAL but REALLY SPECIAL (at the very least, please read)
This leads us to the very clever program written by one Scratch programmer who uses STAMP to accomplish something I have been wanting to do for months -- write text onto the background! Here is the 25 minute video lesson that takes you

thoroughly through all the steps -- http://youtu.be/FpeB5O7UjWU -- it does take a lot of time to dissect and explain the code in a program.

I have used the Haiku program by one former student (Fall 2013) as the demo model to work with. I hope you will find it as cool as I do that you can now write on the screen and achieve this. The programmer who created this code is named Megadrats and he or she uses 26 costumes and the STAMP command so that the poem is *stamped* onto the screen!



Check out this intro program that I wrote; it is much more basic than the one from megadrats -- Putting A,B,C or D onto the Background -- but easier to understand. It lets you stamp words on the screen that contain the first 4 letters of the alphabet.

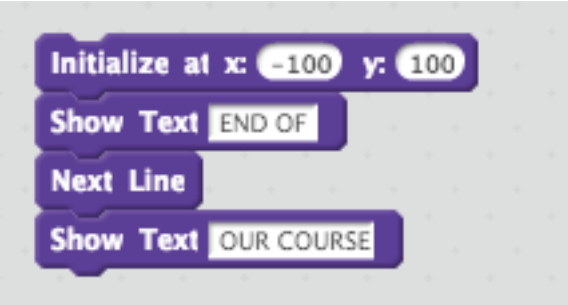


← the C costume is just the letter C which we then STAMP onto the screen!

← We then MOVE 30 pixels and then STAMP an A

Then we STAMP a B and so it looks like CAB to the user!

This is a fascinating topic that allows you to put large text on the screen. In other words, I can type END OF OUR COURSE into my script and I will get this message stamped onto the screen. For example, using this script



I achieved this

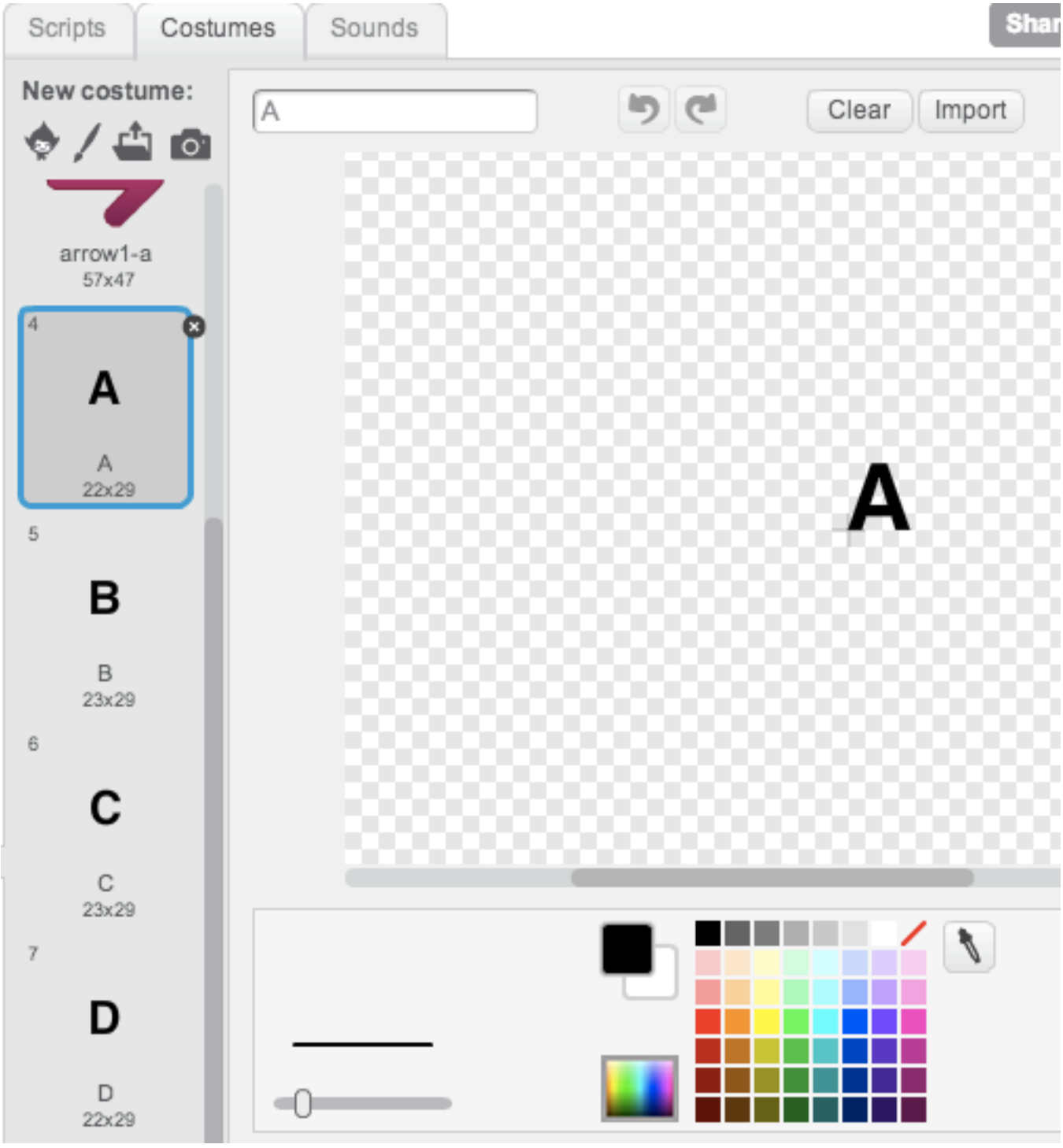
Interested? Read on and watch the videos!

