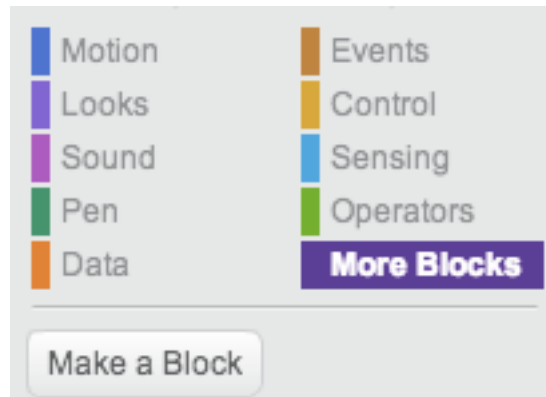


LESSON 6: Feb 17 - 24

Hello everyone! We are going to learn about BYOB this week.
This acronym stands for Build Your Own Blocks.

If you look at the bottom right of the Scratch color choices, you will see it say MORE BLOCKS in Purple.



When you make a new block, you are adding a new word to the Scratch language or defining a new procedure where some work is going to happen.

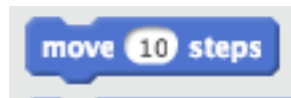
So I can make a block to build a square of any specified size.

In this [first required 4 min video](http://youtu.be/P05EMU9sqSk) at <http://youtu.be/P05EMU9sqSk> I am building a block called SQUARE and using it to achieve Spinning Squares.

Or I can make a block to create a new command. Or I can make a block that handles the question/answer dialogue with the student in a quiz program.

Once again, the BYOB feature of Scratch makes it into a real computer language.

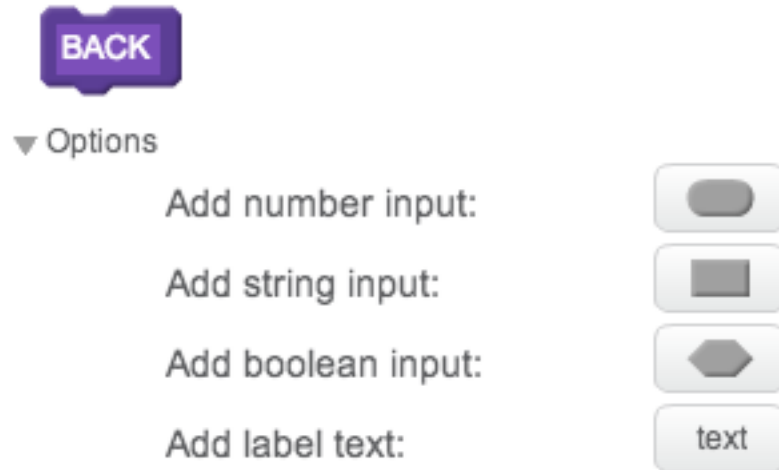
For example, we have the MOTION command



that we have used since our first lesson, but we do not have one that GOES BACKWARD 10 STEPS. So we use MORE BLOCKS to create one, even though we could put a negative number in front of “steps.” When it says MAKE A BLOCK, we choose BACK and then notice there are OPTIONS below.



When we click on OPTIONS, we get several choices, each with a purpose



The first 3 are called INPUTS and they allow us to "pass info" to the new block called BACK. For our purposes now, we want the first type (numeric) so that we can issue commands like BACK 30 or BACK 140. Yes, I know we can MOVE -30 or -140 but I want to pretend we cannot and want to create this NEW command called BACK. This optional 7 minute video at <http://youtu.be/4pNpcn2R1Ck> takes you through building the new COMMAND called BACK. It also has a section on the BACKPACK feature for those of you whom I have not yet shown via screenshare.

Or we might one day create a block called REVERSE that would take a string and reverse the order when we type the command REVERSE (any word).

Boolean refers to True or False and that is yet another input, e.g. we could say RESPOND (X>Y) where X>Y is either true or false.

