Day Four

Introduction to Programming

Exploring Computer Science

Unit 1: Human Computer Interaction
Unit 2: Problem Solving
Unit 3: Web Design
Unit 4: Introduction to Programming

- Day 1: Intro to Scratch
- Days 2-3: Create a simple Scratch program
- Day 4: Dialogue between Scratch sprites
- Days 5-6: Moving sprites in Scratch
- Days 7-8: Event driven programming
- Day 9: Broadcasting events in Scratch
- Day 10-13: Stories in Scratch
- Day 14: Variables
- Day 15: Conditionals
- Day 16-17: And, Or, & Randomness
- Day 18: Rock, Paper, Scissors
- Day 19: Create a timer
- Day 20-23: Creating timing games
- Day 24-30: Final Projects

Unit 5: Computing Applications
Unit 6: Robotics

Intro to Computer Programming

- Programming is the *stereotypical* CS activity.
- But it's really not about screenfuls of odd-looking text.

```
PLEASE DO ,1 <- #13
DO ,1 SUB #1 <- #238
DO ,1 SUB #2 <- #112
DO ,1 SUB #3 <- #112
DO ,1 SUB #4 <- #0
DO ,1 SUB #5 <- #64
DO ,1 SUB #6 <- #238
DO ,1 SUB #7 <- #26
DO ,1 SUB #8 <- #248
DO ,1 SUB #9 <- #168
DO ,1 SUB #10 <- #24
DO ,1 SUB #11 <- #16
DO ,1 SUB #12 <- #158
DO ,1 SUB #13 <- #52
PLEASE READ OUT ,1
PLEASE GIVE UP
```

Intercal

```
                        v
>v"Hello world!"0<
,:
^_25*,@
```

Befunge

What ***popular*** programming languages have you heard of?

Programming languages

- What's popular now? (tiobe.com)

Position Jul 2011	Position Jul 2010	Delta in Position	Programming Language	Ratings Jul 2011	Delta Jul 2010	Status
1	1	=	Java	19.251%	+0.58%	A
2	2	=	C	17.280%	-1.20%	A
3	3	=	C++	9.017%	-1.45%	A
4	5	↑	C#	6.221%	+0.49%	A
5	4	↓	PHP	6.179%	-2.39%	A
6	9	↑↑↑	Objective-C	5.181%	+2.68%	A
7	6	↓	(Visual) Basic	5.106%	-0.41%	A
8	7	↓	Python	3.583%	-0.63%	A
9	8	↓	Perl	2.328%	-0.77%	A
10	10	=	JavaScript	2.242%	-0.19%	A

Where are HTML and CSS?

Intro to Computer Programming

- Here are two more *"Hello, World!"* programs:

```
print 'Hello, World!'
```

Python

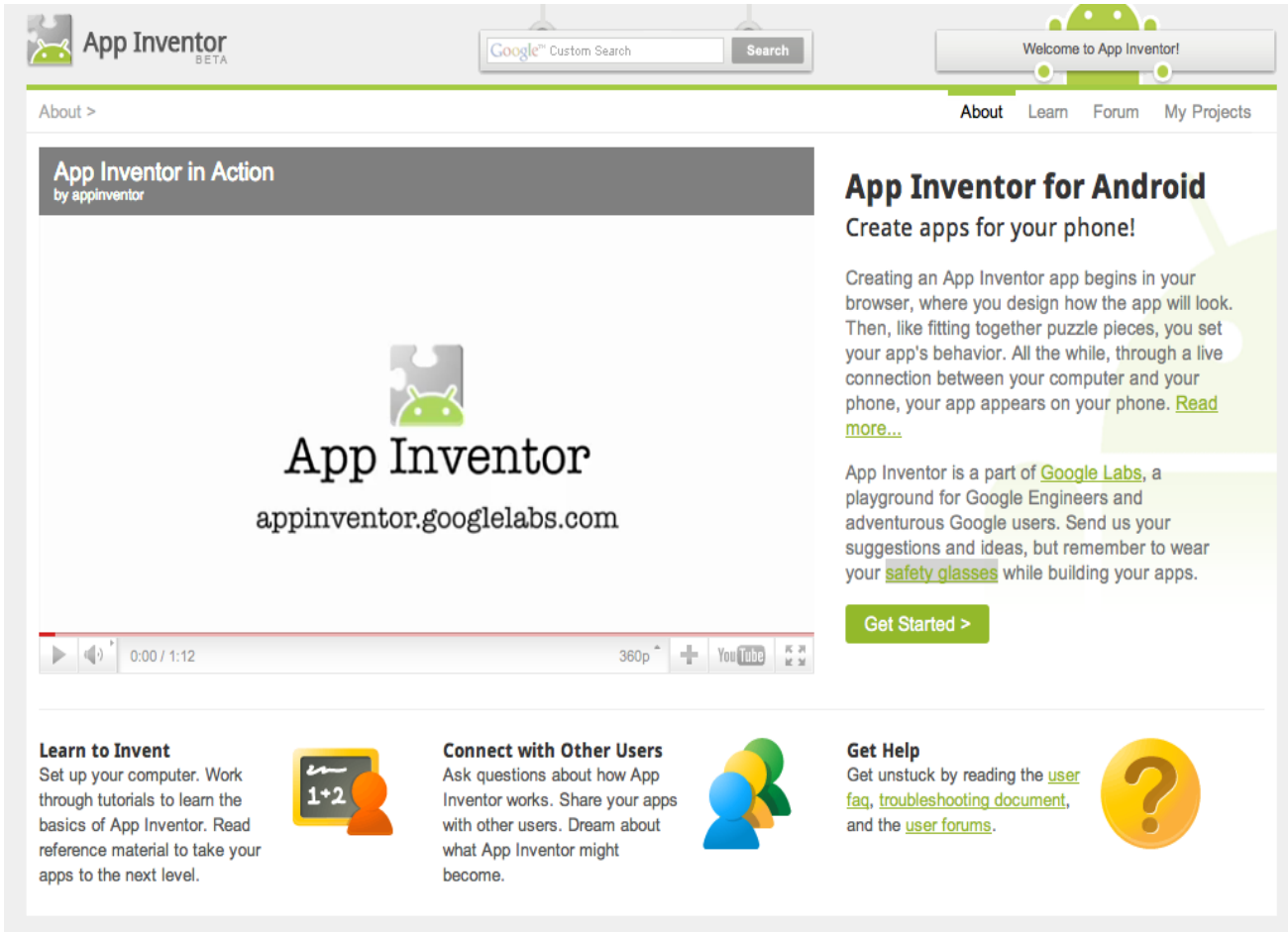
```
int main()
{
  std::cout << "Hello, world!";
}
```

C++

- CS is fundamentally about *problem-solving*,
- and programming is much more like learning a new ***human*** language than students initially suspect...
- programming is just ***translation***

Programming languages

- What does the ***future*** of programming look like?



Graphical interfaces are ever more popular in industry and education.

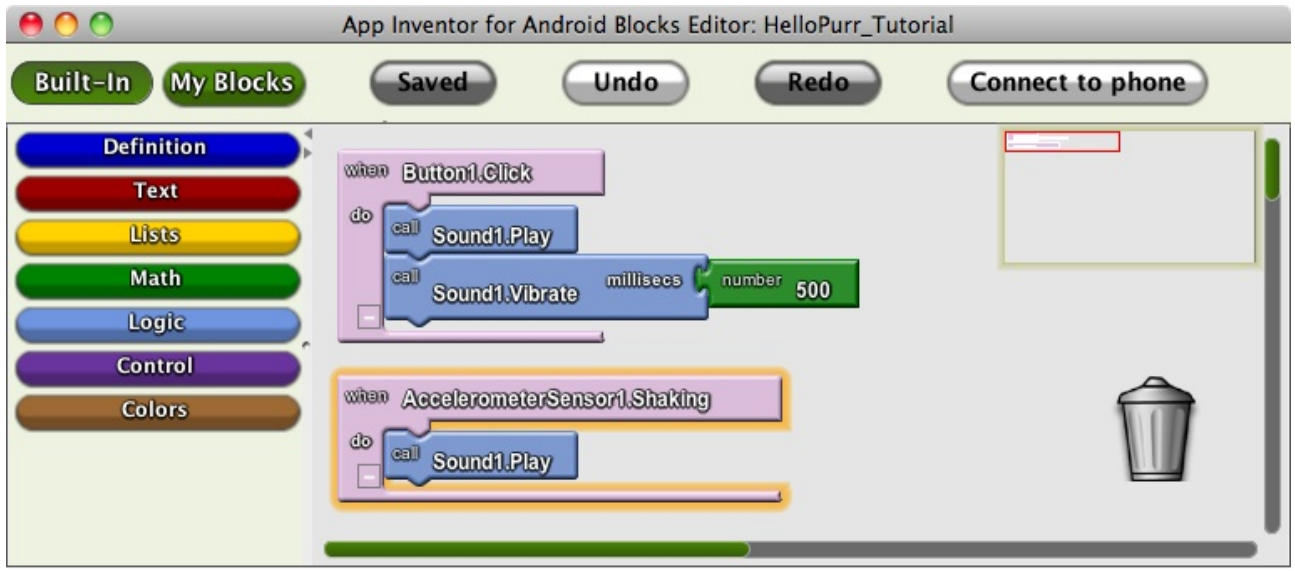
Programming languages

- A "Hello, Kitty" Android Program:

What are these bricks saying?

Programming ***ideas***

- ***Actions*** are the basic building blocks (literally!)...



- The data can be simple.
- The actions can be simple.
- The combinations can be surprising!

Surprising applications?

What looks complicated in biology can often be explained by simple rules

Biology programs, too!

Fractals

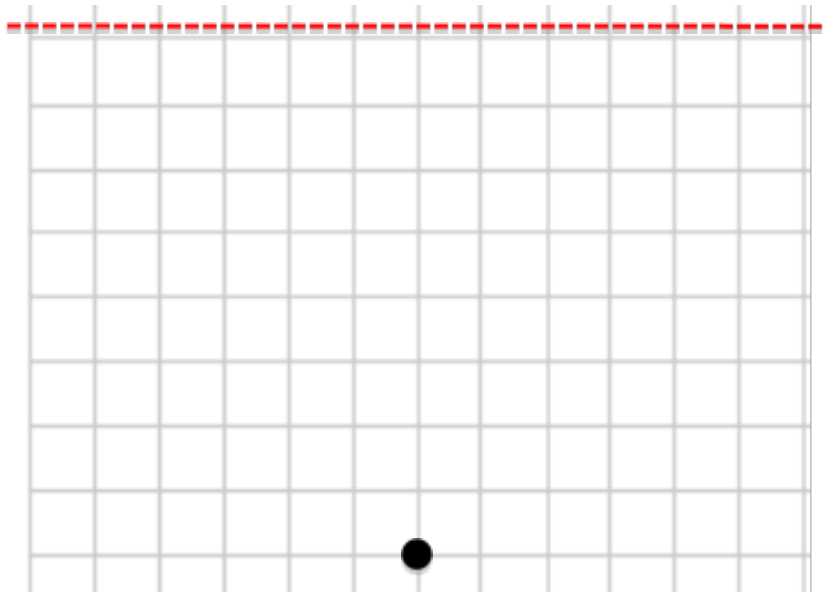
Biology *Rules*?

