COMP 1010- Summer 2015 (A01)

Jim (James) Young

young@cs.umanitoba.ca

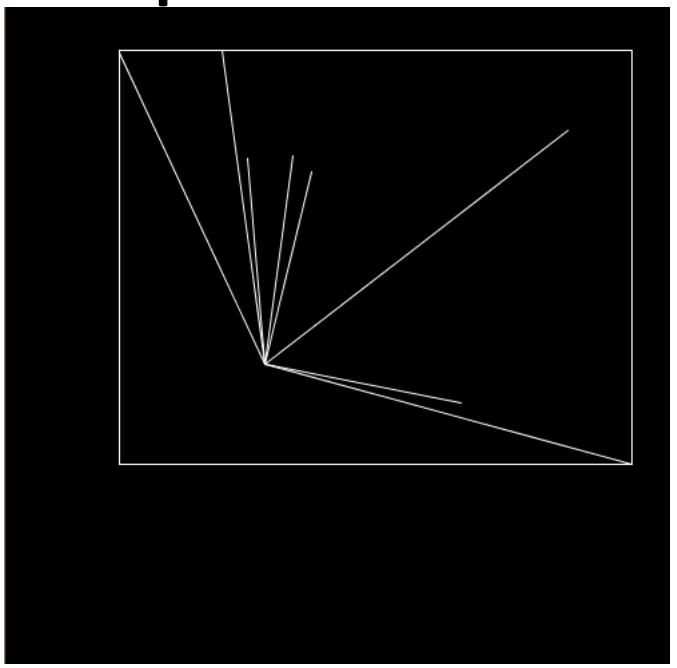
jimyoung.ca

Update: draw a bounding box, a box that fits the points as best as possible

Including the mouse

Find the smallest X and Y of all the points. This is the top left

Find the largest X and Y of all the points. This is the bottom right



Algorithm – smallest in array

Make a variable for smallest

Assume mouse is smallest

Go through array – check if that new number is smaller – if so, save it. (use existing for loop)

Do the same for largest.

Example: mouse explosion!

```
float particleX = -1;
float particleY = -1;
float speedX = 0;
float speedY = 0;
final int MAX_SPEED = 5;

void setup()
{
  size(500, 500);
}


void draw()
{
  background(0);

  if (particleX < 0 || particleX >= width ||
      particleY < 0 || particleY >= height)
      {
        particleX = mouseX;
        particleY = mouseY;
        speedX = random(MAX_SPEED*2)-MAX_SPEED;
        speedY = random(MAX_SPEED*2)-MAX_SPEED;
      }

  particleX += speedX;
  particleY += speedY;

  ellipse(particleX, particleY, 10, 10);
}
```

Update this to use arrays

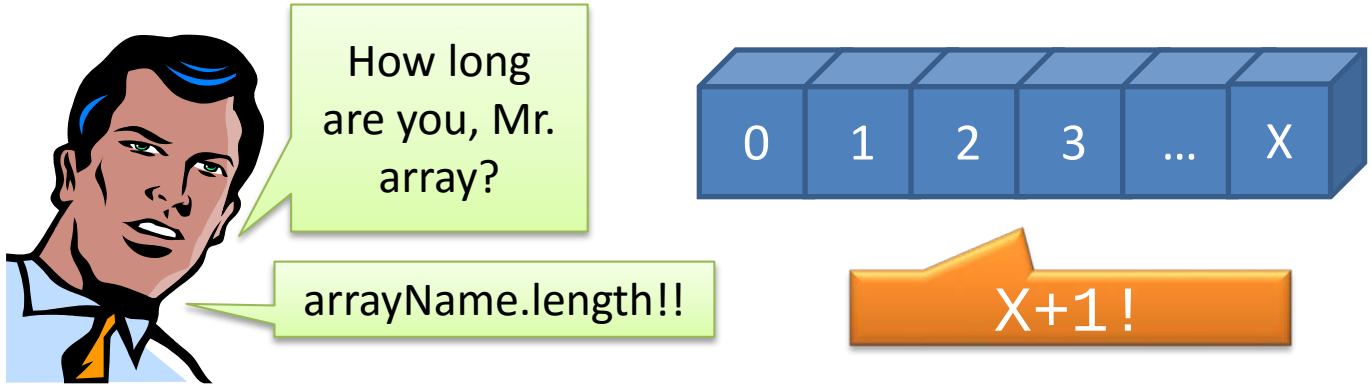
First, make a new global to decide how many balls

Then, *upgrade* the variables to arrays

Every time we use that variable, we now need to use arrays

- instead of doing something just once, we need to do it per array bin.
- use a for loop

length of the array?



Processing provides a mechanism for giving us the length of an array:

variableName.length

```
int[] numbers = new int[100];
println(numbers.length); // result?
```

notice!! no () why is this?

