Arrays

CSE 120 Winter 2019

Instructor: Justin Hsia

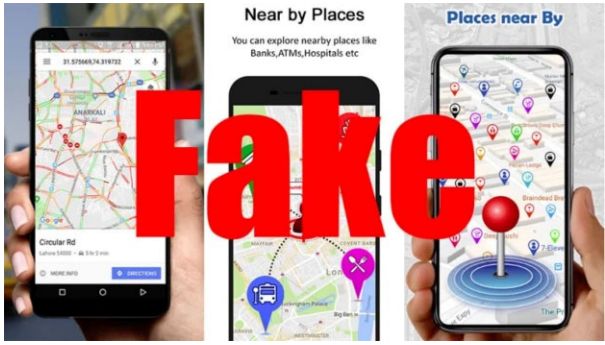
Teaching Assistants: Ann Shan, Eunia Lee, Pei Lee Yap, Sam Wolfson, Travis McGaha

Navigation Apps With Millions of Downloads Exposed as Just Google Maps With Bonus Ads

"One of the purported benefits of modern day app stores is to make it easier for companies to review and ensure that the software you download isn't harmful or malicious.

"[T]he 19 apps he tested were navigation apps with over 1 million installs each, totaling a combined install base of more than 50 million. Sadly, despite claims that these apps can help users map their routes or include tools such as a compass or speedometer, every single app ended up relying on Google Maps or its related API to perform the real work."

- https://gizmodo.com/navigation-apps-with-millions-of-downloads-exposed-as-j-1831869725



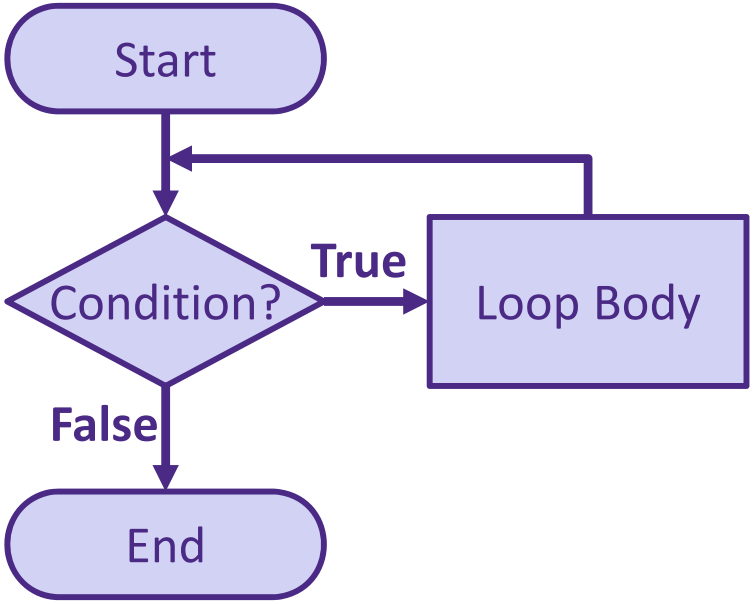
Administrivia

- Assignments:
  - Portfolio Update 1 due tonight (2/6)
  - Creativity Assignment (2/9) – deadline pushed back to Sat
    - We will try to give you feedback on your proposals tonight
- Midterm in class on Monday, 2/11
  - 1 sheet of notes (2-sided, letter, handwritten)
  - Fill-in-the-blank(s), short answer questions, maybe simple drawing
- Living Computers Museum "field trip" upcoming

Loops Worksheet

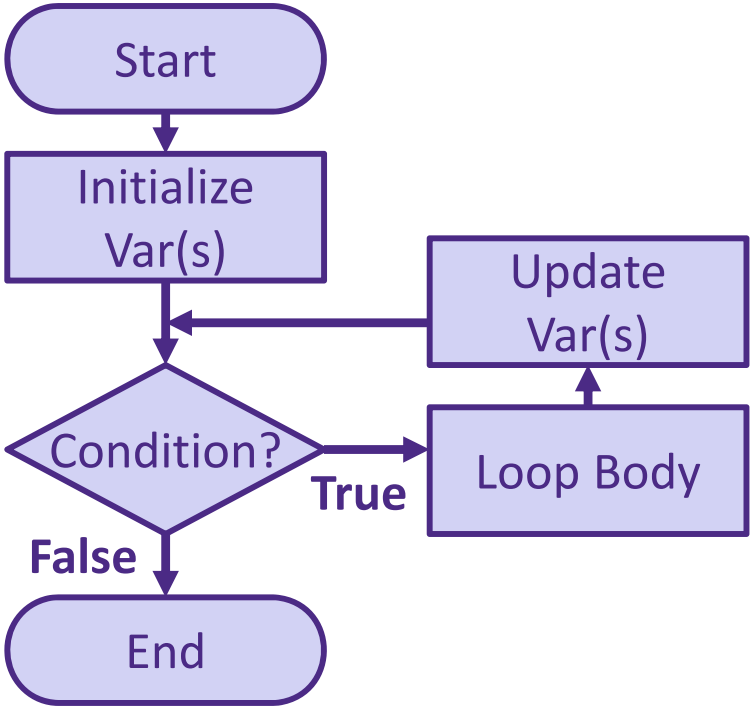
- While-loop:

```
while(condition) {
    // loop body
}
```



