

# Sound and Video

 [coursera.org/learn/programming-fundamentals/supplement/RHNQz/sound-and-video](https://coursera.org/learn/programming-fundamentals/supplement/RHNQz/sound-and-video)

---

Sound is another common aspect of computer use that seems like it is not a number. However, sound is naturally a waveform, which can easily be represented as a sequence of numbers. The most direct numeric representation of a sound wave is to record the “height” of the wave at periodic intervals, forming a sequence of numbers. The frequency of these intervals is called the sampling rate (higher sampling rates result in better quality of the sound), and is typically 22 kHz or 44kHz—that is 22,000 or 44,000 samples per second. Stereo sound simply has 2 sequences of numbers—one for the left channel and one for the right channel. As with images, there are many typical formats (*e.g.*, WAV, AIFF, AAC, *etc.*), some of which are compressed (again, the sound is just numbers, so we can do math on it).

Videos (including those found in this course) again seem to defy the “everything is a number” rule—however, by now, you should see the path to numeric encoding. A video is a sequence of images (called “frames”) and the corresponding sound. We have already seen how to encode images and sound as numbers. The simplest approach would be to encode the video as the sequence of images plus the sound. While this approach gives us a bunch of numbers, it would be huge—one minute of a 512 pixel x 256 pixel video at 32 frames per second with a stereo sound track at 44 kHz would require about 725 Megabytes (almost 1 GB). Correspondingly, all common movie formats (*e.g.*, MP4, MOV, *etc.*) apply compression, not only to the images and sound themselves, but also in terms of not storing the entire image for all frames, but rather storing a way to compute the next frame’s image based on the changes from the previous frame. The following video demonstrates some examples of “everything is a number.”