What looks complicated in biology can often be explained by simple rules

A Tree Program

height = 4 cm

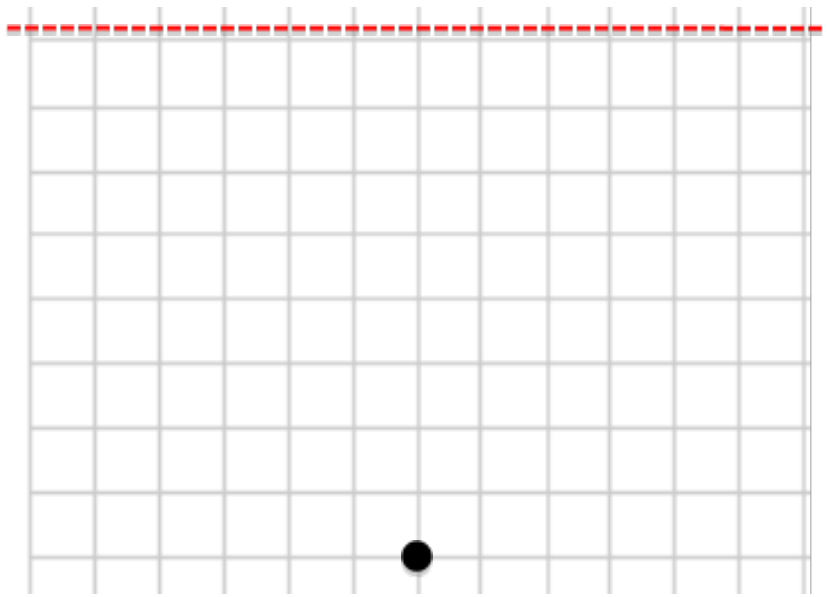
(1) At each new dot:
(2) Draw a **T** with dots on its ends
(3) Divide *height* by 2
Go back to step (1) and continue



A Tree Program

height = 4 cm

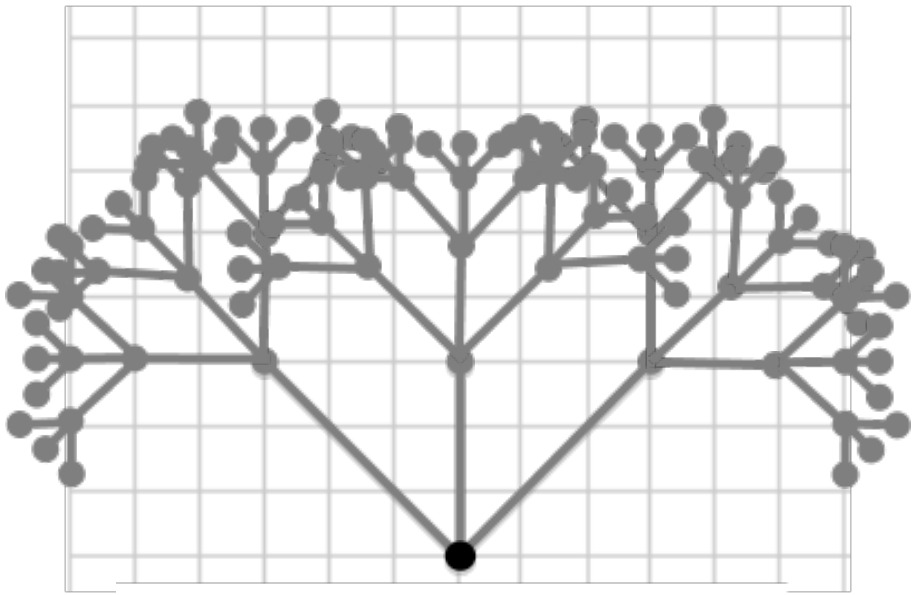
(1) At each new dot:
(2) Draw a **T** with dots on its ends
(3) Divide *height* by 2
Go back to step (1) and continue



---

height = 4 cm

(1) At each new dot:
(2) Draw a **T** with dots on its ends
(3) Divide *height* by 2
Go back to step (1) and continue



Change these rules' underlined parts...

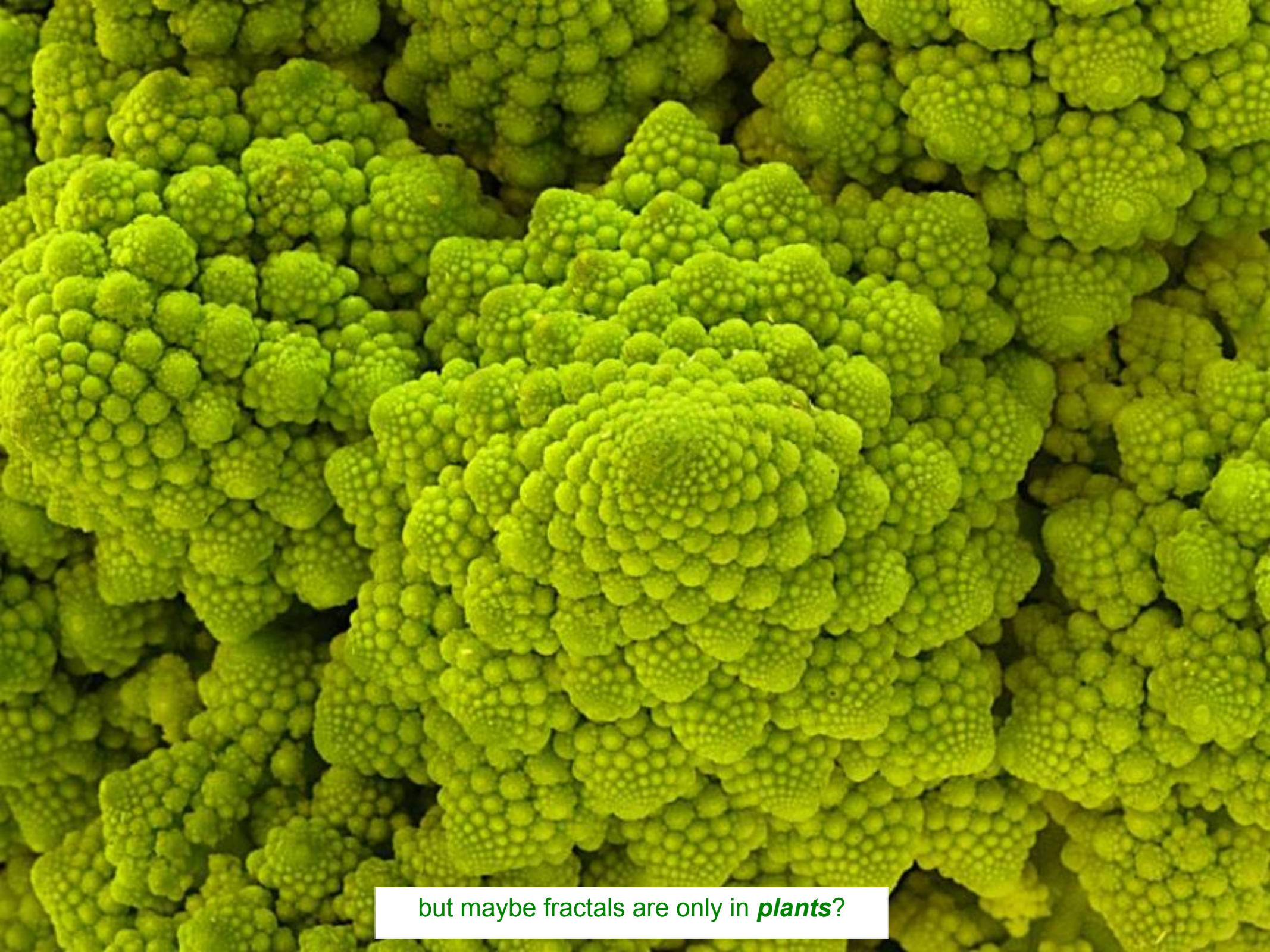
...to create this "tree"

Are these rules for real? Yes... and no.

The rules ***can*** create many different fractal forms

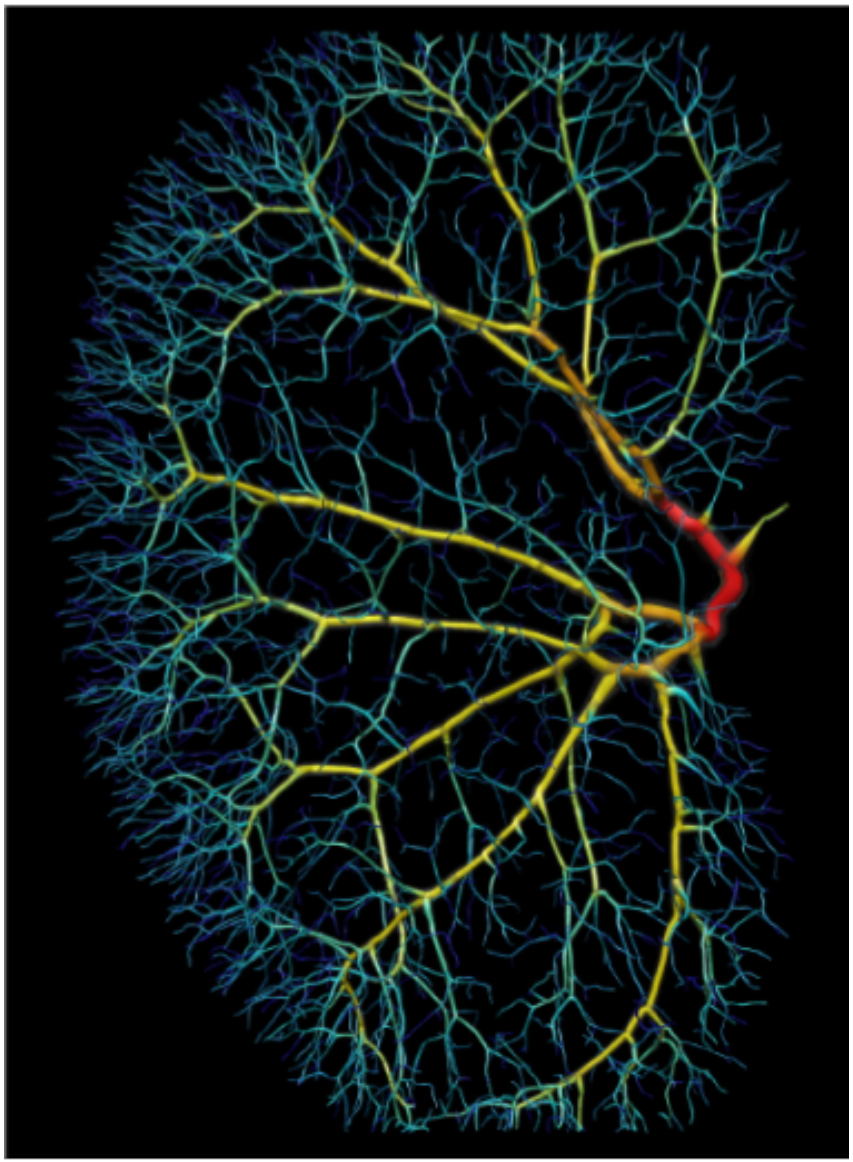
Are these rules for real? **Yes...** and no.

