

Back to Bits

Announcements

- Lab 4 due tomorrow at midnight
 - Ninja session tonight

Today

- How do we convert graphics and videos to digital storage format?
- Colors and animation in Zelle graphics library

Bits and Bytes

- Recall that a **bit** is the smallest unit of digital storage, always in one of two states: 0 or 1.
- A **byte** is 8 bits, so one byte can have $2^8 = 256$ possible states.
 - 00110100, 00110101, 00110110, ...

How much storage?

- A boolean: one bit
- Ints and floats: it depends, but with only 4 bytes we can represent all the ints from -2,147,483,647 to 2,147,483,648
- Strings: using ASCII encoding there are 128 possible characters, so a string needs 7 bits per character.

Graphics

- A digital image is a grid of **pixels**. Each pixel is a tiny square filled in with a particular color.
- Digital colors are defined by how red, green, and blue they are, each on a scale from 0 to 255.
- So one pixel needs 3 bytes. And a 6 megapixel photo (a 2000 x 3000 grid of pixels) needs 18 MBs, or roughly 18 million bytes.

Video

- Digital video is a rapidly changing series of images, something like 60 **frames** (or images) per second.
- For 1080p HD video, it's 60 2.1-megapixel images per second. One second of video thus requires:
 - $60 * 2.1M * 3B = 378 \text{ MB}$.
- A minute of footage would be $60 * 378 \text{ MB} = 22 \text{ GB}$! But that's more than the memory on many smartphones.
- In fact **compression** takes advantage of **redundancy** brings this number way down. *The Martian* in HD is less than 6GB (including audio).

Colors in Zelle graphics

- In the **GraphWin** constructor, we specify the width and height of our grid of pixels.
- Use the **color_rgb(red, green, blue)** function to generate any color you want.
- Or use one of the pre-defined colors. To explore these colors, use the **colorPicker** as described on reference page.

Animation

- Use the **sleep(seconds)** function from the **time** library to do animation in Python.
- The **sleep** function will pause your program for the specified number of seconds (usually a fraction of a second). After that short pause you can move graphics objects by some small distance.
- If the interval and distance are small enough, these discrete steps create the illusion of smooth motion.

Examples