Arrays
1. Declare an array variable to hold elements of a given type
   
   ```java
   String[] names;
   ```

2. Create (size) the array and assign the new array to the array variable
   
   ```java
   names = new String[100];
   ```

3. Assign values to the array locations
   
   ```java
   for (int i=0; i < names.length; i++) {
       names[i] = "George the " + i + "th";
   }
   ```

4. Use elements of the array
   
   ```java
   int r = int(random(names.length));
   println("My new name is " + names[r]);
   ```

Objects
Declare a class = instructions for creating a new object
1. Start the new class declaration with a class keyword, object name and curly brackets

2. Declare required fields inside the curly brackets (if any)
   - Fields are variables declared within the class

3. Declare a constructor, which is executed when a new object is created
   - Similar to a function
   - Same name as class
   - No return type at all (not even void)
   - Declare arguments
   - Add constructor implementation, which may include initializing fields with arguments

4. Declare required methods inside the class (if any)
   - Similar to functions

Using Objects
// MoverBox 
int nbBoxes = 20; 
Mover[] boxes = new Mover[nbBoxes]; // (1) Variable to hold objects 
for (int i=0; i<nbBoxes; i++)
   boxes[i] = new Mover(0.0, 0.0); // (2) Declare and initialize 
box[i].draw(); // (3) Draw the box in each iteration 

// MoverBox 
void setup()
{
    float[] a = new float[3];
    //float[] a = new float[3] { 1.2, 2.3, 3.4 };
    for (int i=0; i<a.length; i++)
    {
        println("a[i] = " + a[i]);
    }
}

void step()
{
    x = x + vx; // Motion
    if (x > width || x < 0.0)
    {
        vx = -vx;
    }
}

void draw()
{
    fill(200);
    rect(x, y, 20, 20);
}

void mousePressed()
{
    // (2) Create a new Box at mouse position and add to the array
    boxes[nextIndex] = new Mover(mouseX, mouseY);
    nextIndex = (nextIndex + 1) % nbBoxes;
}