

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;
void setup () {
  size(400,400);
  y = random(0,height);
}

void draw () {
  background(255);
  fill(255,0,0);
  ellipse(width/2,y,20,20);
  y = y+1;
}
```

create the compartments (one for each ellipse), then initialize each compartment

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

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

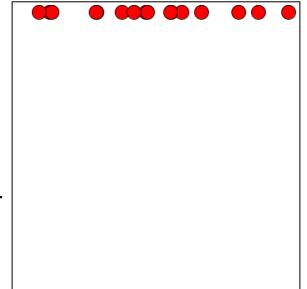
- Make sure each sketch is named as directed and has a comment with the names of your group. Also be sure to save your sketches! (in Linux, this should be in your sketchbook `~/cs120/sketchbook`)
- Copy the entire directory for each sketch (not only the .pde file) into your handin directory (`/classes/cs120/handin/username`). 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.**

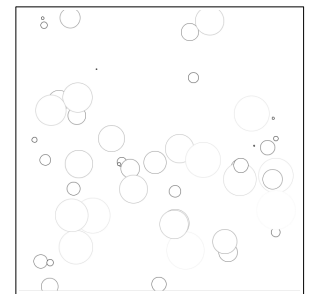
1. Create a new sketch called **sketch_241104a** and paste in the contents of the `starter_goright` sketch from the schedule page. That sketch draws a single circle which starts in the upper left corner of the window and moves right with a constant but randomly-chosen speed.

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.)



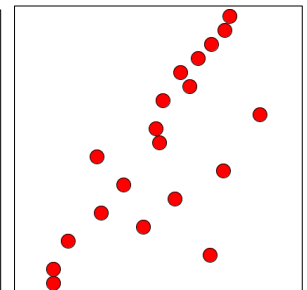
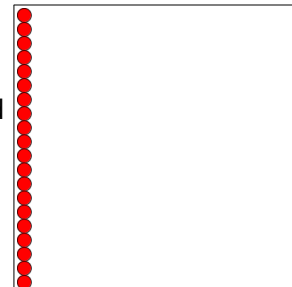
2. Create a new sketch called **sketch_241104b** and paste in the contents of the `starter_raindrop` sketch from the schedule page. That sketch draws a single expanding circle – the circle starts at a random position with a random size and color (grayscale), then expands (increase the size) and fades (make the color more white). When the color is white, it is given a new random position and the size is reset to 0 and the color is reset to black.

Modify the sketch so that there are 50 expanding-and-fading circles, each of which starts with a random position, size, and color and resets when the color reaches white.



3. Save a copy of your sketch from #1 as **sketch_241104c**, 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.



If you have time –

- Create a new sketch called **sketch_241104d** and paste in the contents of the `starter_raindrop` sketch from the schedule page. Modify the sketch so that clicking the mouse toggles between having the circle filled with white and having it filled with its current color. (Note – no arrays here!) Also save a copy of the sketch as **sketch_241104e**. (You should end up with two copies of the same sketch, **sketch_241104d** and **sketch_241104e**.)
- Modify **sketch_241104d** so that there are 50 expanding-and-fading circles. Clicking the mouse should toggle between having all of the circles filled with white and having all of the circles filled with their current colors.
- Modify **sketch_241104e** so that there are 50 expanding-and-fading circles and clicking on a particular circle toggles between having that circle filled with white and having it filled with its current color. (You can tell if a particular circle was clicked on by using the `dist` function – if the distance between the mouse position and the center of the circle is less than the radius of the circle, the click was inside the circle.)