# Fractal

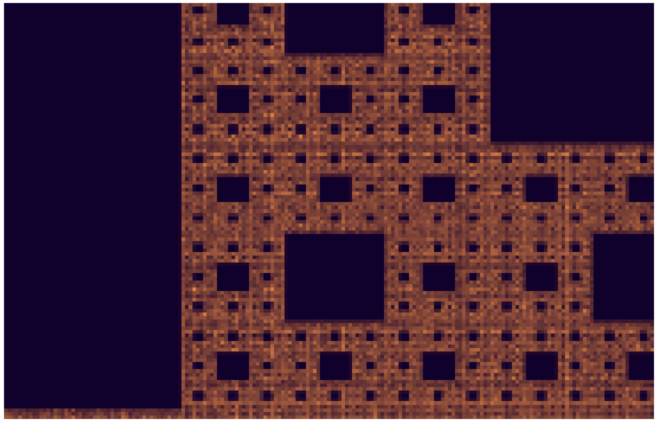
Hill ($2^{nd}$ Edition): 9.1 to 9.3, 9.6

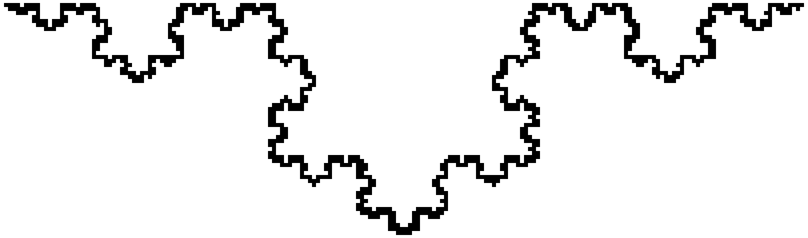
Extra Resource:

1. https://fractalfoundation.org/fractivities/FractalPacks-EducatorsGuide.pdf
2. https://math.bu.edu/DYSYS/chaos-game/node6.html

# Introduction to Fractal

- Self Similarity
  – They appear the same at every scale, no matter how much enlarged.





# Introduction to Fractal

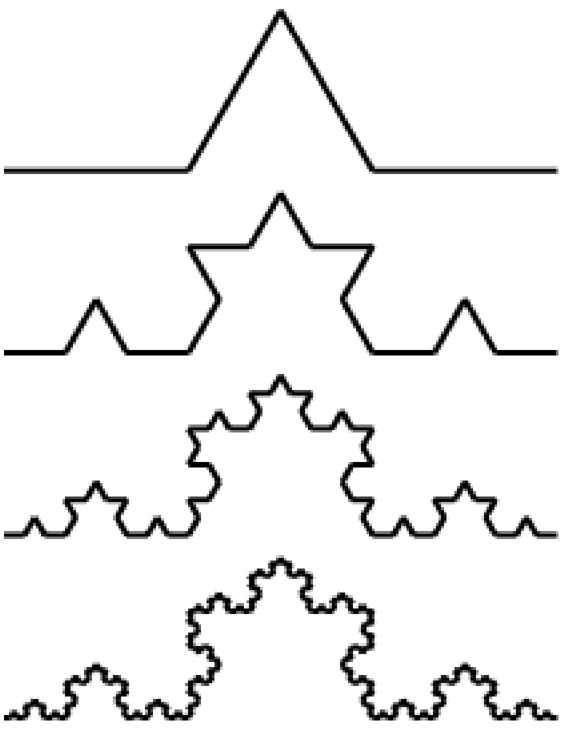
- Infinite length within a finite region
- Different fractals
  - Koch Curve
  - Koch Snowflake
  - Dragon Curve
  - Hilbert Curve
- Implemented using recursive functions

# Introduction to Fractal

- Application:
  - Simulating Clouds, ferns, coastlines, mountains, veins, nerves, parotid gland ducts, etc. all possess these features
  - Fractal geometry is useful for all sorts of things from biology to physics, to cosmology, to even the stock market
  - Fractal math is involved in JPG, MP3, MPG and other digital compression

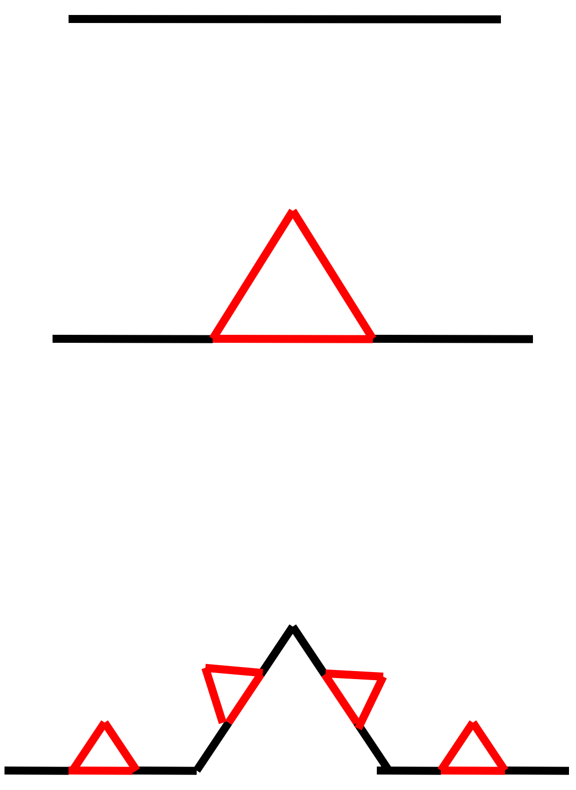
# Koch Curve

- $k_0$ (0-th generation)
- $k_1$ ($1^{st}$ generation)
- $k_2$ ($2^{nd}$ generation)
- $k_3$ ($3^{rd}$ generation)
- $k_4$ ($4^{th}$ generation)



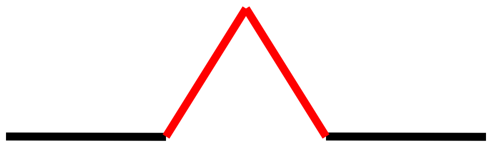
# Koch Curve

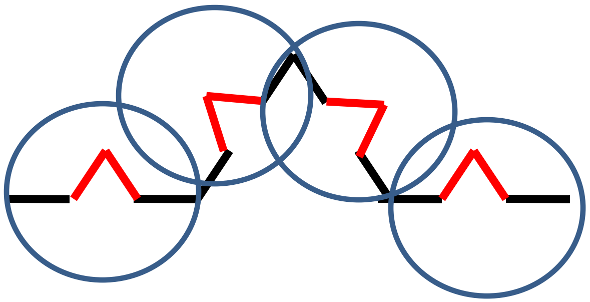
- Initial line: $k_o$
  - horizontal line having unit length
- $1^{st}$ generation: $k_1$
  - divide $k_o$ in three equal parts
  - replace the middle section with triangular bump each side having length 1/3
- $2^{nd}$ generation: $k_2$
  - repeat the process for each of the four line segments



# Drawing Koch Curve

- $k_2$ consists of four $k_1$'s





- To draw $k_2$
  - Draw a smaller version of $k_1$
  - Turn left 60$^o$, and again draw a smaller version of $k_1$
  - Turn right 120$^o$, and again draw a smaller version of $k_1$
  - Turn left 60$^o$, and again draw a smaller version of $k_1$

Recursive!

