

START A MATH + CODING CLUB

page

MATH + CODING?

1

Explore math + coding concepts with Python and Scratch

STEP 1 - ORGANIZE A CLUB

9

Work together. Do more.

STEP 2 - LEARN COMPUTATIONAL THINKING

10

Write algorithms a computer could understand

STEP 3 - LEARN TO CODE

17

Create cool projects with code

STEP 4 - SHARE WITH THE WIDER COMMUNITY

18

"The more we share the more we have." (Leonard Nimoy)



MATH + CODING?

Computer coding and math frequently go hand-in-hand. When you code, you often (although not always) use math ideas to solve problems.

Also, coding helps make learning math fun:

- math concepts become tangible - objects on the screen you can manipulate
- math concepts become dynamic - change a value and see the effect
- you are in control - explore variations and related ideas

Let's explore a few math ideas through code.

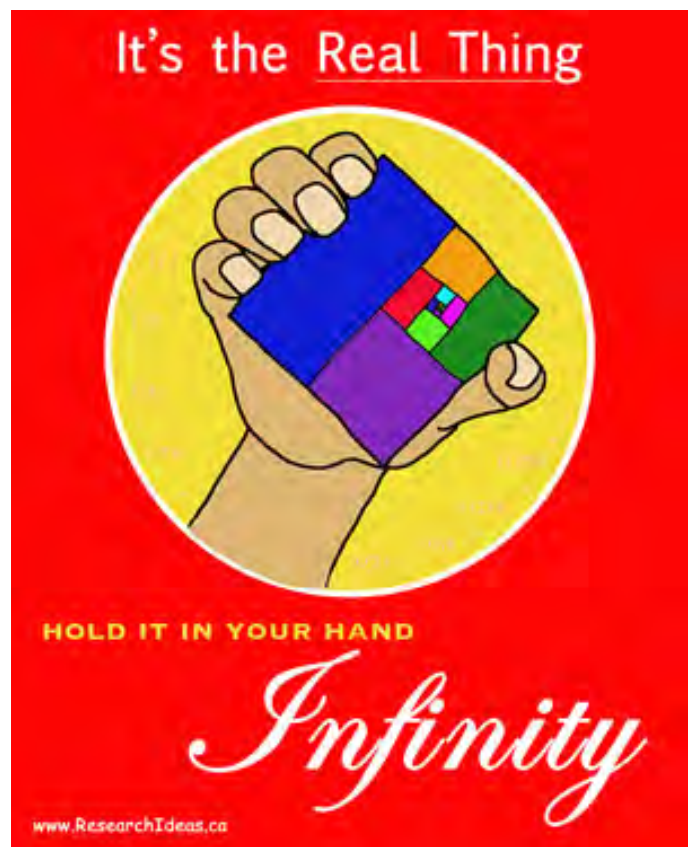
Hold infinity in the palm of your hand

Infinity is big. Infinity takes forever. But, you can hold infinity in the palm of your hand!

Take a square. Shade 1/2 of it one colour. Shade 1/2 of what's left another colour. Shade 1/2 of what's left another colour. You can keep doing this forever, and you would never leave the square. There will be an infinite number of fractions that fit in the square.

Here's another way of saying that an infinite number of fractions can have a finite sum:

$$1/2 + 1/4 + 1/8 + 1/16 + \dots = 1$$



Infinity with Python

Let's explore the sum of $1/2 + 1/4 + 1/8 + 1/16 + \dots$ with computer code, using the Python coding language.

Go to

cscircles.cemc.uwaterloo.ca/console and enter and run the code on the right.

What do you notice about the sum?

Can you explain what the code is doing?

```
1 # *** INFINITY IN YOUR HAND ***
2 sum = 0
3 numerator = 2
4 for stage in range (1,101):
5     fraction = 1/(numerator)
6     sum = sum + fraction
7     print (stage, sum)
8     numerator = numerator*2
```

More infinity with Python

Let's look at these fractions. What would be their sum?

$$1/4 + 1/16 + 1/64 + 1/254 + \dots = ??$$

Let's model this with the code below.

