Iteration

- Iteration is the process of repeating a set of statements
- Often, the statements are slightly altered each time they are executed
- For example, here is code that draws 4 horizontal equally spaced lines

```
line(0, 20, width, 20);
line(0, 40, width, 40);
line(0, 60, width, 60);
line(0, 80, width, 80);
```

- Can we do better? What does 'better' mean?

Iteration

- The previous example is not so good because we did not use any variables!
- Same example using variables

```
int y = 20;
line(0, y, width, y);
y = 40;
line(0, y, width, y);
y = 60;
line(0, y, width, y);
y = 80;
line(0, y, width, y);
```

- What if we wanted lines spaced 30 pixels apart?

Iteration

- Same example using variables

```
int y = 20;
int dy = 20;
line(0, y, width, y);
y = y + dy;
line(0, y, width, y);
y = y + dy;
line(0, y, width, y);
y = y + dy;
line(0, y, width, y);
```

- What if we wanted 5 lines?
- What if we wanted 10 lines?
- What if we wanted 100 lines?
- What if we wanted 1,000,000 lines?

While loop

- Processing has a control structure called the while loop that has the following form

```
while (condition) {
// body of loop
}
```

- Here the statements in the body of the loop are executed while the parenthesized condition is true
- We can simplify and generalize our line drawing example as follows:

```
int y = 20;
while (y <= 80) {
    line(0,y,width,y);
    y = y + 20;
}
```

Loop Requirements

- Every loop needs three things
  - Initialization
  - A termination condition
  - Progress towards termination

```
int y = 20;                    // initialization
while (y <= 80) {              // term cond: y > 80
   line(0,y,width,y);
   y = y + 20;                 // progress -> term
}
```

- All three must be present to have a properly functioning loop
- If we miss one of these, we may form an infinite loop

The map() function

- Processing has a handy little function called `map(`v`,` a_1`,` b_1`,` a_2`,` b_2`)` which remaps the value v in the interval $[a_1, b_1]$ to the value in the interval $[a_2, b_2]$ using the linear mapping

$$\texttt{output} = \frac{b_2 - a_2}{b_1 - a_1} v$$

- This is very handy in doing coodinate transformations, such as a real interval and screen coodinates.
- Let's plot the sine function from $[-2\pi, 2\pi]$ to fit exactly in our display window.
- Let's draw a circle of radius 100 centered inside our window