.length is not a **method**, it is a **property**

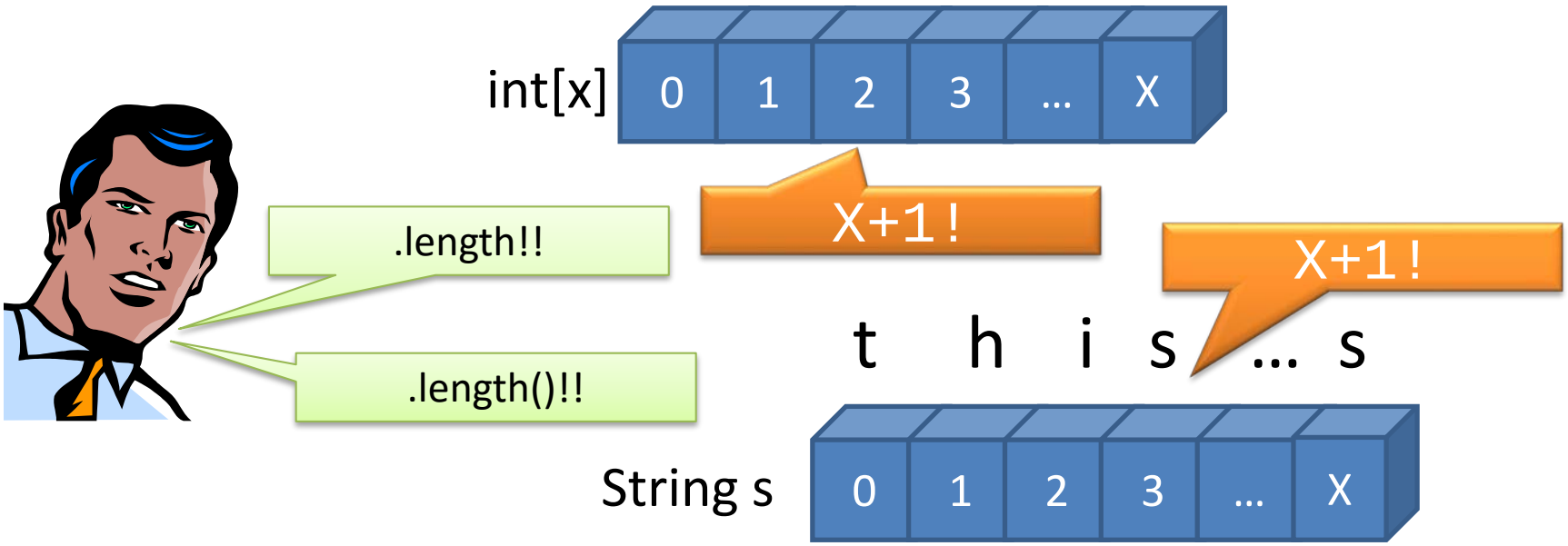
you will learn more about this later, but for now, remember that array length has no ().

array length vs string length

to get the length of a string:

```
stringVariable.length(); // method call
arrayVariable.length;  // property
```

note: this is VERY annoying. have to memorize ☹

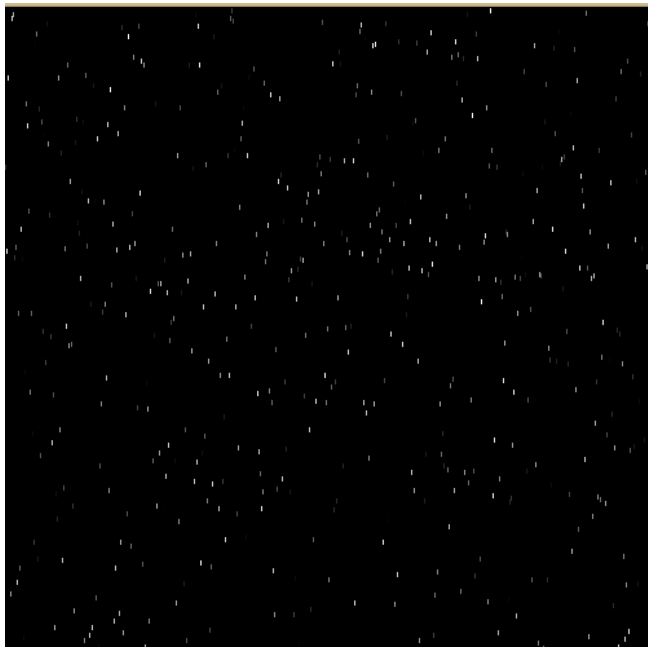


safety

using arrayVariable.length is SAFER because you cannot make a mistake about the length of the array

- you don’t need to remember the length, just ask for it.

Make it rain



Make the drops evenly spread out

Make one array of 500:

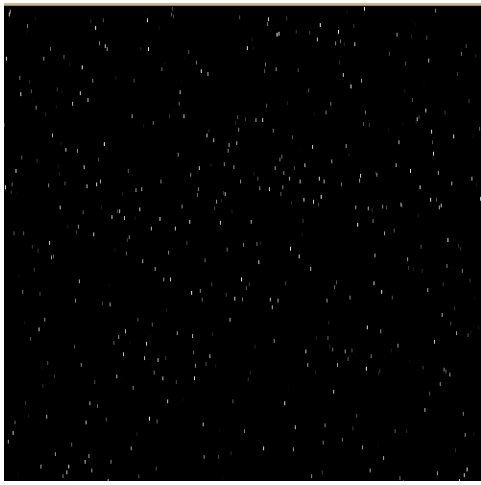
bin number is x coordinate

value stored is y coordinate

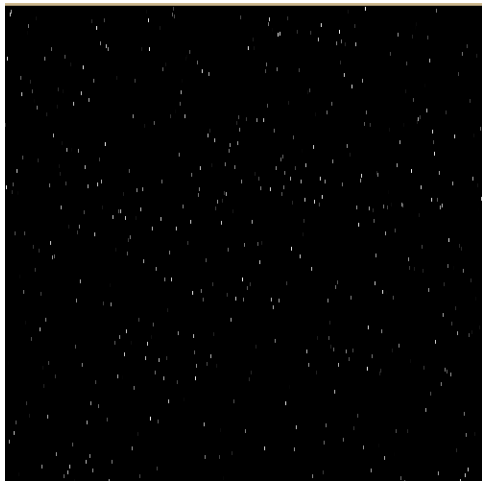
Calculate a random y for each point.

Initialize in startup code

- set the bins to random values (height)



Draw!

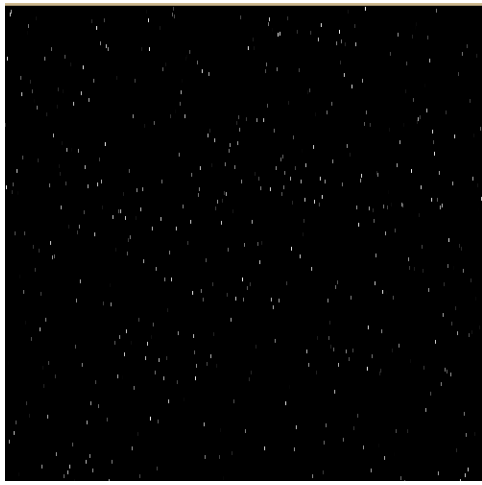


Go through each rain drop

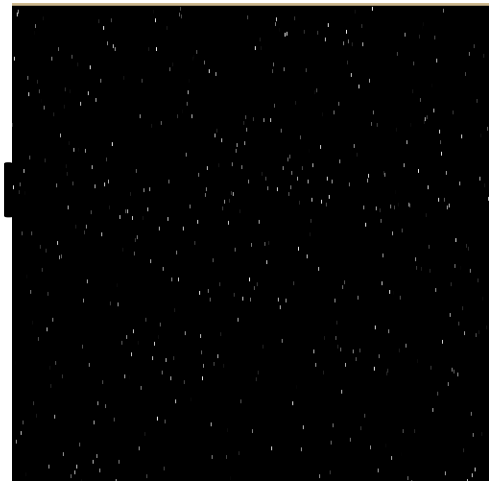
- x is i
- y is data in the bin
- use the length of the drop
- draw the drop!

Animate!!

Add some y value.



Animate!! Make it look layer



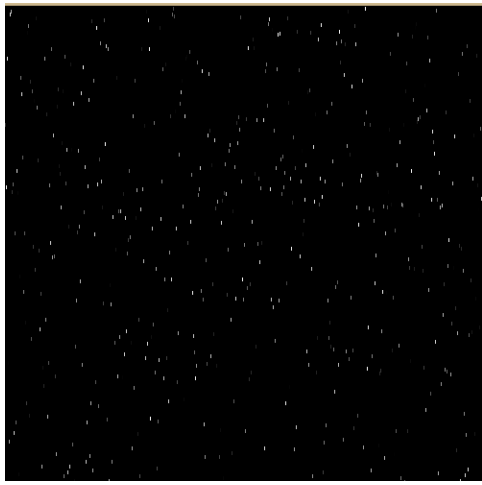
Let’s make it look layered. Use the following formula

drops[i]=(drops[i]+i%3+1)%height;

Exercise: figure it out

Exercise – make slanty rain

Random number of dots



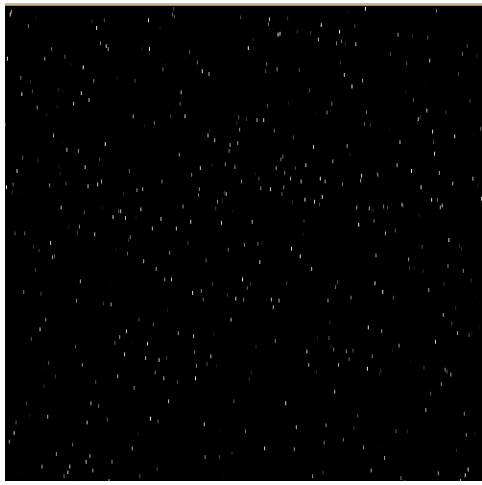
Don’t make a global for # of dots

Calculate the number of dots in startup

define but don’t instantiate array

Also, need a rain-drop length, as we draw streaks

Initialize in startup code



At least 500 drops, and some random amount more – use a local int variable

- create the array
- set the bins to random values (height)

Note that the length of the array was a local variable – in the draw, we ONLY have the .length property to help us.

Grey is boring. Make the rain colorful!

```
stroke(random(256),0,random(256));
```