

Lecture 2:

HOW COMPUTERS WORK

CSC111: Introduction to CS through Programming

R. Jordan Crouser

Assistant Professor of Computer Science

Smith College

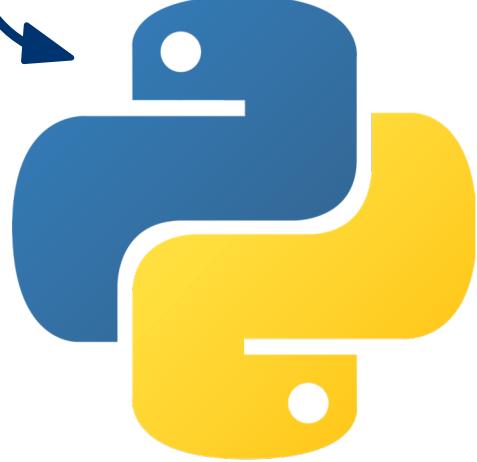


Announcements

- TA hours start **tonight!**
- Weeknights 7:30 to 9:30 in Ford 241
 - Sunday: Jenny, **Georgina**, Anastasia
 - Monday: Jenny
 - Tuesday & Wednesday: **Georgina**, Prayasha
 - Thursday: **Georgina**, Anastasia
- Sunday afternoons 1:30-3:30
 - **Grace**, Jenny, Anastasia

Checking in...

python
installed?

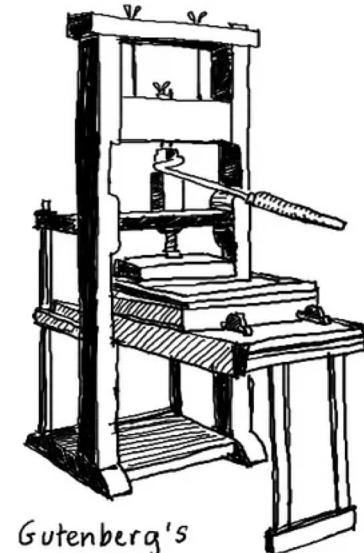
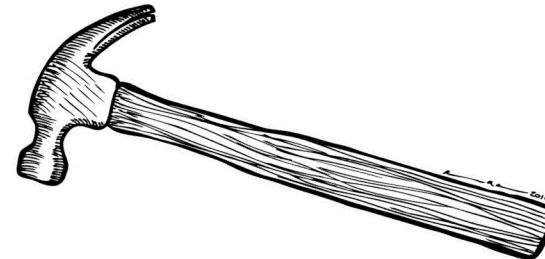
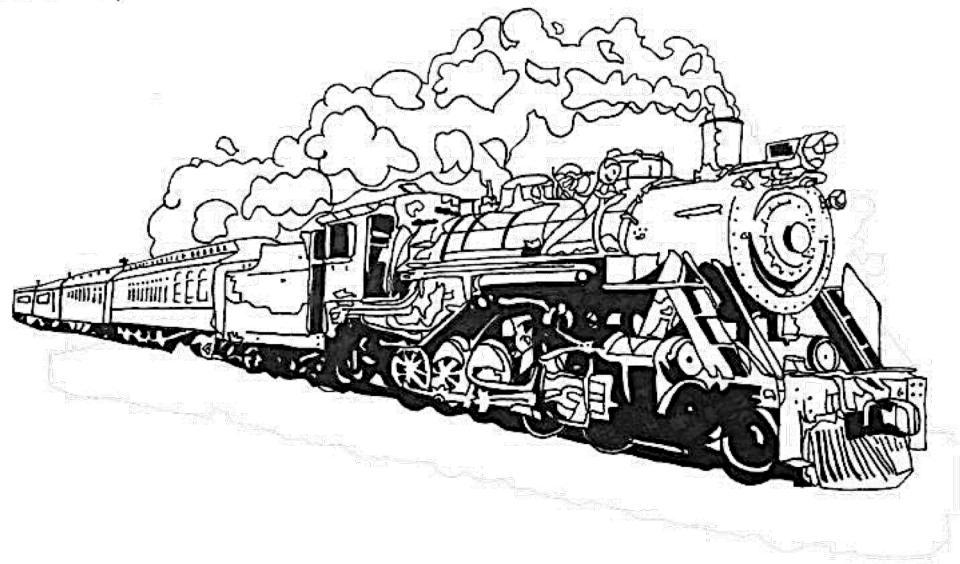
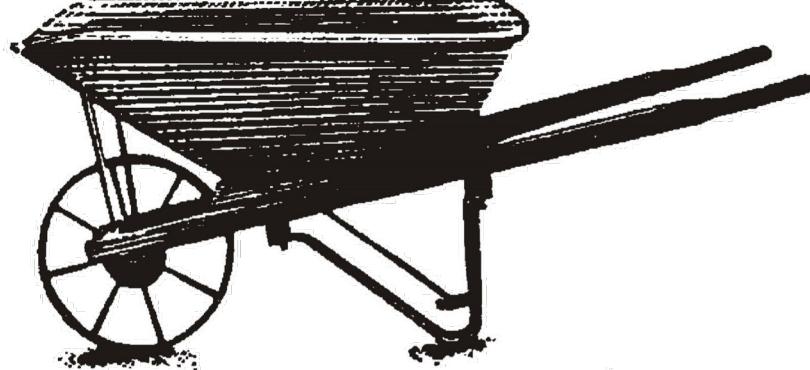


joined
slack?

