

Arrays Recap

Like any other animation variable: declare, initialize, use, update.

Unlike other animation variables, initialize involves two steps: initialize the array variable itself (create the array compartments), and initialize each compartment in the array.

There are three patterns for initializing each compartment:

- initialize every compartment to the same value
- initialize every compartment to a random value
- initialize every compartment to a different value but with a pattern – add additional loop variables to the typical do-something-to-every-slot-of-the-array loop

Array-ifying a Sketch

each ellipse may have a different, unrelated y coordinate so this animation variable becomes an array variable

```
float y;                                float[] y;
void setup () {                          void setup () {
  size(400,400);                          size(400,400);
  y = random(0,height);                    y = new float[10];
  }                                        for ( int i = 0 ; i < y.length ;
  }                                        i = i+1 ) {
  }                                        y[i] = random(0,height);
  }                                        }

void draw () {                            void draw () {
  background(255);                        background(255);
  fill(255,0,0);                          for ( int i = 0 ; i < y.length ;
  ellipse(width/2,y,20,20);                i = i+1 ) {
  }                                        fill(255,0,0);
  }                                        ellipse(width/2,y[i],20,20);
  }                                        }

  y = y+1;                                for ( int i = 0 ; i < y.length ;
  }                                        i = i+1 ) {
  }                                        y[i] = y[i]+1;
  }                                        }

drawing and animating that were done for the one thing are now repeated for each of the things, using the information from the corresponding compartment of the array
```

“Use” and “update” typically involve loops to use and update each compartment in the array.

Array Syntax Recap

- declare an array variable
`int[] a;`
- initialize an array variable (create the array compartments)
`int[] a = new int[10];`
- access one slot of the array
`a[4] // legal index values are 0 through the length
// of the array minus 1`
- the length of the array
`a.length`
- typical do-something-to-every-slot-of-the-array loop

```
for ( int i = 0 ; i < a.length ; i = i+1 ) {
  // do something involving a[i]
}
```

At the End of Class

Hand in whatever you have done during class, even if a sketch is incomplete.

- Make sure each sketch is named as directed and has a comment with the names of your group. Also be sure to save your sketches! (in Linux, this should be in your sketchbook `~/cs120/sketchbook`)
- Copy the entire directory for each sketch (not only the .pde file) into your handin directory (`/classes/cs120/handin/username`). You only need to hand in one copy for the group. (If you are running Processing on your computer instead of using the Linux virtual desktop, you will need to use FileZilla to copy the sketches.)

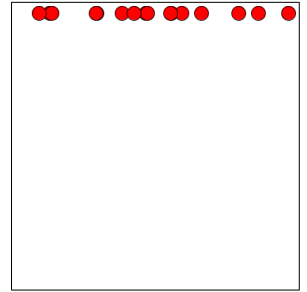
Exercises

For all sketches, be sure to **include a comment with the names of your group at the beginning of the sketch.**

if you finished this exercise on Monday, move on to #2, otherwise save Monday's work with the new name and continue from there

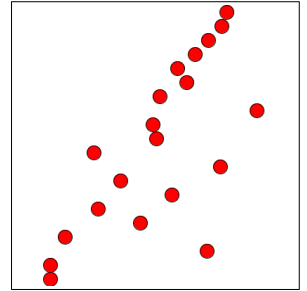
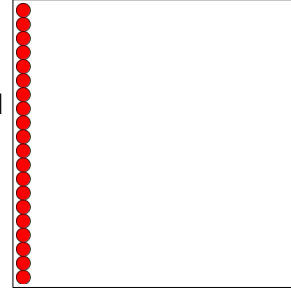
1. Create a new sketch called **sketch_241106a** and paste in the contents of the starter_goright sketch from the schedule page. That sketch draws a single circle which starts in the upper left corner of the window and moves right with a constant but randomly-chosen speed.

Modify the sketch to have 20 circles which start in the upper left corner of the window and then move right with constant but randomly-chosen speeds. (Each circle should have its own randomly-chosen speed.)



2. Save a copy of your sketch from #1 as **sketch_241106b**, then modify the copy so that the circles are drawn with different y coordinates. (The first picture shows the starting point and the second shows things after the sketch has run for a short while.)

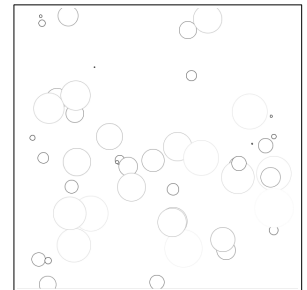
Note that this does not involve any new array variables – the y coordinate of each circle is predictably different from the one right above it.



3. Create a new sketch called **sketch_241106c** and paste in the contents of the starter_raindrop2 sketch from the schedule page. That sketch draws a single expanding circle – the circle starts at a random position with a random size and color (grayscale), then expands (increasing the size) and fades (making the color more white). When the color is white, the circle is given a new random position and the size is reset to 0 and the color is reset to black. Clicking the mouse toggles the circle's fill color between white and the circle's current color.

Modify the sketch so that there are 50 expanding-and-fading circles.

Clicking the mouse should toggle the fill for all of the circles. Which of the animation variables will need to become arrays?



If you have time –

- Save a copy of your sketch from #3 as **sketch_241106d**, then modify the copy so that clicking the mouse on a circle toggles (just) its fill between white and its current color without affecting the other circles. Will you need any more arrays?

You can tell if a particular circle was clicked on by using the `dist` function – the click is inside the circle if the distance between the mouse position and the center of the circle is less than the radius of the circle.

- Create a new sketch called **sketch_241106e** which draws 50 time-delayed circles – the circles should start on the left side of the window with random y coordinates, and every so often another circle should start moving to the right. All of the circles move at the same speed once they start moving.
 - Start with a single circle that starts on the left side of the window with a random y coordinate and moves to the right.
 - To add a random time delay, use a variation of the state machine pattern where there is a (`int`) countdown variable instead of a state variable. Initialize the countdown variable to some value greater than 0, use it by moving the circle only when the countdown has reached 0 (i.e. the countdown variable's value is less than or equal to 0), and update it by subtracting 1. Choose a random value between 0 and 500 for the delay. Since `random()` generates floats, use the `int()` function to get random integers – `int(random(0, 500))`.
 - Finally, modify the sketch so there are 50 circles. Which of the animation variables will need to become arrays?