Arrays

CSE 120 Winter 2020

Instructor: Sam Wolfson

Teaching Assistants: Yae Kubota Eunia Lee Erika Wolfe

YouTuber uses neural networks to upscale 1896 short film to 4K 60 fps

“Someone was able to achieve remarkable results in upscaling a very old, low-resolution, black and white video from 1896 to a crystal-clear 4K video at 60 frames per second. This is one of many experiments that prove AI tech can be used to improve content that would normally seem impossible to "enhance" in any way.”

AI is so big that companies like Google, Apple, Intel, Microsoft, Amazon, and others are all racing to buy or invest in every new and obscure startup in the field to foster talent for their various projects. It's even the subject of ample discussion on the role AI in copyright and trademark law moving forward. It will be interesting to see what's in store for the next decade, and, thanks to Nvidia, we might have a hint - AI-rendered, interactive virtual worlds based on massive libraries of real-world footage.”

https://www.techspot.com/news/83867-youtuber-uses-neural-networks-upscale-1896-short-film.html

https://digg.com/2020/arrival-train-la-ciotat-upscaled

Administrivia

- Assignments:
  - Portfolio Update 1 due tonight (2/~~6~~5)
  - Creativity Assignment (2/7)
    - You should already have feedback for your proposals!

- Guest lecture on Friday: Data Visualization!
  - Reading check due before section, as usual

- Living Computers Museum “field trip” upcoming

Outline

- **Loop Review**
- Arrays
  - Arrays and Loops

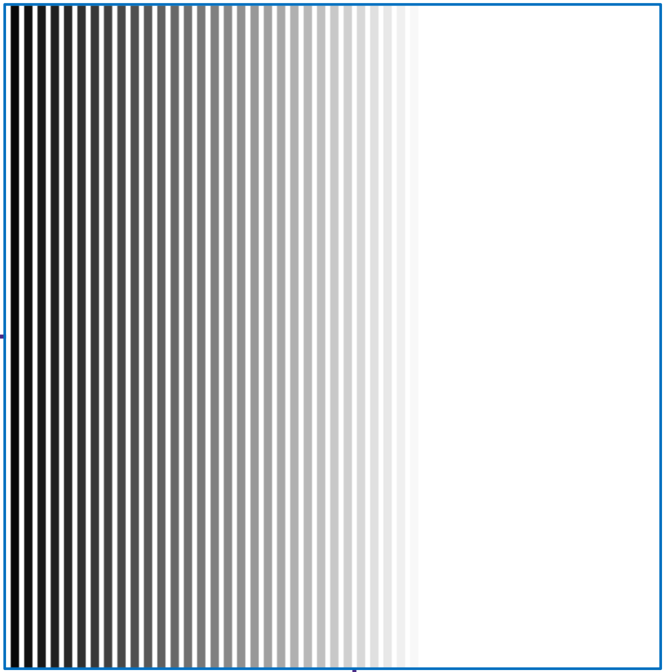
Example: Line Gradient

common change

width lighter

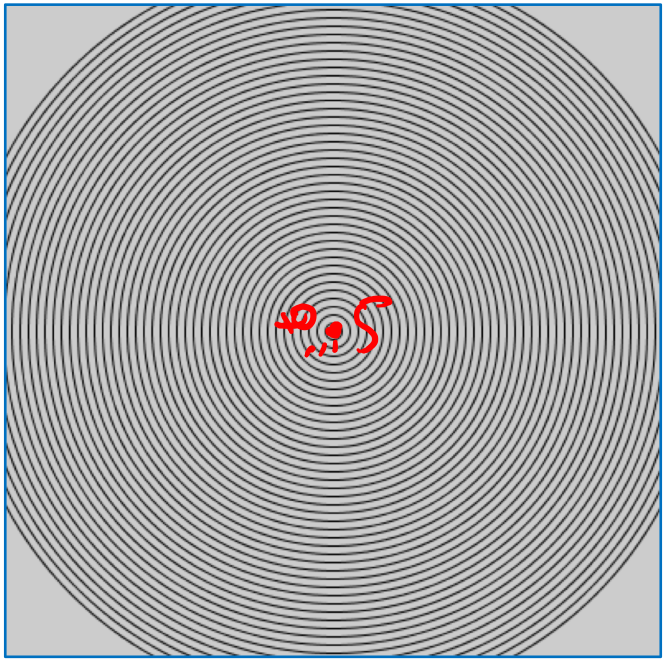
vertical x-pos

```
size(400, 400);

background(255);
strokeWeight(5);

int i = 0;
while (i < 400) {
   stroke(i);
   line(i, 0, i, 400);
   i = i + 8;
}
```



