

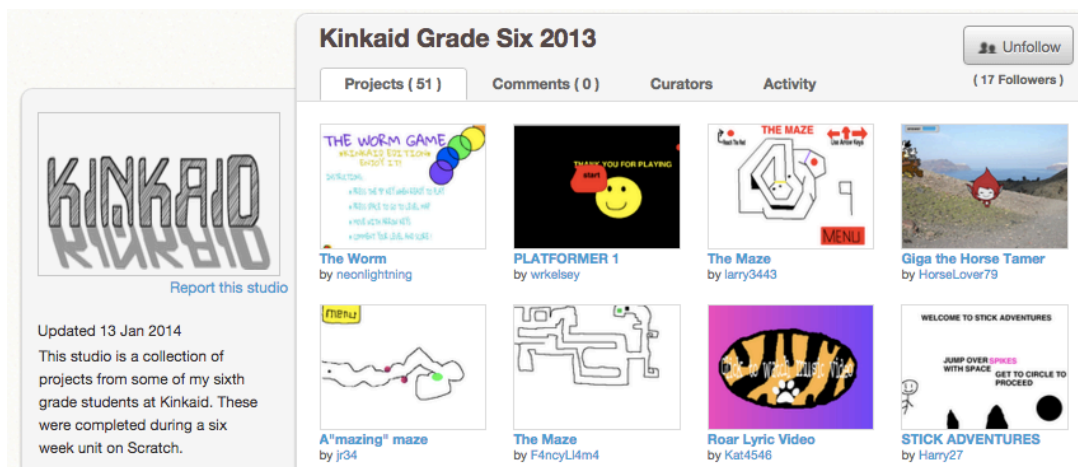
Monday 3/7/16 -- this is the final lesson of our Scratch journey! We will learn about binary numbers and then the key assignment involves making a quiz using the purple block command. I will introduce you to the 3-5-7 Stick Game and hope 1 of you win \$33 from me. All homework is due by Friday 3/18/16 our official last day!

We begin with a short fun assignment called **FIND YOUR FAVORITE** and **EXPLAIN WHY**.

Please go to the Scratch Studio library of sixth graders in Houston TX at a school called The Kinkaid School. The URL is

<https://scratch.mit.edu/studios/303333> and there are 51 Scratch projects by students in Mr. Diedrich's class. I have known Mr. Diedrich for over ten years and love the way his students have created neat Scratch projects.

Exercise 4.1 is to find one program that you like and remix it so it is in your library. Now you can look at the code and check it out more thoroughly even changing a few things if you like. Make sure you know the exact name. Write a sentence on our class webpage that says "I chose \_\_\_\_\_ and think it is cool because \_\_\_\_\_."



## Next let us talk about binary numbers and my 3-5-7 stick game!

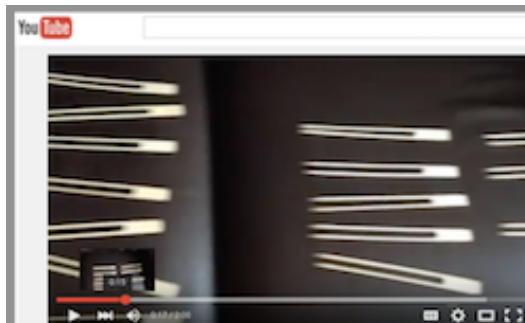
Binary is the language of computers. I have an amazing game to play with you and we will then learn to program it in Scratch. The game is called the 3-5-7 game. You can play it with sticks, with rocks, on paper or on a computer screen.

The rules of the 3-5-7 are simple.

When it is your move, you must take at least 1 stick from 1 row. You can take more than 1 stick but only from 1 row. You can even take all the sticks from 1 row when it is your move. That is it for rules. Whoever takes the last stick loses!

I hereby challenge each and every one of you to play this game with me via a phone call with screen share. When we play, you can choose to go first or you can make me go first. The first student who beats me wins a \$33 gift certificate from Amazon from me and will go into my Scratch Hall of Fame. I am serious!

Here is a short required **INSTRUCTIONAL VIDEO** on playing the game with actual sticks (not on the computer) -- <https://youtu.be/Nvkk4E2EKGy> with short name of [tinyurl.com/lbsticks](http://tinyurl.com/lbsticks)



Why will you probably not win? Because the game is related to binary numbers sometimes called BASE 2. I know binary numbers better than most of you and that is why I probably will beat you.



If you want to play this game, you can use sticks, rocks or paper/pencil. If you want to play on a computer with a friend or relative, use <https://scratch.mit.edu/projects/90654709/> or short version <http://tinyurl.com/beatmrb>

Now I will teach you about binary numbers and maybe that will help you beat me! With binary numbers, the places from **right to left** are 1,2,4,8 and so on.

So the number 6 in binary is 110 since we have a 1 in the twos place and a 1 in the fours place.

Here how to count from 1 to 5 in binary

1 = 001

2 = 010

3 = 011

4 = 100

5 = 101 (if Larry Bird were on a binary basketball team, his teammates would give him a HIGH 101 every time he scored)

Study these details. The picture on the right shows what Larry Bird's uniform number would be if he played on a binary basketball team!

Here are the numbers from 1 to 14 written in **Base**

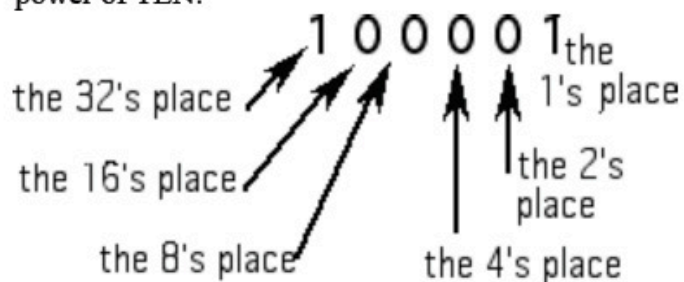
**Two:**

0001 = 1	1000 = 8
0010 = 2	1001 = 9
0011 = 3	1010 = 10
0100 = 4	1011 = 11
0101 = 5	1100 = 12
0110 = 6	1101 = 13
0111 = 7	1110 = 14

Here is Larry Bird's uniform number (33) written as a BASE TWO number:

**100001**

Each position stands for a power of TWO instead of a power of TEN:



And now see if you can convert A, B and C from binary to decimal.

The answers are on our class web page: <http://tinyurl.com/online4kids>



Question #33: Convert (A) the 10100 cookie to decimal \_\_\_\_  
how about (B) the 11110 cookie? \_\_\_\_ and (C) the 10111 cookie? \_\_\_\_



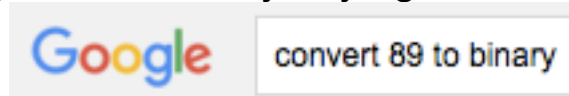
**Example #1: let us change 89 from decimal to binary**

The typical strategy is to write down 8 places, labeling them from 1 on the right to 128 on the left:

<u>128</u>	<u>64</u>	<u>32</u>	<u>16</u>	<u>8</u>	<u>4</u>	<u>2</u>	<u>1</u>
------------	-----------	-----------	-----------	----------	----------	----------	----------

We now distribute the 89 (as if we playing the role of the banker in monopoly and have to pay \$89 but can use only \$128 bills, \$64 bills, \$32 bills, etc. and just ONE of each) We first put a 1 in the 64's place which leaves 89-64 or 25 left to distribute. We next put a 1 in the 16's place, leaving 9 dollars and so on. **We get an answer of 01011001.**

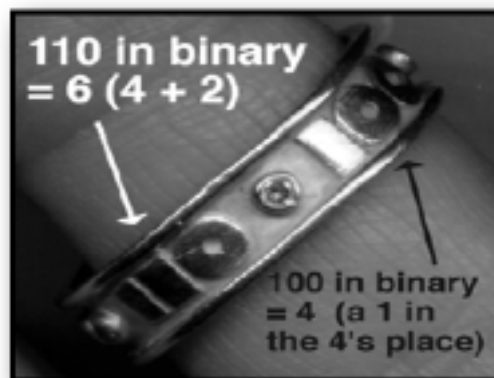
Tip: You can tell google to do this by saying **Convert 89 to binary**

**Example #2: change 01000101 from binary to decimal**

We begin with our 8 positions

<u>128</u>	<u>64</u>	<u>32</u>	<u>16</u>	<u>8</u>	<u>4</u>	<u>2</u>	<u>1</u>
------------	-----------	-----------	-----------	----------	----------	----------	----------

and now translate our 01000101 number into 1 in the 64 place PLUS 1 in the 4 place PLUS 1 in the 1's place, i.e. with our "monopoly money analogy" \$64 + \$4 + \$1. **We get an answer of 69.**



**Example #3:** interpret the months and date on this wedding ring on the right. The 110 represents 6 (June) since there are 1's in the 2's place and the 4's place. The 100 represents 4 since there is a 1 in the 4's place and 0's in the 1's place and the 2's place. We get an answer of June 4th.