# Drawing Koch Curve

- n-th generation consists of four versions of (n-1)-th generation

To draw $k_n$ :

if n = 0 : Draw a straight line

else :

  Draw $k_{n-1}$

  Turn left $60^o$

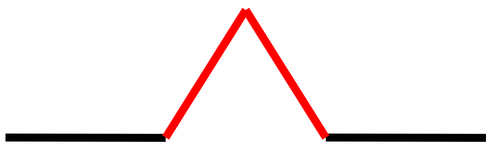
  Draw $k_{n-1}$

  Turn right $120^o$

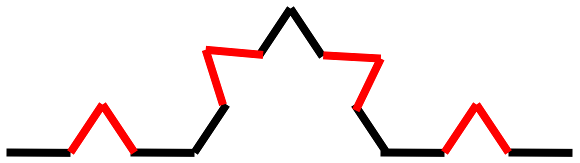
  Draw $k_{n-1}$

  Turn left $60^o$

  Draw $k_{n-1}$



**Length = (4/3)**

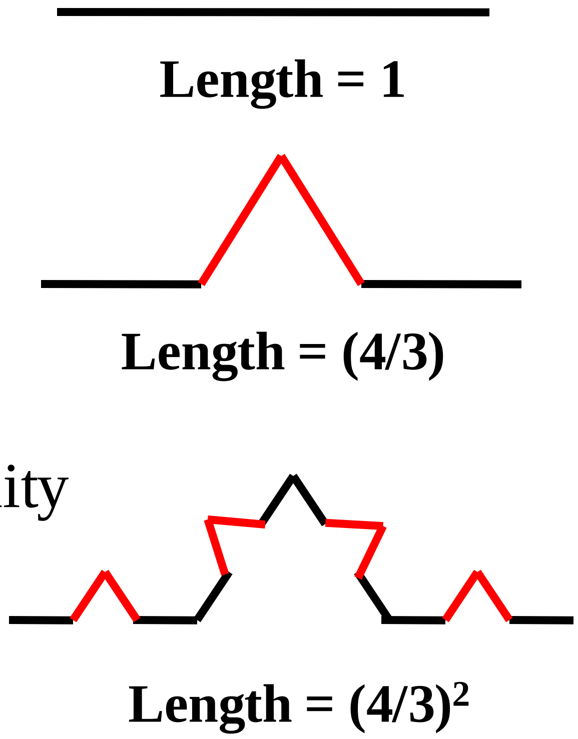


**Length = $(4/3)^2$**

# Length of Koch Curve

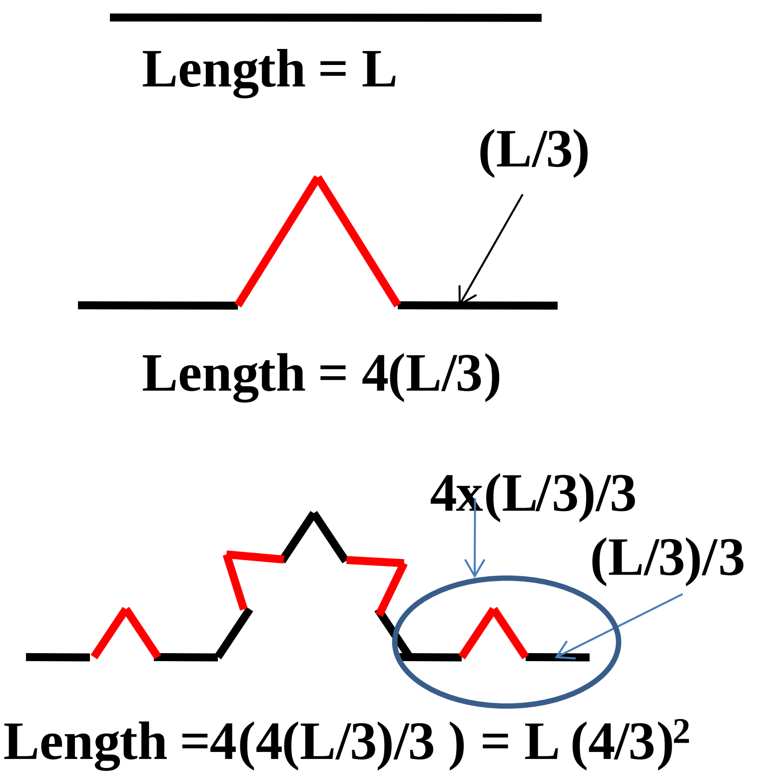
- **Length of curve for $k_n$ is $(4/3)^n$ - How?**
  - Length of $k_0 = 1$
  - Length of $k_1 = 4 \times 1/3 = 4/3$
  - Length of $k_2 = 4\ (4 \times (1/3)/3) = 4^2(1/3)^2$
  - Length of $k_3 = 4^3(1/3)^3$

  …
  - Length of $k_i = 4^i(1/3)^i = (4/3)^i$
    if i tends to infinity, length tends to infinity