Exercise: Circle Loop

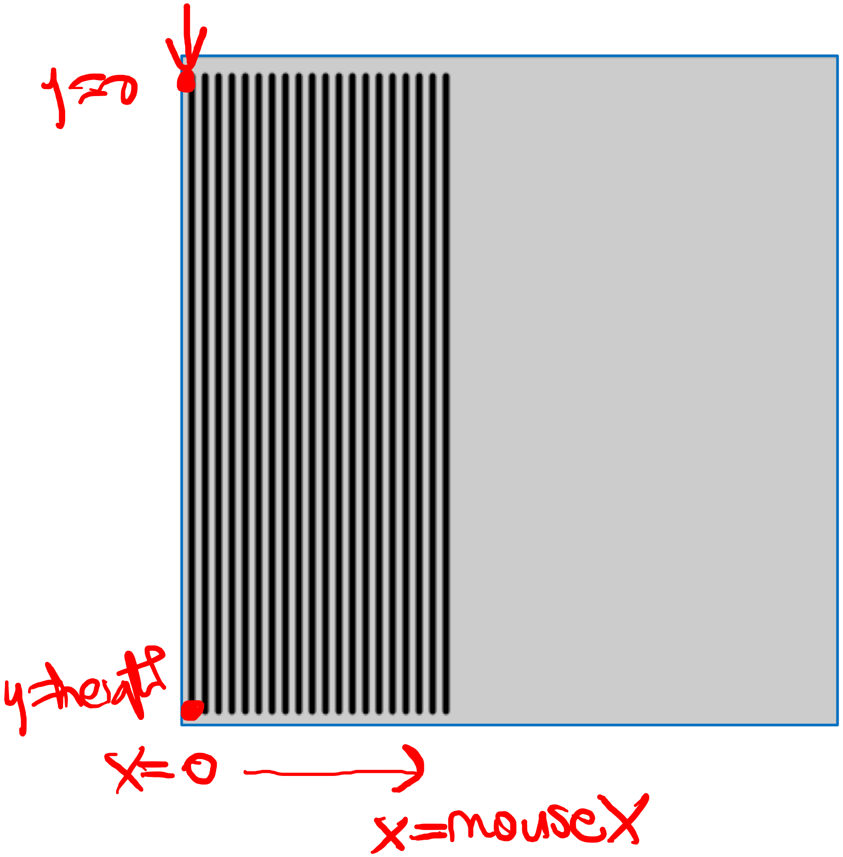
- Consecutive concentric circles differ in diameter by 10:



```
size(400, 400);
noFill();   diameter
int d = 450;
while (d > 10) {
    ellipse( 200 , 200 , d , d );
    d = d - 10;
}
```

Example: Looping with User Interaction?

- Draw lines from left side of screen to the horizontal position of the mouse



```
int x=0;
while (x <= mouseX)
{
  line(x,0,x, height);
  x=x+5;
}
```

Example: Draw Lines to mouseX

```
void setup() {
  size(400, 400);
  strokeWeight(4);
}

void draw() {
  background(204);

  int i = 10;
  while (i < mouseX) {
    line(i, 10, i, 390);
    i = i + 8;
  }
}
```

Outline

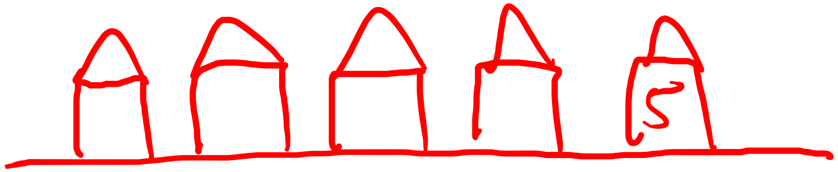
- Loop Review
- **Arrays**
  - **Arrays and Loops**

Arrays

- "Structures" that store many values *of the same datatype*
  - Can think of as a group of related variables

- Arrays store large amounts of data that you can access using a single name
  - Accessing arrays with loops is very convenient

- Analogy: creating a street with lots of houses on it; can access houses via house numbers