Please watch this required 7 minute **INSTRUCTIONAL VIDEO** and get smarter at binary numbers so you can easily count from 1 to 10 in binary.

<http://tinyurl.com/learnbinary33> or <https://youtu.be/1Nre0wtWuU0>



Question #34: Without looking at the previous pages, count from 1 to 5 in binary? \_\_\_\_

**THERE ARE 10 TYPES OF  
PEOPLE IN THE WORLD, THOSE  
WHO UNDERSTAND BINARY  
AND THOSE WHO DON'T.....**

from: [http://imgfave-herokuapp-com.global.ssl.fastly.net/image\\_cache/1336498406725920.jpg](http://imgfave-herokuapp-com.global.ssl.fastly.net/image_cache/1336498406725920.jpg)



Question #35: The picture above can be found on dozens of web sites and web pages. Please explain what this means in one sentence in a way that even the Larry Bird doll could understand:

\_\_\_\_\_

From Mr. Bergen

sbergen33@gmail.com or 781-953-9699



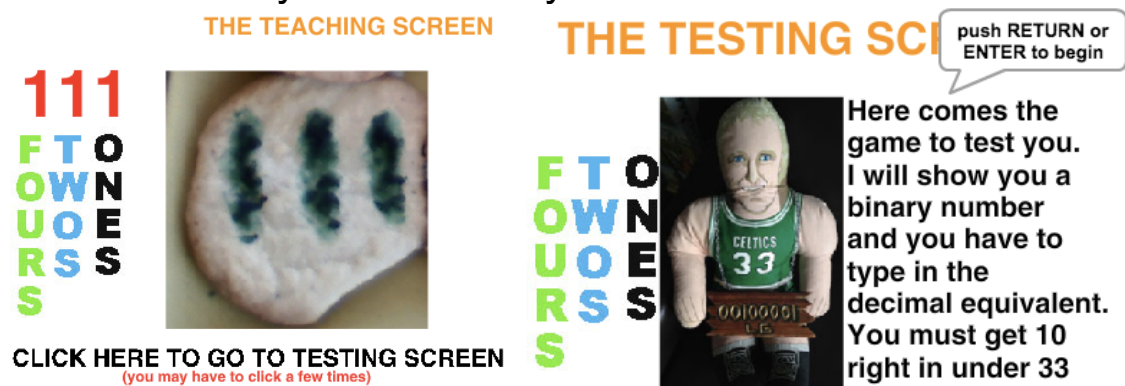
Scratch 4 Kids • Lesson 4 • p 7

Our webpage is <http://tinyurl.com/online4kids2>

Please go to my Scratch game on binary called Binary Numbers: Teaching and Testing.

It is located at <https://scratch.mit.edu/projects/89910084>

with short name of [tinyurl.com/binaryscratch33](http://tinyurl.com/binaryscratch33)



Exercise 4.2 is to play this game until you win. You have to get 10 questions right in 33 seconds. Then you win and find out how many seconds were left on the clock. The bigger the number the better. Put this number onto our class webpage where it says Exercise 4.2 Mr. Bergen's Binary Game.

I promise you that the 3-5-7 game is related to binary numbers, but I am not prepared to tell you how. The only hint I will give you is to write out the three numbers this way in binary:

3 in binary is 11

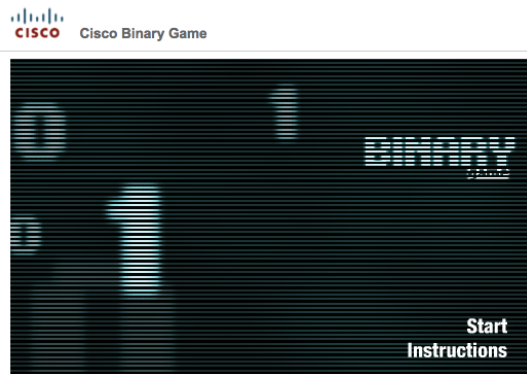
5 in binary is 101

7 in binary is 111

If I told you more, I might lose \$33. We will see if you figure it out by Dec 23rd and if you can beat me. You get 3 shots and can go first or you can have me go first.

Now that you are getting better at binary, play this famous Cisco binary game (much better than mine) at <http://tinyurl.com/binarygame33> and see what you can get as a high score. You can play as many times as you want. Put your high score on our class webpage.

**Exercise 4.3: Put your high score for this Cisco Binary Game on our class webpage where it says Exercise 4.3 Binary Game.**



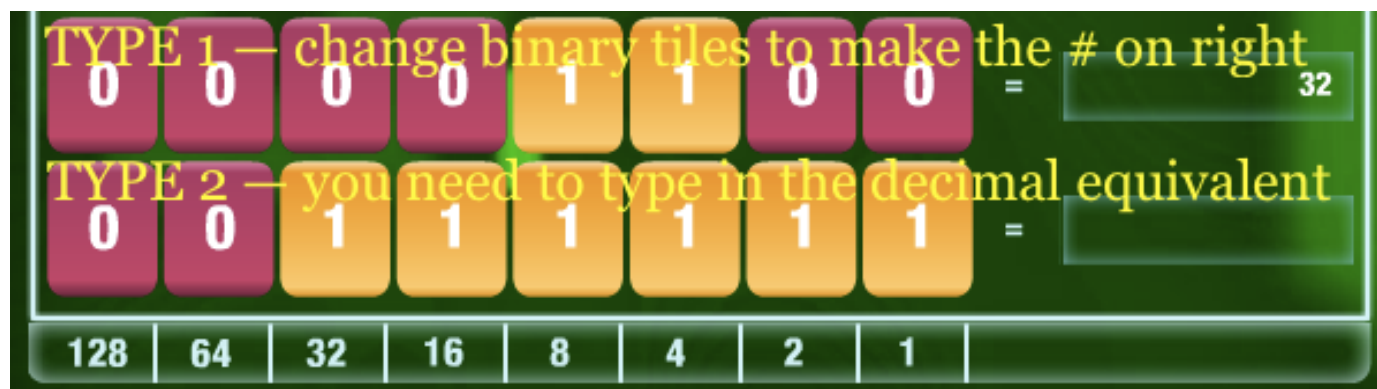
Be honest please! A friend once said that

**“Character is what you do when no one is watching.”**

The game will give you problems of 2 types.

Type 1 -- you are given a number such as 32 and need to change the binary tiles to equal 32.

Type 2 -- when there is no number on the right side, you need to enter on the right the value of the yellow binary tiles. In row 2 below, this answer would be 63. They give you a calculator to enter the number.



Good luck! Let us see who gets the highest score!