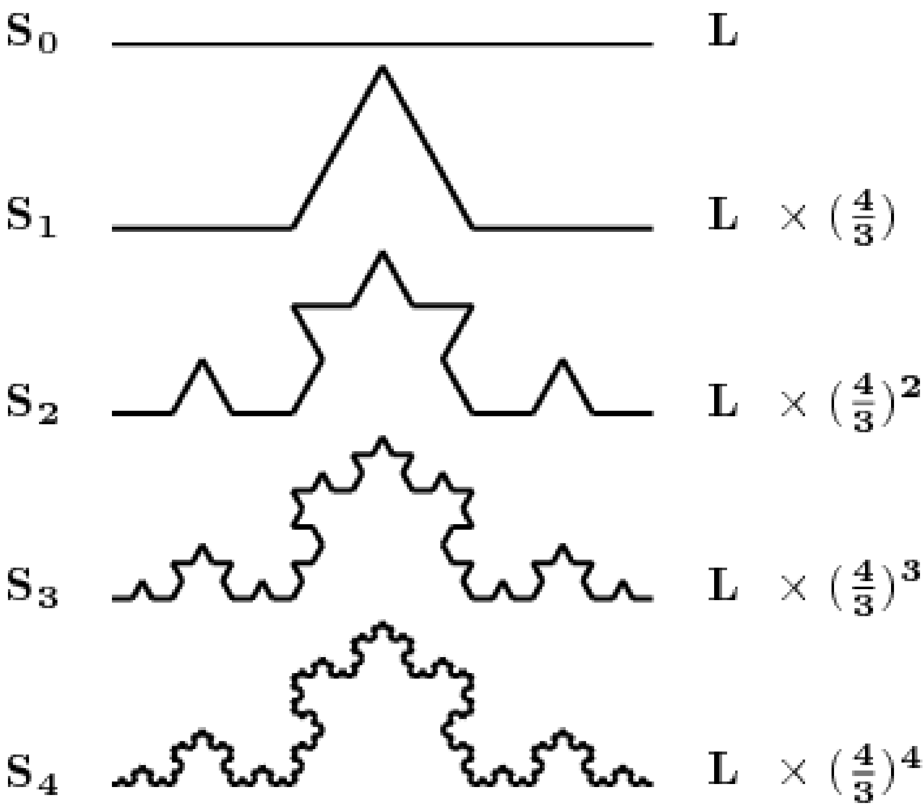
# Length of Koch Curve

- A line of length L is koched
  - $k_o$ has length L instead of unit length



# Length of Koch Curve

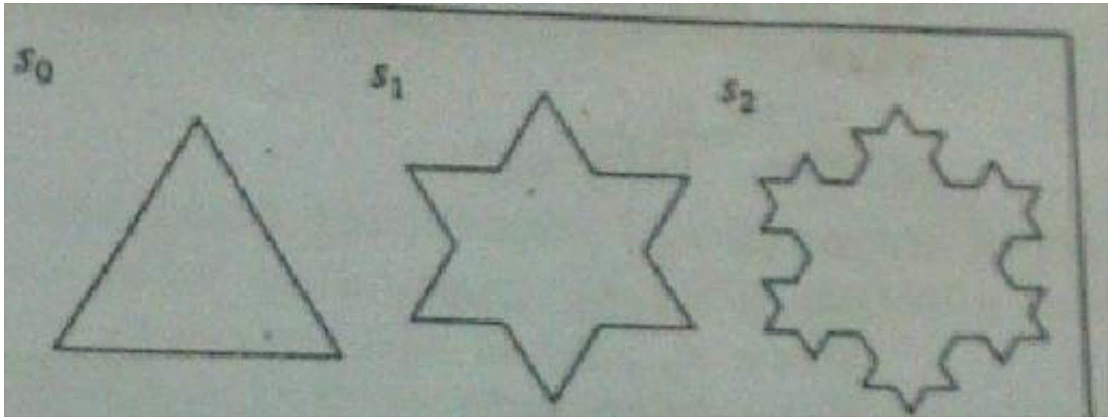
- A line of length L is koched

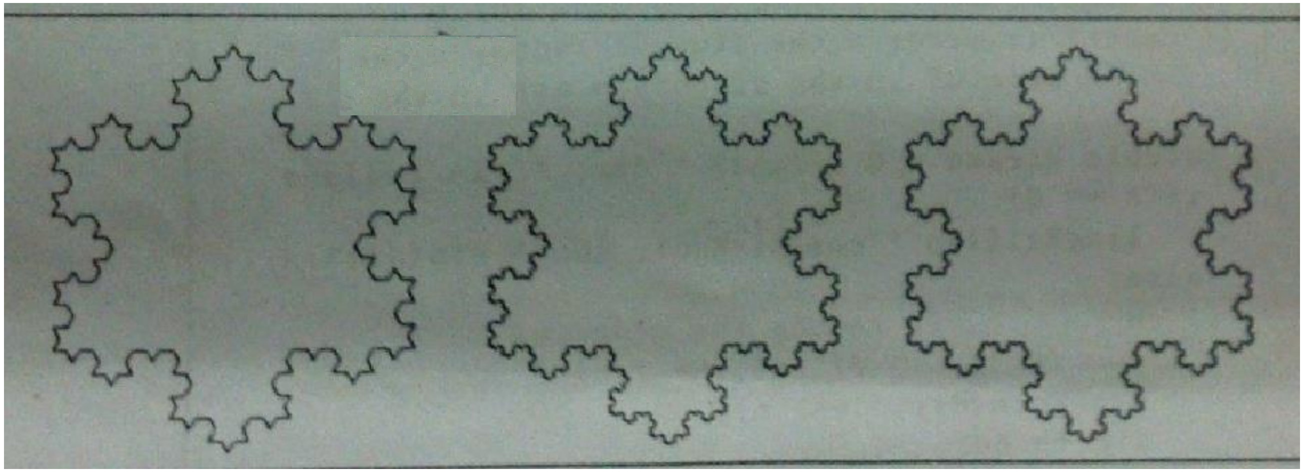



**Length of $k_n$**
**$= L\ (4/3)^n$**

# Koch Snowflake

- Starts with three koch curves joined together

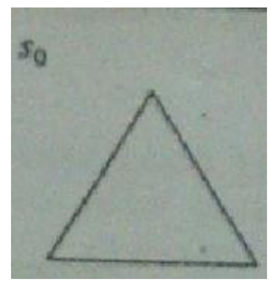


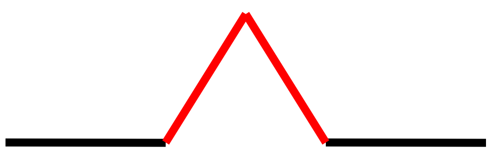
# Perimeter of Koch Snowflake

- Perimeter of koch snowflake curve of $i^{th}$ generation $S_i$ is three times the length of a simple koch curve

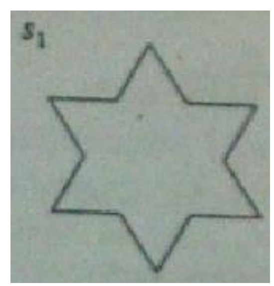
**Length = 1**



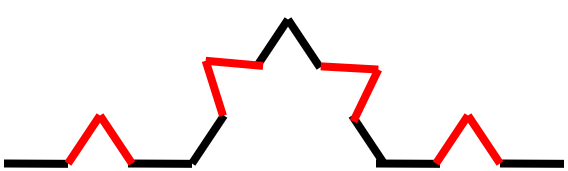
**Perimeter= 3 x 1**



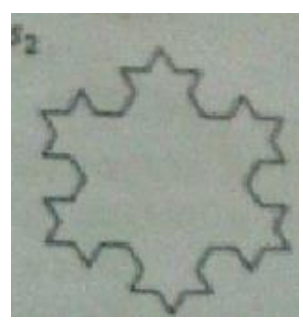
**Length = (4/3)**



**Perimeter= 3 x (4/3)**



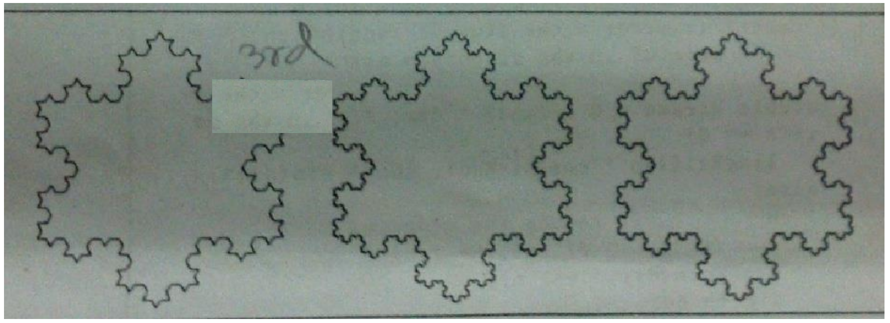
**Length = $(4/3)^2$**



**Perimeter= 3 x $(4/3)^2$**

# Perimeter of Koch Snowflake

- Perimeter of koch snowflake curve of $i^{th}$ generation $S_i$ is three times the length of a simple koch curve:
  **Perimeter of $S_i$ = 3 x length of $k_i$ = $3\ (4/3)^i$**
- As i increases, perimeter increases, boundary becomes rougher



