Variables and For-loops

Variables are the process of replacing concrete values with symbols in order to generalize a computation to work in more than one situation.

```c
void draw()
{
    stroke( 0, 0, 0 );
    fill( 255, 50, 50 );
    int x = 100;
    int y = 200;
    rect( x, y, 150, 50 );
}
```

“Int” stands for “integer”, a whole number with no decimal digits, e.g., 3

“Float” designates a number that can have decimal digits, e.g., 3.14

Variables – using symbols instead of just numbers

We can use variables to capture relationships.

Arithmetic operations in programming are:
+ Addition
- Subtraction
* Multiplication
/ Division
( ) Grouping

Drawing One Rectangle is Pretty Straightforward

```c
rect( 100, 200, 150, 50 );
```

For-loops to the Rescue!

Repeating a code pattern is a recurring theme in programming.

This line is called a “for-loop”. It is very handy for repeating patterns of code. It expresses those patterns as relationships.

The for-loop executes the commands in the curly braces a bunch of times. Its use looks like this:

Keep looping as long as this equation is true

```c
for( x = 0; x < 400; x = x + 10 )
{
    rect( x, 150, 50 );
}
```

But, This Gets Awfully Boring if You Want to Draw 100 Rectangles!

```c
rect( 100, 200, 150, 50 );
rect( 110, 210, 150, 50 );
rect( 120, 220, 150, 50 );
```

For-loops to the Rescue!

```c
for( int x = 0; x < 400; x = x + 10 )
{
    rect( x, 150, 50 );
}
```
For-loops to the Rescue!

The `map()` function

This function takes an input value, the range of values it lives between, and the range of output values. It returns the output value that corresponds to the input value.

So, for example, if we wanted to turn an x value into a red color, we might say:

```
int red = int( map( x, 0, width - 1, 0, 255 ) );
```

More Sophisticated Relationships:

The `map()` function can also do blending by interpolating one forward and the other one backwards.

```
void draw()
{
  stroke( 0, 0, 0 );
  for( int x = 0 ; x < 400 ; x = x + 10 )
  {
    int x = x;
    int red = int( map( x, 0, width, 0, 255 ) );
    int green = int( map( y, 0, width, 0, 255 ) );
    //printf( "x = %d, y = %d, red = %d, green = %d\n", x, y, red, green );
    fill( red, green, 50 );
    rect( x, y, 198, 50 );
  }
}
```