We begin the STAMP command that a few of you have seen but up until now it has not served any real purpose. What STAMP does is to take the current costume at that moment and paints it onto screen at that position.

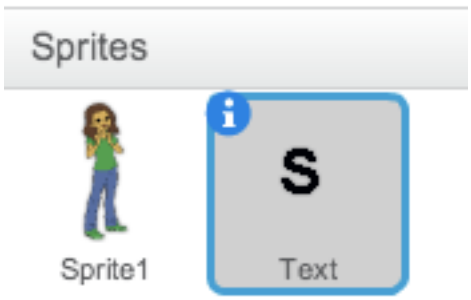
So consider this shared program called **Using the Stamp Command**
http://scratch.mit.edu/projects/14349753/
Got it?

just takes the current costume and "stamps it" on the screen.

Note that the costumes for this script are just single letters on each costume.
The beauty of this approach (thanks, Megadrats) is using the STAMP command!

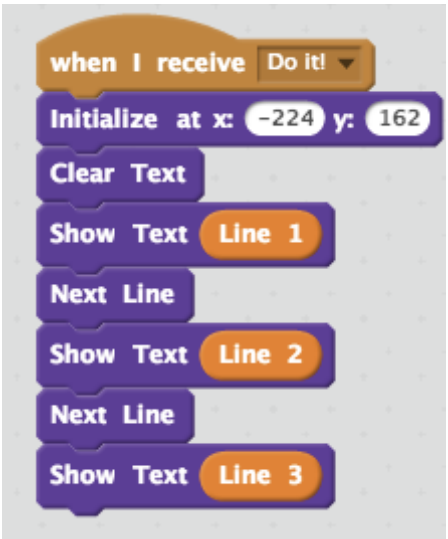


Here are the two Sprites used in this program:
Haiku Poem (Jessica) with Stamping Text by Megadrats