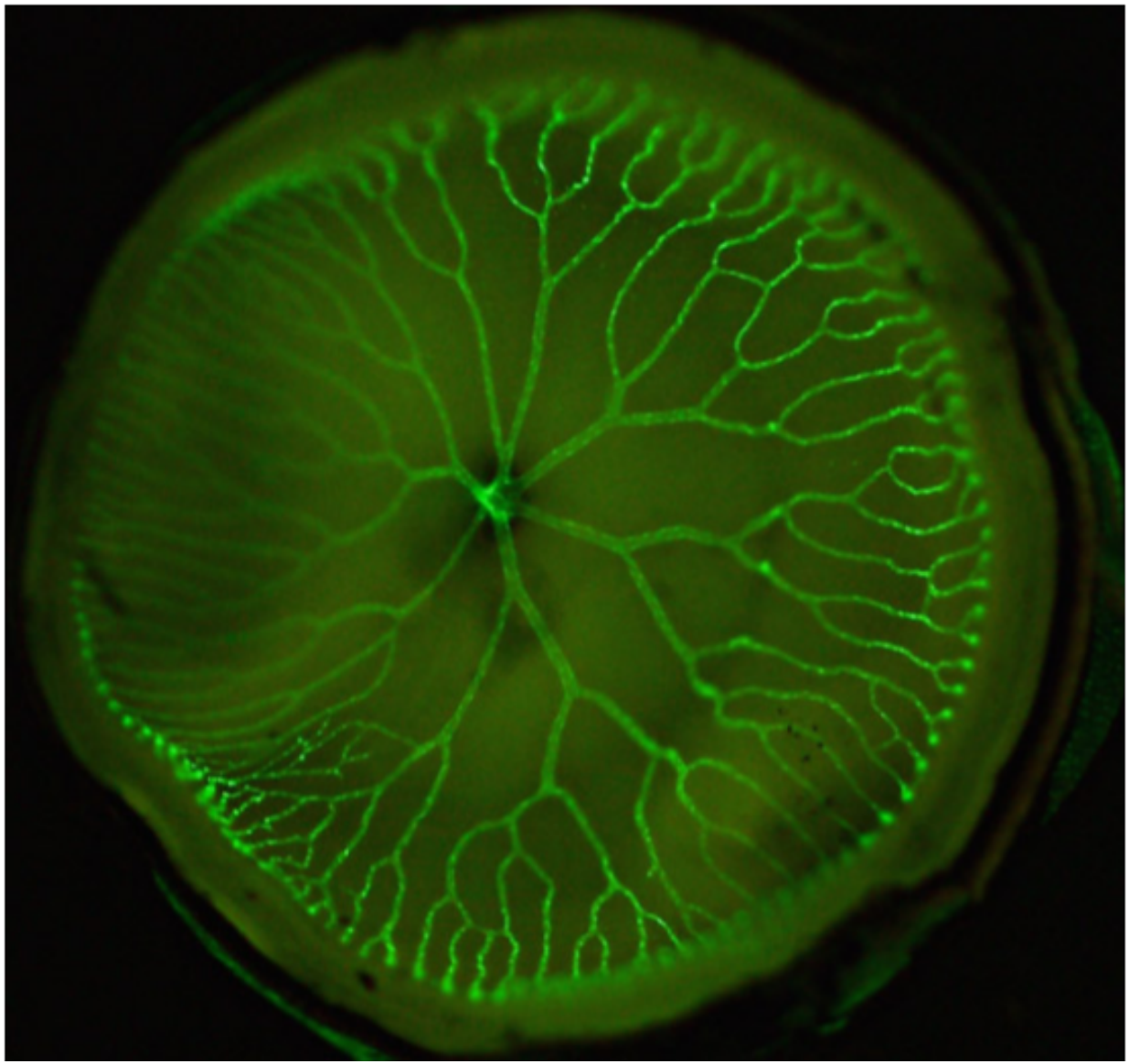
and biology ***does*** create many different fractal forms: Romanesco broccoli



but maybe fractals are only in ***plants***?

Are these rules for real? **Yes...** and no.

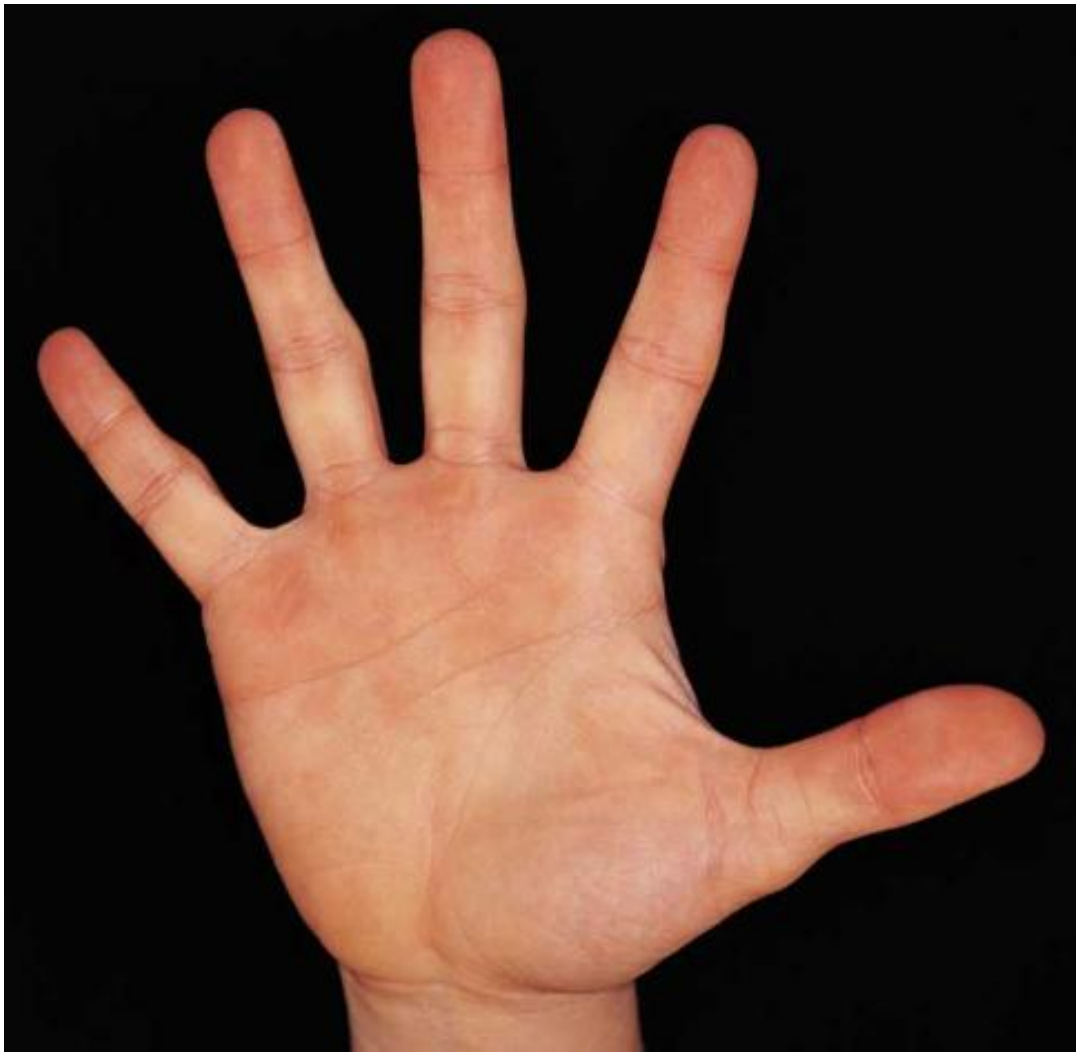




What are these?

Are these rules for real?

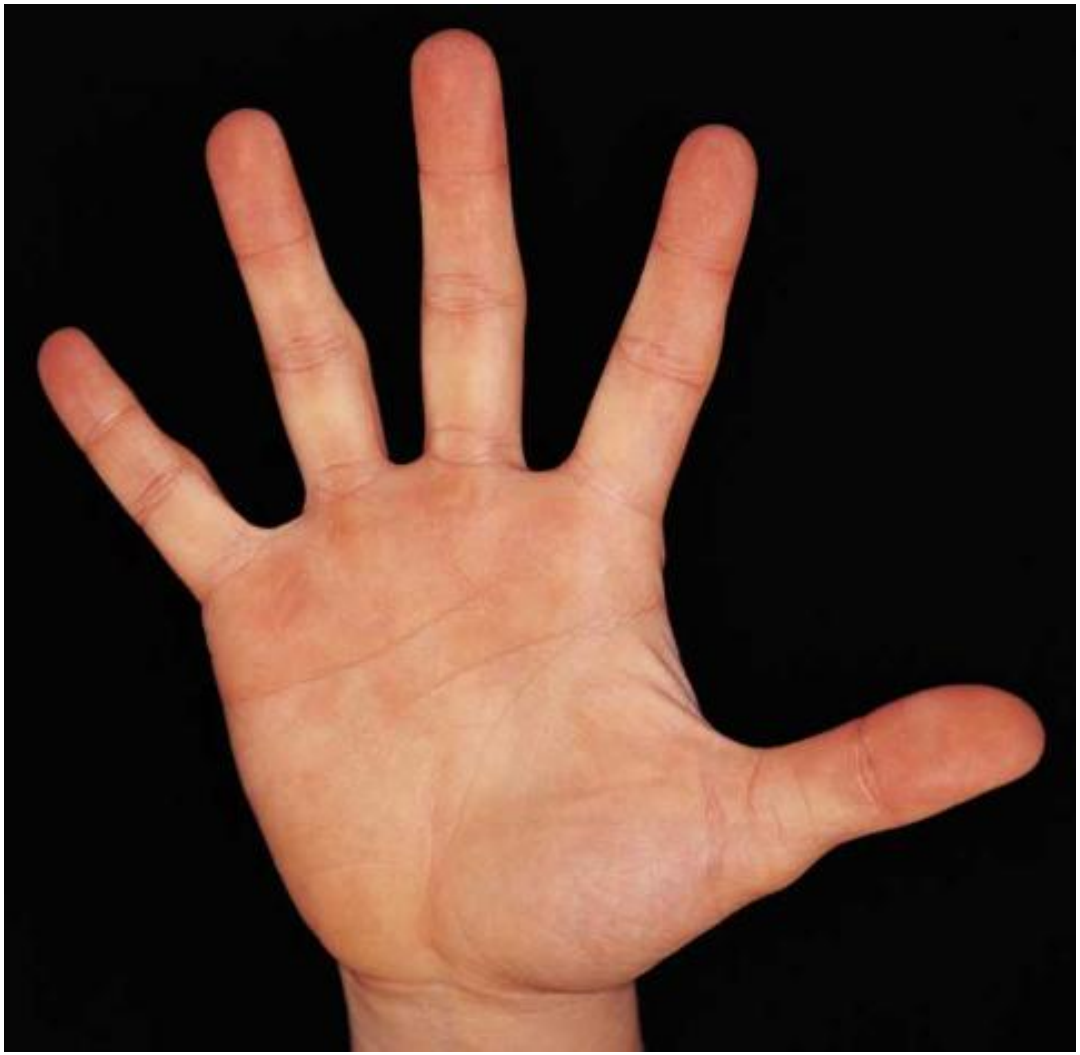
Yes... ***and no.***



All this self-similarity must stop somewhere...

Are these rules for real?