Overview of the week

- Crash course in computers
 - A little history
 - 4 key components
 - Quick hardware demo
 - Boolean logic
- Introduction to Python
- Life skill #1: pair programming

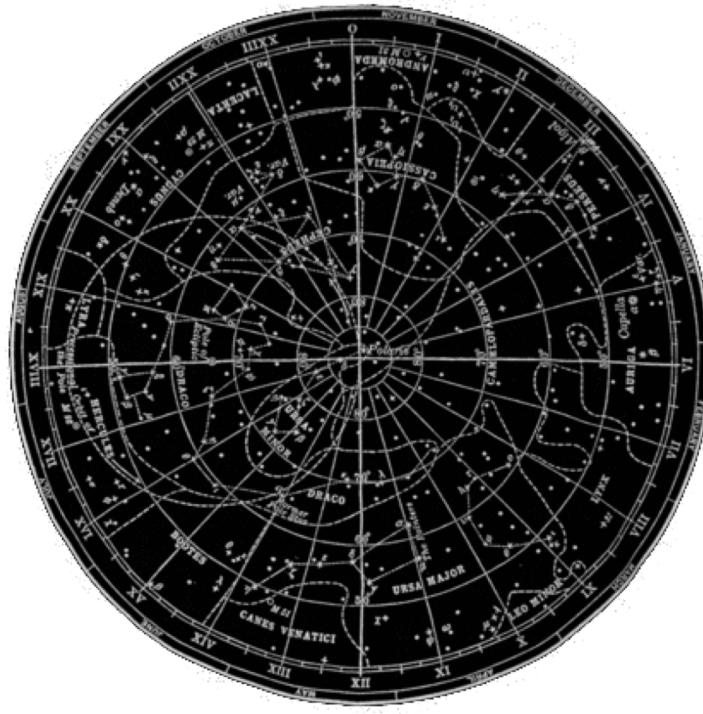
A little history...



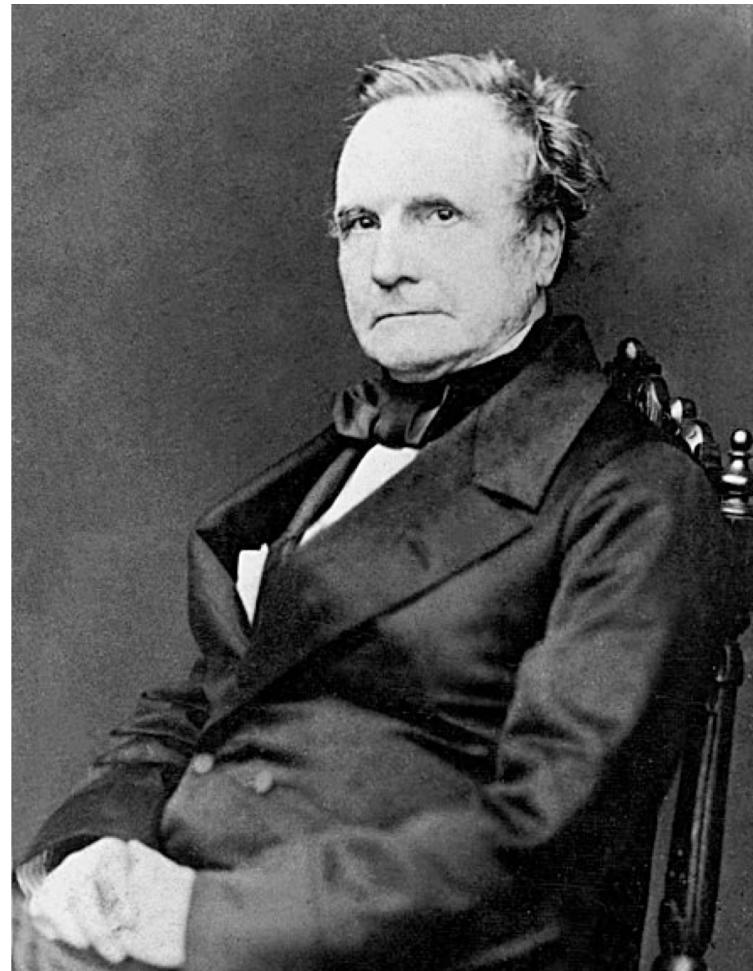
Gutenberg's
Press

A little history...