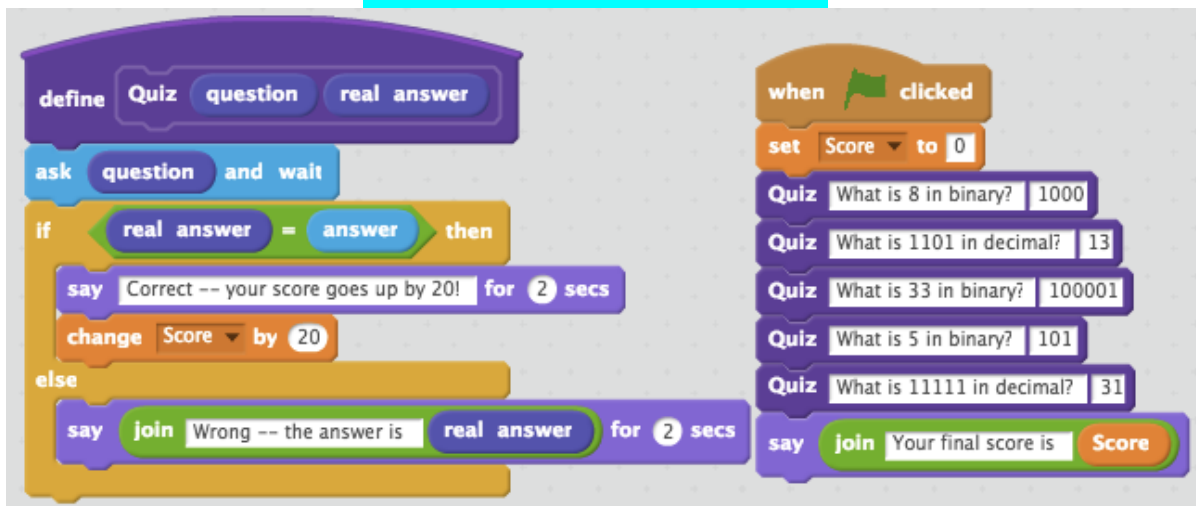


The big project for this week is to create a 5 question quiz on binary.

**Exercise 4.4: Create a 5-question quiz on binary numbers using blocks and using some of the binary cookies I am giving you in the TEMPLATE. Please be as creative and clever as you want! Read carefully the specs for JV, Varsity or SuperVarsity below. This is your big final project!**

The first **INSTRUCTIONAL VIDEO** on 4.4 is 7 minutes and is at <https://youtu.be/MISvheLjBkk> with short name of tinyurl.com/jvlesson4 and emphasizes purple blocks!


The second **INSTRUCTIONAL VIDEO** on 4.4 is 9 minutes and is at <https://youtu.be/QFXIbk5RvNk> and helps to explain PURPLE BLOCKS vs ORANGE VARIABLES. **This video is OPTIONAL.**



My TEMPLATE for JV, Varsity and SuperVarsity is at <https://scratch.mit.edu/projects/91008207> and contains the pictures of the **33 binary cookies** that I baked for my students one year!

Please start with this file and REMIX. Now be creative!

Beware of these reminders when creating your block named Quiz

REMINDER 1: You must click on OPTIONS when you see  and have two STRING inputs that you will then rename the way you want. The word STRING refers to text (words not numbers).

REMINDER 2: the purple variables get **dragged** to their locations in the code below.

JV SPECS: File name is JV Exercise 4.4J Binary Quiz JV by ABC and you must share it of course.

You need to have a counter that adds points when you get each question right. You need to use PURPLE BLOCKS deleting all of my scripts before you start! You must have 5 different things (out of the 8 below) happen as the USER gets 1 right or 2 right or 3 right or 4 right or 5 right.

1. costume animation from lesson 2 page 26 (e.g. dancing ballerina)
2. a new sprite moving from lesson 2 page 27 (e.g. a basketball moving)
3. making a word or phrase animate on the screen (lesson 2 pp 27-28)
4. changing the background (lesson 2, p. 28)
5. adding sounds (lesson 2, p. 29)
6. making 2 Sprites crash (lesson 2, p. 30)
7. making a Sprites grow or shrink (lesson 2, p. 31)
8. stamping snowflakes or other objects (lesson 2, p. 32)

### SPECS FOR VARSITY

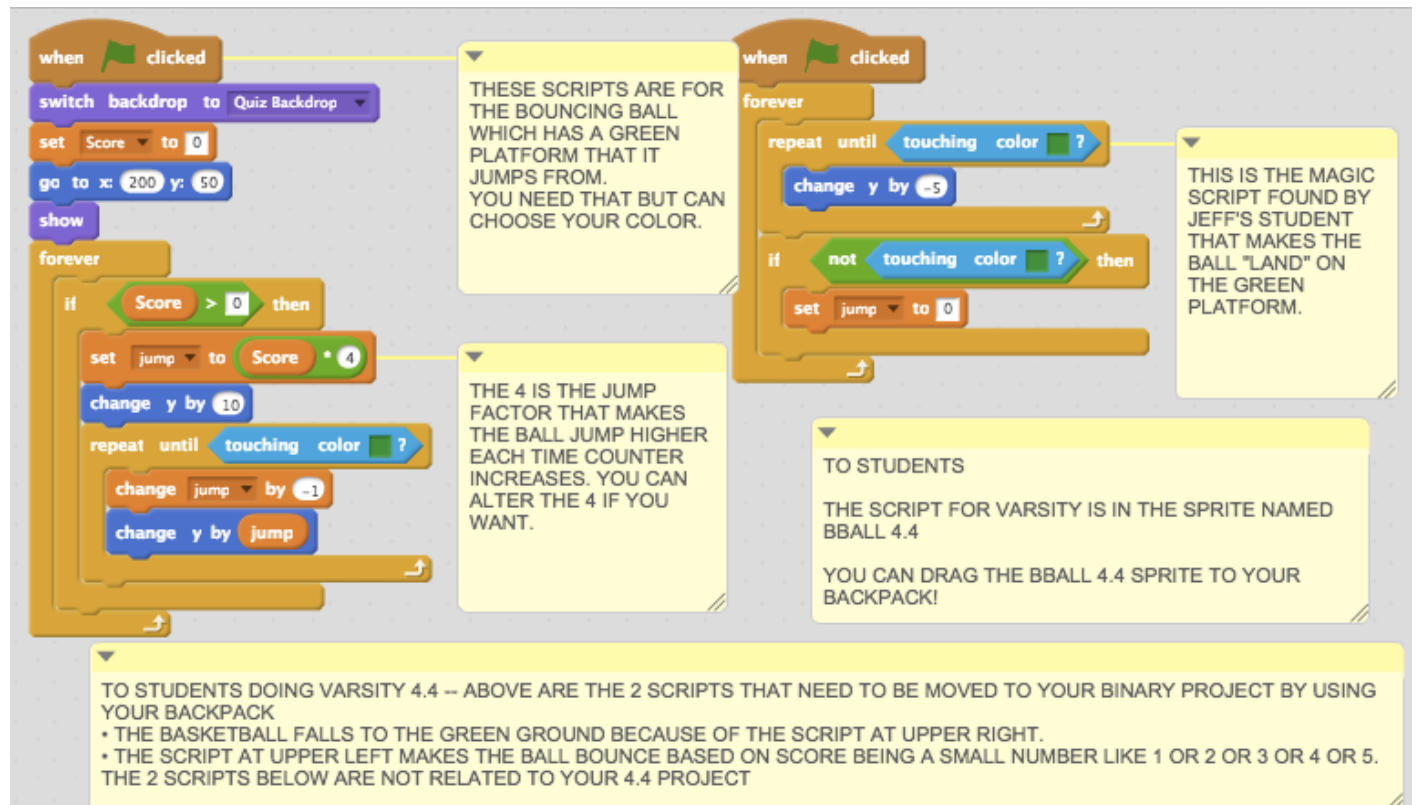
You must do everything required for JV plus add the bouncing ball script I learned from a Kinkaid 6th grader using your backpack to move the SPRITE BBALL 4.4 from my sample at <https://scratch.mit.edu/projects/91364549> to your program. The File name is Varsity Exercise 4.4V Binary Quiz by ABC and share it of course.

THE SCRIPT FOR VARSITY IS IN THE SPRITE NAMED BBALL 4.4



YOU CAN DRAG THE BBALL 4.4 SPRITE TO YOUR BACKPACK!  
(Moving the SPRITE is easier than moving each of the two scripts)

Here are the scripts for BBall 4.4 - - once you drag this Sprite to your backpack and then to your binary program, you need only the top two.



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.

TO STUDENTS

THE SCRIPT FOR VARSITY IS IN THE SPRITE NAMED BBALL 4.4

YOU CAN DRAG THE BBALL 4.4 SPRITE TO YOUR BACKPACK!

TO STUDENTS DOING VARSITY 4.4 -- ABOVE ARE THE 2 SCRIPTS THAT NEED TO BE MOVED TO YOUR BINARY PROJECT BY USING YOUR BACKPACK

- THE BASKETBALL FALLS TO THE GREEN GROUND BECAUSE OF THE SCRIPT AT UPPER RIGHT.
- THE SCRIPT AT UPPER LEFT MAKES THE BALL BOUNCE BASED ON SCORE BEING A SMALL NUMBER LIKE 1 OR 2 OR 3 OR 4 OR 5.