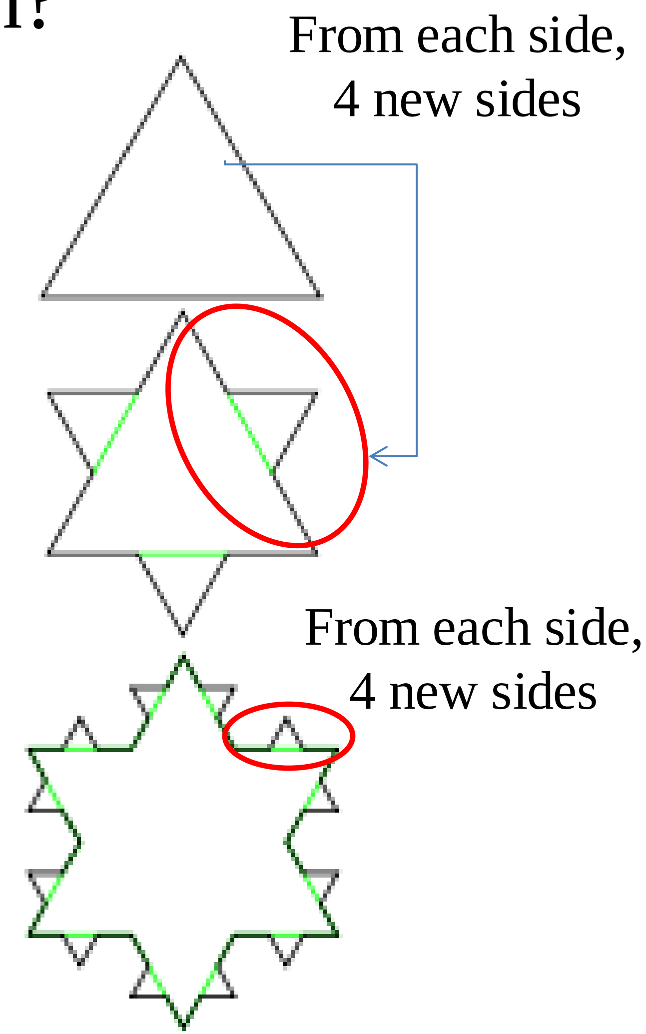

But the area remains bounded!

# Number of Sides of Koch Snowflake

- How many sides at generation n?
  - $n_0 = 3$
  - $n_1 = 4 \text{ x } n_0$ (=12)
  - $n_2 = 4 \text{ x } n_1 = 4 \text{ x } 4 \text{ x } n_0 = 4^2 n_0$
  - $n_3 = 4^3 n_0$

  …

  …
  - $n_n = 4^n n_0 = 3 \text{ x } 4^n$



# Size of the Sides of Koch Snowflake

- Size of sides at iteration n?
- $S_n = 1/3^n$ (formula from koch curve)
  or $L/3^n$
- Perimeter of Koch Snowflake at iteration n:

  $S_n \text{ x } n_n = (1/3^n) \text{ x } (3 \text{ x } 4^n)$

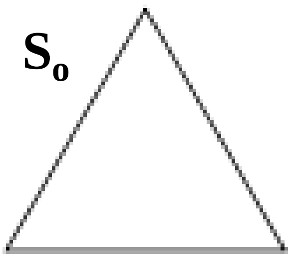
  or $(L/3^n) \text{ x } (3 \text{ x } 4^n)$

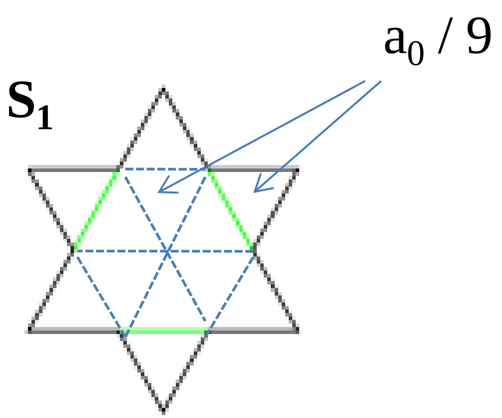
  $= 3L(4/3)^n$

# Area of Koch Snowflake

- Area of $S_o = a_0$
- Area of $S_1 = a_0 + 3(a_0/9)$
  $= a_0 + a_0/3$
  $= a_0(1+1/3)$

- How do we get area of $S_n$?

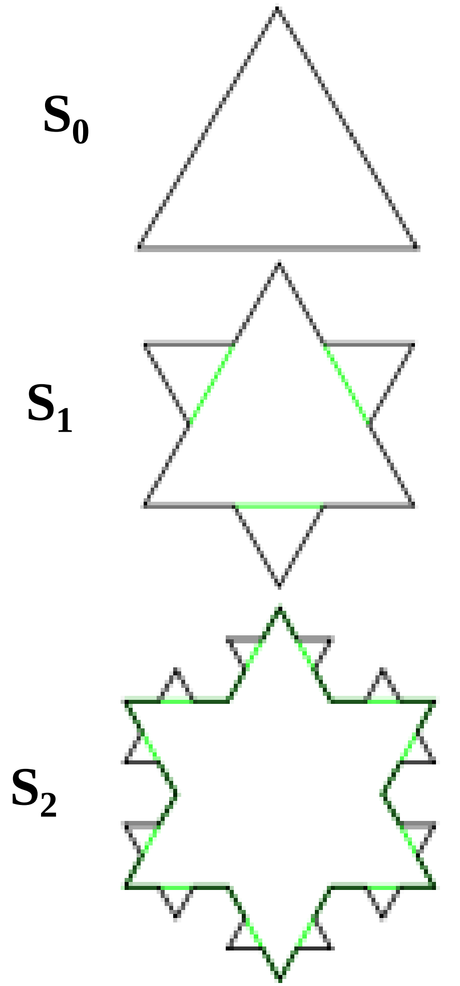


# Area of Koch Snowflake

In each iteration a new bump/triangle is added at each side of previous iteration

- For $S_1$,
  - # of new triangles
  - = # of sides in $S_0$
  - = $n_0 = 3$
- For $S_2$,
  - # of new triangles
  - = # of sides in $S_1$,
  - = $n_1 = 3 \times 4$

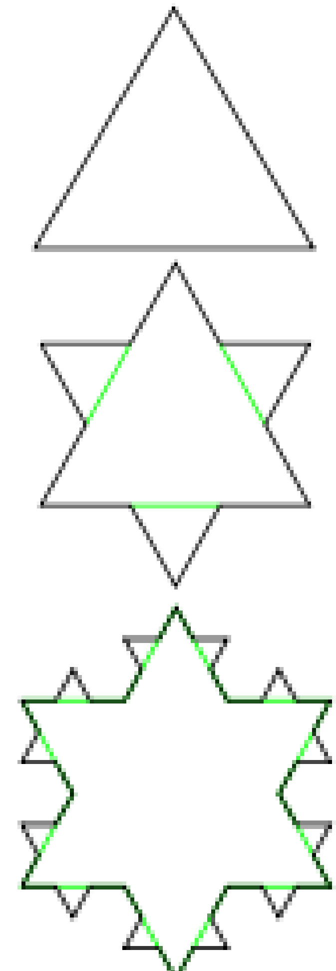


# Area of Koch Snowflake

- For $S_n$,
  – # of new triangles
  = # of sides in $S_{n-1}$,
  = $n_{n-1} = 3 \times 4^{n-1}$

  – New area included
  = $n_{n-1}$ x size of new bump/triangle

- How do we get the size of each of the new bumps/triangles at n-th generation?



