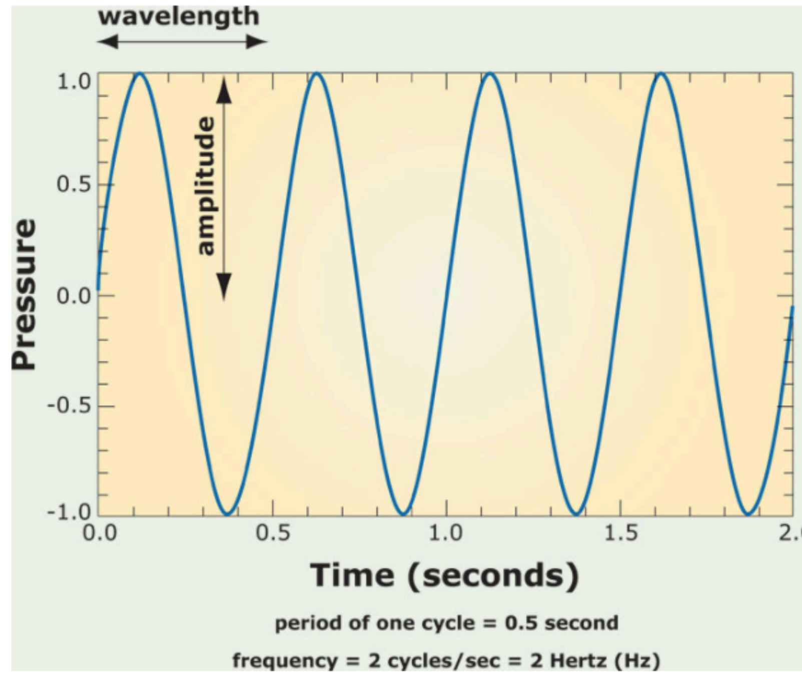
# Understanding Data

**Audio and Video**

# Sound Wave

sound is oscillating pressure travelling through some medium(air/water).

Important sound quantities:




**amplitude**: maximum displacement from equilibrium.

**period**: time to complete one cycle.(sec or sec per cycle)

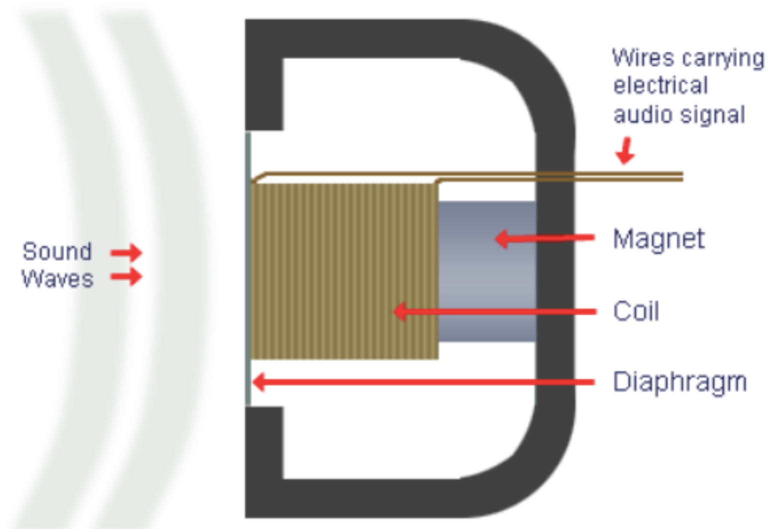
**frequency**: reciprocal of period (cycle per second or hert(hz))
(frequency of a note determines its **pitch**)

# Microphone vs. Speaker

Sound(pressure waves) vibrate a diaphragm(plastic, paper, metal)

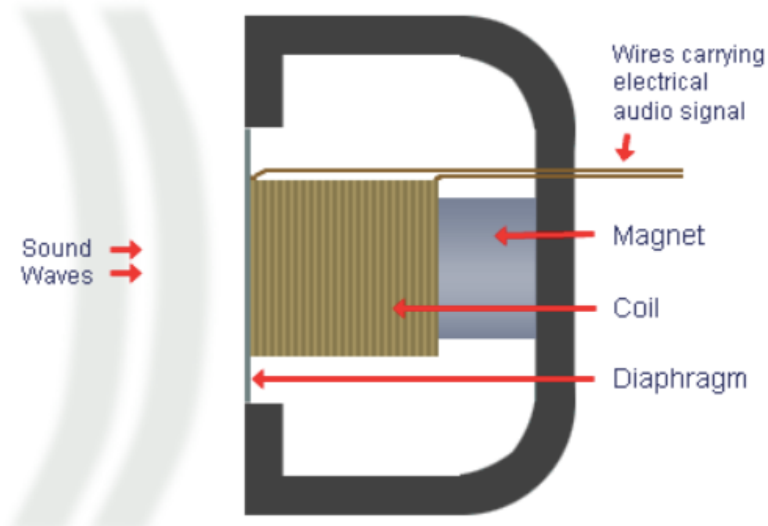
- which moves the electromagnet(coil) back and forth
- and causes the magnetic field to fluctuate
- inducing a current = electrical signal