THE 2 SCRIPTS BELOW ARE NOT RELATED TO YOUR 4.4 PROJECT

SuperVarsity Specs: In addition to the JV and Varsity specs, you need to have the computer generate random numbers from 1 to 33 and then figure out the binary equivalent. File name is Exercise 4.4 SV Binary Quiz by ABC and you must share it of course.

Comments: this is extremely hard and if you succeed, you become the Tom Brady of Scratch programming. You will need to use the IF command to figure out whether the number is bigger than 32, bigger than 16, bigger than 8 and so on. Or maybe you can come up with another algorithm that takes a random decimal under 33 and converts it to a binary number. Good luck!

*If you work on the SuperVarsity challenge and get frustrated, let me know and I will make you an **INSTRUCTIONAL VIDEO**.*

## Next, I need everyone's help in creating the 3-5-7 Stick Game as a program on Scratch.

First we need to create the board. I am pretending that I started doing this and then ran out of time so you can participate in the creation.

Exercise 4.5: Please go to <https://scratch.mit.edu/projects/90384880> or short name [tinyurl.com/displaysticks32](http://tinyurl.com/displaysticks32) and REMIX A COPY of my program so it displays 3 rows of sticks and not just 1 row. After you REMIX, please rename the file as Exercise 4.5 Display Stick Game ABC (using your initials instead of ABC). There is no JV or Varsity for this one.

Here is the solution if you get stumped: <https://scratch.mit.edu/projects/90339254> or short name [tinyurl.com/displaysticks33](http://tinyurl.com/displaysticks33)

- PLEASE FIX THIS PROGRAM THAT I STARTED SO IT WORKS WITH 3 ROWS.
- THIS PROGRAM IS SUPPOSED TO DISPLAY THE 3-5-7 BOARD VIA A BLOCK COMMAND THAT I MADE CALLED DISPLAY BOARD.
- THE IF ROW1>0 COMMAND IS THERE SINCE THE ROW MIGHT BE EMPTY.
- THE VARIABLE HOW FAR GIVES THE X-POSITION FOR EACH STICK
- THE FIRST STICK GOES AT (30,100)
- THE SECOND STICK GOES AT (60,100)
- THE THIRD STICK GOES AT (90,100) AND SO ON.
- THE STAMP COMMAND takes the COSTUME (RED RECTANGLE STICK) AND PAINTS IT ON THE SCREEN AT THAT SPOT.
- YOUR JOB IS MAKE THIS PROGRAM WORK FOR ALL 3 ROWS, PICKING A Y-HEIGHT THAT FITS ON THE SCREEN. REMINDER TO DUPLICATE THE CODE!



Okay we now have the board displayed, but how do 2 players play against each other?

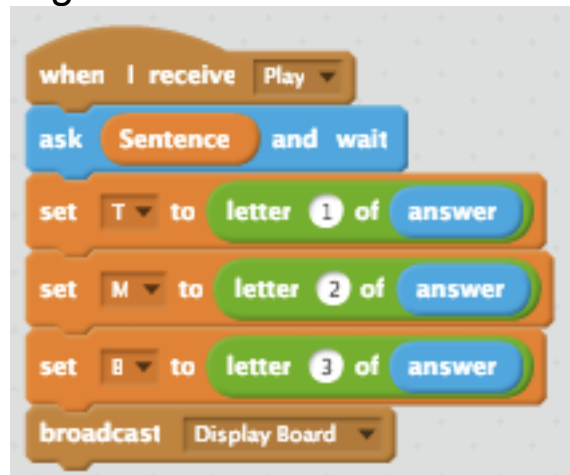
I created 2 costumes -- one for player 1 and one for player 2, typing onto the graphics screen.



I created this code to be the main program switching costumes from 1 player to the other in a FOREVER loop. The subprogram or subroutine called PLAY will contain the logic to allow each player to move. Got it?



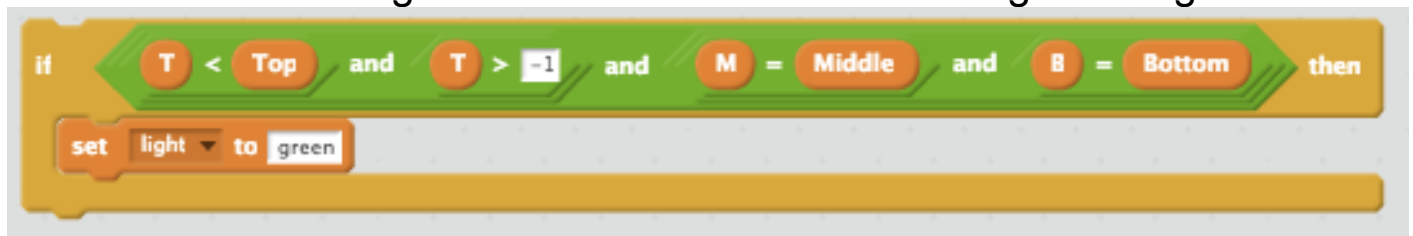
I created 3 variables to hold the number of sticks in each row. When the user types 3 digits for his/her move, this code using the LETTER command grabs each digit and stores in a variable. Here is PLAY:



This works nicely, but now I hit the challenge of how to make sure the player makes a LEGAL MOVE. You are not allowed to go from 357 to 344 for example. You can take from just 1 row. So I decided to make a variable called LIGHT and set LIGHT to RED at the start of the PLAY subroutine. Then I will take the variables T, M and B and test to see if we have a legal move. Here is one of ways we have a legal move. Note that TOP, MIDDLE and BOTTOM represent the existing board.

T, M and B are the 3 characters typed in by the USER. So if the USER types 356 or 306 or 123, then the 3 separate digits become T, M and B.

Meanwhile Top, Middle and Bottom represent the actual 3 rows from the current round of the game. This IF command is testing for a legal move.



So the above statement says if the middle and last numbers stay the same and the first number is smaller but at least zero, the light is green.

## Can you help me out?

Exercise 4.6: Please go to <https://scratch.mit.edu/projects/91332993> or short name [tinyurl.com/error33](http://tinyurl.com/error33) and REMIX A COPY of my program so it works to ERROR CHECK on all 3 rows and not just 1. You will see helpful notes there for you. After you REMIX, please rename the file as Exercise 4.6 Error Check ABC (using your initials instead of ABC). There is no JV or Varsity for this one.

Here is the solution if you get stumped: <https://scratch.mit.edu/projects/91337094> or short name [tinyurl.com/error33solved](http://tinyurl.com/error33solved)

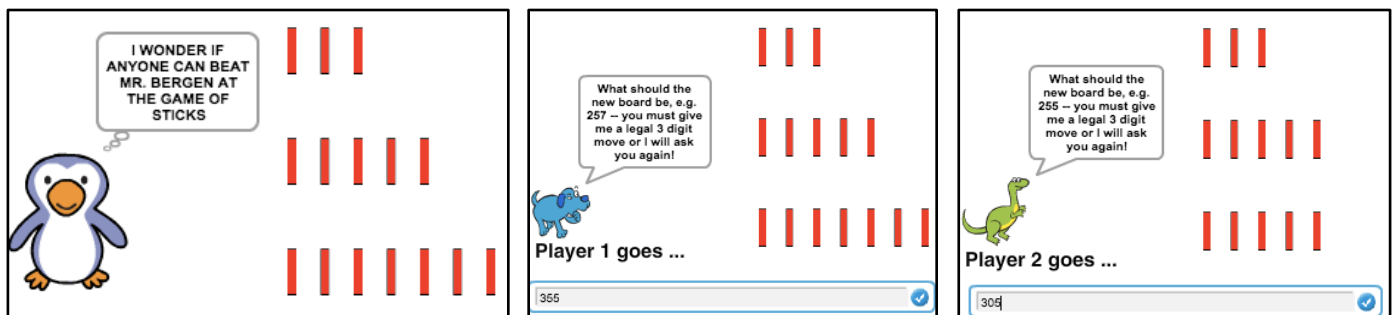
Thanks for helping me work on the game. You now get the privilege of playing it. You can play with a sister, brother, parent or friend.

Here is the finished STICK GAME. When you are ready to play me, we will do a screenshare and be talking on the phone. You will get 3 shots to beat me. You can choose to go first or have me go first. If you succeed and are the first to beat me, you win the \$33 gift certificate!