Yes... ***and no.***

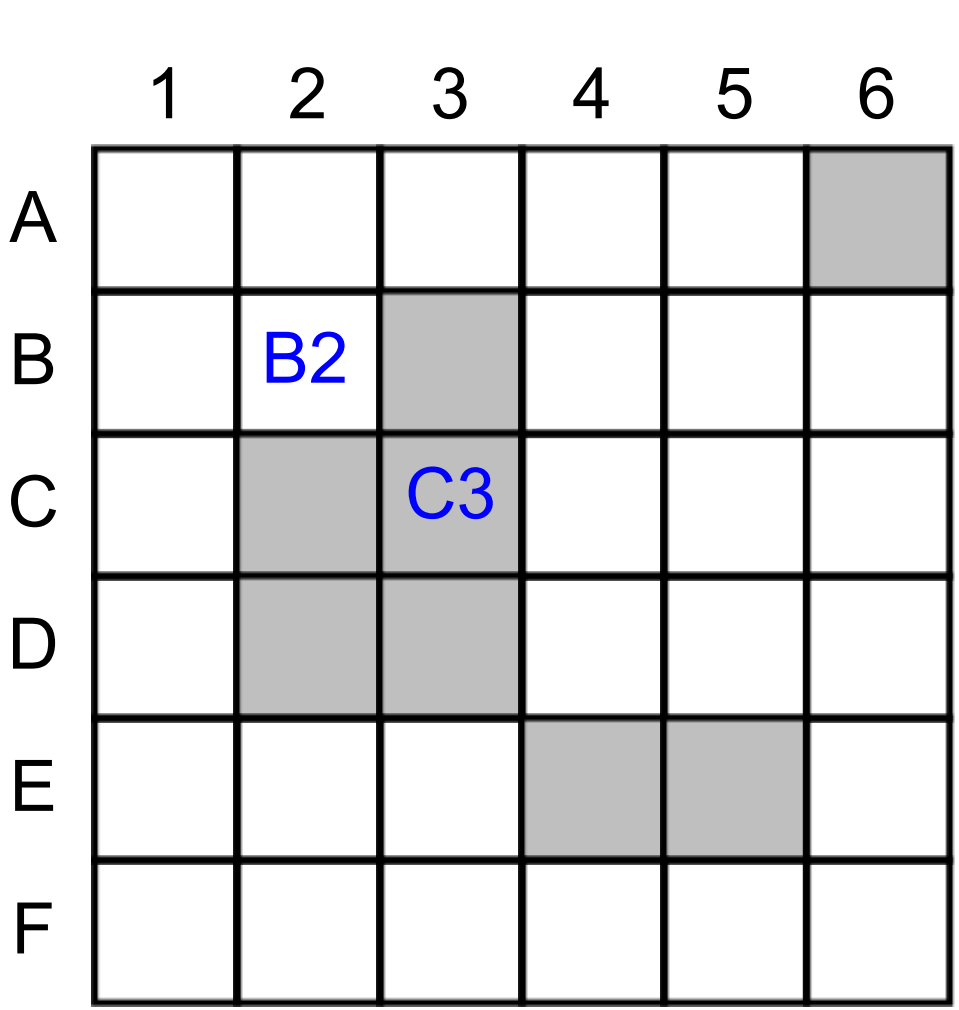


All this self-similarity must stop somewhere...

... or who knows what could happen!?

Simple ***cell*** rules?

Simple *cell* rules



= Living cell

= Empty space

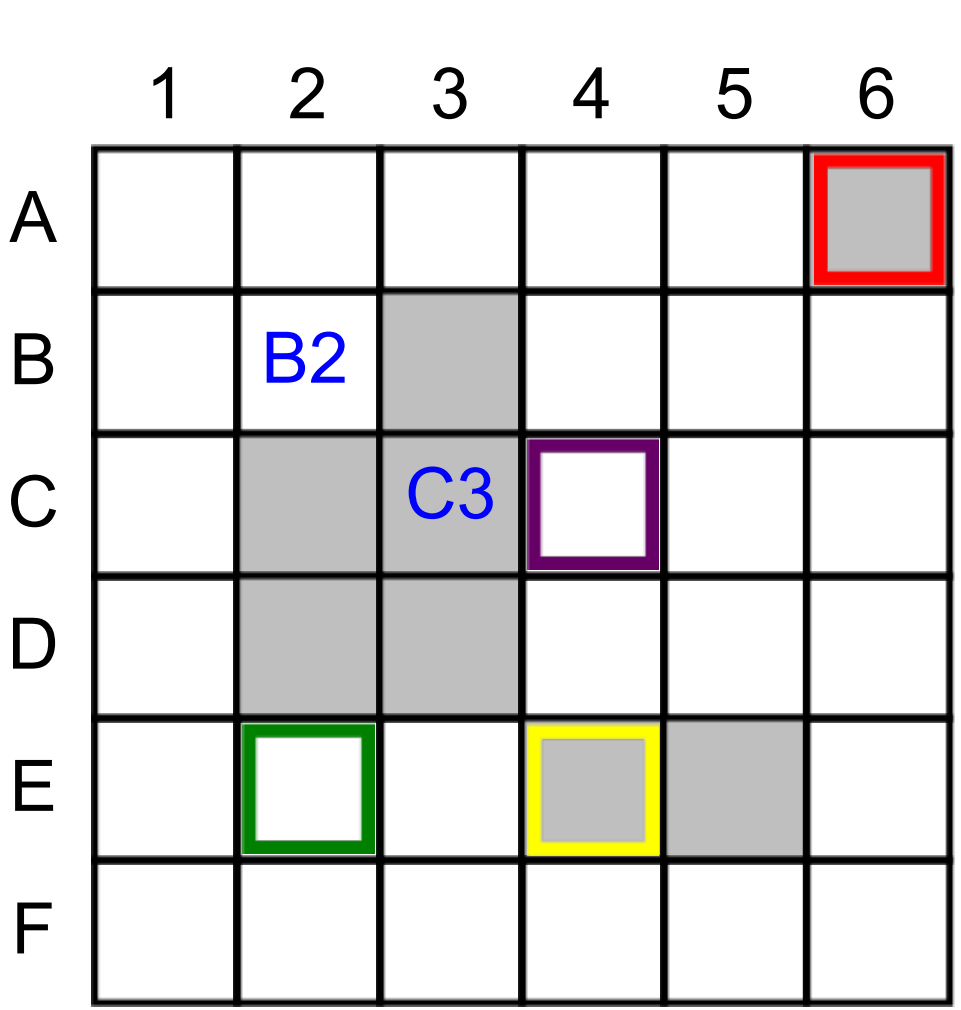
A grid of cells depending on

(1) their rules (DNA)

(2) environment (neighbors)

How many live cells are in this grid?

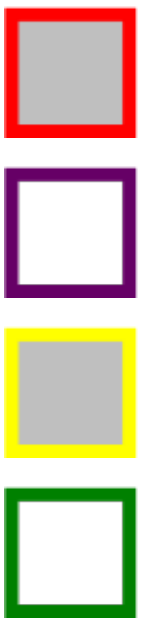
Simple cells



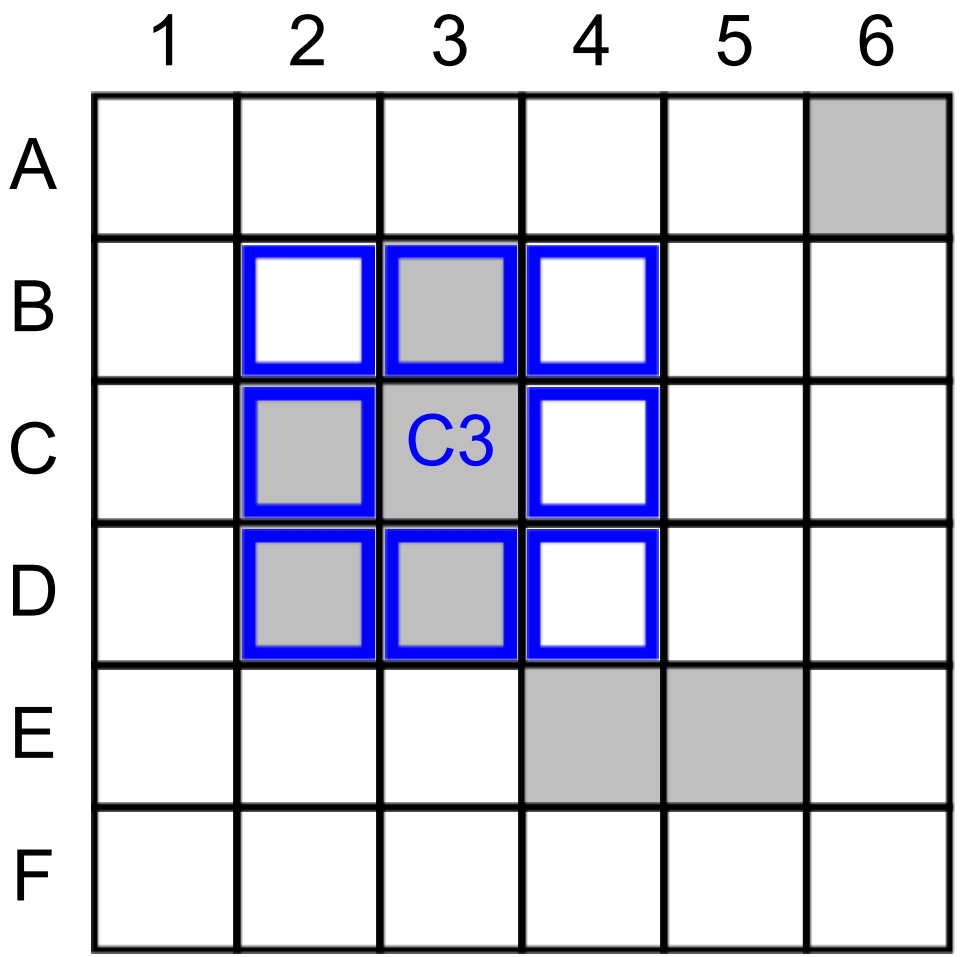
= Living cell

= Empty space

Name	Live or empty?