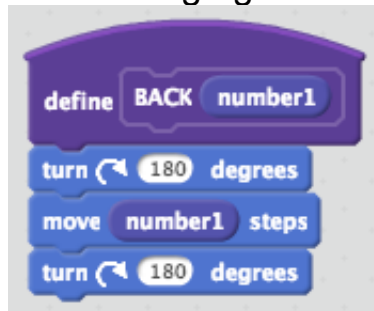
And LABEL TEXT allows you as programmer to put some comments or notes about this block for your future reference or that of someone else if you are on a programming team.

So for our BACK block, we choose NUMERIC and it would look like this:



If you prefer a different variable name to "number1" you can right mouse click on the BACK command under MAKE A BLOCK and change the variable name.

We now put in the script for this command -- turning 180 degrees, moving the specified number of steps and then turning again.

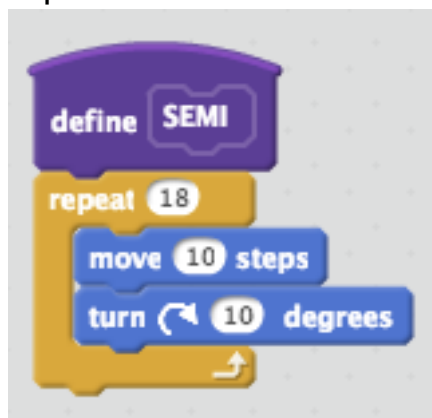


and we now can go back to our main program and issue commands like



and everything acts as if Scratch has a new command!

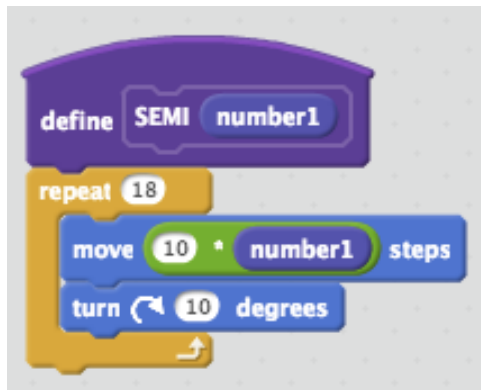
How about a block that draws a SEMICIRCLE which can be useful when drawing letters of the alphabet like "S" or "C" or "G"



which results in

Want to make the SEMI block fancier?

How about a numeric input that specifies whether the semicircle is bigger or smaller than one we just drew? We can EDIT the BLOCK called SEMI to have a numeric input and then use NUMBER1 as a multiplier for the number of steps.



so SEMI .5 is



and SEMI 1.5 is



The fact that both BACK and SEMI use the variable NUMBER1 which was made by the system is not a problem, since it is a "local variable" used internally when the programmer issues the command.

For another example of a BLOCK with a STRING INPUT, consider how useful it would be to have a BLOCK called MUSIC so that you could type in the first few notes of a song and it would play it!

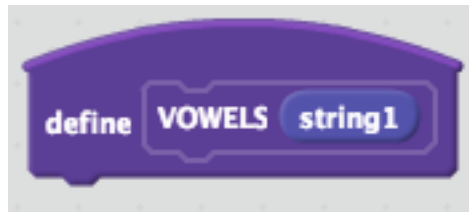


If you had this block working (not impossible to write) then you could put in your program

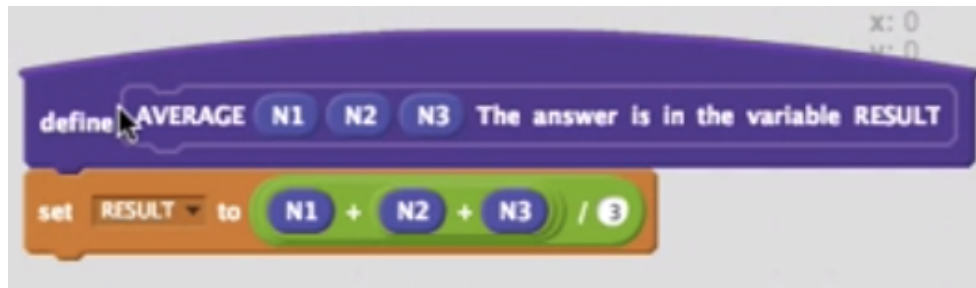


and it would play the first 7 notes of *Doe A Deer*, one of the few songs I remember from my piano playing days as a teenager ;-)