<https://scratch.mit.edu/projects/90654709/> or short version <http://tinyurl.com/beatmrb>

**Here is a sample game. Player 1 typed 355. Player 2 typed 305.**

After you type your move and push RETURN/ENTER then you see the new board on the next screen.



**Then Player 1 typed 301. Player 2 typed 001 and won.**  
Remember you have to type the zeroes when you play.



*Please note that my camp counselor taught me this game when I was 11 years old many years ago. Then I read an article in college about how the game is based on binary numbers. Ever since I read that article, I rarely lose!*

## END OF REQUIRED PORTION OF LESSON 4

Everything on the following pages are bonus lessons and bonus info for you if you finish all items above or after this course is over on Wed 12/23. I have been doing computer programming ever since 1974 and I hope you continue coding in Scratch (or other languages) after this course is over. Stay in touch! Mr. Bergen



## Resources for students and learners

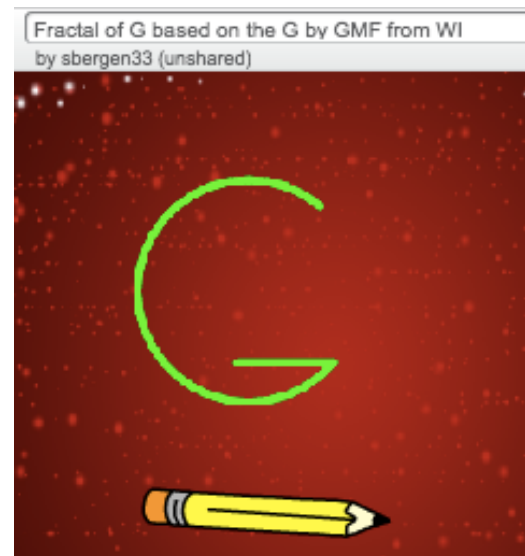
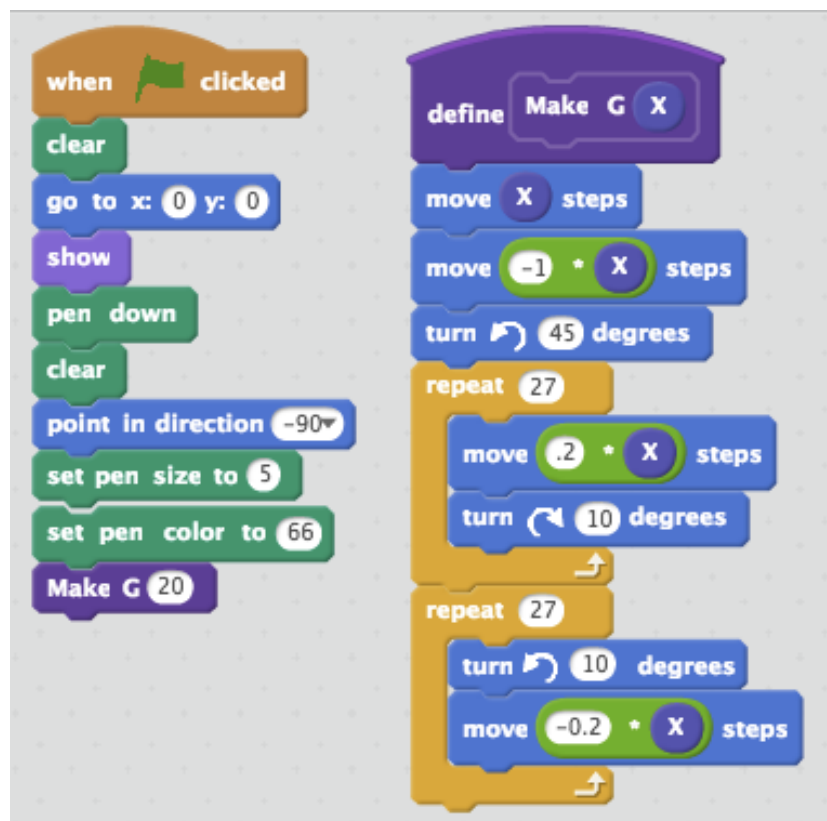
- Tutorials from [code.org](https://code.org)
- hour of code suggestions at <http://askatechteacher.com/2015/11/09/hour-of-code-3/>
- Scratch library of videos by Colleen Lewis, Assistant Professor at Harvey Mudd College CA  
[https://www.youtube.com/playlist?list=PLHqz-wcqDQIExCWk\\_EgaiZ6hLt4QX2Rxn](https://www.youtube.com/playlist?list=PLHqz-wcqDQIExCWk_EgaiZ6hLt4QX2Rxn)
- Book on Scratch <https://www.nostarch.com/learnscratch>
- Code School -- <https://www.codeschool.com>
- Learn Scratch -- <http://learnscratch.org>
- Advanced Level Projects with Scratch: Fun with Fractals  
<http://www.edutopia.org/blog/scratch-programming-advanced-fractal-fun-dylan-ryder>
- Info about Scratch for Parents -- <https://scratch.mit.edu/parents/>
- Scratch Help Cards -- <https://scratch.mit.edu/info/cards/>
- Video Tutorials from MIT -- <https://scratch.mit.edu/help/videos/>



## Optional 4.7 SuperVarsity Challenge: Making a Better Fractal from Your Uppercase Letter

SuperVarsity 4.7: Use your uppercase letter to make a fractal of smaller and smaller letters. This requires using BLOCKS and making sure your SPRITE returns to its original location and orientation. Name your program SV Challenge 4.7 Letter Better Fractal

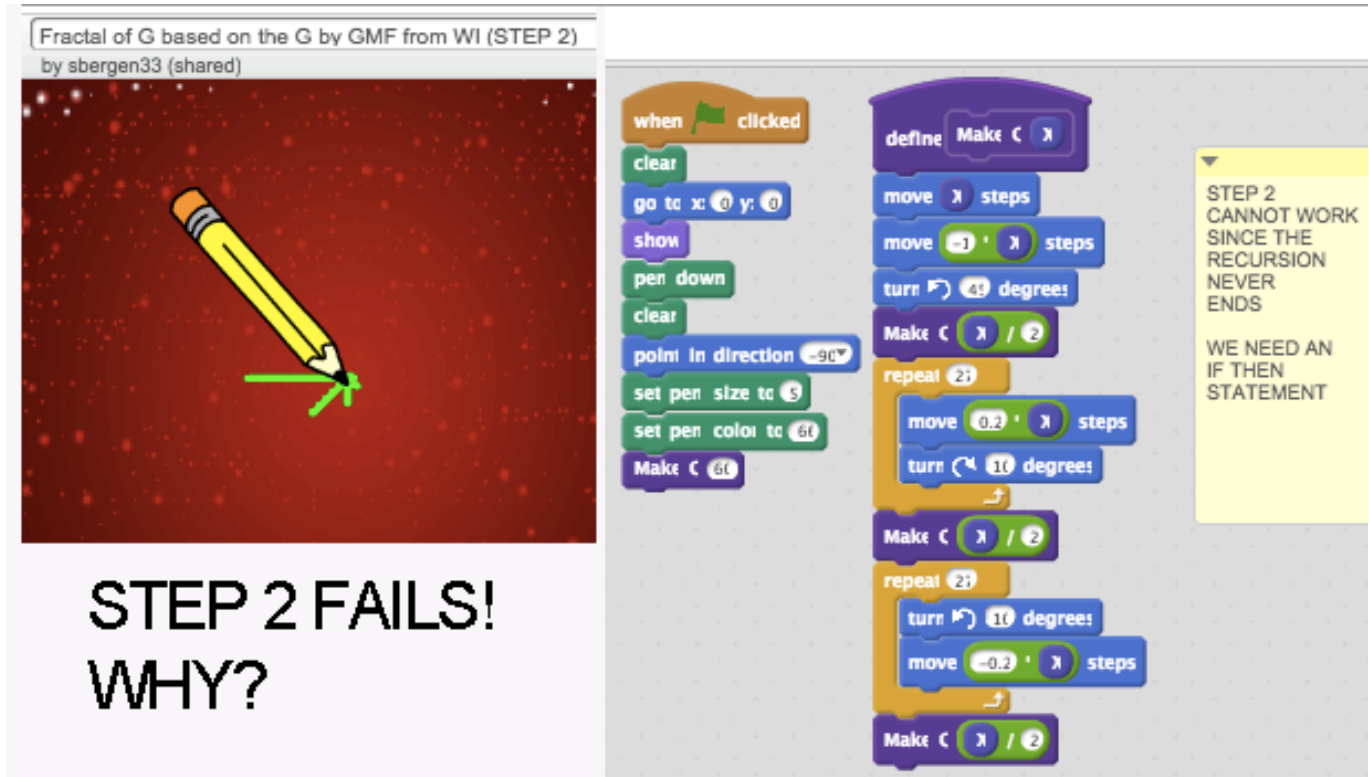
To begin, I took the uppercase "G" from GMF from WI and made it into a block that ends up exactly where it started. This is the crucial step 1.