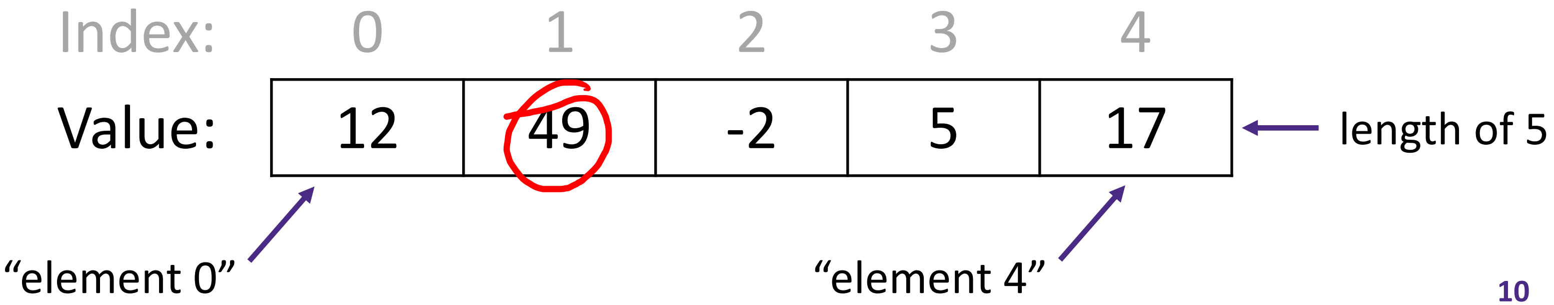


Arrays Terminology

- "Structures" that store many values *of the same datatype*
  - Element: a single value in the array house
  - Index: a number that specifies the location of a particular element of the array house number
    - Start from 0 ☆
  - Length: total number of elements in the array

# of houses on the street

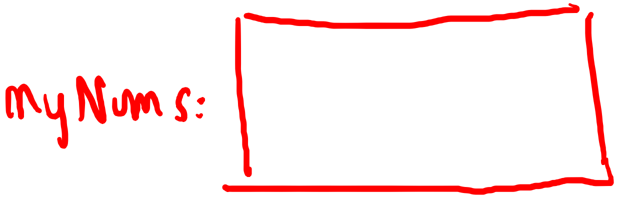
- Example:



Index:	0	1	2	3	4
Value:	12	49	-2	5	17

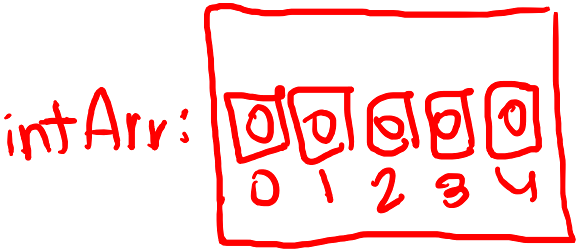
Declaring Arrays in Processing

- Declaration: `type[] name`
  - *Warning: an array variable is NOT an array!*



- Examples:
  - `int[] myNums` declares an array variable for arrays of integers
  - `float[] myData`
  - `color[] rainbow`
    - Could represent the colors of the rainbow in order
  - `boolean[] correct`
    - Could represent correctness on a multiple choice test

Creating Arrays in Processing



- Creation: `new type[num]`
  - Remember that actual indices are from `0` to `num-1`
  - Default value for *all* elements is "zero-equivalent" (0, 0.0, `false`, black)

declaration assignment

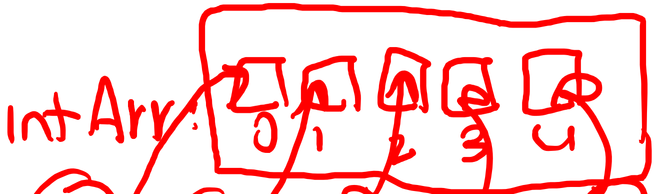
- Examples:
  - `int[] intArr = new int[5];`
    - `intArr` associated with `int` array `{0, 0, 0, 0, 0}`
  - `boolean[] quiz = new boolean[3];`
    - `quiz` associated with `boolean` array `{false, false, false}`
  - `new float[7];`

    - Creates a float array, but you can't use it because you didn't associate it with an array variable!