It's just like the previous code, only we start with a numerator of 4 and then increase it by a multiple of 4.

```
1 # *** AN INFINITE SUM OF 1/3 ***
2 sum = 0
3 numerator = 4
4 for stage in range (1,101):
5     fraction = 1/(numerator)
6     sum = sum + fraction
7     print (stage, sum)
8     numerator = numerator*4
```

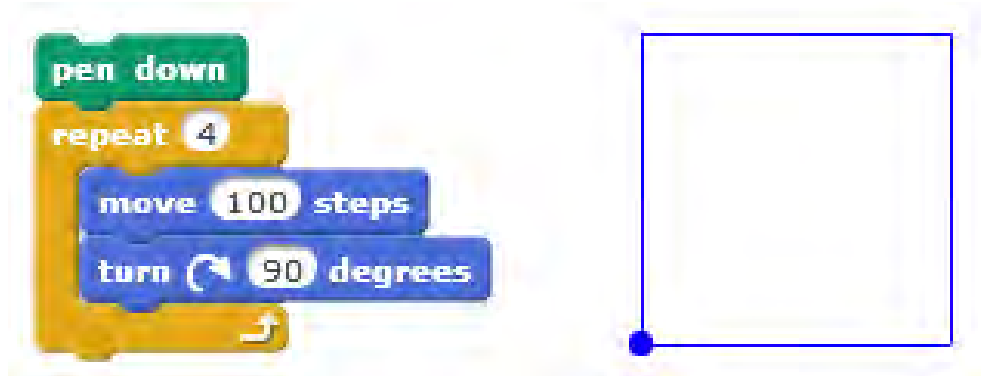
What do you notice about the sum?

Can you explain what the code is doing?