A microphone is also known as a **transducer**; it converts acoustic energy to electrical energy.



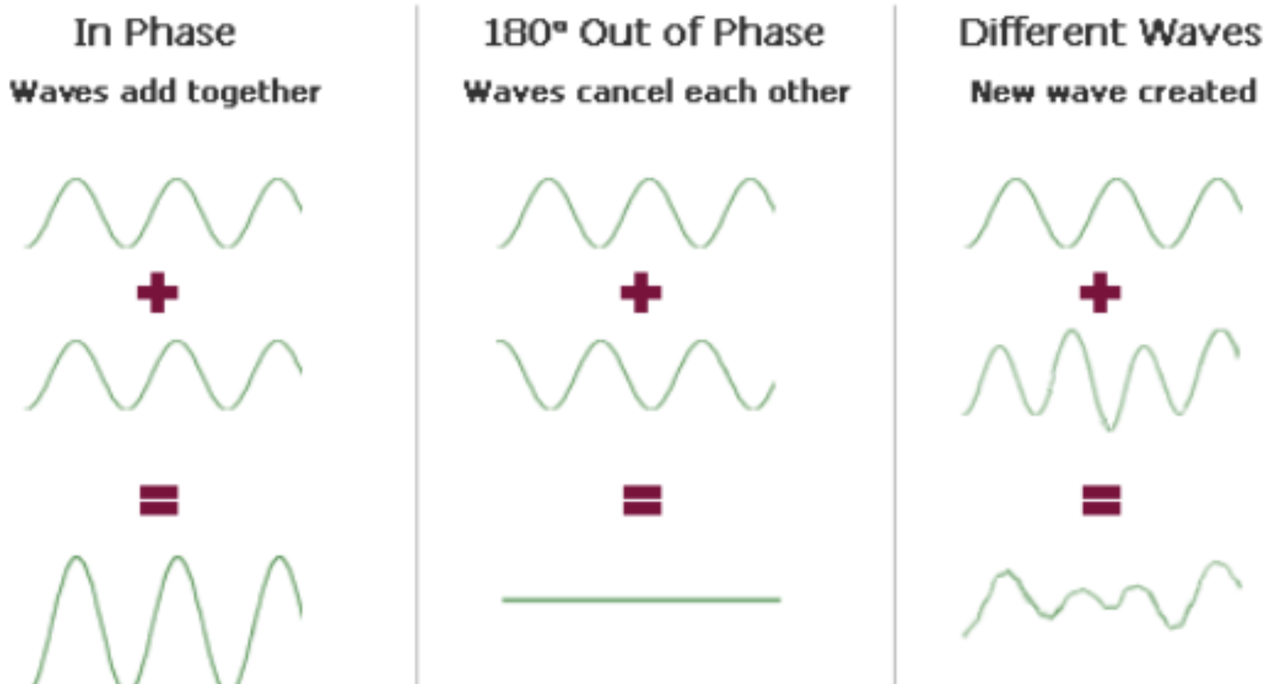
# Microphone vs. Speaker

Speakers work in reverse: An electrical signal cause the magnetic field to change and vibrate the diaphragm to produce the sound waves.



# Sound Wave

- sound waves interact to produce different waves.



# Digital vs. Analog

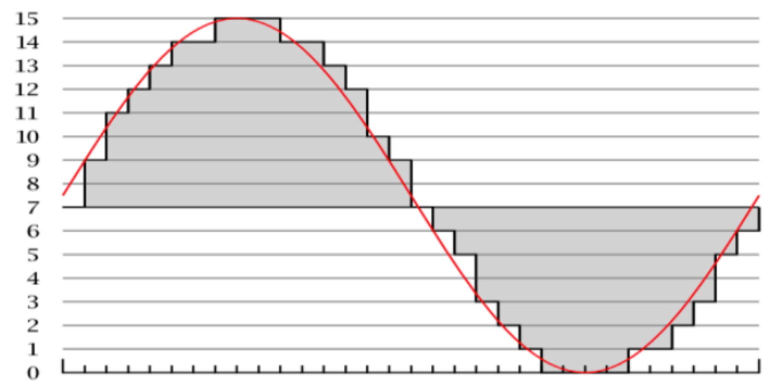
Sound can be modeled as a continuous, **analog** wave or signal.