You can see this program at <https://scratch.mit.edu/projects/91370075>

This is just the beginning of making our fractal -- more to come on the next few pages. The link above in yellow shows just how to make the G in a purple block.

Next, I added the MAKE G X/2 three times to the MAKE procedure, following the principles of recursion.



You can see this program at <https://scratch.mit.edu/projects/91373756>

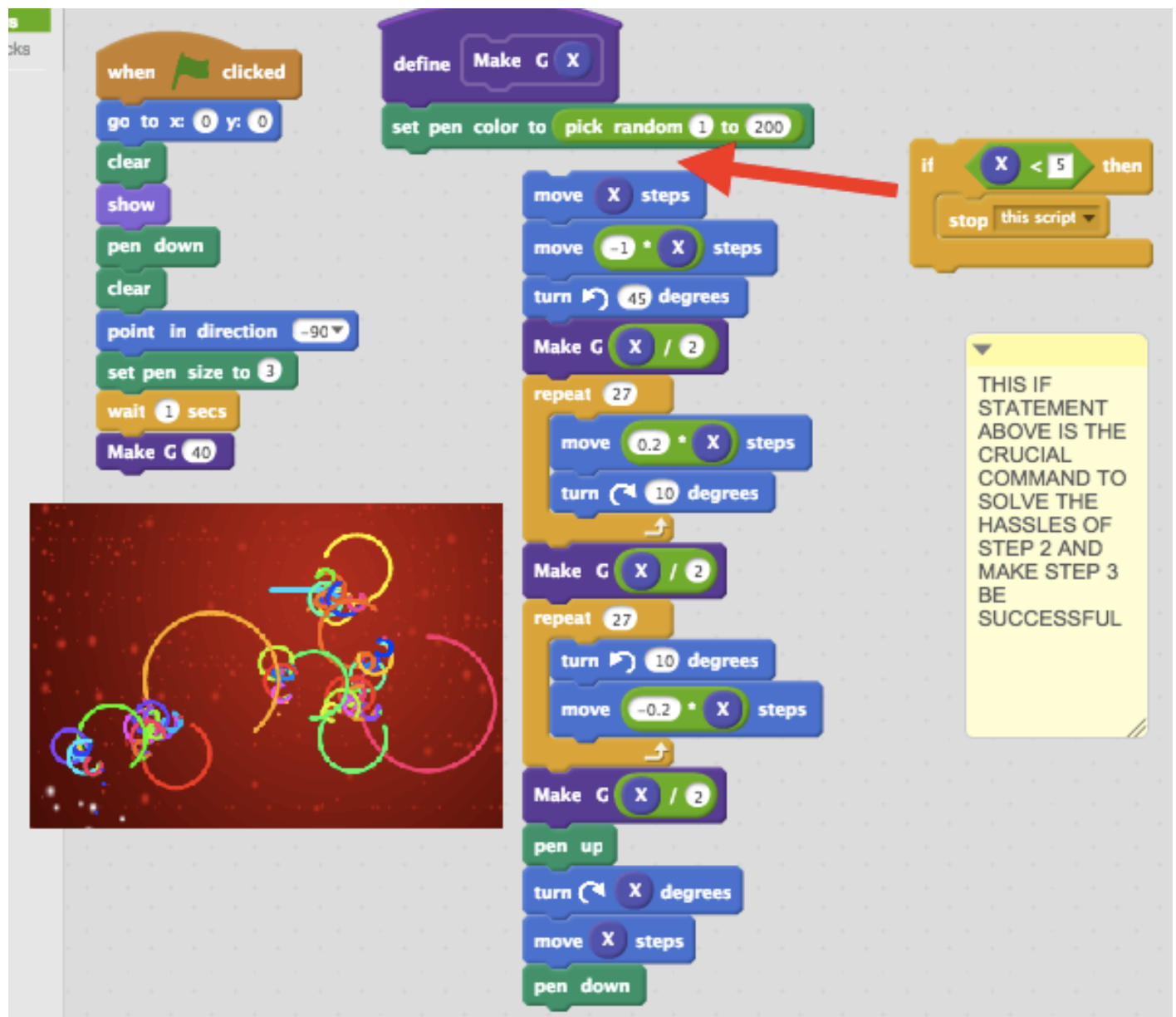
Oh, I know why it failed! I created a recursive program that made smaller and smaller G's. I did not have any code to make it stop!

My block called Make G 60 then calls 3 programs that say Make G 30 and each one then calls 3 programs that say Make G 15 and so on. It goes on and on from 60 to 30 to 15 to 7.5 to 3.75 and it gets smaller and smaller. We need to tell the program "how small we want before stopping."

Note: you might want to read all the 4.8 info on fractals and recursion on the subsequent pages before you continue with 4.7.

**Recursion: "a computer block that calls itself"**

So I created an IF statement that STOPS THIS SCRIPT if  $X < 5$ .

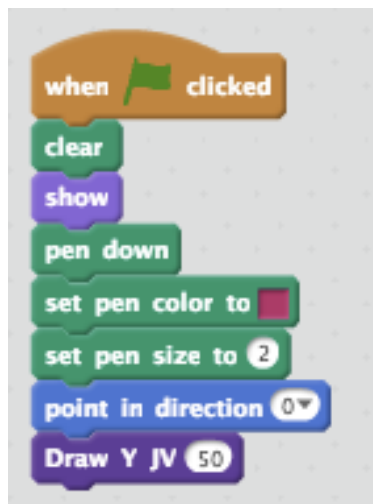


What you need to understand is that MAKE G 40 is one script that calls MAKE G 20 which then calls MAKE G 10. So when you STOP THIS SCRIPT, you are stopping just the sub-program called by recursion. So the G of size 2.5 stops, and now the computer finishes the G of size 5 and so on. In this one program, the nature of RECURSION creates dozens of sub-programs each of which executes and then stops!

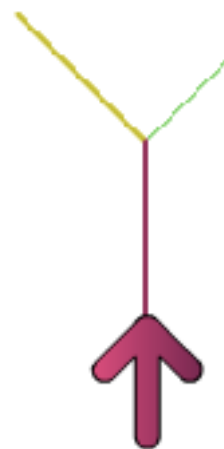
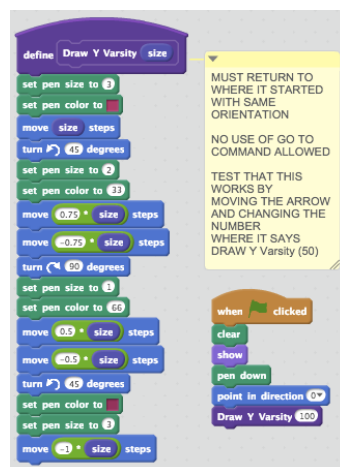
You can see this program at <https://scratch.mit.edu/projects/91373915>

## Optional 4.8 SuperVarsity Challenge: Making a Tree Fractal from the "Y" Block

We will use the "Y" Block you made from 3.2J or 3.2V to create a very cool fractal! If you did not do 3.2J or 3.2V you can REMIX my version and use that.



My JV solution of 3.2J is at <https://scratch.mit.edu/projects/88984345> with short name of [tinyurl.com/branchtree33](http://tinyurl.com/branchtree33) if you need to use it, just get it and REMIX it. You cannot do this challenge unless you have a "Y" done with purple blocks with the arrow going back to its original starting spot.



My Varsity solution of 3.3V is at <https://scratch.mit.edu/projects/88989565> with short name of [tinyurl.com/branchtree34](http://tinyurl.com/branchtree34) if you wish to use it.



