# Variables and Operators Lab

In this lab we'll take the next steps to creating the graph. When we're finished we'll have lines on our graph that will show the overall trend of our survey results. We hope that the 'excellent' and 'good' responses rise over the month and the 'poor' and 'terrible' responses drop.

Excited? Let's do this!

## Creating some number variables

1. Go to your drawTrends() function.
2. Add these variables: leftSideOfGraph, widthOfGraph, topOfGraph, heightOfGraph. Their names are pretty self-explanatory so give them initial values -- estimate what you think they should be.
3. Let's test your guess! Draw a test rectangle kind of like this:

```
context.fillRect(leftSideOfGraph,topOfGraph,widthOfGraph,heightOfGraph);
```

4. Adjust their values so your rectangle fits on your graph.
5. Now leave the variables but remove the rectangle. It was only there for testing the size of your graph.
6. Add a couple more variables: rightSideOfGraph and bottomOfGraph. Calculate these numbers using leftSideOfGraph, widthOfGraph, topOfGraph, and heightOfGraph.
7. Redraw your axes using these new values instead of your hardcoded numbers.
8. Rewrite your labels using these values instead of your hardcoded numbers. You'll want to do a little math like adding to bottomOfGraph to push your x-axis label down and subtracting from leftSideOfGraph to push your y-axis label left.

Coding in this way allows you to reposition or rescale your graph (within reason) by adjusting these variables. In fact, let's test that out.

9. Change the values slightly and re-load your page. Make a few adjustments to prove to yourself that these new variables are being used properly.

## Creating some string variables

Similarly we can adjust colors globally.

10. Add some new variables at the top that represent labelsColor, backgroundLinesColor, axesColor and of course set them equal to some colors you like.
11. Use those variables in place of any colors you have hardcoded previously.
12. While we're here, create these new colors also:

```
var excellentColor = "green";
var goodColor = "blue";
var okayColor = "purple";
var poorColor = "orange";
var terribleColor = "red";
```

13. Use those variables in place of any colors you have hardcoded previously.

```
var xTic=(rightSideOfGraph-leftSideOfGraph)/(29); //pixels per day
var yTic = (bottomOfGraph - topOfGraph) / (100); //pixels per vote
```

We'll use these to draw the trend lines.

## Drawing the overall trend

Let's draw some actual lines on the graph. We'll start out simple; we'll just draw a line from the first data point to the last.

14. Create a new function called plotPoints(). It should receive two strings; the level (Excellent, Good, Okay, Poor, or Terrible), and the color. It will look like this:

```
function plotPoints(level, color) {
  //TODO: Get a reference to the canvas and get the 2d context.
  //TODO: Start the Path
  //TODO: Set the stroke style to the 'color' passed in above
  context.moveTo(getXPointOnGraph(0,level), getYPointOnGraph(0,level));
  context.lineTo(getXPointOnGraph(29,level),getYPointOnGraph(29,level));
  context.stroke();
  //TODO: Close the Path
}
```

15. Lucky you! See all those "TODO" comments? Put in JavaScript code to make that work. (Note: this will not work until you write the functions in the next steps).
16. Now create the getXPointOnGraph() and getYPointOnGraph() functions:

```
function getXPointOnGraph(sequence, level) {
  return leftSideOfGraph + (sequence * xTic);
}
function getYPointOnGraph(sequence, level) {
  var value = raw_survey_results[sequence][level];
  return bottomOfGraph - (value * yTic);
}
```

17. Now call your functions like this:

```
//Draw the Excellent lines in green, Good ones in blue, and so on...
plotPoints("Excellent", excellentColor);
plotPoints("Good", goodColor);
plotPoints("Okay", okayColor);
plotPoints("Poor", poorColor);
plotPoints("Terrible", terribleColor);
```

18. Run and test. You should expect to see five different colored lines on your graph.

Once you see them, you can be finished!