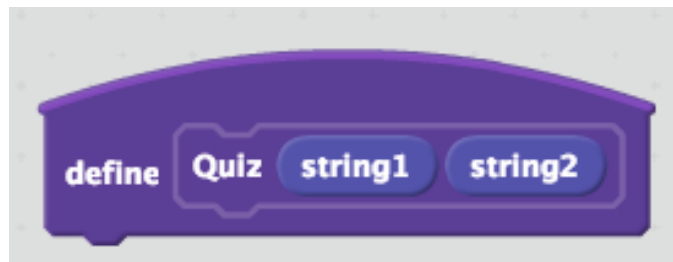
Or you could have a VOWELS block that would count the number of vowels in a string.



Or you could have an AVERAGE BLOCK that automatically finds the average of several numbers. Here is a [required 5.5. minute video](http://youtu.be/I4Qzw3j_TqE) on building the block called AVERAGE -- http://youtu.be/I4Qzw3j_TqE -- using three numeric parameters. Hopefully this video will solidify your awareness of BYOB.

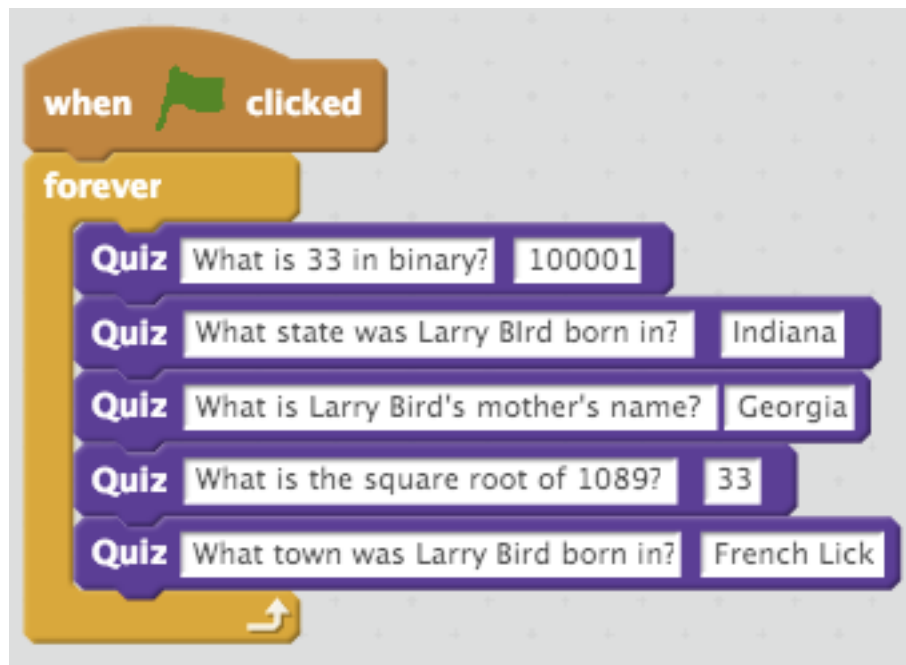


Here is one last example that you will weave into your [6.1 HW](#) so it is the most important example. I am going to use the BYOB feature to create a quiz game that will ask a bunch of questions of the user.