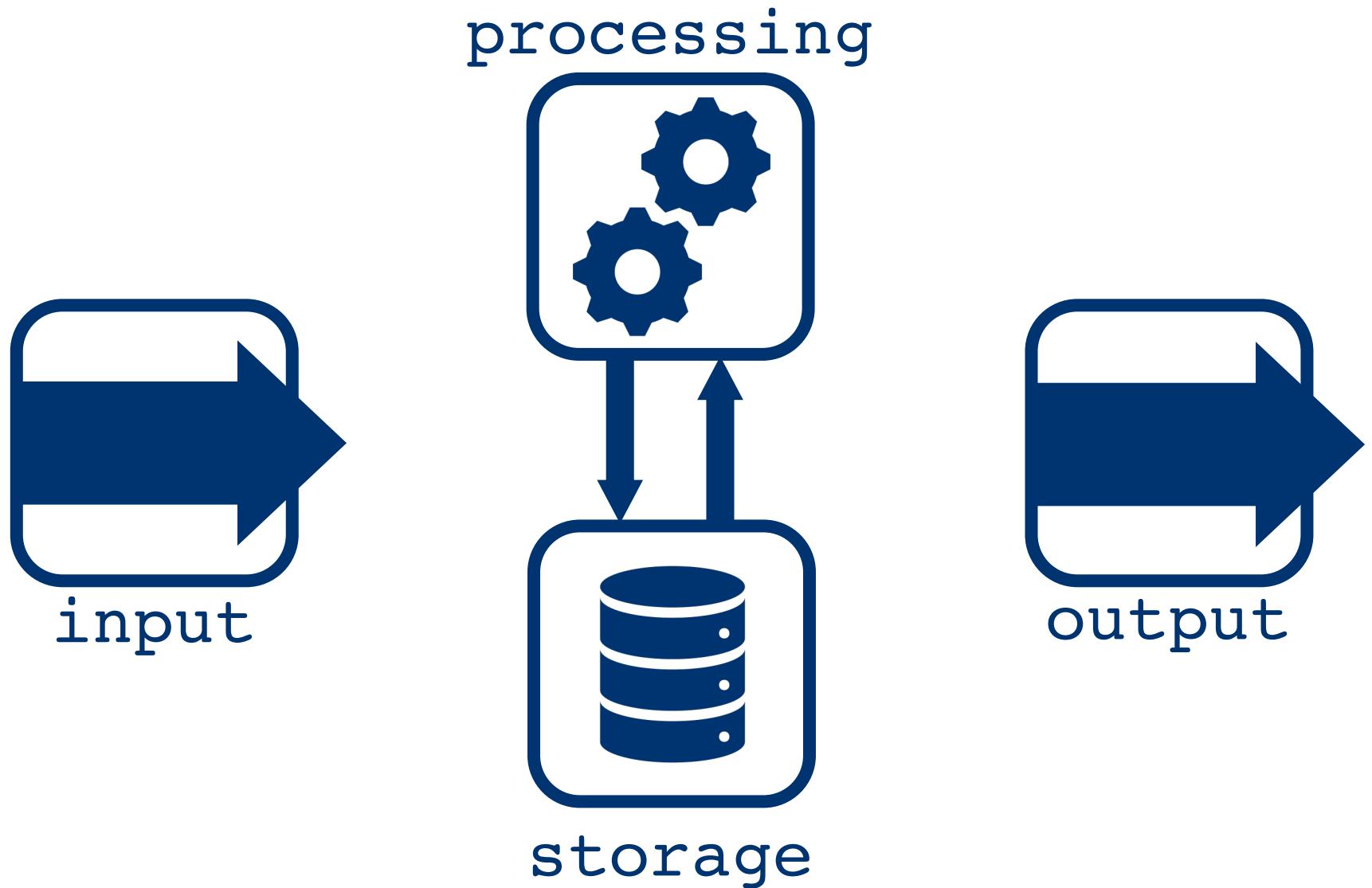
$$(x + 3)^2 = 4$$



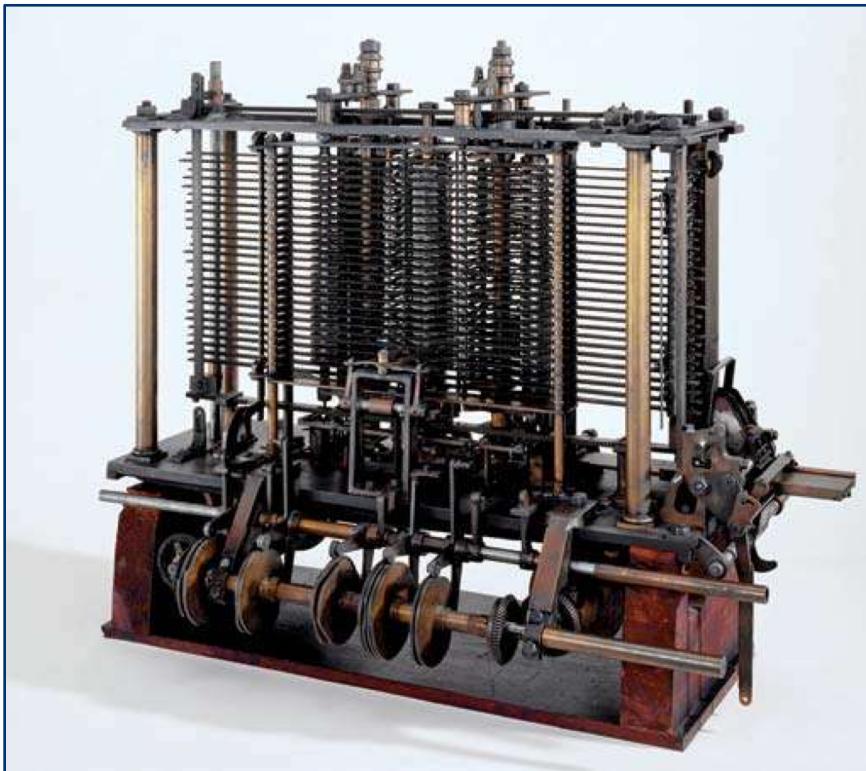
A little history...