Neighbor cells



Cell C3 has **8** neighbors

= a living cell

= empty space

Living neighbors

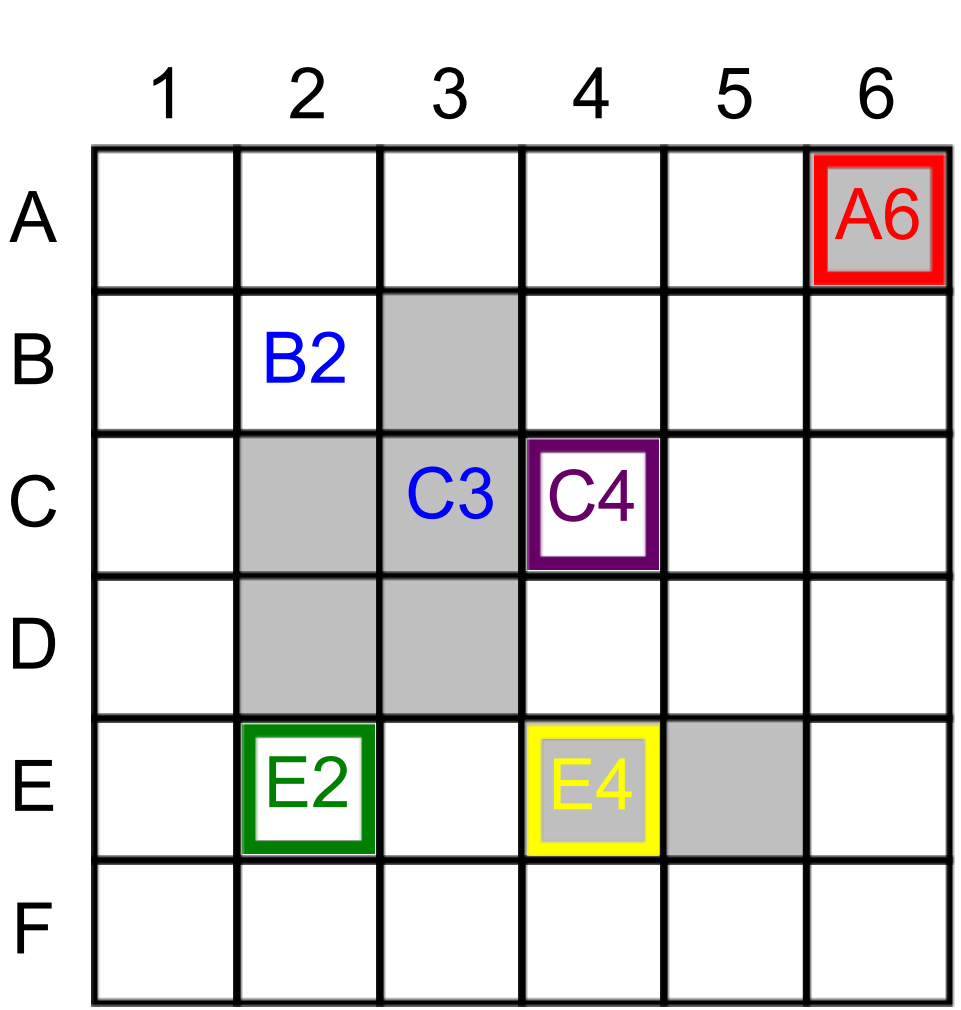
Empty neighbors

C
3

How many are living?

How many are empty?

Neighbor cells



Each cell's future depends on its living neighbors

= Living cell

= Empty space

# of Living neighbors

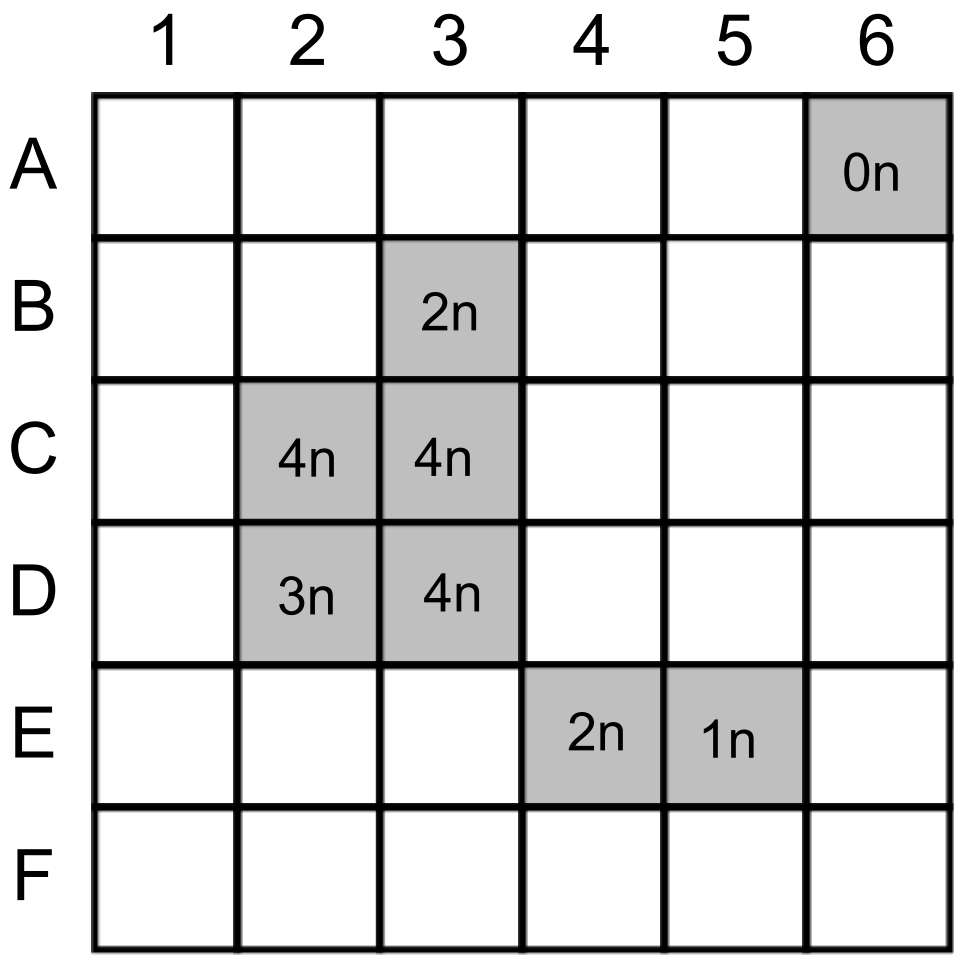
A6

C4

E4

E2

Two rules

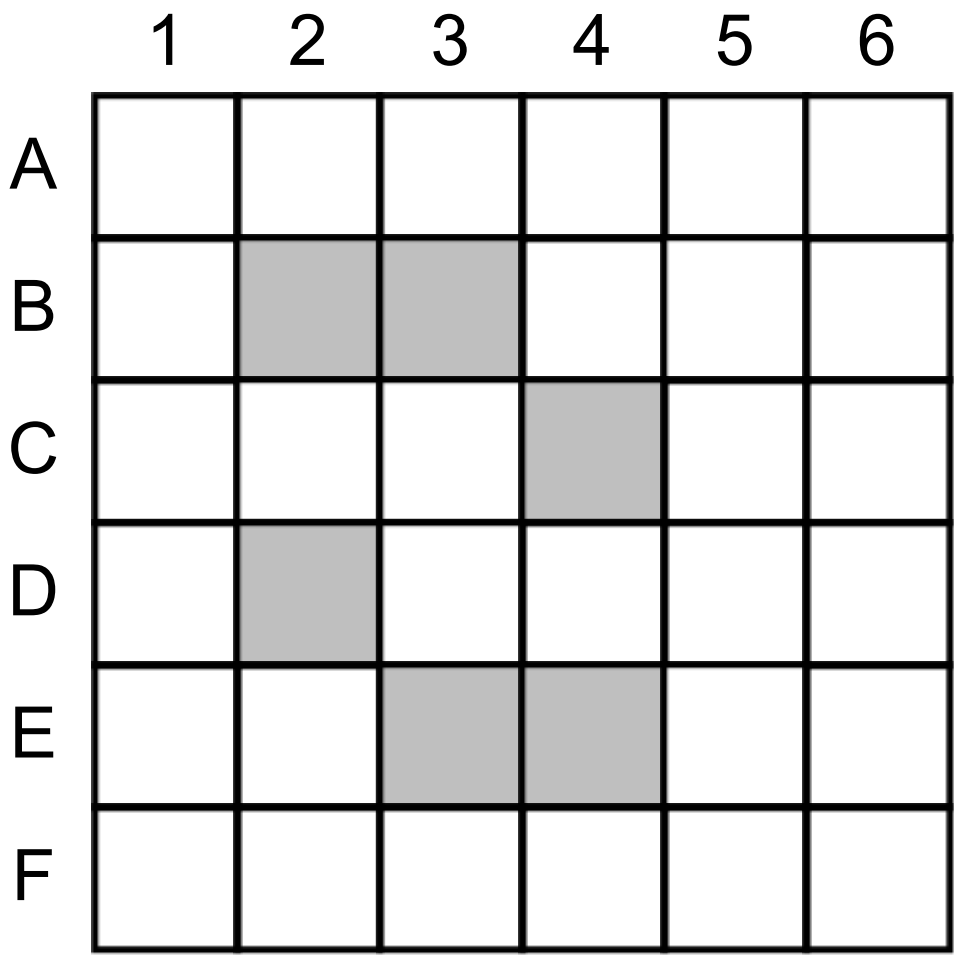


BEFORE

A living cell with ***2 or 3*** living neighbors ***survives***. Others die.

An empty cell with exactly 3 living neighbors ***comes to life***. No others do.

Two rules



AFTER

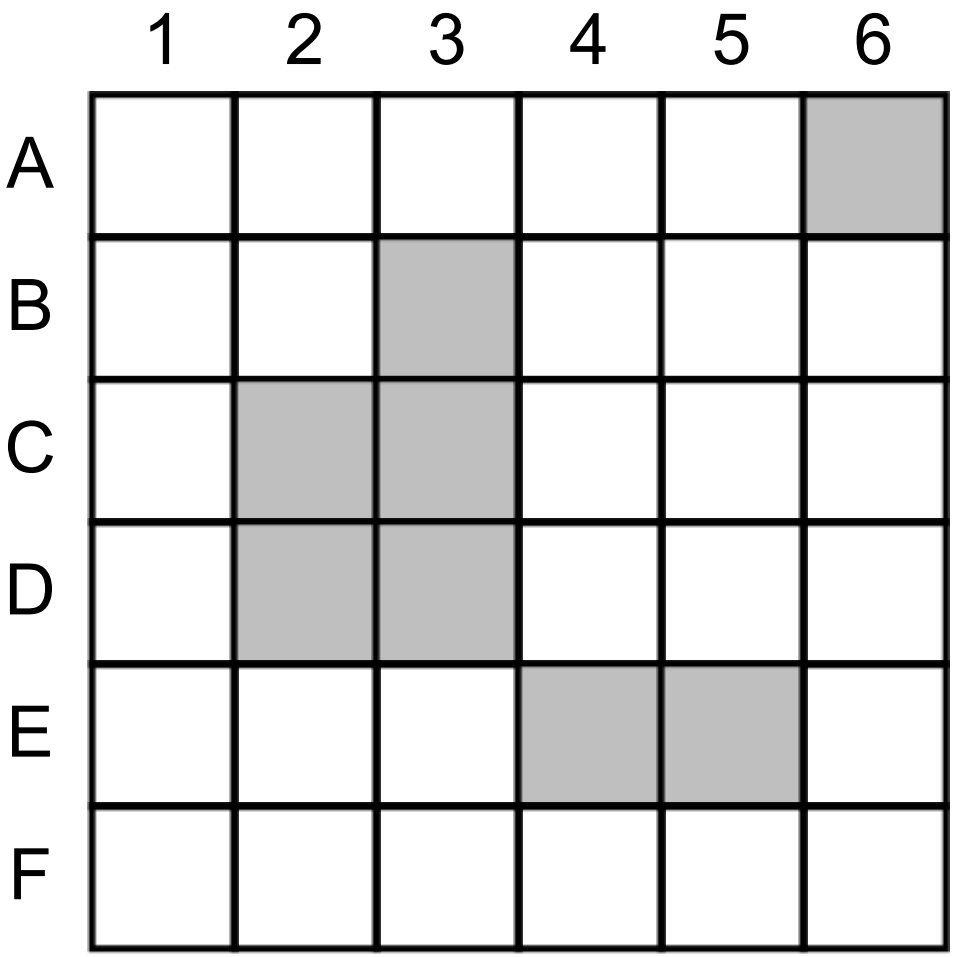
A living cell with ***2 or 3*** living neighbors ***survives***. Others die.

