CODE TO NOTE

INTERNAL PULL-UP RESISTOR:

```
pinMode(pin, INPUT_PULLUP);
```

To declare a standard input, use the line `pinMode(pin, INPUT);`. If you would like to use one of the RedBoard's built-in pull-up 20kΩ resistors, it would look like this: `pinMode(pin, INPUT_PULLUP);`. The advantage of external pull-ups is being able to choose a more exact value for the resistor.

DIGITAL INPUT:

```
digitalRead(pin);
```

Check to see if an input pin is reading HIGH (5V) or LOW (0V). Returns TRUE (1) or FALSE (0) depending on the reading.

IS EQUAL TO:

```
if(digitalRead(pin) == LOW)
```

This is another logical operator. The "is equal to" symbol `==` can be confusing. Two equals signs are the same as asking, "Are these two values equal to one another?" Contrarily, one equals sign means assigning a particular value to a variable. Don't forget to add the second equals sign if you are comparing two values.

CODING CHALLENGES

CHANGE THE KEY OF EACH BUTTON: Use the frequency table in the comment section at the end of the code to change the notes that each button plays.

PLAY MORE THAN THREE NOTES WITH IF STATEMENTS: By using combinations of buttons, you can play up to seven notes of the scale. You can do this in a few ways. To get more practice with **if** statements, try adding seven **if** statements and using the Boolean AND `&&` operator to represent all of the combinations of keys.

PLAY MORE THAN THREE NOTES WITH BINARY MATH: You can use a clever math equation to play more than three notes with your three keys. By multiplying each key by a different number, then adding up all of these numbers, you can make a math equation that produces a different number for each combination of keys.