So we are now ready to make our own fractal since we have a BLOCK that draws an UPPERCASE Y and returns to its original location.

To make a fractal, we need to understand a new word and new concept called **recursion**.

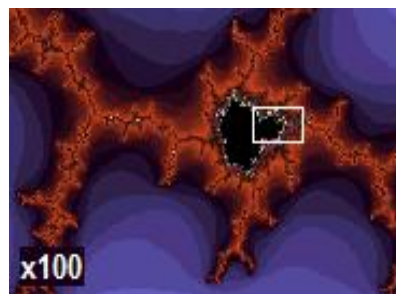
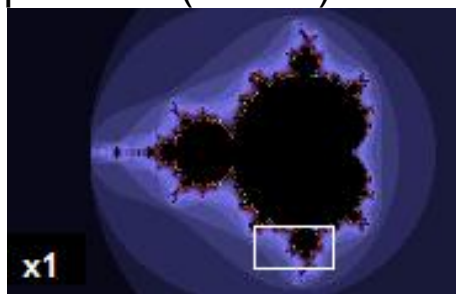
## Recursion: "a computer block that calls itself"

Please watch my silly 3 minute **INSTRUCTIONAL VIDEO** --

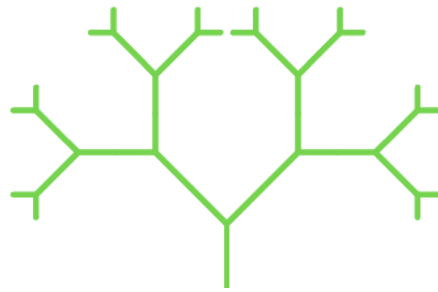
<http://youtu.be/r40MHcsWxql> with short name of tinyurl.com/lbsilly explaining with my Larry Bird doll what recursion is and why it is not the same as repetition using the REPEAT command.



Look at this cool fractal below. When you zoom in to this fractal on the left you get the graphic in the middle and then when you zoom in again, you get the graphic on the right. Each "zoom" yields a shape that is proportional (similar) to the original one.



So our journey is to make our own fractal.



Video Part 1 (5 min) <http://youtu.be/WQfEd5AcnVY>

This gives you the orientation to what a fractal is and an overview of the one we are going to code from scratch using Scratch.

Video Lesson Fractals Part 2 (18 min)

<http://youtu.be/LKkVzmBB5Zs>

This takes you through the nitty gritty programming for creating a tree fractal.

The above two videos will show you how to make this primitive "tree fractal" which of course I have shared with you in my Scratch library/studio

<http://scratch.mit.edu/projects/13740603/>

program name: Recursion with Fractals (Make Your Own Fractal 2013)

**Here is a better step-by-step presentation of creating your first fractal that has simpler and more efficient code that corresponds to below.**

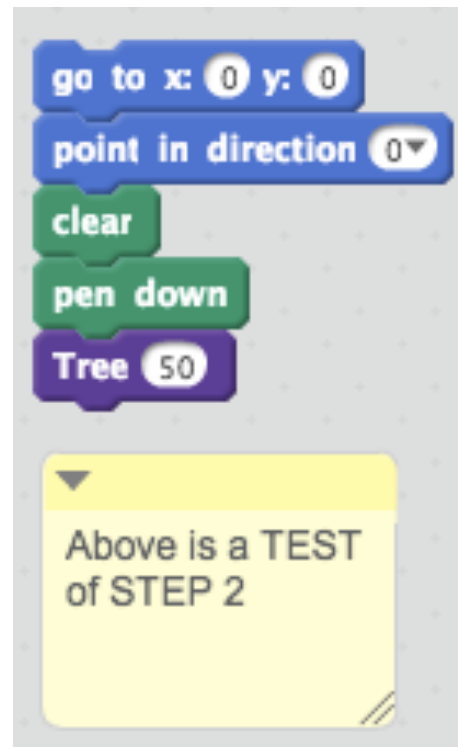
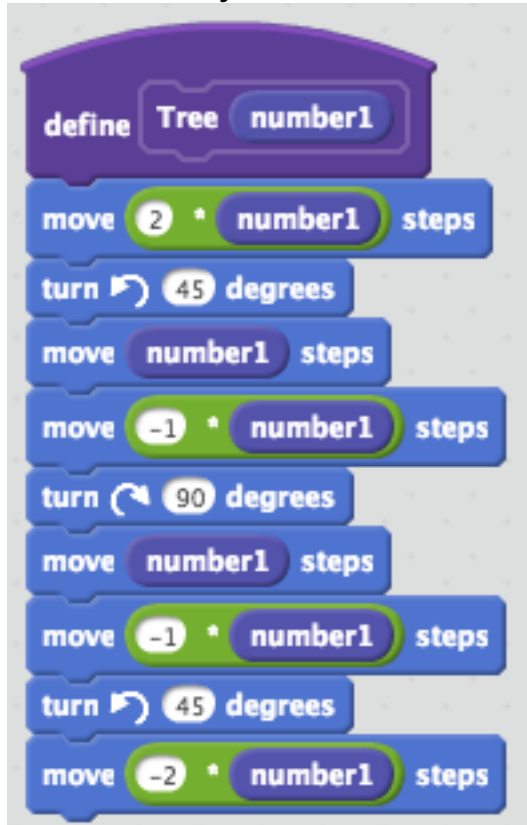
<http://scratch.mit.edu/projects/32791442>

program name: My First Fractal (step by step 2014)

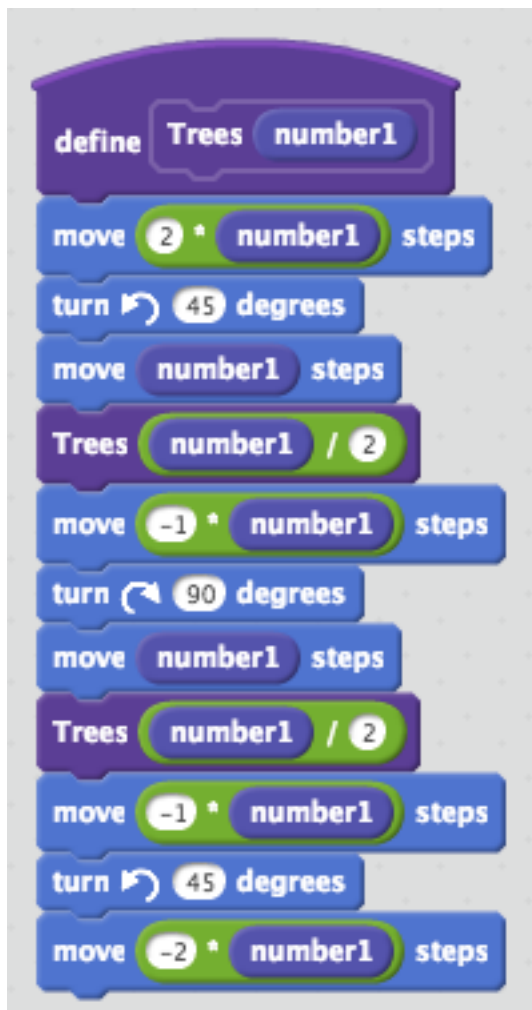
STEP 1 We begin by making an uppercase Y with each "limb" equal to half of the main trunk. Observe that the cat comes back to the starting spot (very important).

STEP 2 We now turn this into a new block called TREE that makes a variable sized "Y" where in this example NUMBER1 is taking on the value of 50 as in the STEP 1 example.

Make sure you test STEP 2 to get the same Y as you had in STEP 1.



STEP 3 Below is STEP 3 where in my demo I modified the TREE block, calling it TREES just so I can show you both STEP 2 and STEP 3. You do NOT need to have both the TREE and the TREES blocks. You can succeed with one block that is like TREE modified in step 3 and step 4.