Computers can only understand **digital** signals (discrete or finite signal(0s and 1s).)

- analog signals are continuous and can take on an infinite possible values(the real numbers)
- digital signals are finite.
- For example, 8 bit colors can take on one of 256 discrete, finite possibilities. But actual colors can take on any of an infinite possible values or shades.

**Sampling** allow computers to approximate analog signals such as sound.

The number of samples is the **sampling rate**, the higher the rate the better the quality.



# Sampling

CD-quality has a rate of 44,100 samples per second(44,100 Hz or 44.1 kHz).

Analog Low sampling rate High sampling rate

The possible values for each sample is determined by the **bit depth**.

- CD usually has 16-bit audio which means it can take on one of 2^16 or 65,536 values.
- This is analogous to 8-bit color for example taking on 2^8=256 possible values.
- More bits means more accuracy but more memory.

# MP3

A popular format to store audio is MP3(MPEG-2 Audio Layer III)

CD-quality(**PCM, pulse code modulation**) sounds require a lot of memory.

- Each of 44,100 samples require 16 bits(32 bits for left and right channel), that's approximately 0.2 MB per second. A 3-mins song would require about 30MB of memory! MP3 is a lossy compression format and only need 3MB!

MP3 compression uses **psychoacoustics** to take advantage of limitations in our biology and psychology of sound.

- human hearing range is 20 Hz to 20 kHz
- hard to differentiate between similar sounds(similar frequencies).
- humans are only good at differentiating pitches in the 2 kHz and 5 kHz range.
- in compression, sounds outside of human hearing range is thrown out and similar sounds can be combined.

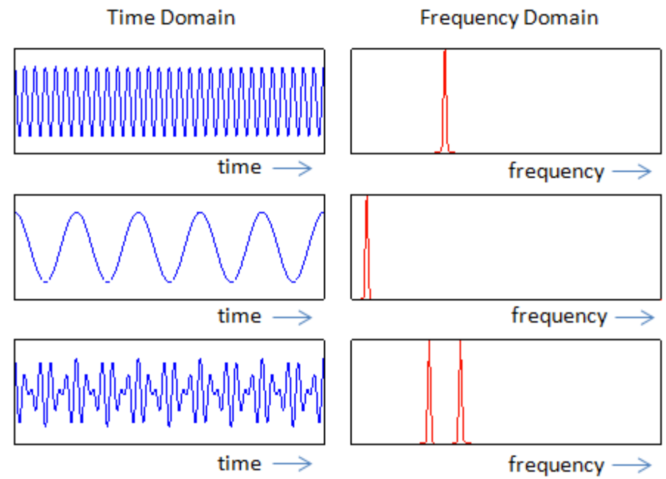
# Compression

In order to determine which data from the PCM encoding can be thrown away, your computer uses two different tools.

**Discrete Fourier Transform(DFT)**: convert sound from the time domain to frequency domain and filter out unnecessary frequencies.

The other is called the **Modified Discrete Cosine Transform**, or MDCT for short, which analyzes the entire spectrum of frequencies of the audio.

The result of this process is a visualization called a **spectrogram** which is essentially a fingerprint of the sound's frequencies. (Shazam app uses these techniques to identify songs.)




We'll use Python to do audio analysis as above in the next lecture.

# Popular Formats

Other popular audio formats include AAC, WMA, WAV, FLAC.

AAC(Advanced Audio Coding) was developed by Apple for iTunes but is currently adopted by many devices.

- more efficient compression and better quality than MP3

Windows Media Audio(WMA) is a proprietary format developed by Microsoft but has not been as popular as AAC or MP3.

FLAC(Free Lossless Audio Codec) is a lossless compression format.

Waveform Audio(WAV) was also developed by Windows

- uncompressed

# Video Codecs

When dealing with video on your computer, we're concerned with two different things: **codecs** and **containers**.

We saw that PCM audio can take up a lot of space, but uncompressed video is even larger. Without any compression, an hour of high quality video can be hundreds of gigabytes large!

The role of the codec is to do just that, as the codec (a combination of the words "encode" and "decode") is a program responsible for compressing and decompressing video.