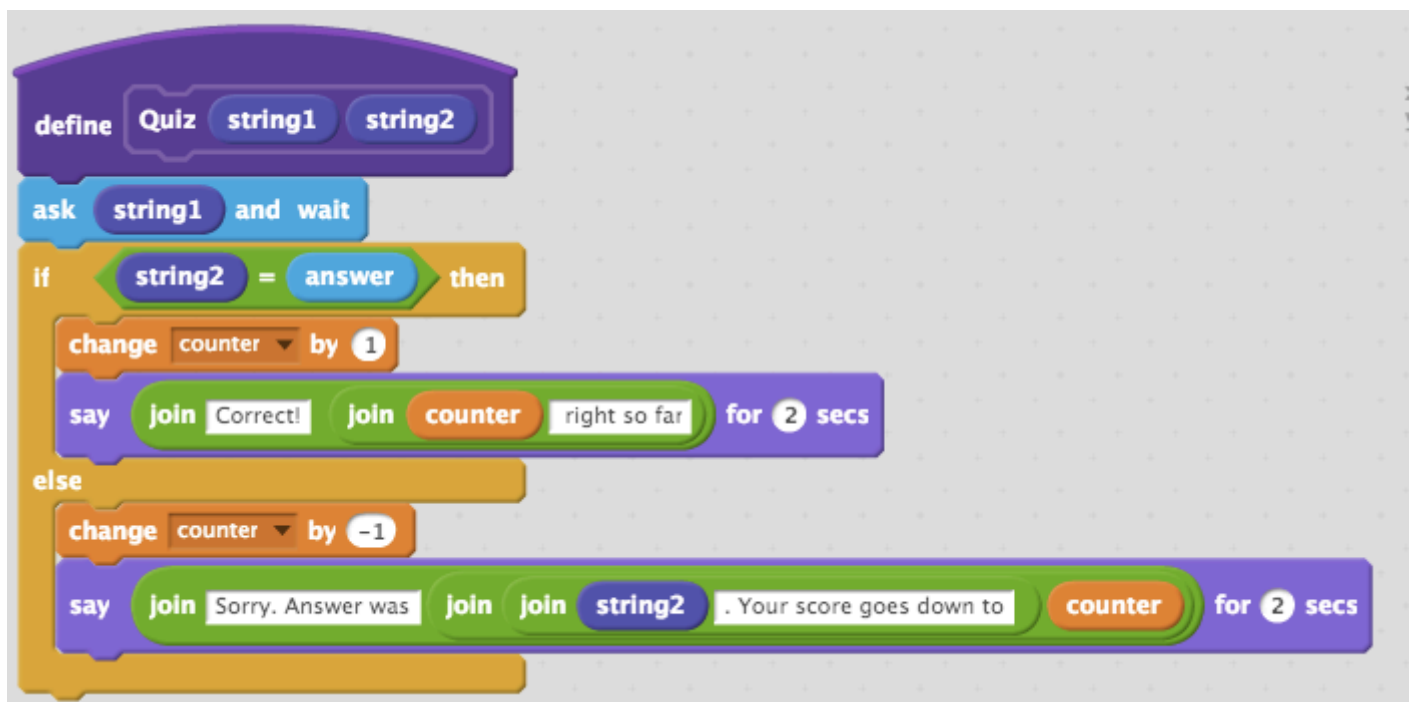
The two variables STRING1 and STRING2 are called PARAMETERS. We are passing these parameters from the main program to the BLOCK and saying "hey Block, please take charge of asking the question called STRING1 to the user and evaluate what he or she responds based on the answer I am giving you in STRING2."

One more point of advocacy regarding BLOCKS. It allows you as the programmer to keep your code shorter and more efficient. The code I have written for my quiz program would have to be replicated 5 times in my main program if I did not have blocks. Just imagine if you were charged \$1 per line of code! Efficiency in coding is a plus.