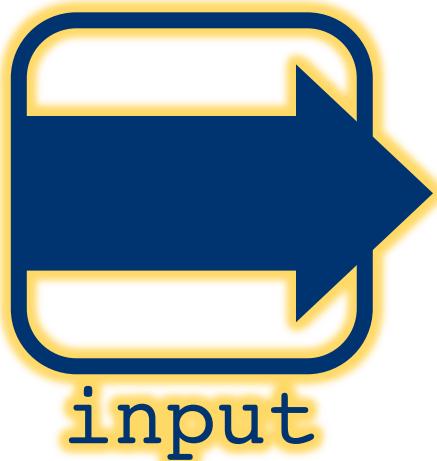
4 basic tasks



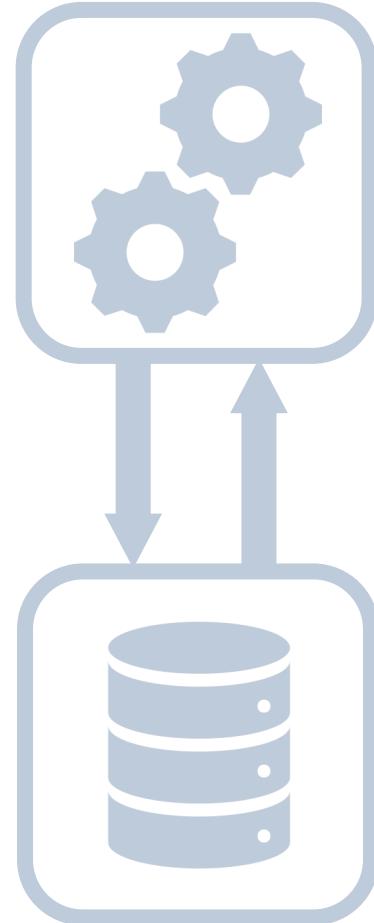
Then and now



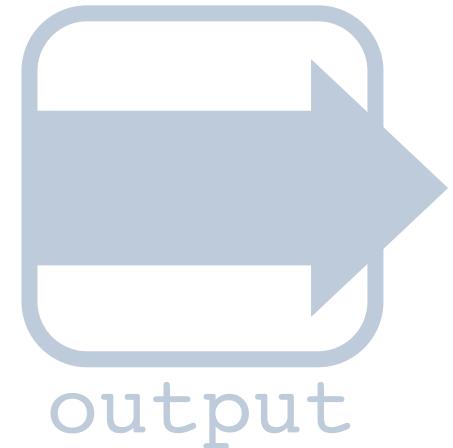
Input



processing



storage



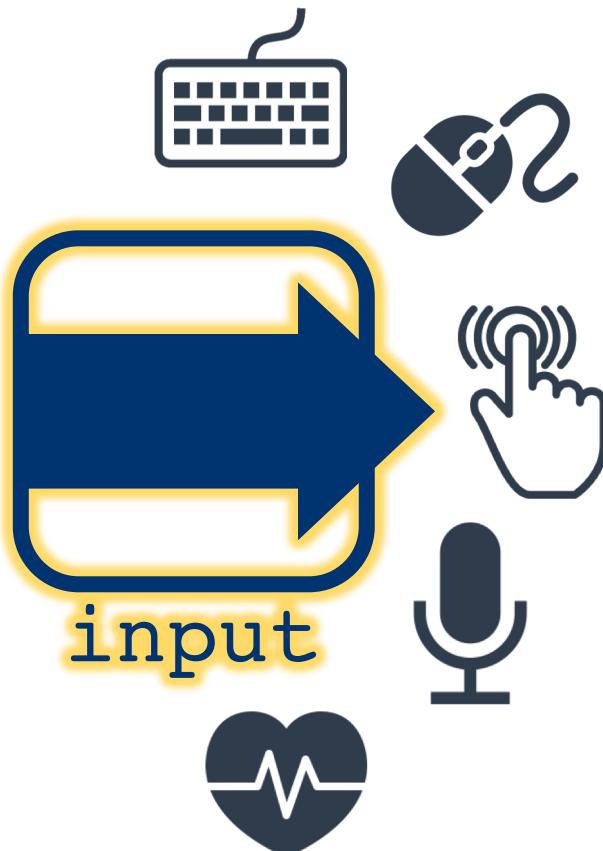
output

Input

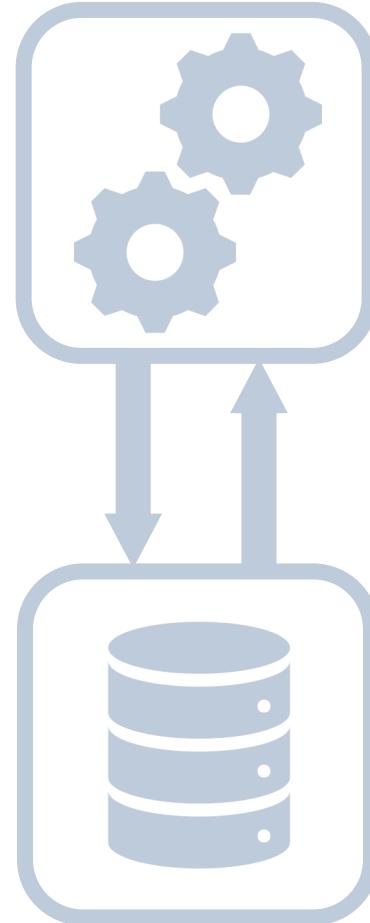
What are some ways that you
get information **into** a computer?



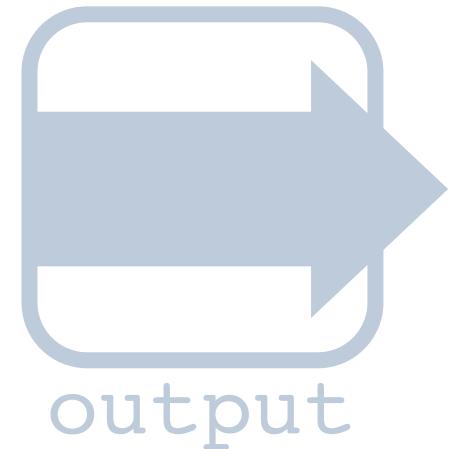
Input



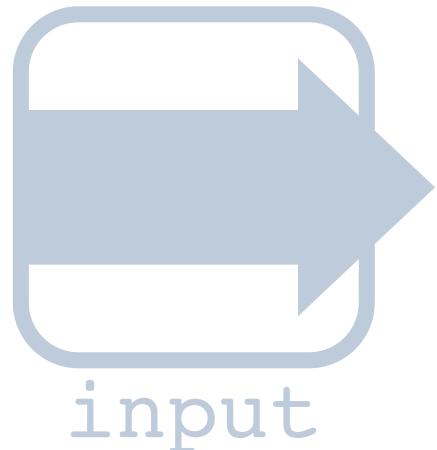
processing



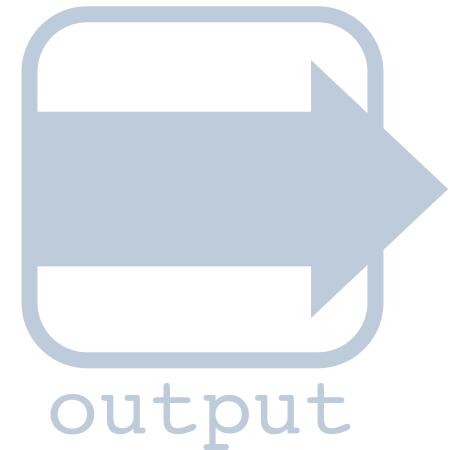
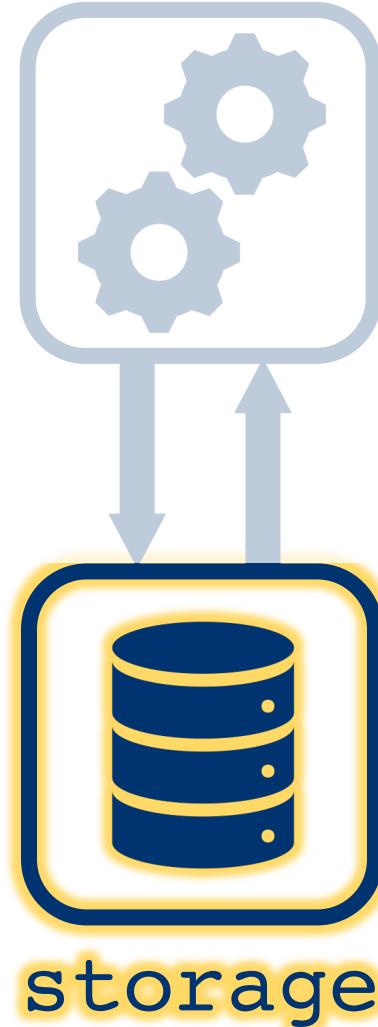
storage