An empty cell with exactly 3 living neighbors ***comes to life***. No others do.

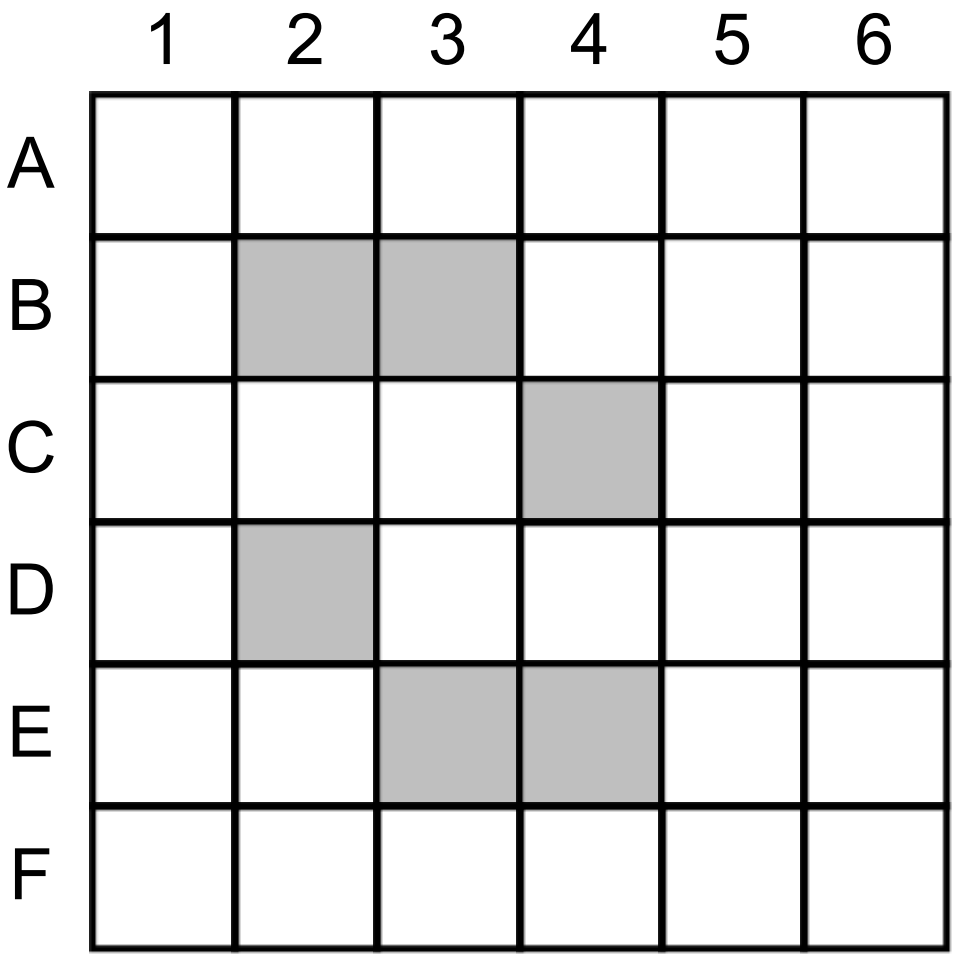
Rules of *Life*

A living cell with ***2 or 3*** living neighbors ***survives***. Others die.

An empty cell with exactly 3 living neighbors ***comes to life***.



BEFORE

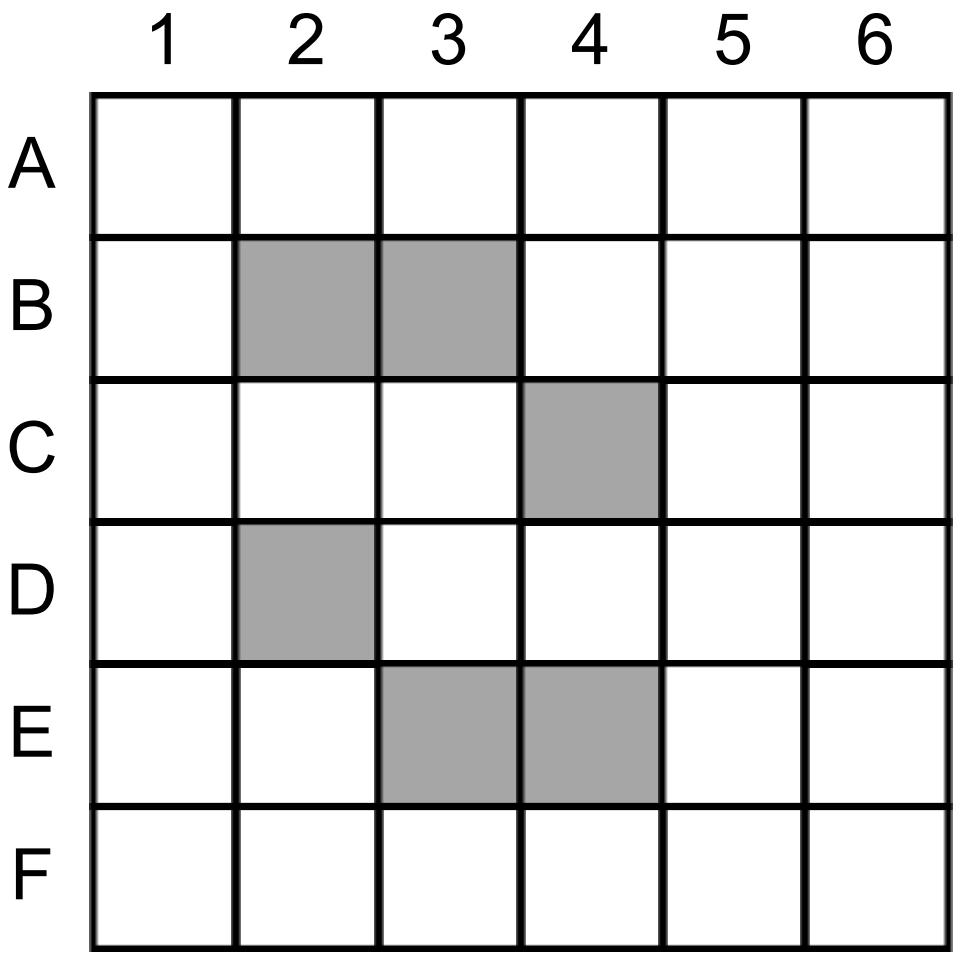


AFTER

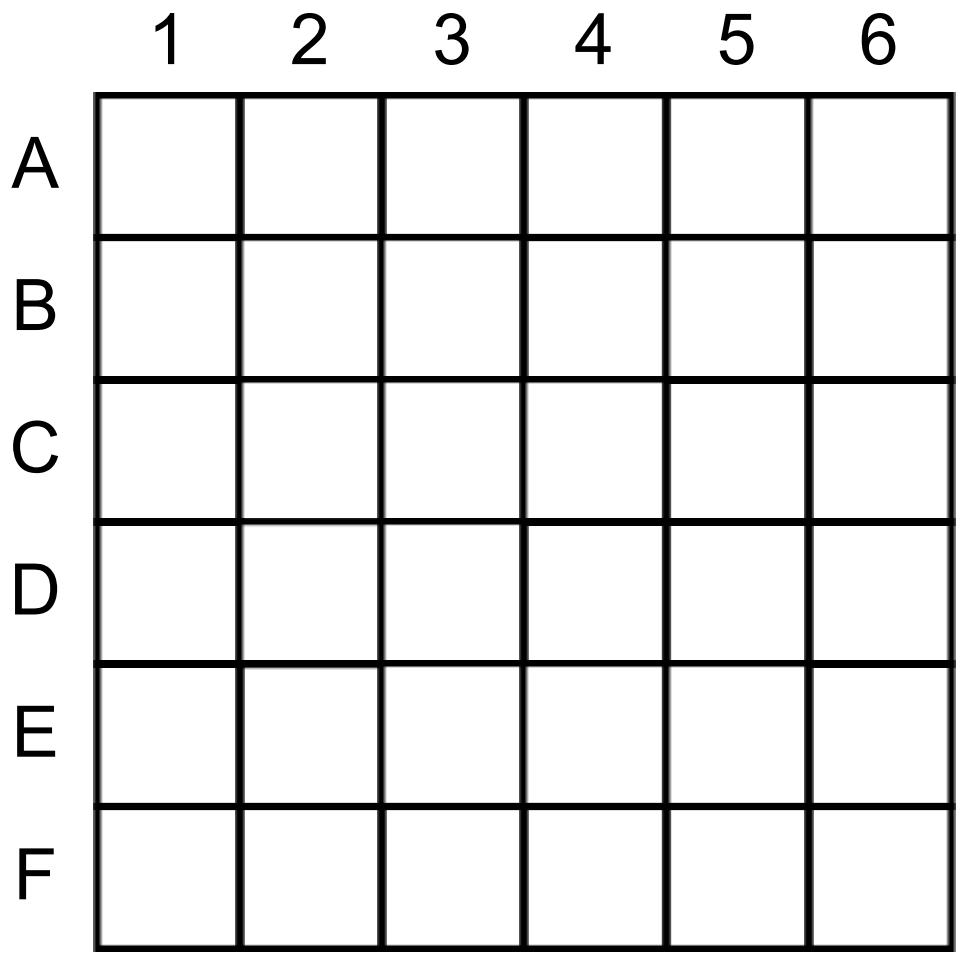
Rules of *Life*

A living cell with ***2 or 3*** living neighbors ***survives***. Others die.

An empty cell with exactly 3 living neighbors ***comes to life***.



BEFORE



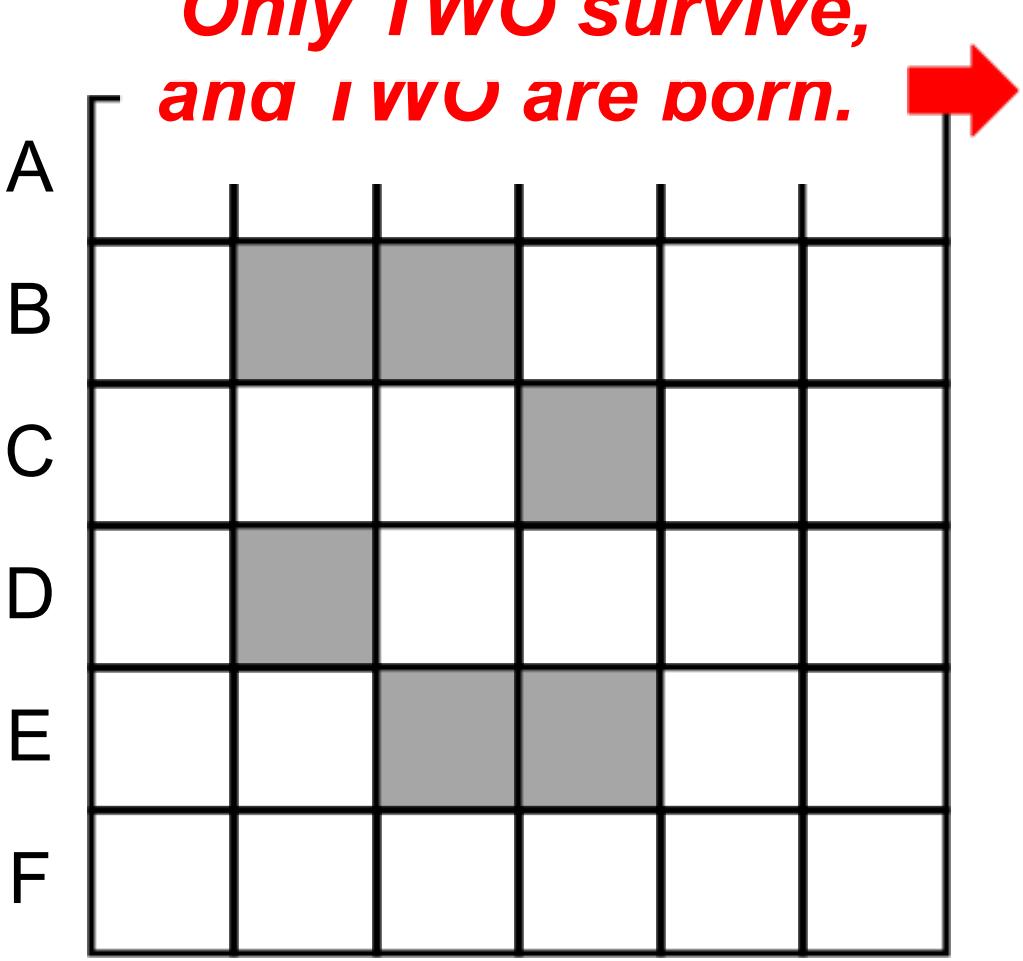
Fill in the next generation here.