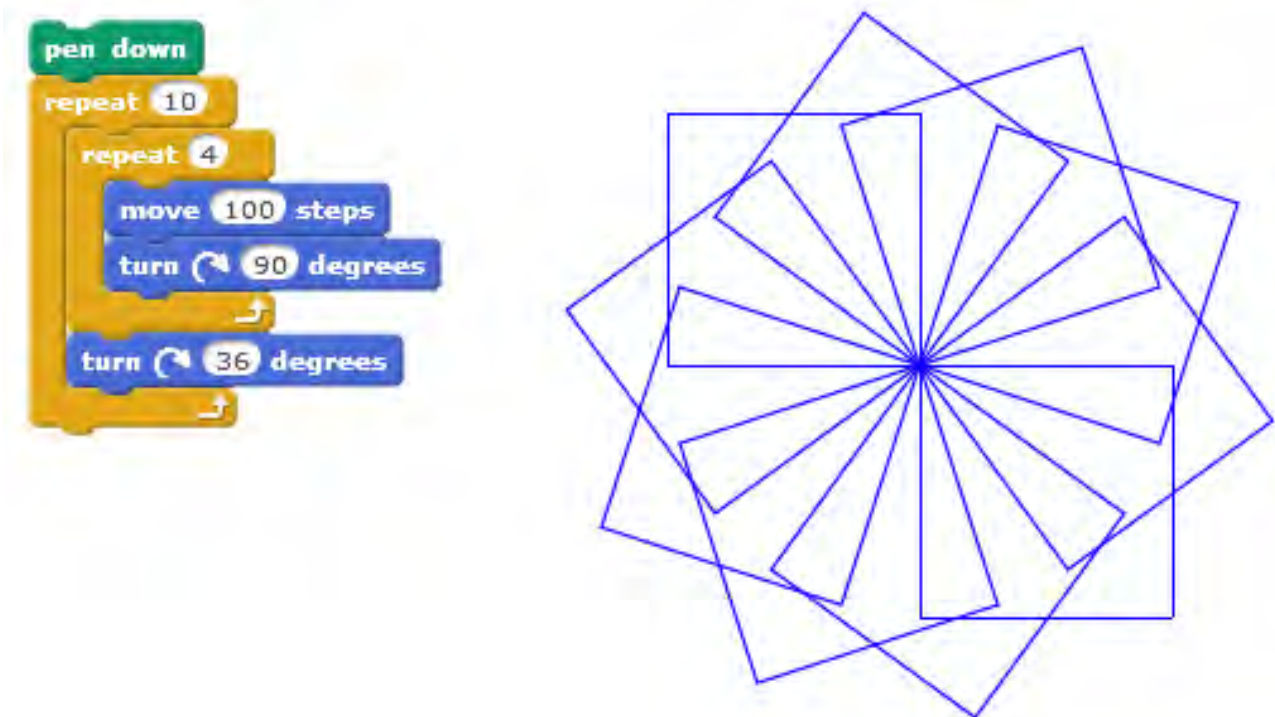
Draw squares with Scratch

Let's draw a square using the Scratch coding language.

Go to scratch.mit.edu and click on Create. Compose the code below.

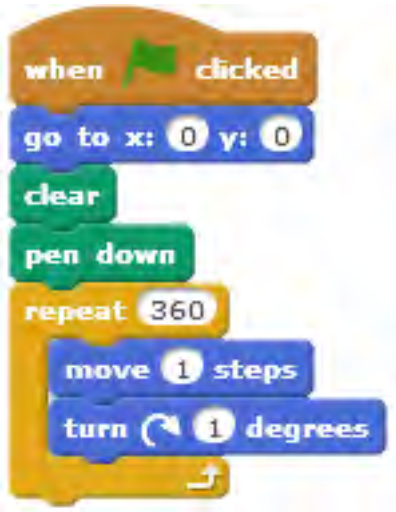


Now edit and run the code to get this pattern.

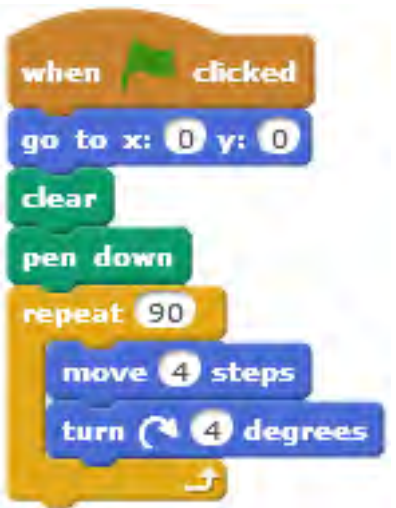


Draw circles with Scratch

Enter this code to draw a circle.



Enter this code to draw a circle faster.



Make a block with Scratch

Let's create a new block called

draw circle

- in the **Scripts** menu, click on **More Blocks**
 - click on **New Block**
 - in the dialogue box that pops up, type "draw circle" for the name of the new block



- click OK – **define: draw circle** will appear in the coding window



- drag the repeat block that draws a circle and place it as shown below

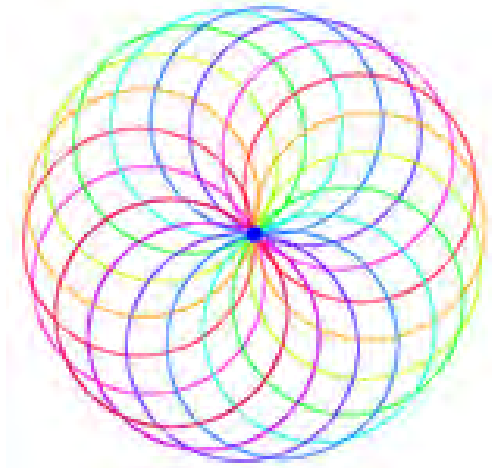


Lots of circles, in colour! (with Scratch)

Enter the code below to draw a circles pattern.

```
when green flag clicked
  go to x: 0 y: 0
  clear
  pen down
  repeat 20
    draw circle
    turn 20 degrees
    set pen color to pick random 1 to 256

define draw circle
  repeat 90
    move 4 steps
    turn 4 degrees
```