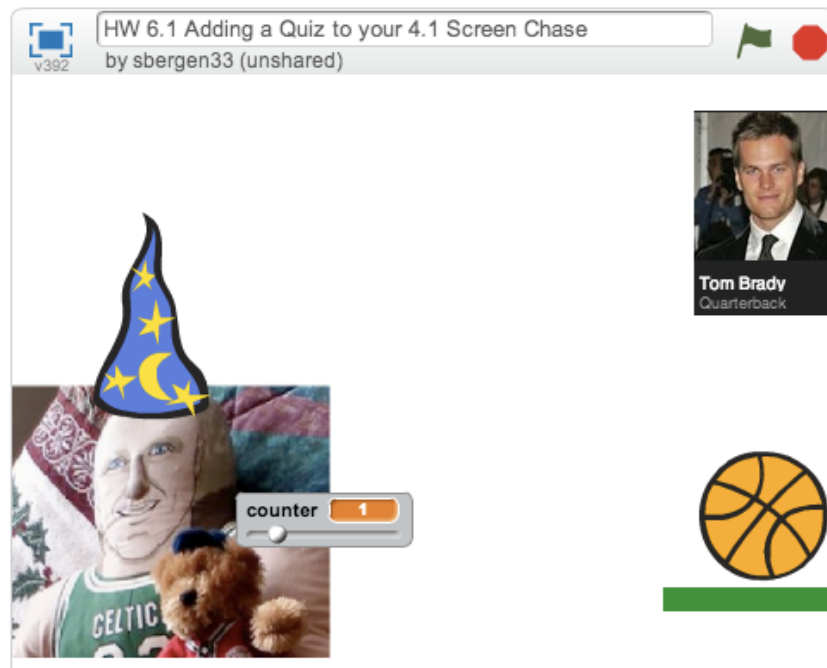
Instead, this is my main program where I have my five questions. The block called QUIZ takes care of all the work. In other computer languages, what we call blocks in Scratch might be called SUBROUTINES or PROCEDURES.

Here are the details of what goes on within the block called QUIZ.



Notice how the computer asks the question called STRING1 of the user and then evaluates whether STRING2 match ANSWER, either adding or subtracting a point from the user's score via the variable COUNTER.

Now using the COUNTER, you can do what you want on your screen. I have chosen to program a basketball to start bouncing and eventually reach Tom Brady on this screen.



The script for the ball is based on the script that one of Jeff's students (from Houston) used. Even though I can't say I understand it 100%, I can now use it effectively and want you to do the same. I have programmed the use of the COUNTER to make the ball bounce a little higher each time you get one quiz question correct. I want you to do something similar in the context of your program, bouncing a fish, a volleyball or an object from Downton Abbey.

Last week, I shared an optional video on the bouncing ball that you may not have watched due to time constraints. This week, it is **required** since you will be using it in **HW 6.1** – <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 community. The central idea of the descent of the ball is that it "falls down" a little bit more each nanosecond, perhaps 2 pixels in one nanosecond, then 3 pixels in the next nanosecond, then 4 and so on, creating the illusion of gravity. **Please work at understanding this!**

Finally, let us learn about BROADCASTING a MESSAGE -- another new topic we have never discussed before but I have shown some of you via screenshare.

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



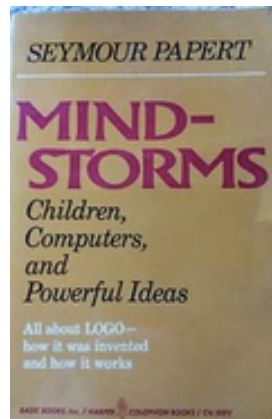
This idea is that this command gives us a method for Sprite #1 to interact with Sprite #2. For example, I could take my Tom Brady program and change it so that when Tom goes to one of the corners, he calls one of the players and "broadcasts" a message. This way that player and ONLY that player chases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

Very, very powerful stuff and fun stuff which makes me think yet again of Seymour Papert's Logo book called *Mindstorms: Children, Computers and Powerful Ideas*.



I may have told you this before, but reading this book back in 1982 -- when I was working as a programmer for a company called The Williamson Group in Cambridge MA -- was one of the things that made me quit my job and start the teacher training company first called ***The Teaching Company*** that then became ***Summercore***.

In my program called HW 6.1 Adding a Quiz to your 4.1 Screen Chase, you will see all of these elements. You can remix the program and use your backpack and move any scripts you want into your own 4.1 Screen Chase which will now become 6.1.

So your HW 6.1 is an enhancement to your 4.1 project.

When done, I want you to have your 6.1 include

- a bouncing ball or bouncing anything
- a quiz program that keeps track of points using BLOCKS
- the reward of seeing your 4.1 Screen Chase animation happen after a certain number of points are scored using BROADCAST
- a scrolling message using the technique shown in Lesson 5

Please send an email to your partner and me when you are ready for us to see your 6.1 creation.