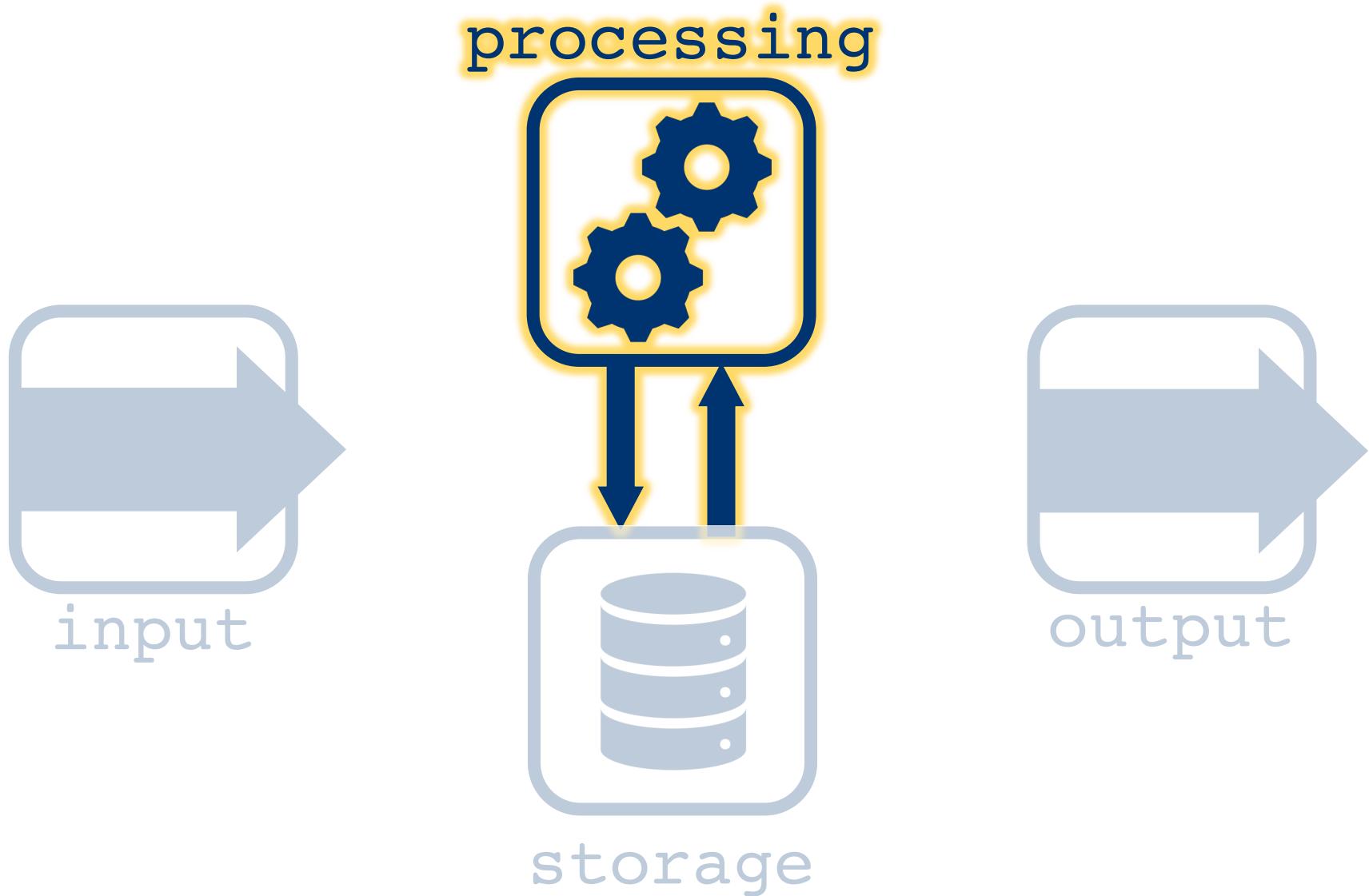
Storage



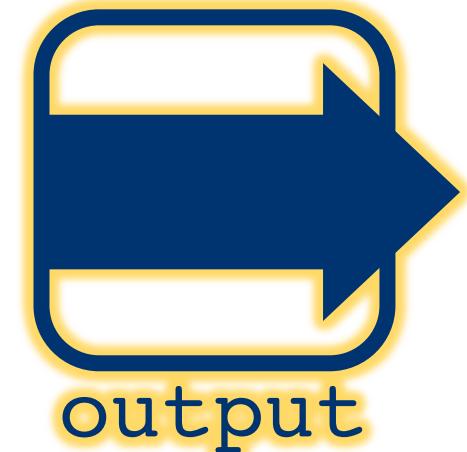
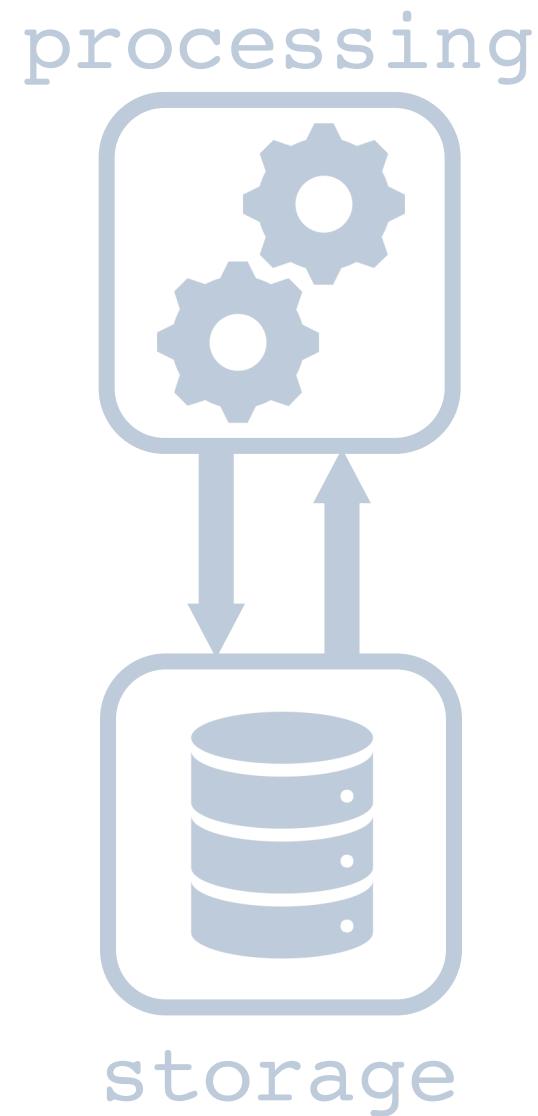
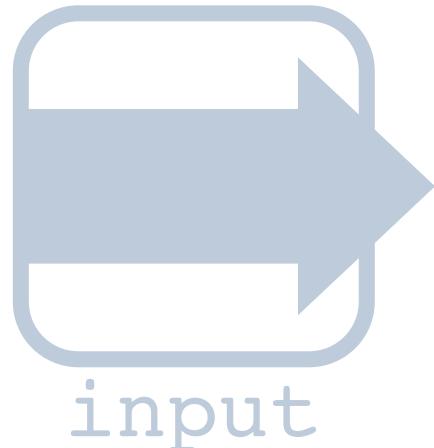
processing



Processing



Output

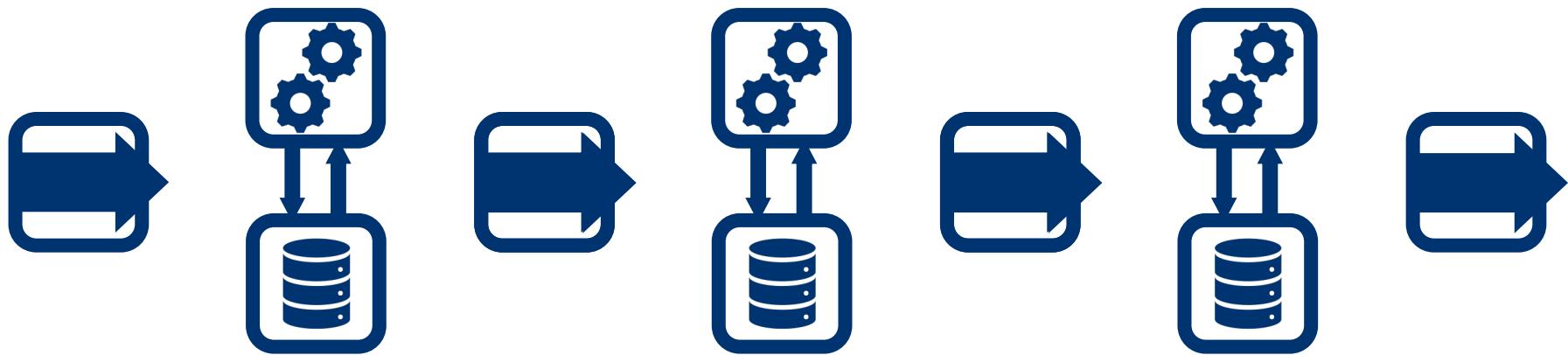


Output

What are some ways that you