Probability with 2 coins with Python

The Python code below simulates tossing 2 coins 100 times and counting the number of times each of these outcomes occur:

- 2 heads (HH)
- 1 head, 1 tail (HT)
- 2 tails (TT)

Go to cscircles.cemc.uwaterloo.ca/console and enter and run the code below. What do you notice? Do you understand what the code is doing?

```
1 # *** PROBABILITY WITH 2 COINS ***
2 import random
3 HH = 0
4 HT = 0
5 TT = 0
6 for x in range(1,101):
7     coin1 = random.randint(1,2)
8     coin2 = random.randint(1,2)
9     sum = coin1 + coin2
10    if sum == 2:
11        HH = HH + 1
12    elif sum == 3:
13        HT = HT + 1
14    elif sum == 4:
15        TT = TT + 1
16 print("")
17 print("TOTALS")
18 print("HH =",HH, "/100")
19 print("HT =",HT, "/100")
20 print("TT =",TT, "/100")
```


Math + coding simulations and games

Explore math surprises with code.

See www.mathNcode.com

Computer Code for a Growing Pattern

VIEW: Code Flowchart Text OUTPUT: Table Graph Grid

```

repeat for stage = 1 to 8
  BLUE = stage x 3
  RED = 4
  plot BLUE, RED
  
```

Plotting Random Pairs - Game

VIEW: Code Flowchart

```

x + y = 8
repeat 6
  roll die for x
  calculate y
  plot (x, y)
  
```

TEST NEXT

Question: 1/5
Current Score: 0/3 (0%)
High Score: 0/0
Level: 1/3 Goal: 20%

Select values to match the graph on the right.
Click TEST to check your prediction.

Making Math Waves - Game

VIEW: Code Flowchart Speed:

```

diameter = 13
height = 13
hour = 1
repeat 12
  climb to hour
  plot climb
  hour = hour + 1
  
```

TEST NEXT

Question: 1/5
Current Score: 0/5
High Score: 0/5
Level: 1/3 Goal: 1/5

Select values to match the graph on the right.
Click TEST to check your prediction.

Probabilities - Simulation

VIEW: Code Flowchart Speed:

```

# of dice = 2
loop
  roll dice
  frog = sum rolled
  move frog
  if frog in pond
    then exit loop
  
```

Repeating Patterns - Simulation

VIEW: Code Flowchart Speed:

```

grid = 5
core length = 4
core = ●●●●
repeat 10
  stamp core
  
```

```

when clicked
  set Counter to 1
  clear
  go to x: Left, y: Bottom
  repeat 3
    point in direction 90
    repeat Counter
      stamp
      change y by 10
    turn 45 degrees
    repeat 2
      stamp
      change y by 10
  change Counter by 1
  change x by 50
  set y to Bottom
  
```

STEP 1 - ORGANIZE A CLUB

Working with others on a common interest is a lot of fun!

Here are some suggestions for organizing a math + coding club.



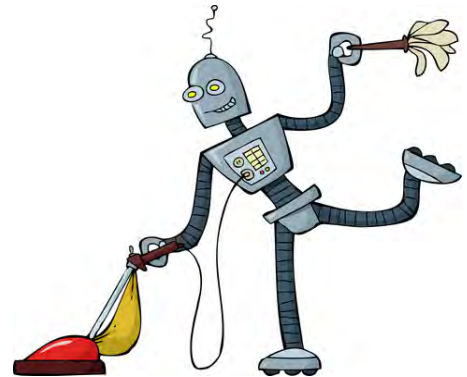
1. Chat with others that might be interested.
2. Decide what you want to do in this club.
3. Create a catchy name and a brief/clear description of the club.
4. Talk to your teacher or principal about the club.
5. Advertise a first meeting time/place.
 - a. Discuss the goals of the club.
 - b. Decide how often you want to meet, and when/where.
 - c. Create a members list.
6. After a couple of meetings, elect leaders, such as:
 - a. a president
 - b. a vice-president
 - c. a secretary (to record what happens at your meetings)
 - d. an event organizer.
7. Advertise your club.
 - a. Finalize the name and brief description of the club
 - b. Create a club motto or slogan
 - c. Create posters to let others know about your club.
 - d. Create a blog or website for your club. Talk to your teacher for advice on Internet safety.