On the next two pages -- to help you -- I am including the scripts of the bouncing ball and wizards hat from my 6.1. Please make this program your own by doing your own thing with these techniques in your own creative way!



Here is the script of the Bouncing Ball Sprite1 that animates more and more based on the counter.



THESE SCRIPTS ARE FOR THE BOUNCING BALL WHICH HAS A GREEN PLATFORM THAT IT JUMPS FROM. YOU NEED THAT BUT CAN CHOOSE YOUR COLOR.

THIS IS THE MAGIC SCRIPT FOUND BY JEFF'S STUDENT THAT MAKES THE BALL "LAND" ON THE GREEN PLATFORM.

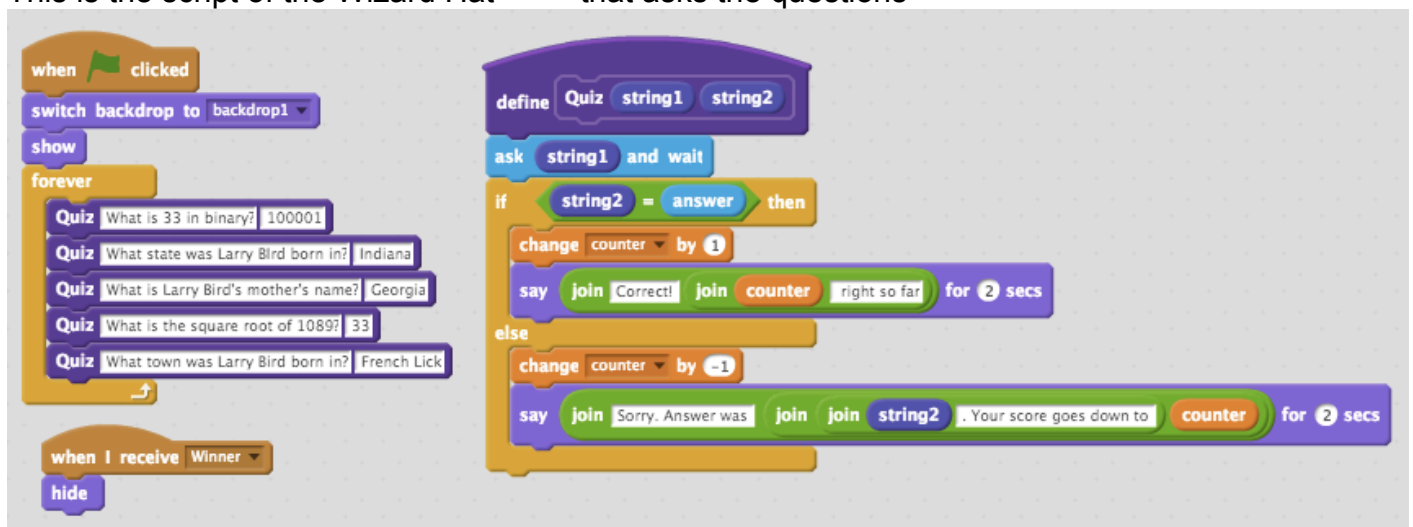
THE 4 IS THE JUMP FACTOR THAT MAKES THE BALL JUMP HIGHER EACH TIME COUNTER INCREASES. YOU CAN ALTER THE 4 IF YOU WANT.

THIS IS THE KEY SCRIPT THAT GOES TO THE TOM BRADY 4.1 ANIMATION SCREEN ONCE THE USER GETS OVER 5 POINTS.