- For-loop:

```
for(init; cond; update){
    // loop body
}
```



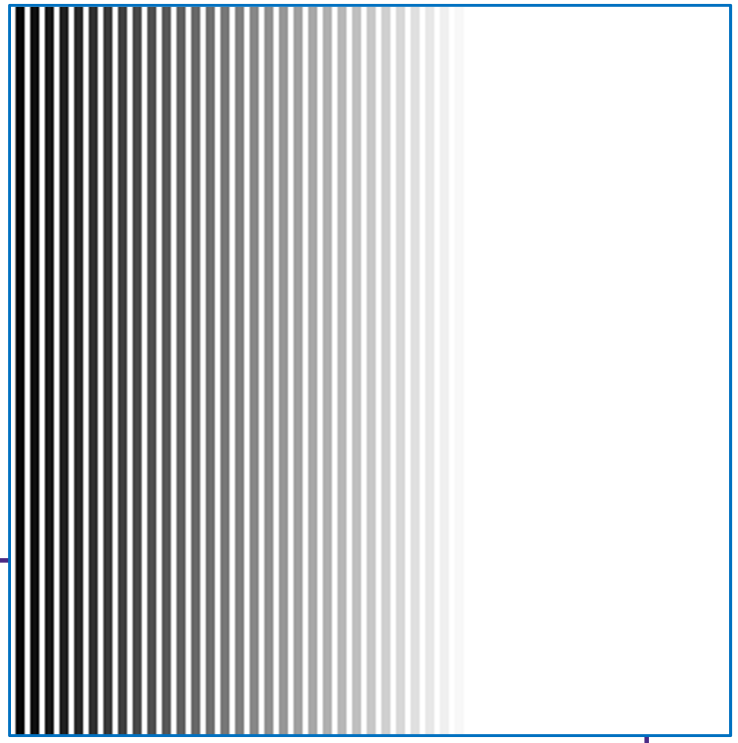
Outline

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

Example: Line Gradient

Common:
vertical lines

Change:
transparency / color
x-position



```
size(400, 400);

background(255);
strokeWeight(5);

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

line color

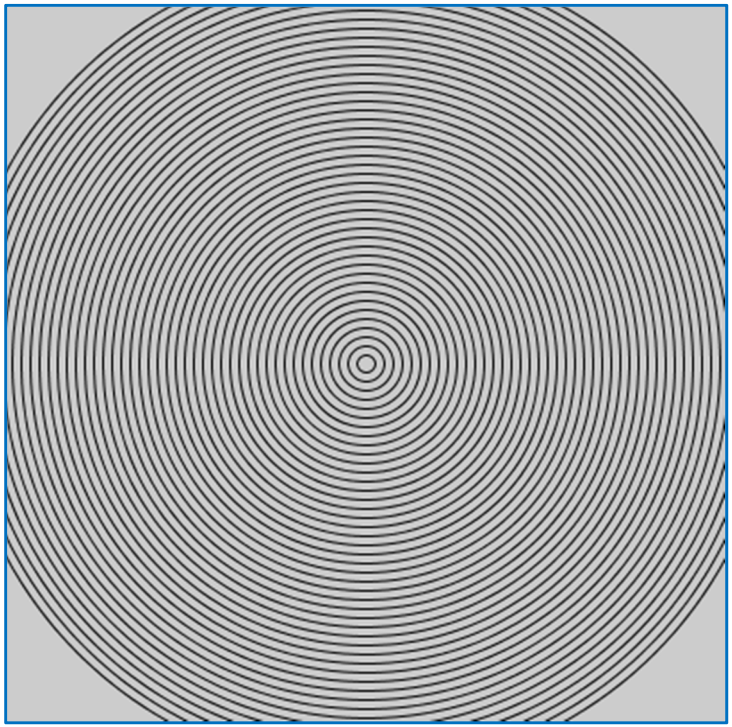
x-position

Exercise: Circle Loop

- Consecutive concentric circles differ in diameter by 10:

Common: concentric circles

Change: radius/size



