# Sound Check Experiment

This is a preliminary test for making sound with javascript. Right now, the code will be tangled together in an attempt to understand how everything works. The actual program will have much more separation of parts.

## Global Constants

The Lists will be constant throughout the whole program. The program will randomly select an item from one of these lists depending on the situation, and use that value to affect the sound in some way.

```javascript
const LIST_OF_WAVEFORM_TYPES = [
    'sine', 'square', 'sawtooth', 'triangle'
];
```

## Audio Context

Quote from Mozilla Developer - Audio Context

> The AudioContext interface represents an audio-processing graph built from audio modules linked together, each represented by an AudioNode. An audio context controls both the creation of the nodes it contains and the execution of the audio processing, or decoding. You need to create an AudioContext before you do anything else, as everything happens inside a context.

```javascript
var audioContext = new AudioContext();
var osc;
var waveformType = "square";
var is_sound_on = false;
```

## Button Events - Start and Stop the Sound

These functions simply start and stop the sound, and are intended to be used by button events on the HTML page. This is part of the interface, and should always be present in some form, just in case somebody wants to shut off the sound.

```javascript
var startSound = function() {
    if (is_sound_on) {return;}
    osc = audioContext.createOscillator();
    osc.type = waveformType;
    osc.connect(audioContext.destination);
    osc.start(audioContext.currentTime);
    is_sound_on = true;
    changeSoundStatus('True');
};
var endSound = function() {
    if (osc == undefined) {return;}
    osc.stop(audioContext.currentTime);
    is_sound_on = false;
    changeSoundStatus('False');
};
```

If we are using this script with the HTML webpage (which we are), then starting or stopping the sound can change the text in the interface. If, for some reason, we aren't in the webpage, then that function simply won't return anything.

```javascript
var changeSoundStatus = function (status) {
    var span = document.getElementById('sound_status');
    span.innerHTML = status;
    span.className = status;
};
var humanHearingCheck = function(frequency) {
    return;
};
```