STEP 2 - LEARN COMPUTATIONAL THINKING

What is computational thinking?

To solve a problem computationally means to write an algorithm--a set of solution steps--that a computer or a robot could follow.

Most fields of study today use computational thinking to design and test solutions using computers.



Having fun

Let's have some fun with computational thinking.

BOYS AND GIRLS

Can you follow the algorithm below?

```
When your teacher says "GO"  
  If You = Boy  
    Clap 5 times  
  Else  
    Say "Girls rule"  
  End If
```



Colours

Try this algorithm.

```
Look at the colour of your shirt  
When your teacher says "GO"  
  Stand up  
  If Your Shirt = Blue  
    Clap twice  
    Sit down  
  Else If Your Shirt = Green  
    Turn around once  
  Else  
    Do a little dance  
  End If
```



Your own

Have fun. Write your own algorithm!

Food

What is your favourite food?

Decide on your favourite food

Repeat 5 times

When your teacher says "GO"

If Favourite = Pizza

Say "I love Pizza"

Else If Favourite = Pasta

Say "Pasta is the best"

Else

Clap 3 times

End If

End Repeat



Square dance

Face the front of the class

Repeat 4 times

When your teacher says "GO"

Walk 2 steps

Clap twice

Turn right

End Repeat



Square dance #2

Face the front of the class

Number = 1

Repeat 4 times

When your teacher says "GO"

Walk 2 steps

Clap Number times

Turn right

Increase Number by 1

End Repeat

Coin toss

Perform the algorithm and record the heads/tails count.

Which has the better chance of winning: heads or tails? Why?

```
Toss a coin
When your teacher says "GO"
  If Toss = Heads
    Stand up
  Else
    Sit down
  End If
```



Coin toss - 5 times

How is this algorithm similar to the previous one; and how is it different?

Perform the algorithm and record the heads/tails count.

Which has the better chance of winning: heads or tails? Why?

```
Repeat 5 times
  Toss a coin
  When your teacher says "GO"
    If Toss = Heads
      Stand up
    Else
      Sit down
    End If
  Wait for your teacher to record the tally
End Repeat
```

2 coin toss

Perform the algorithm and record the HH, TT and HT count.

Which has the better chance of winning: HH, TT or HT? Why?

```
Toss 2 coins
When your teacher says "GO"
  If Toss = 2 Heads
    Raise 2 hands
  Else If Toss = 2 Tails
    Raise 0 hands
  Else
    Raise 1 hand
  End If
```



2 coin toss - 5 times

How is this algorithm similar to the previous one; and how is it different?

Perform the algorithm and record the HH, TT, HT count.

Which has the better chance of winning: HH, TT or HT? Why?