$S_0$

$S_1$

$S_2$

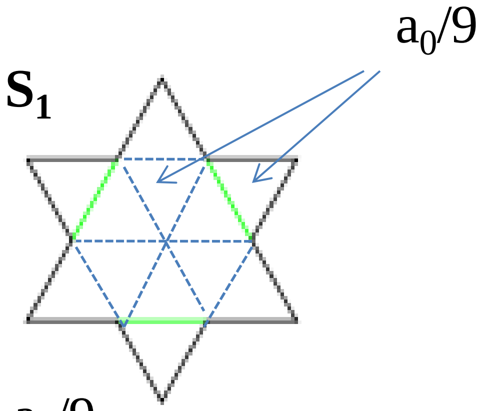
# Area of Koch Snowflake

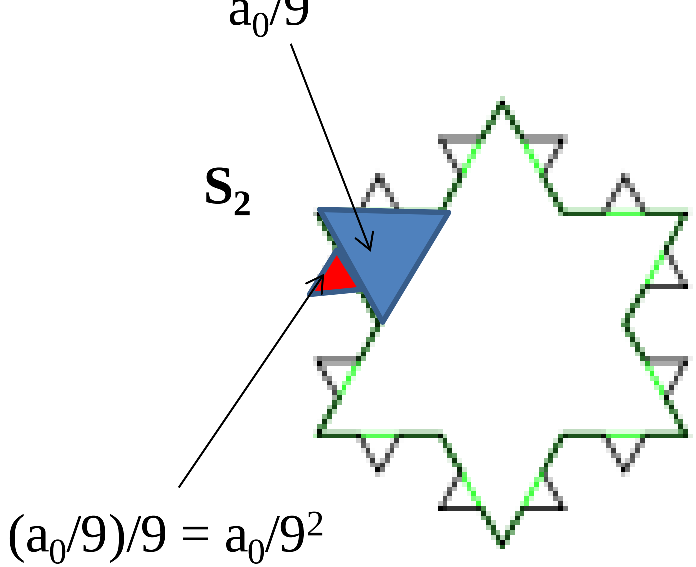
- Area of each small bump in $S_1$ $= a_0 / 9$
- Area of each small bump in $S_2$ $= a_0 / 9^2$
- Area of each small bump in $S_3$ $= a_0 / 9^3$

  …

  …

- Area of each small bump in $S_n$ $= a_0 / 9^n$



# Area of Koch Snowflake

- For $S_n$,
  - # of new triangles
  
    = # of sides in $S_{n-1}$,
    
    = $n_{n-1} = 3 \times 4^{n-1}$

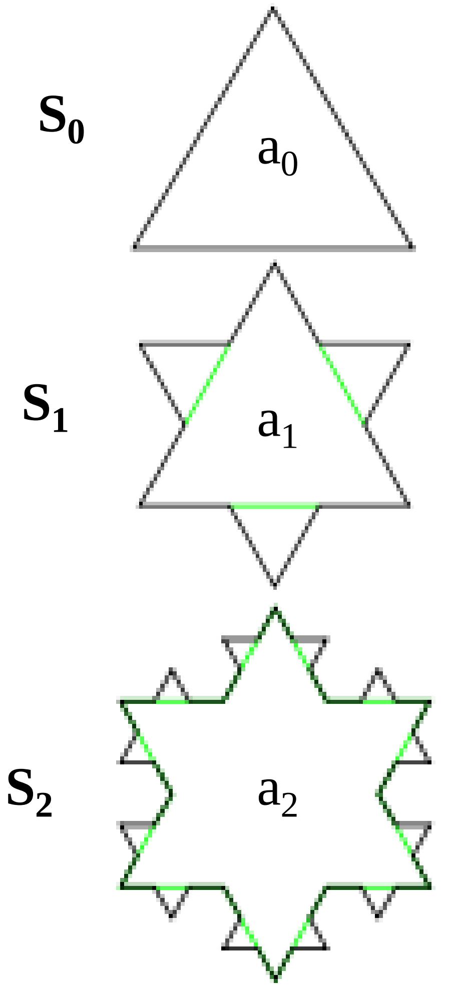
  - New area included
  
    $= n_{n-1}$ x size of new bump/triangle
    
    $= n_{n-1} \times S_n$
    
    $= (3 \times 4^{n-1})\ (a_0 / 9^n)$
    
    $= a_0\ (3/4)\ (4/9)^n$

# Area of Koch Snowflake

- Total area of $S_1 = a_1$
  $= $ Total area of $S_0$ + New area for $S_1$
  $= a_0 + a_0(3/4)\,(4/9)^1$
- Total area of $S_2 = a_2$
  $= $ Total area of $S_1$ + New area for $S_2$
  $= a_1 + a_0(3/4)\,(4/9)^2$
  $= a_0 + a_0(3/4)\,(4/9)^1 + a_0(3/4)\,(4/9)^2$

…

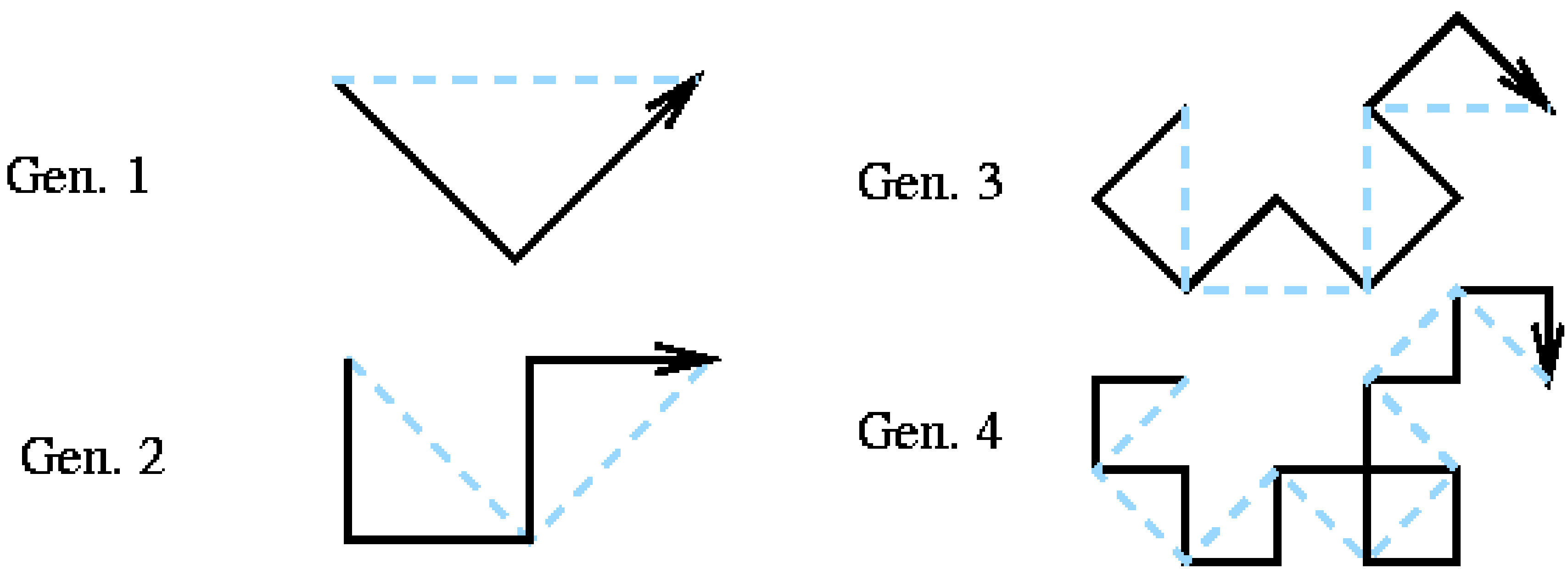
- Total area of $S_n = a_n$
  $= a_{n-1} + a_0(3/4)\,(4/9)^n$
  $= a_0 + a_0(3/4)\,((4/9)^1 + (4/9)^2 + \ldots + (4/9)^n)$
  $= a_0/5\,(8 - 3(4/9)^n)$