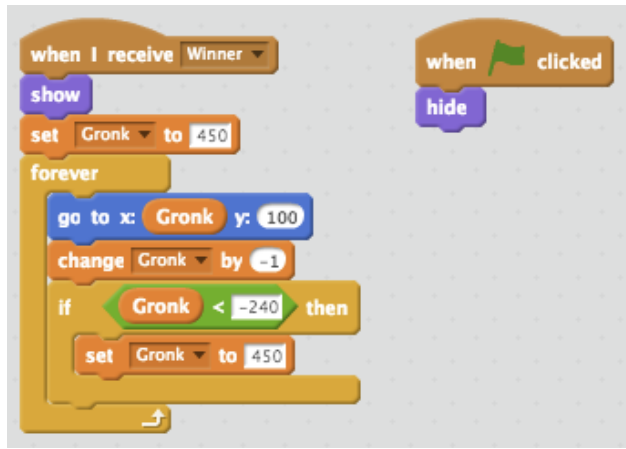
Try your best to understand the script on the left. Pretend that COUNTER = 2. This makes JUMP start out as 10. So in the bottom REPEAT loop, it changes Y by 9, then 8 then 7 until 1 which is why the ball GOES UP. Then JUMP is -1 so it descends 1 pixel in the first nanosecond, then descends 2 pixels in the second nanosecond, then 3, then 4 which means it speeds up until it touches "and lands" on the green. I am in awe of whomever was this clever to first program this!



This is the script of the Wizard Hat Wizard Hat that asks the questions



Here is the script of the SCROLLING TEXT



My Gronk is back (in 2014-2015) and
you're gonna be in trouble

PUSH 1-4 TO MAKE TOM SCRAMBLE

Finally, HW 6.2 is to communicate with your partner by phone, email, skype or google hangout and to jointly pick one of the sixth grade projects by Jeff's students in Houston TX. The two of you will end up teaching the rest of the class about this project the week of March 3-7 when we hit our last week of our course.

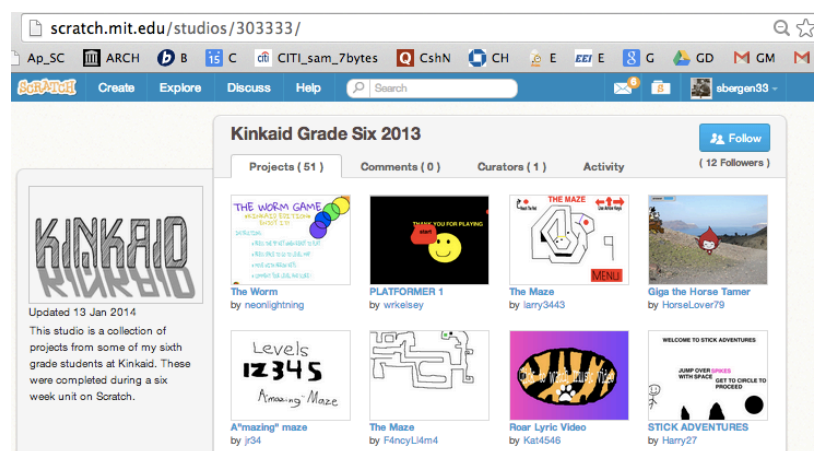
The projects are at <http://scratch.mit.edu/studios/303333/> but also on our January webpage direct under the list of names and partners.

You and your partner should pick one that

a) you like (I think this will be easy)

b) can decipher the code and make sense of how the program works (not so easy)

You and your partner will be able to choose amongst various options for turning in your work on March 3rd. This assignment is called 8.1 and is due on 3/3.

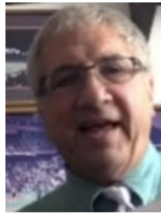




summercore

Have a good week. Two more lessons to go everyone after today -- Spring is coming! Our last lesson will be Monday 3/3 and our course officially ends 3/10 which I call JH Day. JH Day on 3/10 always takes place exactly four days before PI Day on 3/14. My guess is that none of you know what JH Day is but if you've made it through this winter, you should celebrate on 3/10.

Steve



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



p.s. for any compulsive types, here is an OPTIONAL bonus problem for week 6

HW 6.3 Optional and Challenging

Create a program that has two lists, one for SUIT and one for CARD. The four items of the SUIT list are Spades, Hearts, Diamonds, Clubs. You then create a BLOCK called RANDOMCARD which picks a random card. Now use this block to pick 5 random cards making sure there are no duplicates. That is the toughest part!