

Arrays

1. Declare an array variable to hold elements of a given type

```
String[] names;
```

2. Create (size) the array and assign the new array to the array variable

```
names = new String[100];
```

3. Assign values to the array locations

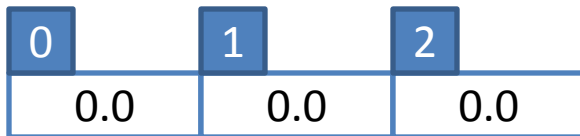
```
for (int i=0; i < names.length; i++) {  
    names[i] = "George the " + i + "th";  
}
```

4. Use elements of the array

```
int r = int( random( names.length ) );  
println("My new name is " + names[r]);
```

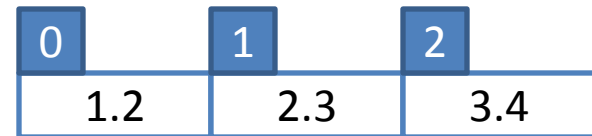
Arrays

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



A diagram of a memory array with three slots. Each slot is represented by a blue box with a white header containing an index (0, 1, or 2) and a white body containing the value 0.0. Red arrows from the code above point to these boxes: one from the first parameter '3' to the first slot, and one from the 'a[i]' expression in the loop to the third slot.

0	1	2
0.0	0.0	0.0



A diagram of a memory array with three slots, similar to the one above but with populated values. Each slot is represented by a blue box with a white header containing an index (0, 1, or 2) and a white body containing the values 1.2, 2.3, and 3.4 respectively. Red arrows from the code above point to these boxes: one from the first parameter '3' to the first slot, and one from the 'a[i]' expression in the loop to the third slot.

0	1	2
1.2	2.3	3.4

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

```
// A simple Box class
class Mover {
    float x, y, vx, vy;

    Mover(float tx, float ty) {
        x = tx;    // x position
        y = ty;    // y position
        vx = 1.0;  // x velocity
        vy = 0.0;  // y velocity
    }

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

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

1. Start the new class declaration with a class keyword, object type name and curly brackets
2. Declare required fields inside the class (if any)
3. Declare a constructor, which is executed when a new object is created
4. Declare required methods inside the class (if any)

Using Objects

```
// BoxMover
int nBoxes = 20;
Mover[] boxes = new Mover[nBoxes];    // (1) Variable to hold objects

int nextIndex = 0;

void setup() {
    size(500, 500);
    rectMode(CENTER);
}

void draw() {
    background(0);

    for (int i=0; i<boxes.length; i++) {
        if (boxes[i] != null) {
            boxes[i].step();              // (3) Call object methods
            boxes[i].draw();
        }
    }
}

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