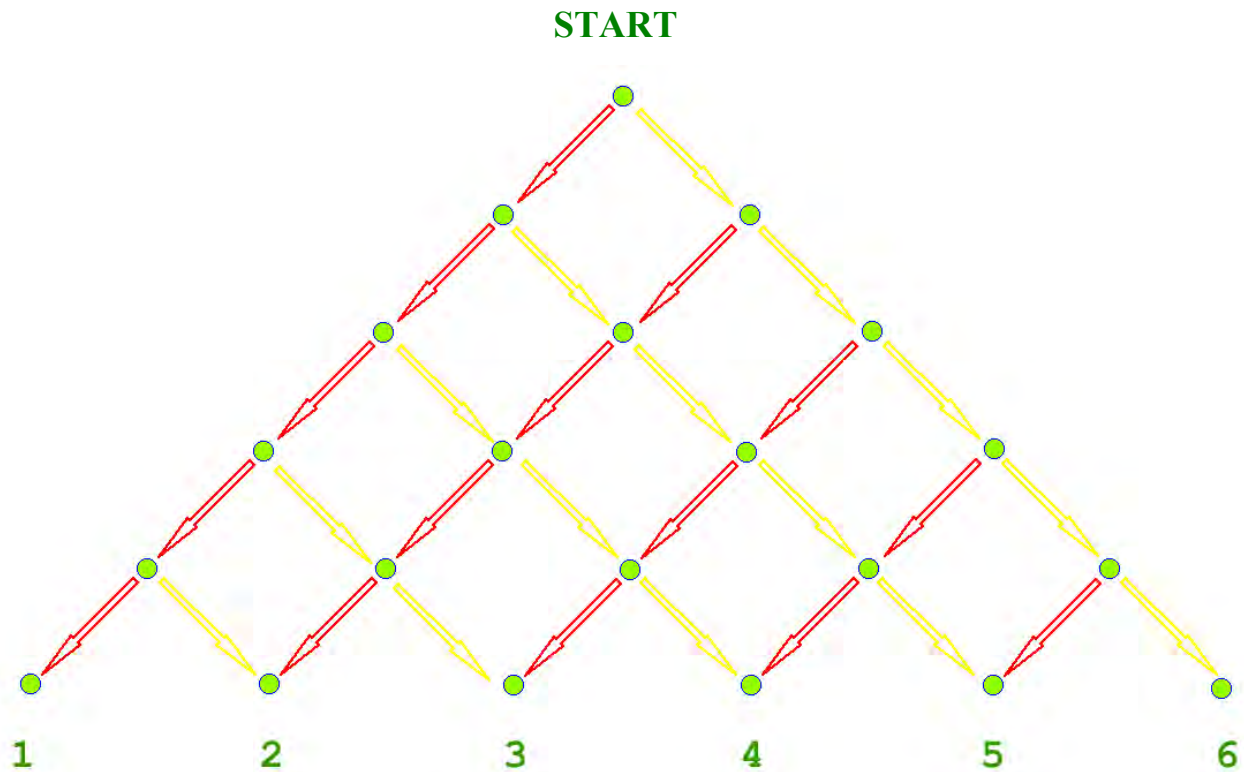
```
Repeat 5 times
  Toss 2 coins
  When your teacher says "GO"
    If Toss = 2 Heads
      Raise 2 hands
    Else If Toss = 2 Tails
      Raise 0 hands
    Else
      Raise 1 hand
    End If
  Wait for your teacher to record the tally
End repeat
```

Coin toss path

Perform the algorithm and record your result.

At which number are you most likely to end up? 1, 2, 3, 4, 5, or 6?

```
Start at the top of the path
Repeat until bottom of the path
  Toss coin
  If coin = HEADS
    Go 1 step along RED path
  Else
    Go 1 step along YELLOW path
  End If
End Repeat
Record your number
```



WHAT GRADE?

All of the computational thinking activities on pages 11-15 were first done with grade one students! They can also be done in higher grades. See higher grade connections on pages 16-17

Pascal's Triangle

Pascal's Triangle is named after the French mathematician Blaise Pascal (1623-1662).

Can you see a connection between the numbers in Pascal's Triangle and the number of ways of getting to each of the outcomes in the previous activity?