Rules of *Life*

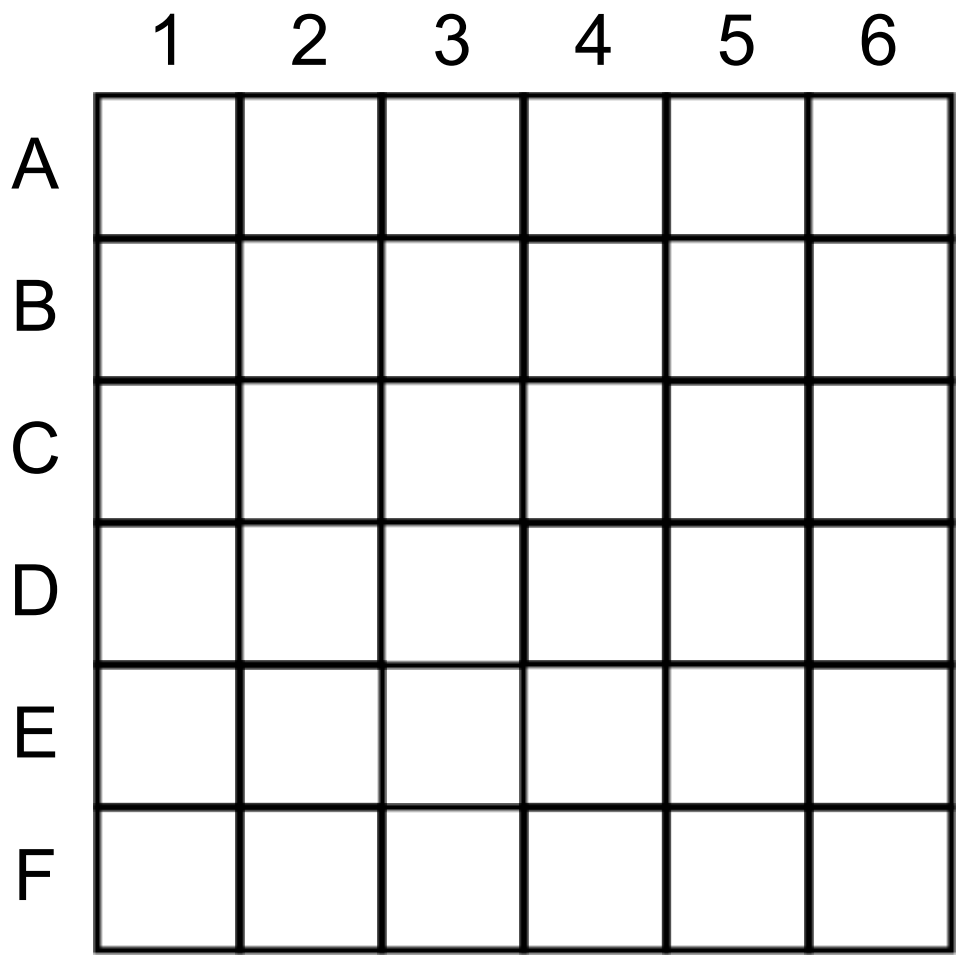
A living cell with ***2 or 3*** living neighbors ***survives***. Others die.

An empty cell with exactly 3 living neighbors ***comes to life***.

Only TWO survive, and TWO are born.



BEFORE

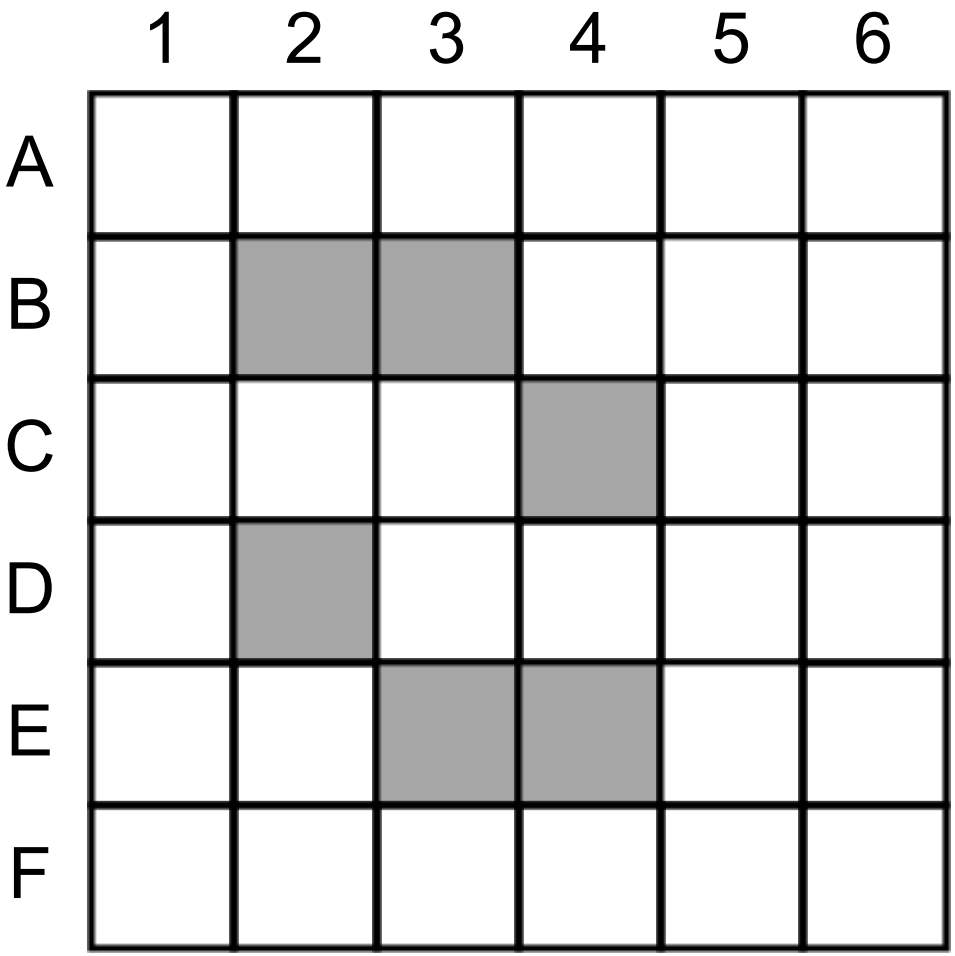


Fill in the next generation here.

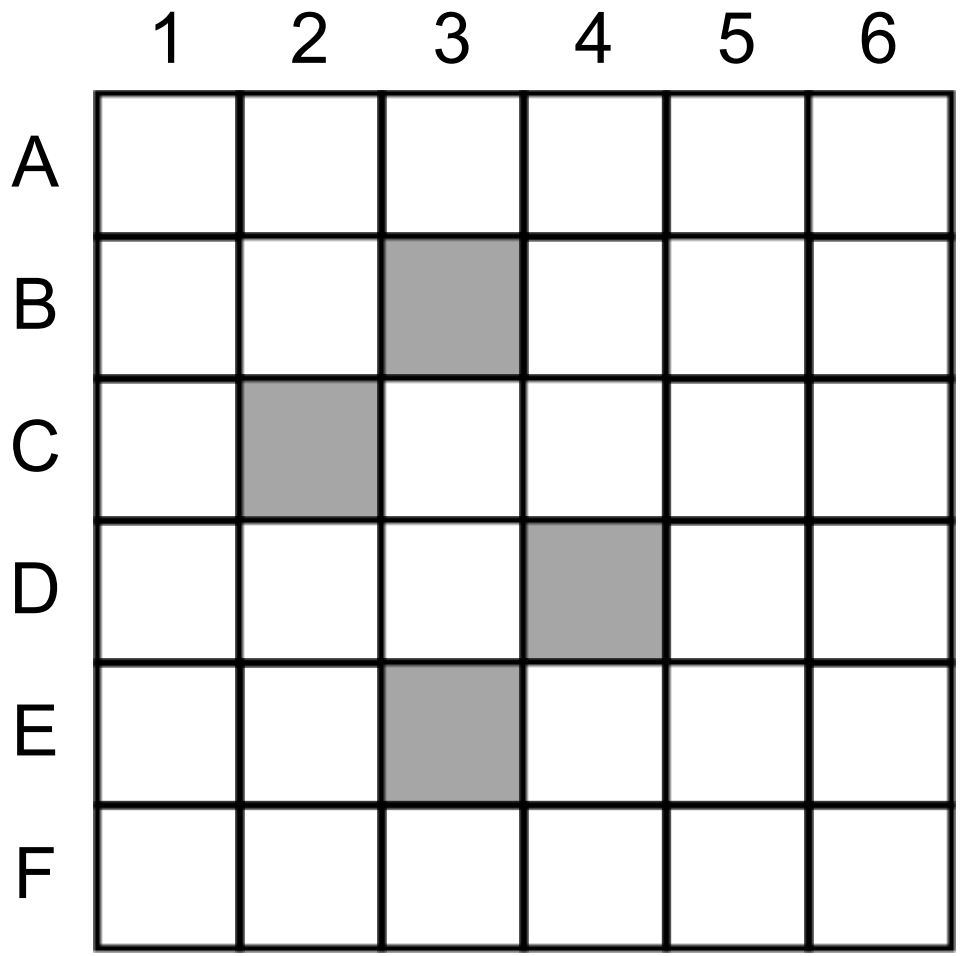
Rules of *Life*

A living cell with ***2 or 3*** living neighbors ***survives***. Others die.

An empty cell with exactly 3 living neighbors ***comes to life***.

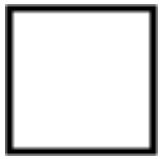


BEFORE



AFTER

next...



empty space

Simple data

living cells

+

Basic rules

A living cell with ***2 or 3*** living neighbors ***survives***. Others die.

An empty cell with exactly 3 living neighbors ***comes to life***.