Initializing Arrays in Processing

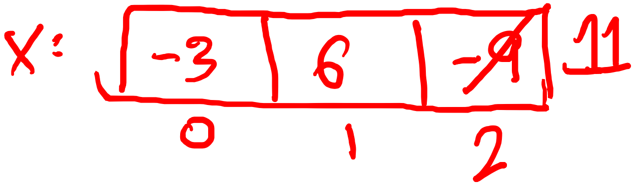
- Initialization: `{elem0, elem1, …, elemN}`
  - The "curly brace" notation can be used to create an array with specified/initialized element values
  - Don't confuse this usage of { } with that for a code block

- Examples:



  - `int[] intArr = {12, 49, -2, 5, 17};`
    - Create a new array with specified values and associate it with the array variable `intArr`
  - `intArr = {1, 2, 3, 4};`
    - *Discards* the old array and associates the array variable `intArr` with this new array!

Using Arrays in Processing



- Use element: `name[index]`
  - In *expression*, uses value of that index of the array
  - In *assignment*, modifies value of that index of the array

int i = x[1];

i is 6

x[2] = 11;

- Get length: `name.length`

x.length // 3

- Example:

0 1 2 5

```
int[] intArr = {12, 49, ~~-2~~, 5, 17};
println(intArr[0]);          // prints 12 to console
intArr[2] = intArr.length;  // changes -2 to 5
```