```
size(400, 400);
noFill();
for(int d = 450; d > 0; d = d - 10) {
    ellipse( width/2, height/2, d, d );
}
```

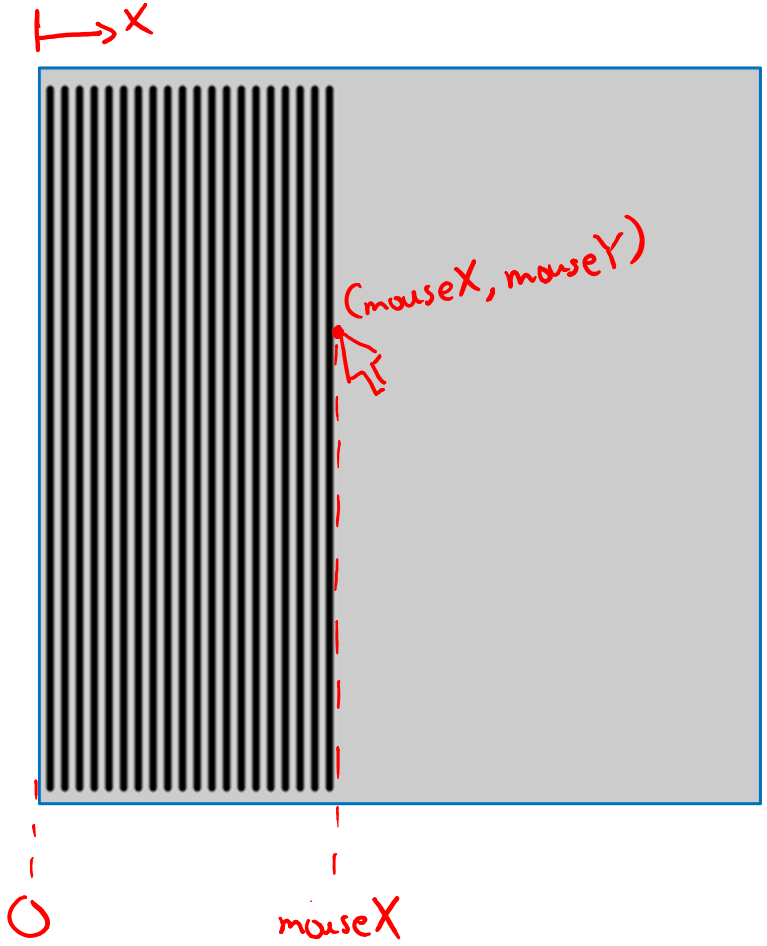
also works: `int d = 10; d <= 450; d = d + 10`

initialization; condition; update

center X, center Y, width, height

Example: Looping with User Interaction?

- Draw lines from left side of screen to the horizontal position of the mouse: draw lines from x=0 to mouseX

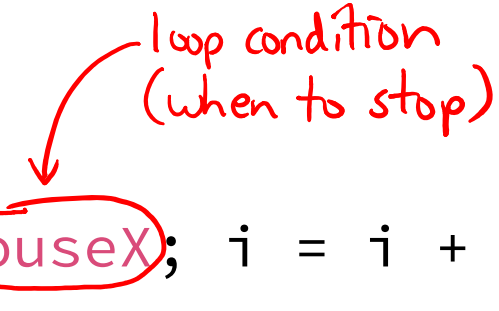


Example: Draw Lines to mouseX

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

void draw() {
  background(204);

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



Outline

- For-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

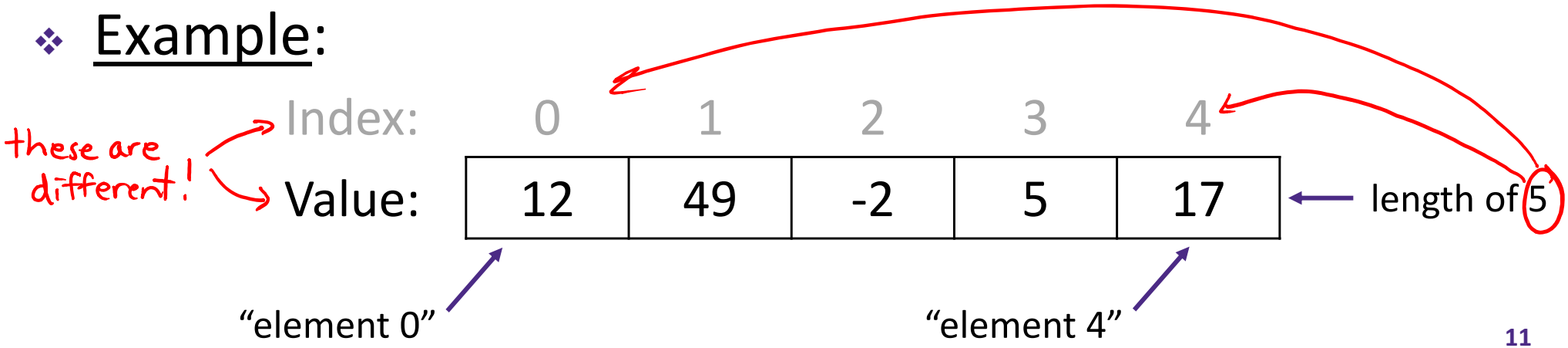
- Arrays store large amounts of data that you can access using a single name
  - Accessing arrays with loops is very convenient

- 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 (a house)
  - Index: a number that specifies the location of a particular element of the array (house number)
    - Start from 0, so numbered 0 to length - 1
  - Length: total number of elements in the array (# of houses on the street)

- Example:



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

like building num houses on your street

- 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)

color(0)