get information **out** of a computer?



Networks



D E M O

T Y M E

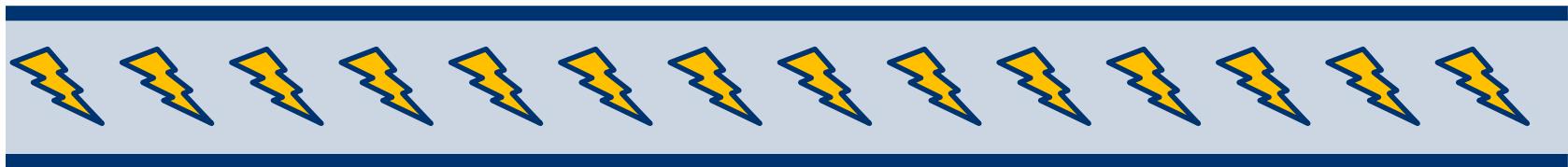
Discussion

How is information represented
using **electricity?**



One wire: a “bit”

“off”



“on”

Multiple bits

- 1 bit:



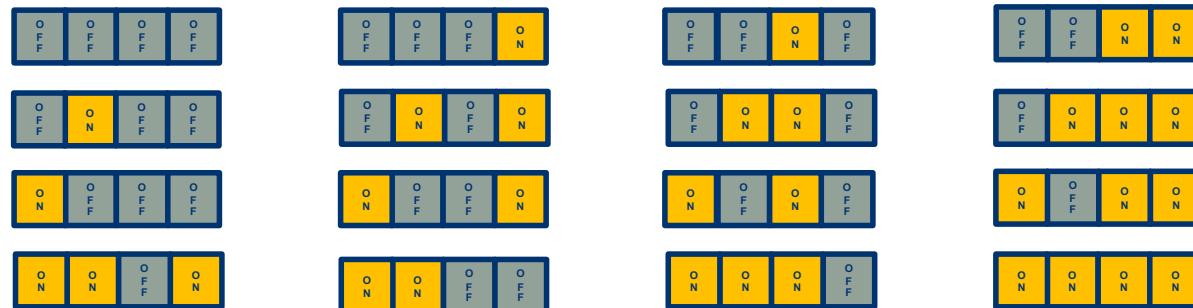
- 2 bits:



- 3 bits:



- 4 bits:



Using bits to represent numbers

- In **base-10**, each place represents a power of 10, and each digit can take on a value from 0 to 9:

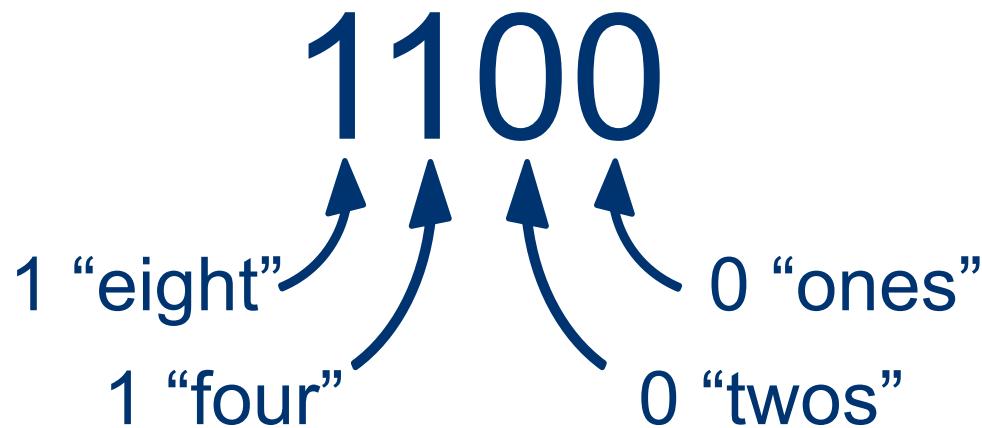
12

1 “ten” 2 “ones”

$$(1 * 10) + (2 * 1) = 12$$

Using bits to represent numbers

- In **base-2 (“binary”)**, each place represents a power of 2, and each digit can take on a value of either 0 or 1:

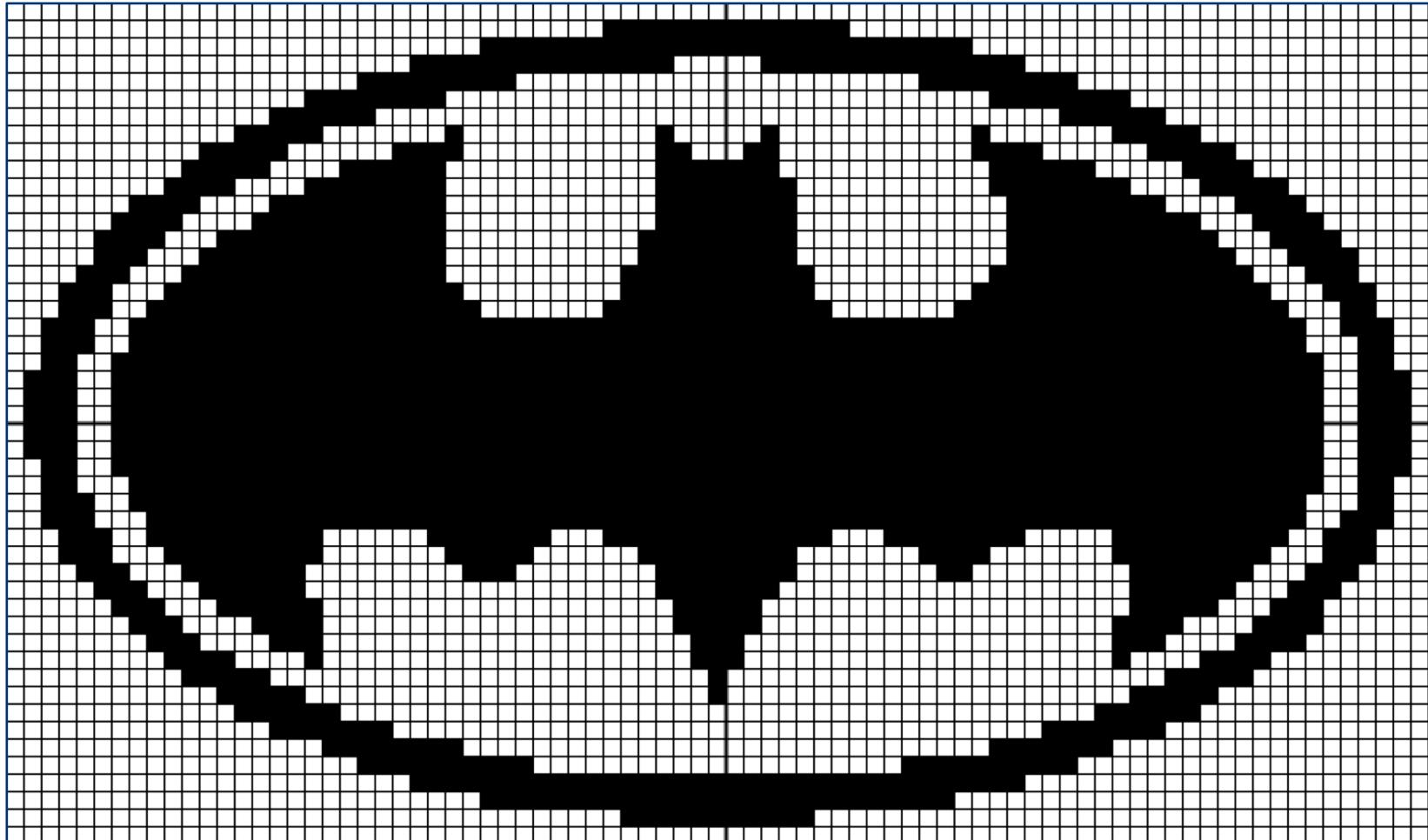


$$(1 * 8) + (1 * 4) + (0 * 2) + (0 * 1) = 12$$

How much can we represent?

- With 8 bits*, we can represent the numbers 0 to 255
*8 bits is called a “byte”
- With 32 bits, we can represent numbers **> 4 billion**
- With 266 bits, we can represent **more unique numbers** than there are believed to be **atoms in the universe**

Binary images?