# Dragon Curve

- Starting from a base segment, replace each segment by **2 segments with a right angle** and with a rotation of 45° alternatively to the right and to the left.

Gen. 0

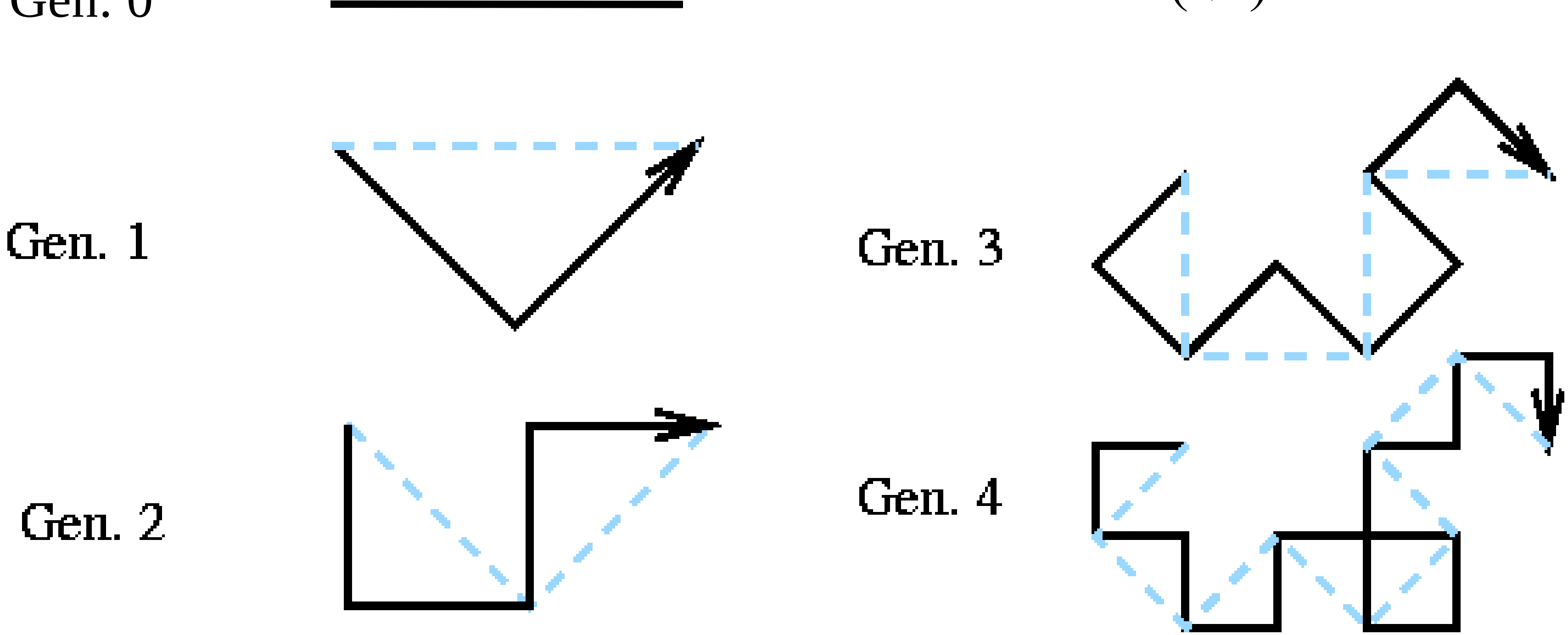


# Dragon Curve

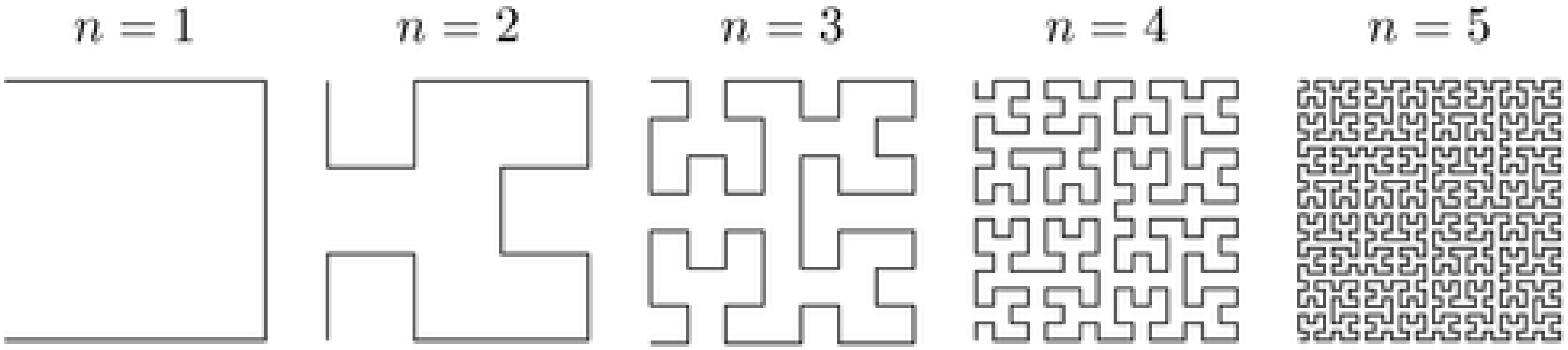
- Length
  - Gen 0: $L$; Gen 1: $2 \times (L/\sqrt{2}) = \sqrt{2}\, L$; Gen 2: $2 \times (2 \times (L/\sqrt{2})/\sqrt{2})$

    $= L\,(2 \times 2\,/\,2) = 2\,L$;

    … Gen n: $(\sqrt{2})^n\, L$



# Hilbert Curve

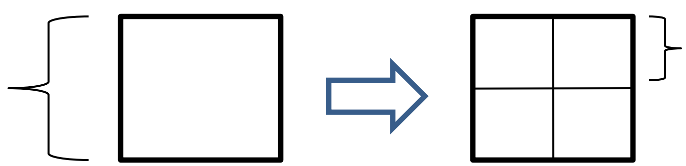



Hilbert proved that, as the order tends to infinity, the infinitely thin, continuous line of the curve passes *through every point of the unit square*.

