

Name: _____

Boids I Worksheet

Exercise 2

In order to have 100 boids, which of the animation variables in `starter_boids` will become array variables?

In this case, the neighborhood radius and angle and the maximum acceleration and speed will be species-specific rather than different for individual boids.

```
PVector pos, vel;           // boid's position and velocity
float radius, angle;        // boid's neighborhood
float maxforce, maxspeed;    // boid's maximum acceleration and speed
```

(continued on the next page)

Draw a box around the following chunks of code below:

- the initialization of each animation variable that becomes an array variable
- drawing the boid
- updating the animation variables for the boid

When considering what goes into each box, note the steps 2-4 are all part of updating the boid and logically should be kept together. Also note that the local variable `steer` is involved in both step 2 and step 3 so that those steps must be kept together.

```
void setup () {
    size(600, 600);

    // start with random position and velocity
    pos = new PVector(random(0, width), random(0, height));
    vel = new PVector(random(-1, 1), random(-1, 1));

    // other boid parameters
    radius = 60;
    angle = radians(135);
    maxforce = .2;
    maxspeed = 3;
}
```

```
void draw () {

    // 1 - draw scene
    background(255);
    drawBoid(pos, vel, 255, 0, 0);

    // 2 - compute net steering force
    // a - compute steering force for each behavior

    // b - combine forces (one behavior)
    PVector steer = new PVector(0,0);

    // c - limit the size of the force that can be applied
    steer.limit(maxforce);

    // 3 - update boid's velocity
    // a - add net steering force
    vel.add(steer);
    // b - limit boid's max speed
    vel.limit(maxspeed);

    // 4 - update boid's position
    // a - update position by adding velocity
    pos.add(vel);
    // b - wrap at edges of window
    if ( pos.x > width ) {
        pos.x = 0;
    } else if ( pos.x < 0 ) {
        pos.x = width;
    }
    if ( pos.y > height ) {
        pos.y = 0;
    } else if ( pos.y < 0 ) {
        pos.y = height;
    }
}
```

Exercise 3

The rules for how each boid moves are given in the handout:

- If there's an obstacle in range, the boid should avoid it.
- Otherwise if there's potential for colliding with another boid, avoid the collision.
- Otherwise if the boid is near the mouse (within 200 pixels), seek on the mouse position.
- Otherwise wander.

This is already written in language that closely matches the structure of an `if` statement. Identify how you are going to translate each condition into code – write the specific condition (in Processing) that you will use in your sketch.

there's an obstacle in range	
there's potential for colliding with another boid	
the boid is within 200 pixels of the mouse position	