Arrays

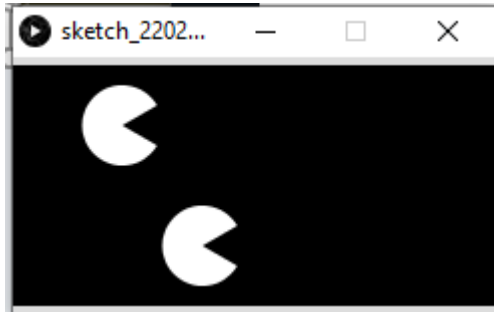
The University of Mount Union

Too many variables

```
float x1 = -20;
float x2 = 20;

void setup() {
  size(240, 120);
  noStroke();
}

void draw() {
  background(0);
  x1 += 0.5;
  x2 += 0.5;
  arc(x1, 30, 40, 40, 0.52, 5.76);
  arc(x2, 90, 40, 40, 0.52, 5.76);
}
```

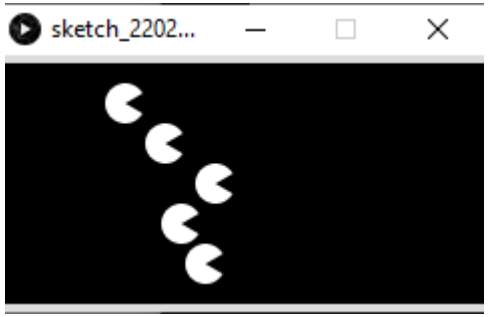


Way too many variables

```
float x2 = 10;
float x3 = 35;
float x4 = 18;
float x5 = 30;

void setup() {
  size(240, 120);
  noStroke();
}

void draw() {
  background(0);
  x1 += 0.5;
  x2 += 0.5;
  x3 += 0.5;
  x4 += 0.5;
  x5 += 0.5;
  arc(x1, 20, 20, 20, 0.52, 5.76);
  arc(x2, 40, 20, 20, 0.52, 5.76);
  arc(x3, 60, 20, 20, 0.52, 5.76);
  arc(x4, 80, 20, 20, 0.52, 5.76);
  arc(x5, 100, 20, 20, 0.52, 5.76);
}
```



What can I do? Use an array!

sketch_220215a

```
float[] x = new float[3000];

void setup() {
  size(240, 120);
  noStroke();
  fill(255, 200);
  for (int i = 0; i < x.length; i++) {
    x[i] = random(-1000, 200);
  }
}

void draw() {
  background(0);
  for (int i = 0; i < x.length; i++) {
    x[i] += 0.5;
    float y = i * 0.04;
    arc(x[i], y, 12, 12, 0.52, 5.76);
  }
}
```

A Conceptual Look

```
int[] years = { 1920, 1972, 1980, 1996, 2010 };
```

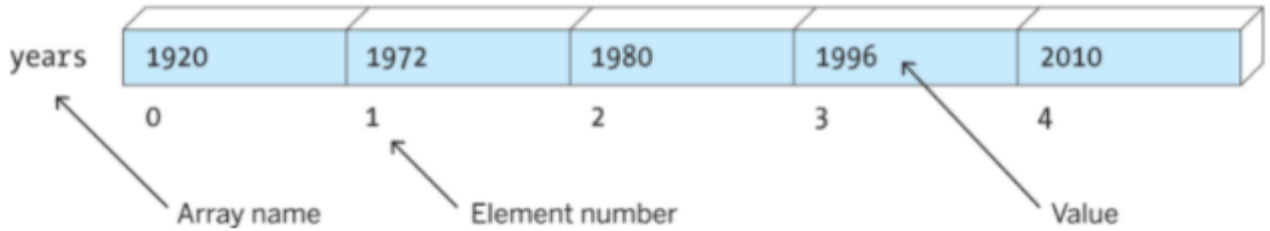


Figure 11-1. An array is a list of one or more variables that share the same name

Each element in this array can be generically referred to as `years[i]` pronounced "years sub i"

A specific element can be referred to like this: `years[3]` or "years sub 3." The current value of `years[3]` is 1996.

It might seem strange to start at 0 instead of 1, but think of it as distance from the beginning of the array.