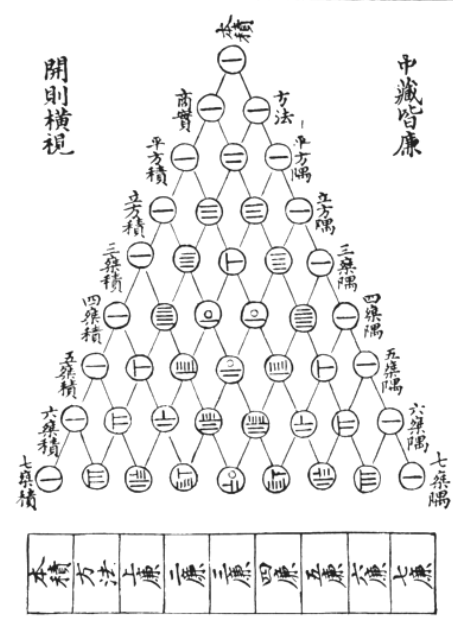


However, this triangular array of numbers was known well before Pascal's time, in India, Persia and China.

The image on the right shows the Yanghui Triangle, published in 1303 in Chinese, and named after the mathematician Yang Hui (1238–1298).

There are many patterns in Pascal's Triangle, in addition to the ones that relate to probability outcomes.

古法七乘方圖



Probability, algebra & Pascal's Triangle

There is a neat connection between (1) the outcomes when tossing coins, (2) algebra, and (3) Pascal's Triangle.

Can you see the connection below?

Complete the missing rows.

PROBABILITY

ALGEBRA

PASCAL'S TRIANGLE

0 coins

--

$$(H+T)^0 = 1$$

1

1 coin

H, T

$$(H+T)^1 = H+T$$

1

1

2 coins

HH, 2 HT, TT

$$(H+T)^2 = H^2+2HT+T^2$$

1

2

1

3 coins

HHH, 3 HHT, 3 HTT, TTT

$$(H+T)^3 = H^3+3H^2T+3HT^2+T^3$$

1

3

3

1

4 coins

5 coins

Binomial Theorem

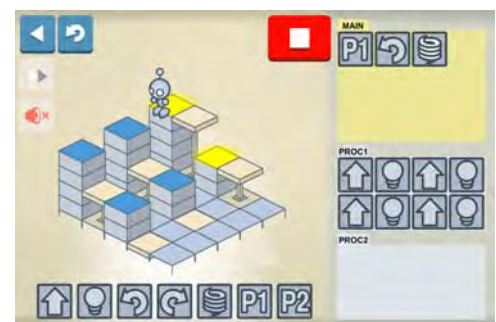
All of this also connects to the Binomial Theorem, which is studied in high school:

$$(x+y)^n = \binom{n}{0}x^n y^0 + \binom{n}{1}x^{n-1}y^1 + \dots + \binom{n}{n-1}x^1 y^{n-1} + \binom{n}{n}x^0 y^n$$

Light Bot

Practice your computational thinking with Lightbot.

See www.lightbot.com



STEP 3 - LEARN TO CODE

You learn best when you learn for a purpose.

Decide on a coding project. Then learn the coding methods you need to complete the project.

Two cool coding languages to start with are Scratch and Python.

WHAT IS PYTHON?

- Python is a widely used computer coding language.
- It is designed to make computer code readable and easy-to-understand.
- It also allows for expressing concepts in fewer lines of code than other popular computer coding languages.

WHAT IS SCRATCH?

- Scratch is a child-friendly, drag-and-drop computer coding language.
- Students attach code to objects (sprites) to create interactive simulations, stories and games.

Also explore game and app development environments, such as:

- GameMaker (www.yoyogames.com)
- App Inventor (www.appinventor.mit.edu).



STEP 4 - SHARE WITH THE WIDER COMMUNITY

Organize a Math + Coding evening for your community.

Before:

- Come up with a catchy name for your event.
- Decide what you want to share.
- Talk to your teacher or principal.
- Set a date that works for your school.
- Create posters.
- Advertise to the school community through a newsletter.
- Inform your local press.

During:

- Set up stations where you share your projects.
- Create stations where you give people opportunities to learn to code - such as the activities in this booklet.
- Have a feedback station, where people can write notes about your event.
- Take pictures.

After:

- Write a summary for your website/blog.
- Share the summary with local media.

Get \$200 support for materials and snacks, funded by the Fields Institute. See www.researchideas.ca/coding-events