# Dimension of Fractals

- Fractals have infinite length but occupied in finite region
- Lets apply this concept for simple line, square, and cube.
- A straight line having unit length is divided into N equal segments, then length of each side is $r = 1/N$
- A square having unit length sides, divided into N equal squares, then side of each small squares is $r = 1/(N)^{1/2}$

For N=4, we see

1  ½ → $1/(4)^{1/2}$

$r = 1/N^{1/D}$
D= Dimension

- Similarly, a cube having unit length sides if divided into N equal cubes, then each side will have length $r = 1/(N)^{1/3}$

You can check out Hausdorff dimension, fractional dimension for more details if interested!!

# Dimension of Fractals

- $r = 1/N^{1/D}$
- $N^{1/D} = 1/r$
- $\log N^{1/D} = \log (1/r)$
- $(1/D) \log N = \log (1/r)$
- $\boxed{D = \log N / \log (1/r)}$

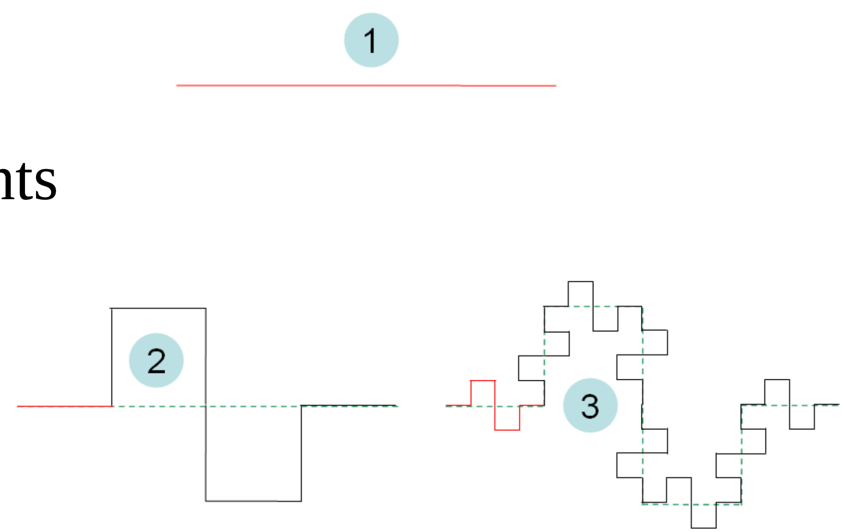
# Dimension of Fractals

**Koch Curve:**

$k_0$ to $k_1$ → the base segment is divided into **N = 4** equal segments each having length **r = 1/3**

So dimension is D = log N / log (1/r) = log 4 / log (3) = 1.26

**Quadratic Koch Curve:**




$Q_0$ to $Q_1$ → the base segment is divided into **N = 8** equal segments each having length **r = 1/4**

So dimension is D
= log N / log (1/r)
= log 8 / log (4) = 1.5

# Dimension of Fractals

**Dragon Curve:**

$d_0$ to $d_1$ → the base segment is divided into **N = 2** equal segments each having length $\mathbf{r = 1/2^{1/2}}$

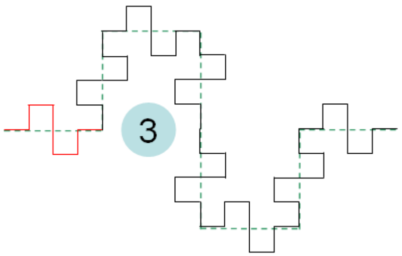
So dimension is D $= \log N / \log (1/r) = \log 2 / \log (\mathbf{2^{1/2}})$
$= 2$ → dimension of square!

# Dimension of Fractals

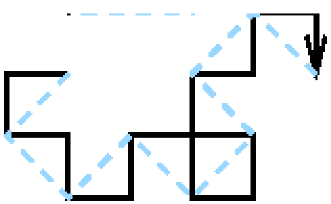
So what we have learned?

- Koch curve has dimension 1.26
- Quadratic koch curve has dimension 1.5
- Dragon curve has dimension 2

S 4




Gen. 4



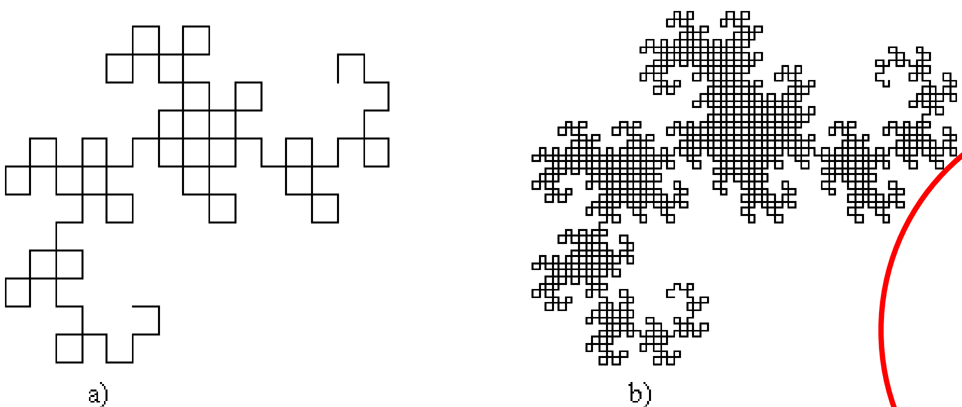



Figure 1. The Dragon Curve: a) after 7 folds, b) after 11 folds

Curve becomes more plane filling as the dimension increases

# Peano Curves

- Fractal curves that have dimension 2 are called **Peano** curves
  - Dragon curve
  - Gosper curve
  - Hilbert curve

# String Production Rule to Draw Fractals

- The pseudocode can be converted to a set of String Production Rules (L-System)
- Koch Curve

F → F – F + + F – F

F = go forward

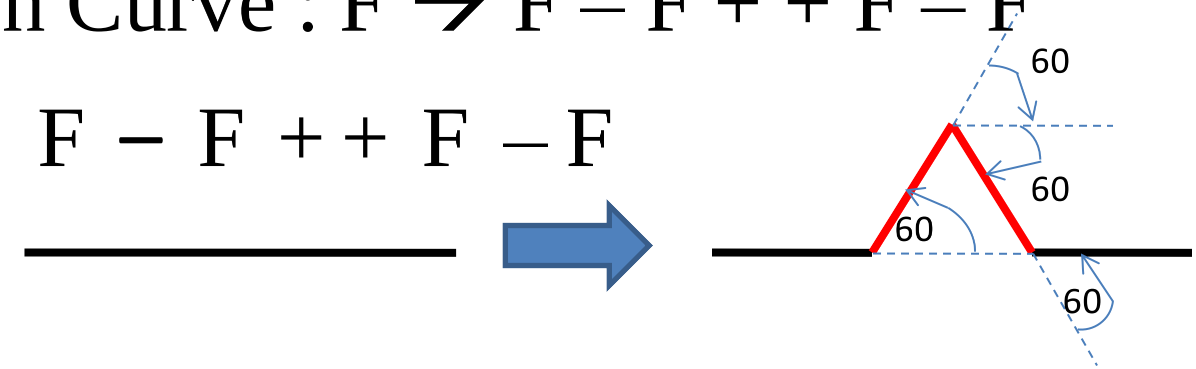
– = turn left through angle A degrees [A = 60]