Declaring & Instantiating

sketch_220215a

```
1  float[] x = new float[3000];
2
3  void setup() {
4    size(240, 120);
5    noStroke();
6    fill(255, 200);
7    for (int i = 0; i < x.length; i++) {
8      x[i] = random(-1000, 200);
9    }
10 }
11
12 void draw() {
13   background(0);
14   for (int i = 0; i < x.length; i++) {
15     x[i] += 0.5;
16     float y = i * 0.04;
17     arc(x[i], y, 12, 12, 0.52, 5.76);
18   }
19 }
```

This line **declares** and **instantiates** (or, in the book's language, *creates*) the array, giving it a name and a length. It has slots for 3000 floating-point variables, called **elements** of the array.

All the elements of an array must be the same data type!

This could also be done in two separate lines, with

```
float[] x;
```

at the top and

```
x = new float[3000];
```

in `setup()`.

Instantiating in `setup()` is fine, but avoid instantiating in `draw()` as it slows down the frame rate.

Assigning Values

sketch_220215a

```
1  float[] x = new float[3000];
2
3  void setup() {
4    size(240, 120);
5    noStroke();
6    fill(255, 200);
7    for (int i = 0; i < x.length; i++) {
8      x[i] = random(-1000, 200);
9    }
10 }
11
12 void draw() {
13   background(0);
14   for (int i = 0; i < x.length; i++) {
15     x[i] += 0.5;
16     float y = i * 0.04;
17     arc(x[i], y, 12, 12, 0.52, 5.76);
18   }
19 }
```

This stores the length of the array.

This `for` loop is assigning random values between -1000 and 200 to each element of the array. We start at 0 because the first **index value** of an array is 0. Since this array has 3000 elements, its last index value is 2999.

The first element would be written like this:
`x[0]`
And the last like this:
`x[2999]`

Why do we check for `i<x.length` and not `i<=x.length`?

Because we started with 0, not 1!

Using an array

sketch_220215a

```
1  float[] x = new float[3000];
2
3  void setup() {
4    size(240, 120);
5    noStroke();
6    fill(255, 200);
7    for (int i = 0; i < x.length; i++) {
8      x[i] = random(-1000, 200);
9    }
10 }
11
12 void draw() {
13   background(0);
14   for (int i = 0; i < x.length; i++) {
15     x[i] += 0.5;
16     float y = i * 0.04;
17     arc(x[i], y, 12, 12, 0.52, 5.76);
18   }
19 }
```

This `for` loop adds 0.5 to each element of the array every time `draw()` runs (60 times per second).

Using an array (continued)

sketch_220215a

```
1  float[] x = new float[3000];
2
3  void setup() {
4    size(240, 120);
5    noStroke();
6    fill(255, 200);
7    for (int i = 0; i < x.length; i++) {
8      x[i] = random(-1000, 200);
9    }
10 }
11
12 void draw() {
13   background(0);
14   for (int i = 0; i < x.length; i++) {
15     x[i] += 0.5;
16     float y = i * 0.04;
17     arc(x[i], y, 12, 12, 0.52, 5.76);
18   }
19 }
```

This line assigns a value to **y** each time the loop runs: the current value of **i** multiplied by 0.04. The values of **y** will thus range from 0 (0 * 0.04) to 119.96 (2999 * 0.04), which is very close to the height of the sketch (120).

Using an array (continued again)

sketch_220215a

```
1  float[] x = new float[3000];
2
3  void setup() {
4    size(240, 120);
5    noStroke();
6    fill(255, 200);
7    for (int i = 0; i < x.length; i++) {
8      x[i] = random(-1000, 200);
9    }
10 }
11
12 void draw() {
13   background(0);
14   for (int i = 0; i < x.length; i++) {
15     x[i] += 0.5;
16     float y = i * 0.04;
17     arc(x[i], y, 12, 12, 0.52, 5.76);
18   }
19 }
```

Here we draw our Pac-Men. Each arc is centered at `(x[i], y)` initially and moves horizontally 0.5 pixels every time a frame runs. Remember that the frame rate is 60 frames/second, so each Pan-Man moves 30 pixels to the right horizontally each second. This movement occurs because of the

`x[i] += 0.5;`

line.