CODE TO NOTE

CHARACTER VARIABLES:

```
void play(char note, int
beats)
```

The `char`, or **character**, variable stores character values. In this sketch, the **`play()`** function gets passed two variables: a character variable that represents the musical note we want to play and an integer variable that represents how long to play that note. A second array takes the character variable and associates a frequency value to it. This makes programming a song easier as you can just reference the character and not the exact frequency.

TONE FUNCTION:

```
tone(pin, frequency,
duration);
```

The `tone()` function will pulse power to a pin at a specific frequency. The duration controls how long the sound will play. `tone()` can be used on any digital pin.

DECLARING AN ARRAY:

```
array_name[array_size];
```

To declare an **array**, you must give it a name, then either tell Arduino how many positions the array will have or assign a list of values to the array. An array must contain all the same type of variables and be declared as such.

CALLING AN ARRAY:

```
array_name[index_#];
```

To call one of the values in an array, simply type the name of the array and the index of the value. Don't forget the index starts at 0, not 1, so to call the first element, use **`array_name[0];`**.

CODING CHALLENGES

CHANGE THE TEMPO OF THE SONG: Experiment with the `beatLength;` variable to change the tempo of the song.

MAKE YOUR OWN SONG: Try changing the notes to make a different song. Spaces “ █ ” can be used for rests in the song.