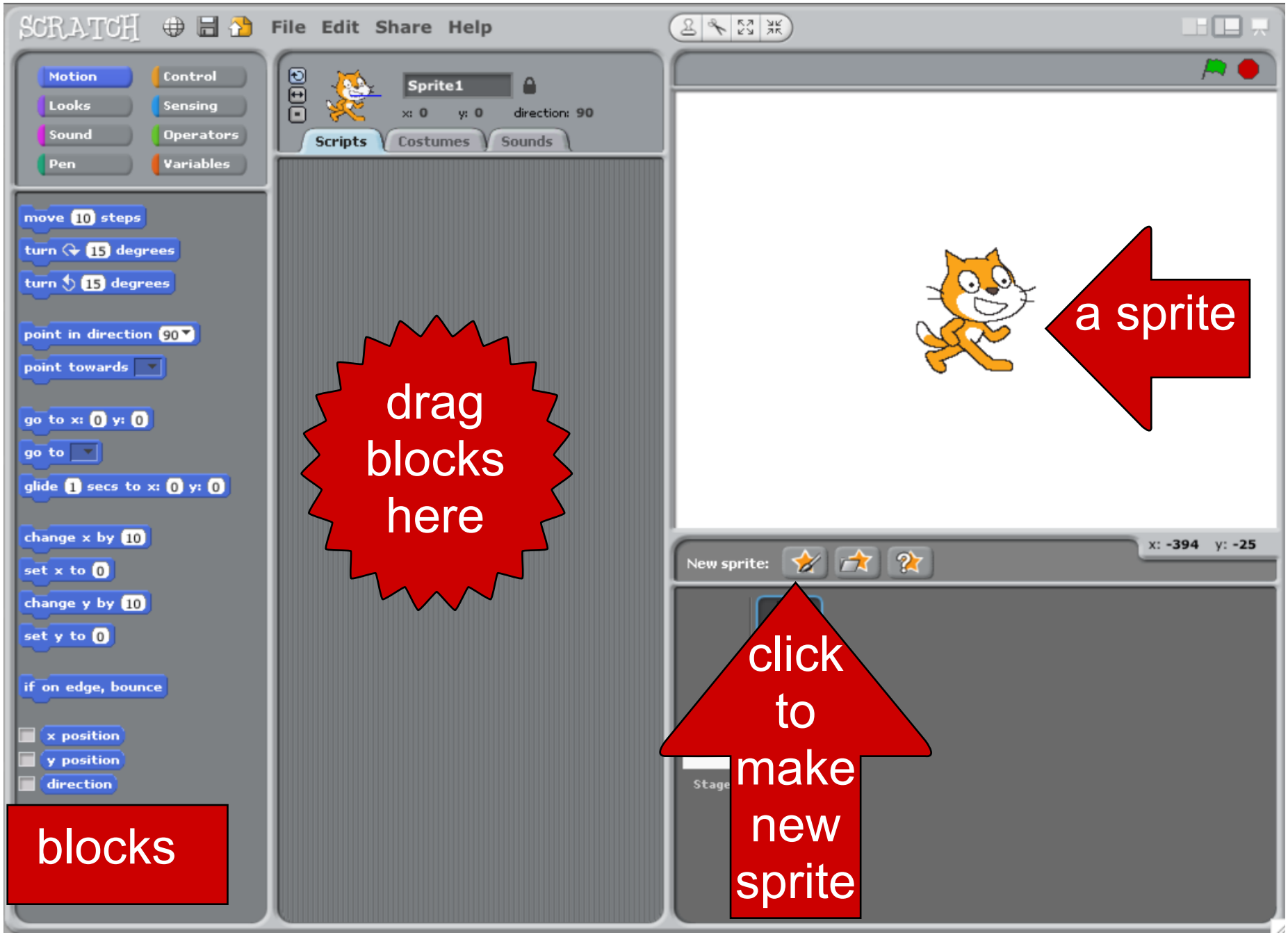
=

Complex behavior

"Game of Life"

Scratch !

Intro to Scratch



Moving Sprites

There are three ways to move sprites:

move _ steps | go to x: _ y: _ | glide _ secs to x: _ y: _

Open moving.sb (in "Scratch-Files").

Look at page 146 of the ECS Curriculum and discuss the first 9 questions with your partner.

Event Driven Programming

- Events are usually an input from the user.
- Scripts are triggered an event happens.
- alphabetlearning.sb implements all three of the blocks at right. See what it does!

Scratch blocks

Blocks click together, kind of like virtual legos.

For more information about any block, right click it and select "help."

What do these two scripts do?
How are they different?

Dialogue between Sprites

Create a short scene where two sprites talk to each other. (A knock-knock joke would work well.)

Take turns! Have one partner make the scripts for Sprite1 and the other partner make the scripts for Sprite2.

Remember:

- the Sprites should take turns speaking
- all scripts start with a curved block
- You can type in the white boxes contained in the blocks

Here are some blocks you might find useful:

Broadcasting Events

Events can also be broadcast between scripts.

Broadcasting Activity- Who likes acting?

Broadcasting Events

What are the advantages of broadcasting events?

Open summer.sb and

- add at least one new sprite
- make the basketball scene where
  - the background changes
  - the costume of the sprite changes

At this point, students get the opportunity to make a story project.

What is a variable?

What does the word "variable" mean in English and in mathematical terms?

A variable is a name that represents a value that can be changed. For example, the points accumulated in a game could be called "points".

(See "variable example.sb" or "monkey game.sb")

Monkey Game

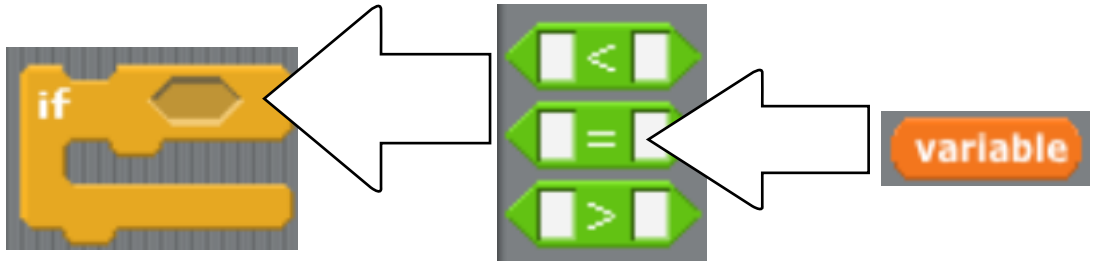
Edit the file monkey game.sb to add more features such as

- another food that the monkey can eat (which can be worth more points)
- a win condition so that once the player reaches a certain number of points, it tells them

(If you are really stuck, ask us, or check out monkey game solution.sb)

Conditionals

What are some ways we use the word "if" in English?

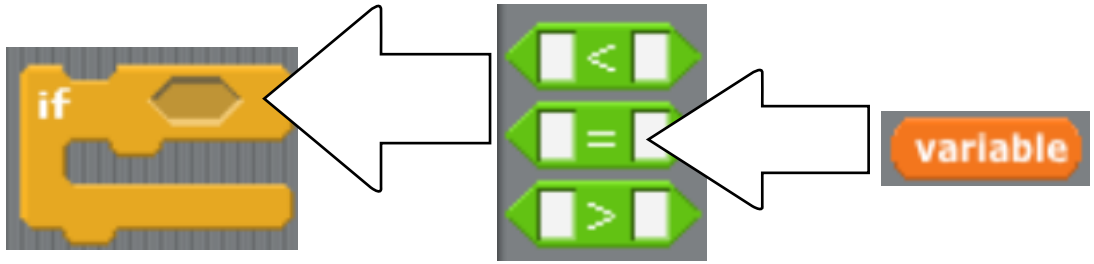


In Scratch, the if block tells the computer,

"Evaluate the statement in the hexagon. If it is true, do everything inside the if block."

Conditionals

What are some ways we use the word "if" in English?

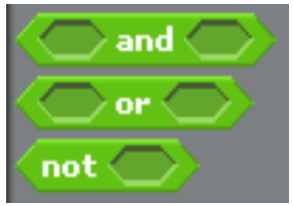


Check out age.sb and edit it to have conditions for other ages.

Note: Click the checkbox next to the variable to remove it from the display.

Click on the display to change between views (slider, value, etc).

AND, OR & Randomness

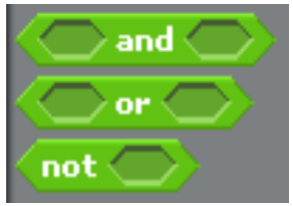


Simon Says!

Raise your hand if the symbols on your card are...

- ovals AND red
- NOT solid OR green
- NOT (red OR diamonds)
- NOT ((green OR striped) AND (purple OR squiggle))

AND, OR & Randomness



Simon Says!

Look at grades solution.sb for examples of "and" block.

Check out dice.sb for an example of randomness.

Now, using rps starter.sb and what you have learned about variables, conditionals and randomness, edit the rock, paper, scissors game!

(All you have to edit is the "determine winner" script on the computer sprite.)

Lunch Demos

Scratch Demos

- Projects/
  - Animation/
    - Daydream
    - Jellyfish
    - Trampoline
  - Games/
    - Bug on a plate
  - Interactive Art/
    - Kaleidoscope
    - WHEE
    - Zen Rock Garden

Timers for Timed Games

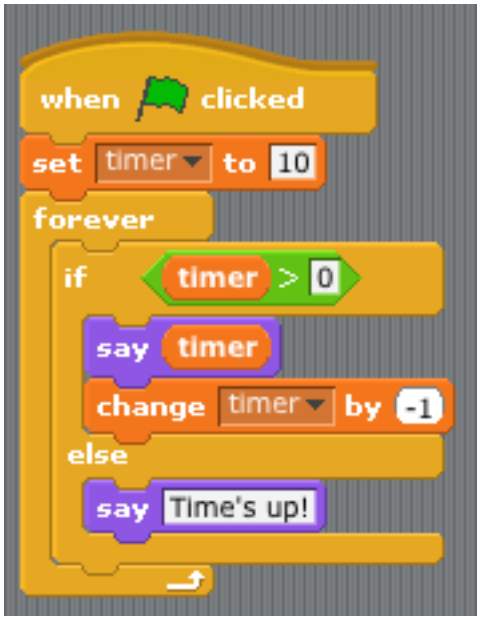
To create a timer:

1. Create a variable called timer.
2. When the flag is clicked, initialize the timer to 10.
3. Continually wait 1 second and check if the timer = 0 and
    a. output the current time either with a sprite, or just show the variable,
    b. if the timer = 0, make either the background or a sprite say "Time's Up!"
4. When the flag is clicked, everything should start over.
5. Be creative!

Hint: Make sure the timer stops at 0 and does not continue to go negative.
TRY IT!

Timers

Here are some examples of timers built in Scratch:



Match the timers above with the results below:

- Counts too quickly
- Works correctly as a timer
- Says "10" then stops
- The same as one of the above

Timed Game

Make a game that uses your timer!

There are many examples in the Scratch examples folder (like the ones from lunch).

Brainstorm ideas for games together!
Some ideas are:

- collect as many stars as you can in 10 seconds
- answer as many questions as you can in 30 seconds
- complete a maze within a time limit

When you're done, you can play each other's games!

Final Unit 4 Project

Students complete either a game project or community project (informational story telling) in Scratch.