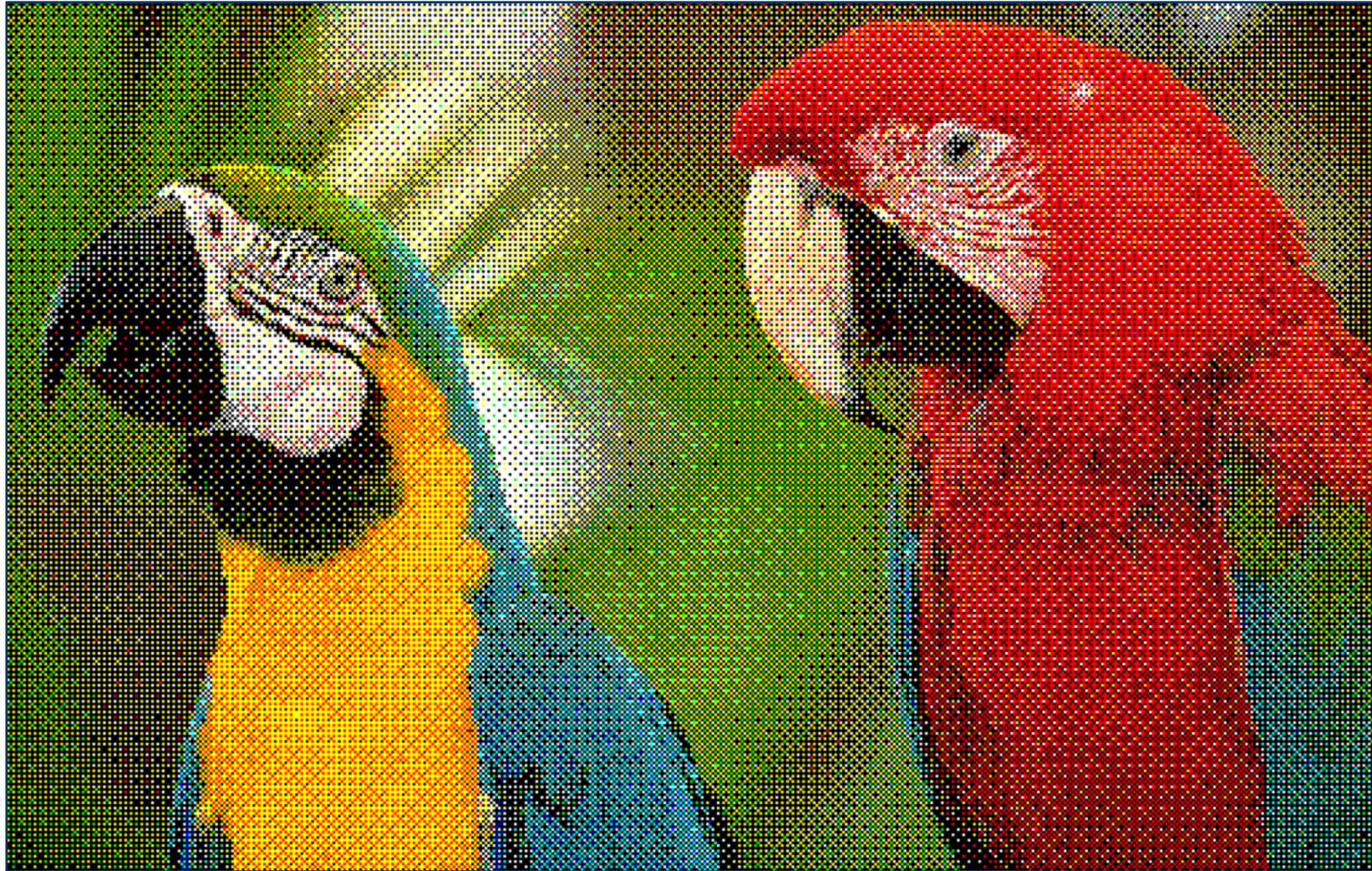
[0 , 1]

Greyscale images?



[0 : 255]

What about color?



([0:255], [0:255], [0:255])

The tradeoff

- iPhone camera (8 megapixels): 3296×2472 pixels
 - each requires 4 bytes to represent RGB + opacity
 - 32,590,848 bits \approx 4MB (1MB = 1024 bytes, 1 byte = 8 bits)
- HD video: 1920 x 1080 pixels, 30fps
 - 5 minutes of video = 300 seconds = 9000 frames
 - 2,073,600 bits per frame \approx 2.33GB (1GB = 1024MB)

The good news

- “High level” programming languages like Python mean we don’t have to write in “low level” binary
- Instead, we write statements like:

```
print("hello")
```

Overview of the week

- ✓ Crash course in computers

- ✓ A little history

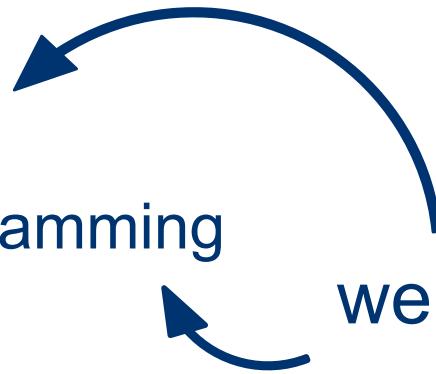
- ✓ 4 key components

- ✓ Quick hardware demo

- ✓ Boolean logic

- Introduction to Python

- Life skill #1: pair programming



we'll do a bit of both
on Wednesday to
set you up for lab

Homework #1: how does it **do** that?



Homework #1: how does it **do** that?

- Over the next few days: **track your computation**
(laptop, tablet, smartphone, ATM, distance tracker on the treadmill,...)
- **Pay attention** to what you're doing, and to what's happening in response
- Find one example where you **don't know how it works**
- In a ~1 page reflection:
 - describe **what you were doing**
 - describe the part you **don't really understand**
 - give your best explanation for **what you think is going on**
- Submit your **PDF** on Moodle by **11:55pm on Sunday**