5

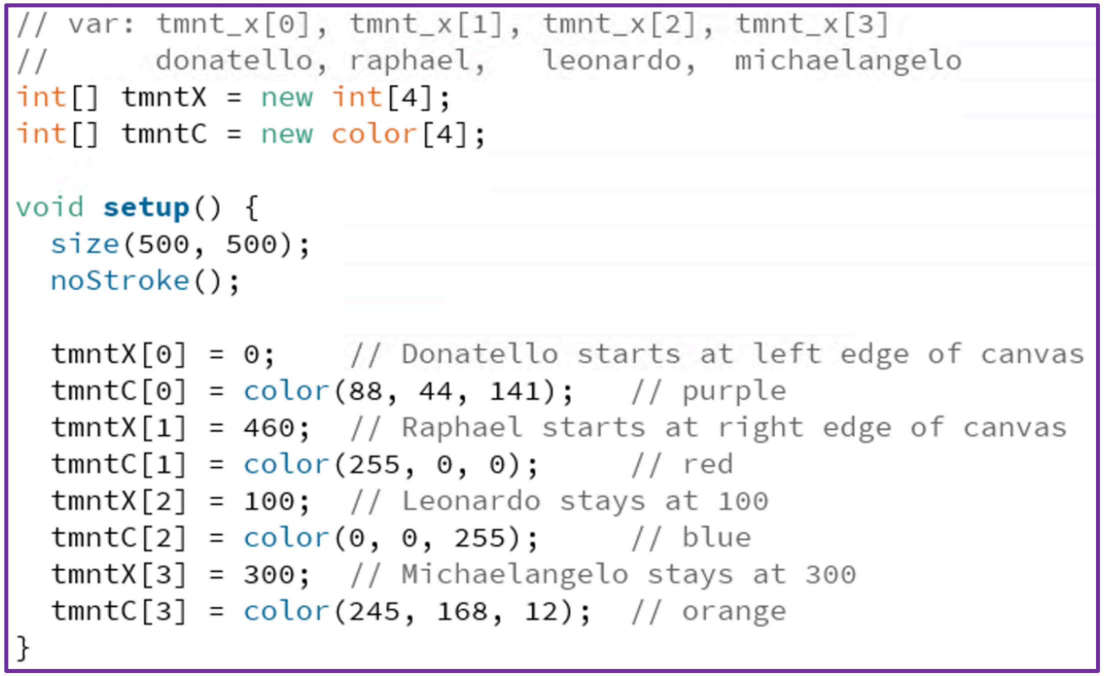
Index:	0	1	2	3	4
Value:	12	49	-2	5	17

Arrays and Loops

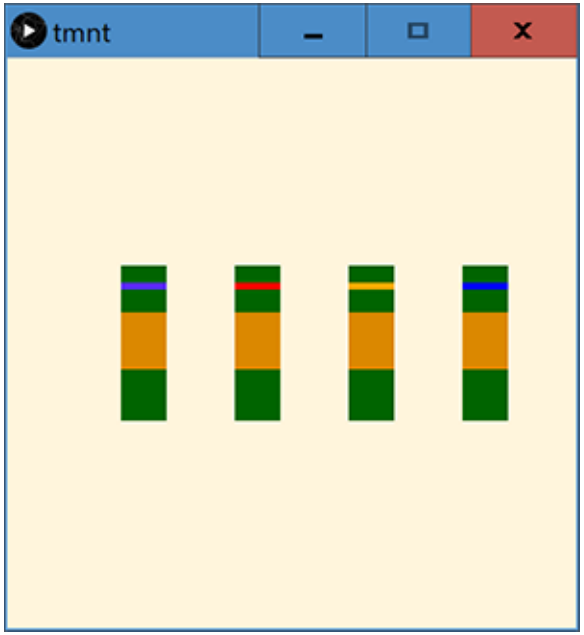
- Arrays are very convenient to use with loops!
  - Loops usually change *numerical value* in loop variable (*e.g.* Update: `i = i + 1`)
  - You access array element values using a *numerical value* (*e.g.* `intArr[i]`)
  - Loops provide an easy way to perform the same/similar action on all/many elements of an array

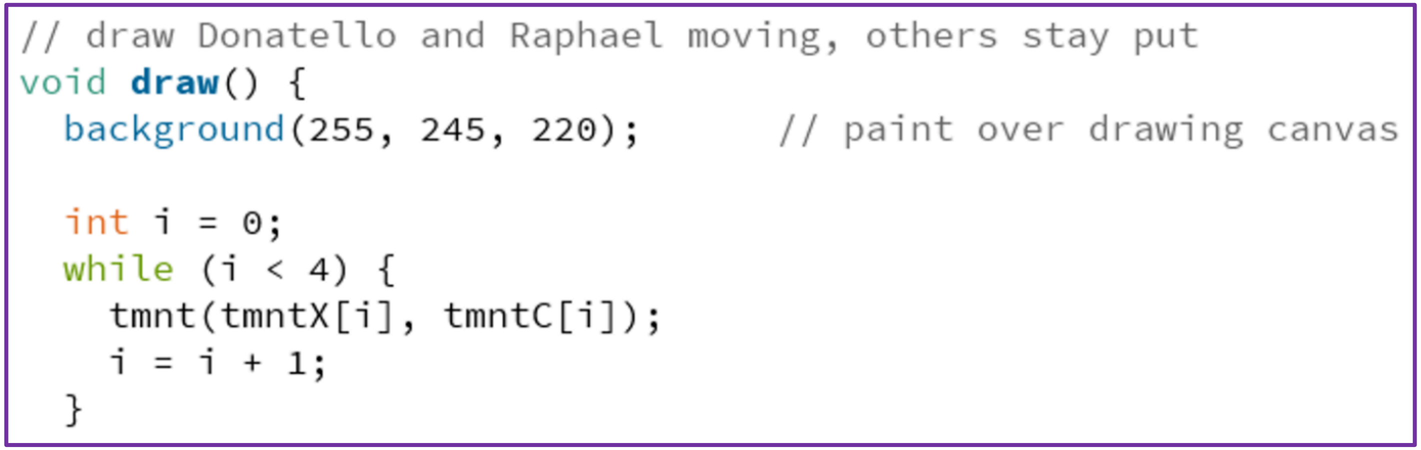
- Examples:
  - Read each array value to use in a drawing command
  - Sum up the numbers in an array

Example: TMNT



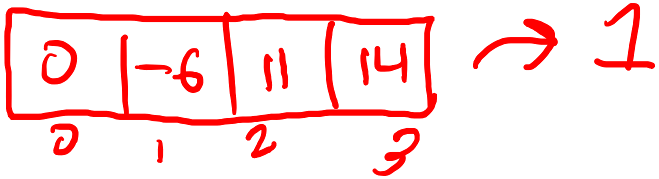
```
// var: tmnt_x[0], tmnt_x[1], tmnt_x[2], tmnt_x[3]
//      donatello, raphael,   leonardo,  michaelangelo
int[] tmntX = new int[4];
int[] tmntC = new color[4];

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

  tmntX[0] = 0;    // Donatello starts at left edge of canvas
  tmntC[0] = color(88, 44, 141);   // purple
  tmntX[1] = 460;  // Raphael starts at right edge of canvas
  tmntC[1] = color(255, 0, 0);     // red
  tmntX[2] = 100;  // Leonardo stays at 100
  tmntC[2] = color(0, 0, 255);     // blue
  tmntX[3] = 300;  // Michaelangelo stays at 300
  tmntC[3] = color(245, 168, 12);  // orange
}
```