- 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!

(houses built, but no street name!)

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 (READ)
  - In *assignment*, modifies value of that index of the array (WRITE)

- Get length: `name.length`

  "dot notation"

- Example:

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

access value

change value

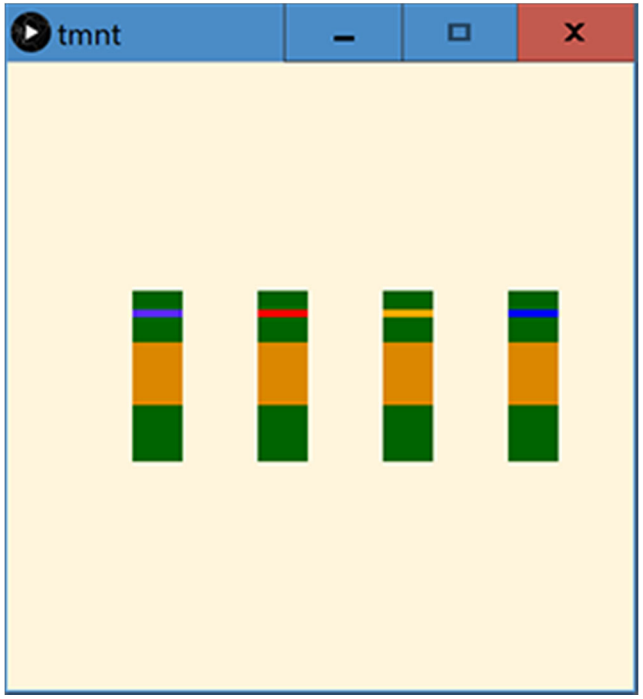
Index:	0	1	2	3	4
Value:	12	49	~~-2~~ 5	5	17

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




```
// array order: {don, raf, mic, leo}
int[] tmnt_x = {100,200,300,400};
color[] tmnt_c = {color(88,44,1410),color(255,0,0),color(255,171,3),colo

// draw TMNT using arrays
void draw() {
  background(255,245,220);       // paint over drawing canvas
  for(int i=0; i<tmnt_x.length; i=i+1) {
    tmnt(tmnt_x[i],tmnt_c[i]);
  }
```

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 find_smallest(float[] list) {
  int smallest = 0;
  for(int i = 0; i < list.length; i = i + 1) {
    if(list[i] < list[smallest]) {
      smallest = i;
    }
  }
  return smallest;
}
```

Arrays Visual Demo


```
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;
```

index: 0 1 2 3

ar | 0 | 0 | 0 | 0 |

ar | 0 | 1 | 0 | 0 |

ar | 0 | 1 | 0 | 2 |

true

ar | 0 | 1 | 4 | 2 |

ar | 3 | 1 | 4 | 2 |

error! there is no ar[4]

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`

same as regular variables