Day 2:

Numpy Notes (Basic)

- Vectorization

```
import numpy as np

# common way
%%timeit

# list comprehension
[np.exp(1) for i in range(10000)]
--> 46.4 ms +/- 10 ms per loop...

# better way
%%timeit

np.exp(np.ones)
--> 195 microseconds +/- 10.6 microseconds per loop
```

```
import numpy as np

zip(range(1000), range(1000, 2000))

# common way
%%timeit

[xi*yi for xi, yi in zip(range(1000), range(1000, 2000))]
--> 380 microseconds +/- 67.2 microseconds per loop

# better way
%%timeit

np.arange(1000)*np.arange(1000, 2000)
--> 17.8 microseconds +/- 1.81 microseconds per loop
```

- Numpy Arrays

```
import numpy as np

np.arange(10)
--> [0 1 2 3 4 5 6 7 8 9]
```

```
x = np.array([12, 13, 100, 9])
x[1]
--> 13

x.shape
--> [4]

x = np.arange(9)
print(x)
--> [0 1 2 3 4 5 6 7 8]

x = x.reshape(3, 3)
print(x)
--> [[0, 1, 2], [3, 4, 5], [6, 7, 8]]

x.shape
--> [3, 3]

x[1, 2]
--> 5
```

- Numpy Slicing

```
import numpy as np

x = np.arange(27)
x
--> [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26]

x = x.reshape(3, 3, 3)
x[2, :, :]
--> array([[19, 19, 20],
          [21, 22, 23],
          [24, 25, 26]])
```

Sound

Physics of Sound

- Air is made up O_2, N_2 and other molecules
- Molecules that float around and bounce off matter due to electrons repelling
    - Pressure - increases with the force/strength with which molecules would collide with surface (average rate + increase/decrease in rate collision)

$$P = \frac{nRT}{V}$$

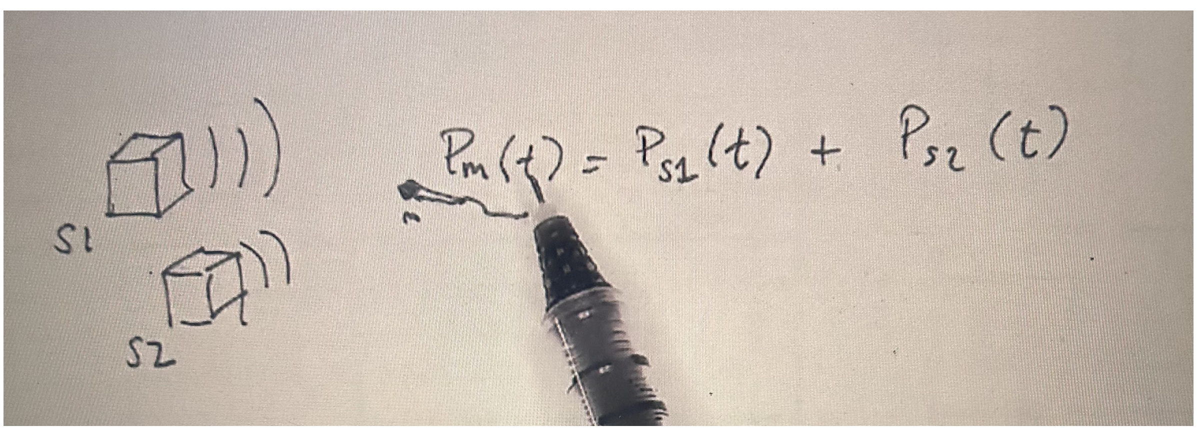
- Ways to increase pressure in container:
  - increase temperature
  - decrease volume
  - increase number of particles in container

- What do we hear when we perceive sound?
  - When we clap, we create a “peak” and a “trough” of molecules. That is, an area of many molecules and an area of a low number of molecules
  - In the process of stabilizing this pressure back to the average rate, the areas of pressure propagate and create sound

$$P(t) = Acos(2\pi ft + \phi)$$

$$P(t) = Acos(\frac{2\pi t}{T} + \phi - \frac{\pi}{2})$$

  - A - Amplitude (“loudness”); in Pascals: N/m^2
  - f - frequency
  - phi - phase shift



Recording an Audio Signal

- In physical world, sound is physical, but recording equipment can only hold so much data. Therefore, we sample the audio over regular intervals
- The continuous quantity is stored as a set of samples
- $f(t) := [f(0), f(T), f(2T), f(3T)...]$ where T is the sampling rate
- $N_s = floor(T_d * f) + 1$ where T_d is the total duration of the recording
- Nyquist Frequency - the minimum sampling frequency required for one to reliably reproduce the most information of audio
- The sampling is not the only discretized quantity—the amplitudes are also discrete (the nature of the data is fundamentally changing, none of the inputs and outputs are continuous).
- Example Audio Standard
    - 16-bit / 44100 Hz ~ 16 bits is the amount of data that can be alocated to store the information (2^{16} is the number of possible states that can be stored; usually using floating point arithmetic), while 44100 Hz is the sampling frequency or the amount of times per second the audio is sampled
- A (dynamic) microphone works by analyzing the change in flow of electrons in a wire induced by movement of a magnet by surrounding pressure
- Faraday’s Law
- Pulse-code Modulation
    - Pulse code modulation (PCM) is a digital representation of an analog signal. It's a method for converting an analog signal into a digital signal so that it can be transmitted through a digital communication network.
    - PCM works by sampling the amplitude of the analog signal at regular intervals. The sampled analog data is then changed to binary data, which is represented by only two possible states: high and low (0 and 1).