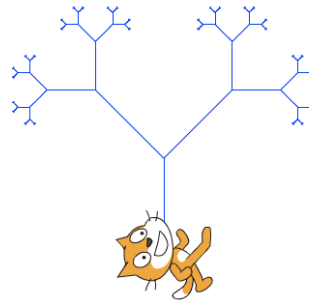
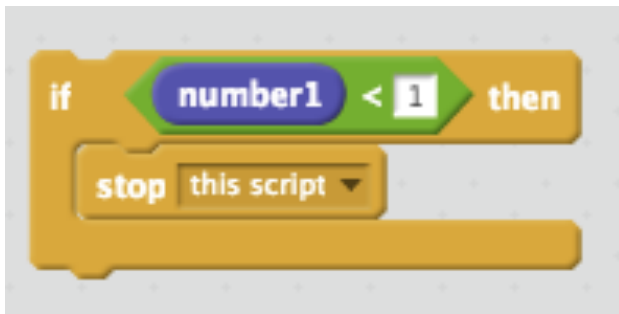
## STEP 3 is a failure!

This STEP 3 is very important even though it is a failure. You can see that the computer is trying to make the "Y" but it never finishes. At the top LEFT of each subsequent "Y" it tries to make another "Y" that is smaller. But it never finishes so what we have in STEP 3 is a series of "left sides of each Y."

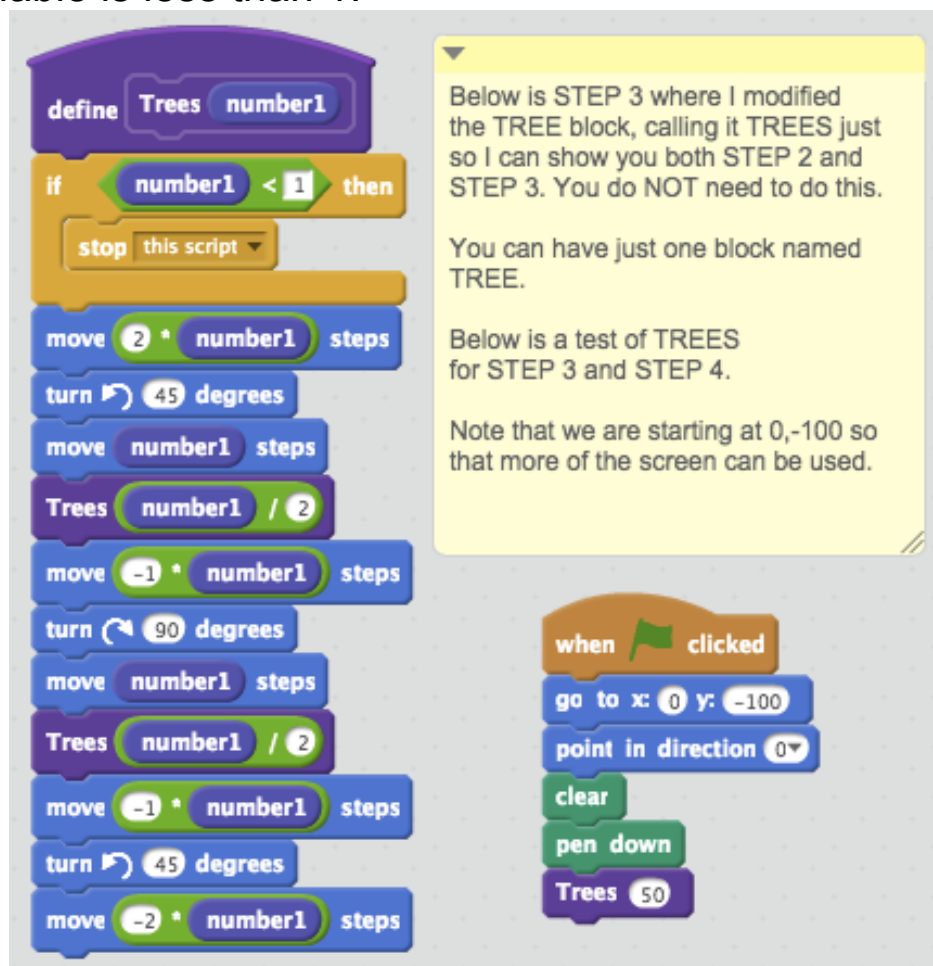
STEP 4: We are so close to success. We need just ONE IF STATEMENT to STOP the recursion when number 1 gets "too small" which you are welcome to define.

We add this one statement to our loop





and now we have it since now each subsequent and smaller "Y" is done until the variable is less than 1.



SuperVarsity 4.8: Using the "Y" you made with BLOCKS either 3.2J or 3.2V follow the logic of this lesson and create your own fractal. Name your file SV4.8 ABC Fractal with my Y.

## Fractal Alphabet: A Lot of Fun

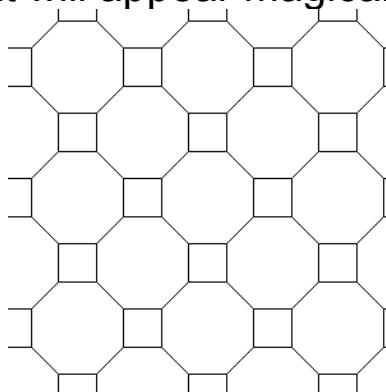
Check out <https://scratch.mit.edu/projects/77600038>

This is so much fun that I can watch this for hours!



## Another Tessellation Optional Challenge 4.9

You will see this on many bathroom floors -- a tessellation of octagons (8 sides) and squares. You can do this with the same algorithm I taught you. Make an octagon. Go around more than 8 times, i.e. REPEAT 8-16 using a random number that must be even-- 8 or 10 or 12 or 14. Then turn the INTERIOR ANGLE of the octagon. You need to calculate this. The exterior angle of an octagon is 45 degrees since  $360/8 = 45$ . So the interior angle is 135 degrees. Good luck! What is so interesting is that you can ignore the squares that will appear magically between the hexagons.



From: <http://gwydir.demon.co.uk/jo/tess/octsq.gif>

**SuperVarsity 4.9: Create a program that makes the tessellation of octagons and squares. Name your file SV4.9 ABC Octagon Tess.**

## Optional Lesson on Lists and Poetry in Scratch SuperVarsity Challenge 4.10

In this topic we use Scratch to write poetry! I am very excited to teach you this based on something I did in 1973-74 when I was young and started teaching programming at an independent school in Charlottesville VA.

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 the 7 minute **INSTRUCTIONAL VIDEO** on 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!**

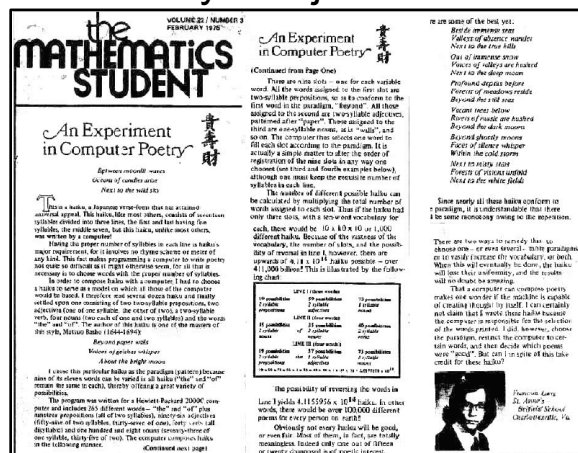
**Prep**

- 1 **Next to**
- 2 **Beyond**
- 3 **Beside**
- 4 **Above**
- 5 **Below**
- 6 **Within**

set chosen to item pick random 1 to 6 of Prep

set chosen to item random of Prep

Here is the article that my student Francois Lang published in 1975. He was a high student of mine in my first job in Charlottesville VA.





So Francois took a famous poem such as one of these by Matsuo Basho

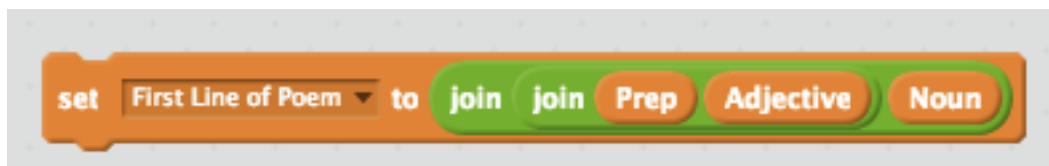
Next to misty stars  
Forests of visions unfold  
Next to the white fields

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

and then programmed the computer to 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.



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 ... Mr. Bergen 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



SuperVarsity 4.10: Create on Scratch a program that writes an original Haiku randomly the way Francois did in 1974. Name your file SV4.10 ABC Poetry. You can see and study a sample at <https://scratch.mit.edu/projects/14015872>

