Lab 10

- the idea of array-ifying is that the sketch still has the same steps, but the steps now involve multiple things
  - "do step" → "do step to all things"
- prefer a loop structure that reflects that –
  - "do step 1" → "do step 1 to all things"
  - "do step 2" → "do step 2 to all things"

  instead of
  - "do step 1", "do step 2"→ "do step1, step2 to all things"

Lab 10

```
// an object whose animation starts after some period of time

float x, y;  // ellipse's position
int timer;   // remaining delay until ellipse starts moving

void setup () {
  size(400, 400);

  // ellipse starts in a random position
  x = random(0, width);
  y = random(0, height);

  // ellipse will start moving after 200 frames
  timer = 200;
}

void draw () {
  background(255);

  // always draw the ellipse
  fill(255, 0, 0);
  ellipse(x, y, 10, 10);

  // always decrease the timer
  timer = timer-1;

  // only move the ellipse if the timer has gotten to 0 (or below)
  if ( timer <= 0 ) {
    x = x+1;
  }
}
```

properties of the object

one step "initialize the object's properties" with three substeps
→ prefer one loop initializing all three properties in the body

`draw()` as a whole is two steps – "draw the frame", "update the things that change"

"draw the frame" can be viewed as simply "draw the object" with two substeps for "draw the object"
→ prefer one loop with both `fill` and `ellipse` in the body

"draw the frame" can also be viewed as two steps "set fill color, draw the object"
→ prefer `fill` before the loop, then one loop with `ellipse` in the body

"update the things that change" is one step "update the object's properties" with two substeps
→ prefer one loop with all up the update in the body

Lab 10

- prefer loop variables to formulas – easier to figure out

- Change the initialization of the delays so that the first circle starts to move after 40 frames, the next after 80 frames, the next after 120 frames, etc.

```
for ( int i = 0 ; i < x.length ; i = i+1 ) {
  x[i] = random(0,width);
  y[i] = random(0,height);
  timer[i] = (i+1)*40;
}

for ( int i = 0, delay = 40 ; i < x.length ; i = i+1, delay = delay+40 ) {
  x[i] = random(0,width);
  y[i] = random(0,height);
  timer[i] = delay;
}
```

both of these are correct solutions, but with the `delay` loop variable is it easier to correctly start the first circle after 40 frames instead of immediately – e.g. a common bug with the formula is to use `i*40` instead of `(i+1)*40`