+ = turn right through angle A degrees  [A = 60]

# String Production Rule to Draw Fractals

- Koch Curve : F $\rightarrow$ F – F + + F – F

F $\rightarrow$ F – F + + F – F

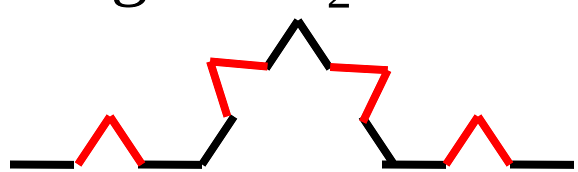



This gives us first generation

- $k_1$ = F – F + + F – F

Replacing each F with same rules gives $k_2$

- $k_2$ = (F – F + + F – F) –
  (F – F + + F – F) + +
  (F – F + + F – F) – (F – F + + F – F)

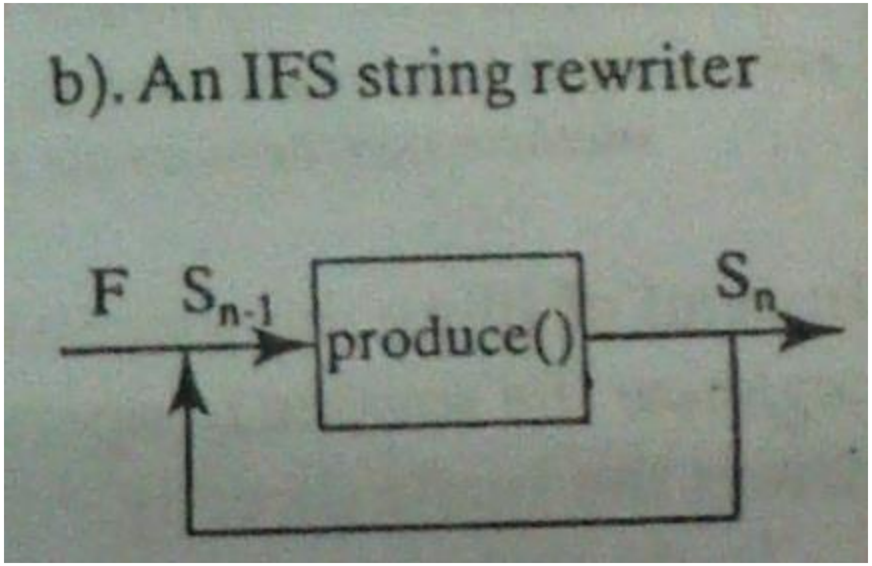


Applying repeatedly gives latter generations

# String Production Rule to Draw Fractals

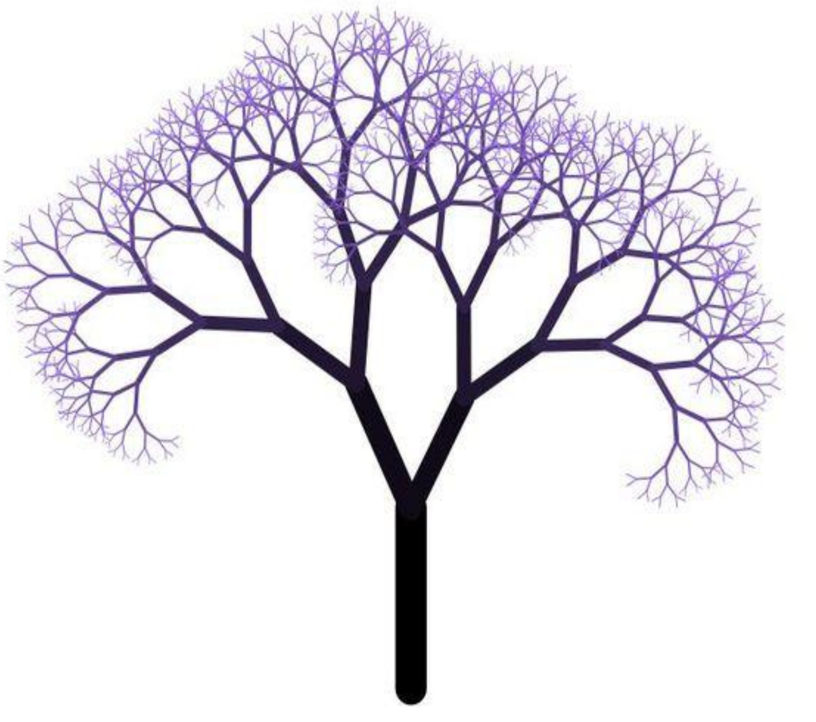
This technique is called Iterated Function System (IFS)

- A string is repeatedly fed back into the same function to produce the next higher order object



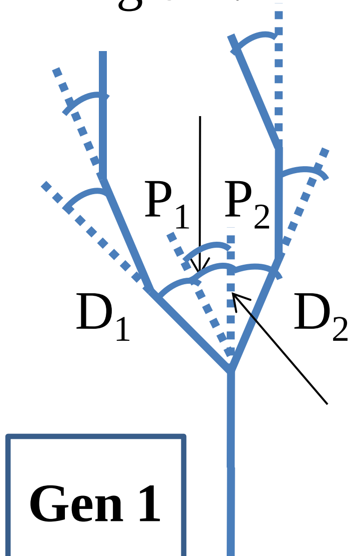
# Allowing Branching

- String production rule with special symbol for saving current state and restore to saved state
- Current State
  - Current position, CP
  - Current  direction, CD
- Saving current state: [
- Restore the last saved state: ]
- Implementation: Using Stack
  - Save state: Push
  - Restore last saved state: Pop

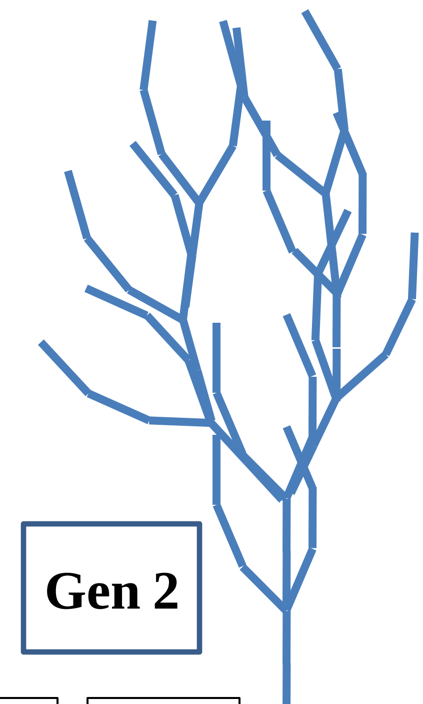


# Allowing Branching

- Example: Fractal Tree
- F → "FF – [ – F + F + F ] + [ + F – F – F ] "
- Atom : F
- Angle A: $22^o$




Replace each **F**
by the same rule




Saved State, Current State = $P_1, D_1$ | Ø, $P_1$ | $P_2, D_2$ | Ø, $P_2$

Thank you ☺