# Video Codecs

Today, the most popular standard for compressing videos is called **H.264**, sometimes referred to as AVC, or advanced video coding.

For example, all Blu-Ray players must be able to decode video that has been encoded according to the H.264 standard

There are lots of codecs out on the Internet that compress video using H.264, including x264 and DivX.

- software like VLC has built in codecs that can generally play any video file.

# Containers

Video files are packaged up into a **container** which include video/audio files, captions, DVD menus, etc..

Popular video containers today include AVI, MKV, MP4, and MOV, and different container formats support different codecs for audio and video.

Containers give us a bit of flexibility in creating videos, since we get to choose which codec we want to use for the video and which codec we want to use for the audio, though not all containers support all codecs.

However, because codecs and containers are separate concepts, it could be the case that your media player supports a certain container format but not the codec used for the video in the container, which means you won't be able to play the video!

# Containers Vs Codecs


Image from (http://www.pitivi.org/manual/codecscontainers.html)

# Video Compression

When we looked at lossless compression for images, the basic idea was that compressed images avoided storing redundant data for similar pixels.

This same principle is applied to video compression.

In a movie or television show, it's usually the case that the entire scene isn't changing all at once. Instead, there's usually some subject in the video that's moving on a more static background.

So, if nothing in the background is changing between frames, then we don't need to redundantly store that data!

# Video Compression

Notice the chair on the left is static, it never moves in these frames. We don't need to save those data over and over again.

That means we can actually save something that looks more like this:

# Streaming

**Streaming** gives us the ability to watch a video (or listen to an audio track) as it's being downloaded.

a YouTube video file will be downloaded by the browser in small chunks (think less than 10 seconds of video).

So that the video remains smooth as you're watching it, your browser actually needs to make sure it stays a bit ahead of you as the viewer.

- the browser download ahead into a **buffer**.
- "buffering" related errors are due to slow internet connection.
- for slow connections, the server can choose a lower resolution video.(**adaptive bitrate streaming**)

Common video resolutions are 1920x1080, 1280x720, and 854x480.

- These are common denoted by 1080p, 720p and 480p(height of video).
- Notice that in all three cases, the ratio between width and height is 16:9, which is known as the image's aspect ratio. (4:3 is an older ratio)

# Sequential vs Parallel vs Distributed

**Sequential computing** is a computational model in which operations are performed in order one at a time. A sequential solution takes as long as the sum of all of its steps.

**Parallel computing** is a computational model where the program is broken into multiple smaller sequential computing operations, some of which are performed simultaneously. A parallel computing solution takes as long as its sequential tasks plus the longest of its parallel tasks.

Solutions that use parallel computing can scale more effectively than solutions that use sequential computing. (e.g. in machine learning where you need to work with large amounts of data, parallel computing can speed up the computer's "learning" process.)

# Sequential vs Parallel vs Distributed

**Distributed computing** is a computational model in which multiple devices are used to run a program.

Distributed computing allows problems to be solved that could not be solved on a single computer because of either the processing time or storage needs involved.

Distributed computing allows much larger problems to be solved quicker than they could be solved using a single computer.

# GPU

One component of the video card is the **GPU, or Graphics Processing Unit**, which is essentially a CPU dedicated just to graphics-related computations.

Working with graphics and animation requires many mathematical computations(e.g. matrix operations) that are resource intensive. Parallel computing can lead to more efficient solutions to computing problems.

GPUs take parallelism to a whole new level by running thousands of computations at the same time.

It also helps if these problems are independent of one another, so everything can be solved at the same time with minimal need to combine everything back together. Luckily, both of these conditions are really common in graphics!

- For example, figuring out the color of the pixel in the top-left of your screen doesn't depend on figuring out the color of the pixel in the bottom-right of your screen.

# Homework

1)Read and reread these lecture notes.

2)Watch or Read(Required):

a)MP3 Compression.

https://arstechnica.com/features/2007/10/the-audiofile-understanding-mp3-compression/

b)CPU vs GPU Painting.

https://youtu.be/-P28LKWTzrI

c) Sampling

https://www.youtube.com/watch?v=zC5KFnSUPNo

d) Analog vs Digital.

https://www.youtube.com/watch?v=btgAUdbj85E

# References

This lecture is a summary of a lecture from an OpenCourseWare course below.

Computer Science E-1 at Harvard Extension School

Understanding Computers and the Internet

by Tommy MacWilliam

https://www.youtube.com/watch?v=-R2uBgcw600