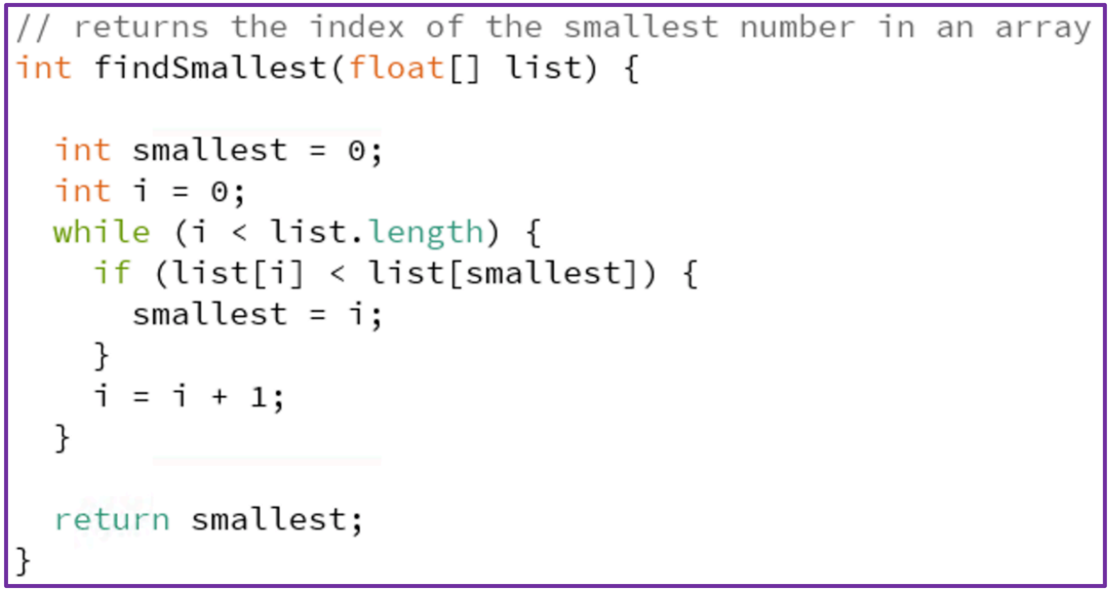


```
// draw Donatello and Raphael moving, others stay put
void draw() {
  background(255, 245, 220);      // paint over drawing canvas

  int i = 0;
  while (i < 4) {
    tmnt(tmntX[i], tmntC[i]);
    i = i + 1;
  }
```

Example: Index of Smallest Number



- Algorithm:
  - Keep track of the *index* of the smallest number seen so far
    - Start with index 0
  - Check each *element* 1-by-1; if number is smaller, then update the smallest index



```
// returns the index of the smallest number in an array
int findSmallest(float[] list) {

  int smallest = 0;
  int i = 0;
  while (i < list.length) {
    if (list[i] < list[smallest]) {
      smallest = i;
    }
    i = i + 1;
  }

  return smallest;
}
```

Arrays Visual Demo

Rd: "read" **Wr: "write"**

```
1)    int[] ar;
2)    ar = new int[4];
3)    ar[1] = 1;
4)    ar[3] = ar[1] + 1;
5)    if (ar[2] < ar[1]) {
          ar[2] = 4;
      }
6)    ar[0] = ar[1] + ar[3];
7)    ar[4] = 0;
```

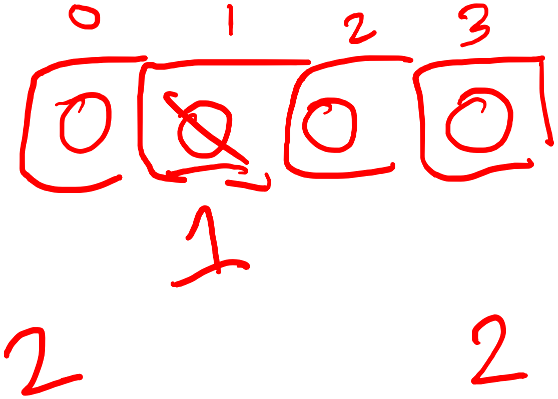
read

→ 2

0

1

→ true



Arrays Visual Demo

- Demo Notes:
  - Array creation only gives you as many buckets as you request
  - Creation of array automatically initializes it
  - When WRITING, replace ball in bucket
  - When READING, take ball of same color (*leave current one in bucket*)
  - Trying to use an invalid index (off either end) will result in an `ArrayIndexOutOfBoundsException`