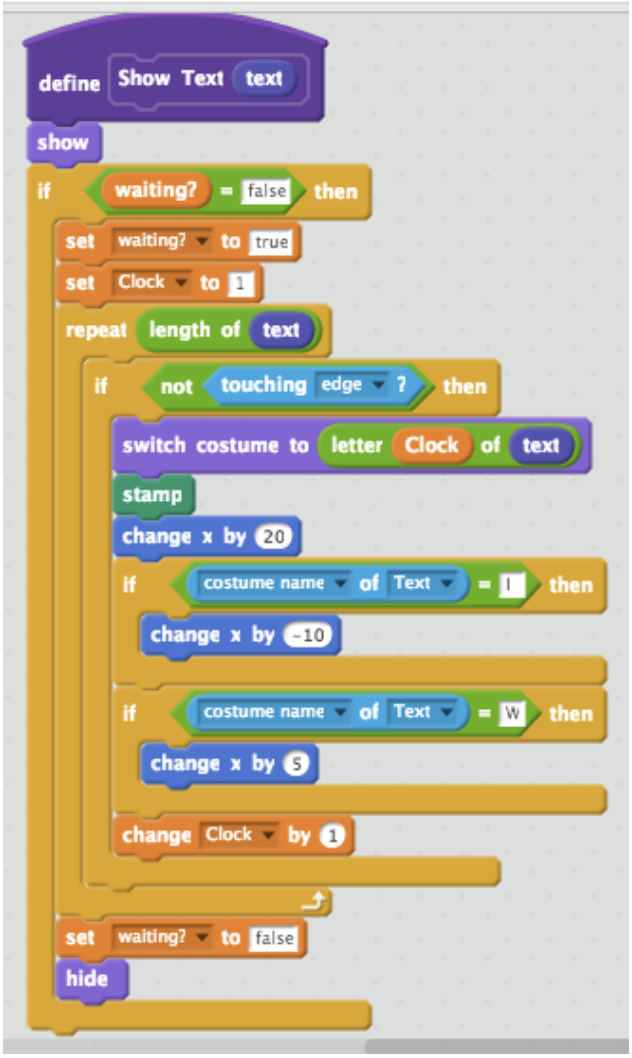
Sprite1 is Jessica's Haiku program but it ends with the BROADCAST command that I added called Do It!

This corresponds to a script in the "S" Sprite program below. This program uses the construction of Megadrats who makes the work available to all of us and encourages us to use it. The video will show you step by step exactly how the BLOCKS called NEXT LINE (go to next line) and SHOW TEXT (type this line onto the background) work.



The INITIALIZE block by Megadrats specifies the location to begin the typing.

Here is the most important script from Megadrafts. The variable TEXT is the stuff to be printed on the screen. Notice the purple command to SWITCH THE COSTUME. What is it switching to? If the word TEXT is CAB, then it switches to Costume C and stamps, then it switches to Costume A and stamps and so on.



I was unable to figure out the significance of the flag called WAITING? Maybe you can help me make sense of that one. However, the reason for the IF THEN statements has to do with the width of I (narrow letter) and W (wide letter) so Megadrats is adjusting the spacing accordingly. The program from Megadrats is now in my library as **Make Text in your Games! (Custom Blocks) remix**. Take it and put it in your library and in your backpack. As an optional assignment if you want one, it would be to use the Megadrats program as part of one of your programs to add large text.

INDEX of HIGHLIGHTS can be found on the last page

☑ HW 5.0A Watch the 20 minute interview with Scratch teacher Jeff from Houston (p1)
☑ HW 5.0B Send an email to your partner and me on something you found interesting from Jeff's talk (p1)
☑ 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. (p4)
☑ HW 5.1B OPTIONAL CHALLENGE: Have computer count # of vowels in the word. (p4)
☑ HW 5.1C OPTIONAL AND FAIRLY HARD CHALLENGE: Have computer share the individual letters in reverse order. (p4)
☑ HW 5.2A REQUIRED Video on scrolling is 3 minutes. (p5)
☑ HW 5.2B Enhance your 4.1 project to have a scrolling message (p7)
☑ HW 5.2C: Enhance your 4.1 project with a new sprite and costume with your name and STAMP it onto the 4.1 canvas (p9)
☑ HW 5.3A REQUIRED Video on LISTS is 7 minutes. (p10)
☑ HW 5.3B Enhance 4.1 with LISTS to produce one line of random poetry (p14)
☑ HW 5.4A 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. (p16)
☑ HW 5.4B OPTIONAL DIFFICULT CHALLENGE Modify the program above so that it results in a random sequence of the students each time you run the program
☑ HW 5.5A REQUIRED Watch the 5 minute VIDEO at that shows you how Jeff's student created the bouncing ball.
☑ HW 5.5B REQUIRED Send an email to your partner and to me in which you either
a) ask a question about the bouncing ball
or
b) provide an insight about the bouncing ball script (p17)
☑ HW 5.5C OPTIONAL: get the code on the bouncing ball by remixing from my library at http://scratch.mit.edu/projects/17552738/ and then implement it in your 4.1 program (p17)
☑ OPTIONAL 5.6 -- several articles about Coding (p19)
☑ OPTIONAL 5.6 -- more tricks with the STAMP command (p20)
☑ OPTIONAL 5.7 -- snowflakes with the STAMP command (p21)
☑ OPTIONAL -- writing text with the STAMP command by Megadrats (p23)