

Introduction

Goals for ULAB CS curriculum

- Basics of Python language
- Familiar with the terminal
- Python Packages : numpy and matplotlib

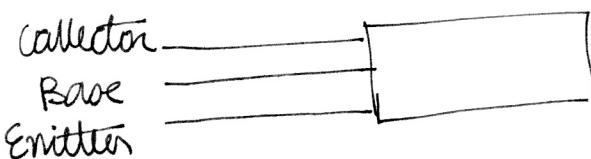
These lectures alone will not be enough  $\Rightarrow$  practice

But, you have help in ULAB.

- staff and mentors
- Questions channel on slack

Q: How does a computer compute?

① Transistor : semiconductor device with outputs that reacts to inputs



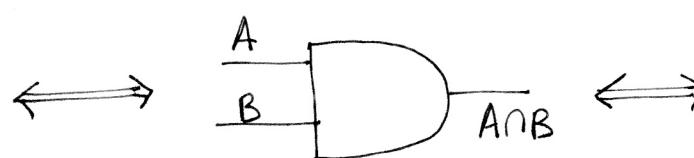
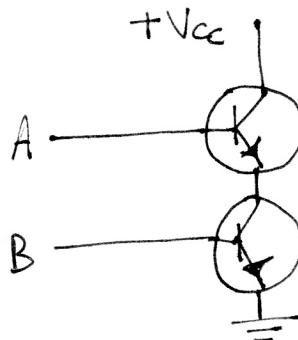
Base HIGH:



Base LOW:

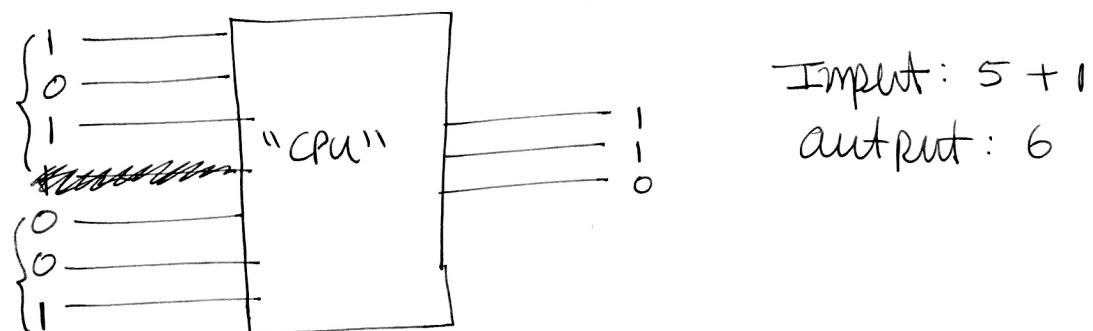


② logic gate : collection of transistors that can compute logical operations. EX: ~~AND~~ AND Gate



A	B	out
0	0	0
0	1	0
1	0	0
1	1	1

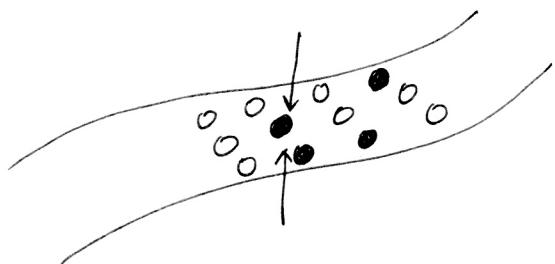
③ CPU: A collection of interconnected logic gates that can perform complex operations. Simplified example,



Are we done? Can we go home?

No: this is painful. Is there a better way to input and output data?

Past: Punch cards formed physical connections



But this is still painful!

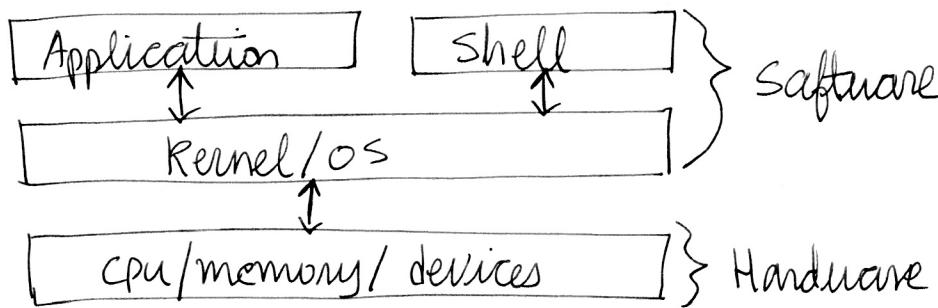
New: operating systems with

- text-based command lines
- graphical user interfaces

## The Full computing stack

just

The modern computer is much more complex than just a CPU



Hardware: we've already discussed hardware!

~~Assembly~~: How do we tell the computer what to do?  
The assembly language is a low-level

Machine code: low level of programming language, written in binary, that a computer can execute directly. E.g. .exe files on Windows.

Kernel: The kernel is the core program that ~~contains~~ ~~all of the functions of the computer~~ interfaces between the software and the hardware. It is the heart of the operating system.

Booting: on startup:

Press power button → BIOS/UEFI is loaded. This program performs POST and starts the bootloader

→ bootloader starts the kernel

→ kernel configures all hardware and starts os

## Interact with the Kernel | 2 ways:

- shell / command line interface
- graphical user interface (GUI)

### The GUI

- what most people think of ~~tell~~ as the OS
- Easy to use
- limited because there ~~is no scripting~~ can only be so many GUI functions (can't program the GUI, you just have to use it). e.g. I have a folder with 1,000 PDF files and 1000 JPG files. I want to move all PDF files into another folder.
- Not practical for remote access.

### The Shell

A text-based interface to interact with the Kernel.

- greater control over OS
- shell scripting

How does the shell work? You type a text command. The kernel executes the command and displays any output as text. (What's actually happening: each command causes the corresponding executable program to be run).

Windows: Default is DOS [require students to download gitbash before hand]

Mac / Linux: Bash (Bourne - again shell)

\* Students should start shell now

## Demo Bash commands

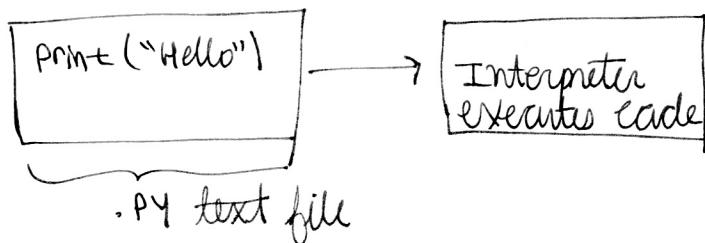
- pwd
  - demo moving all pdf files to a new folder  
(pg. 4)
  - ls
  - cd
- ~~mkdir~~  
~~mv~~
- \* students will practice in HW
  - \* students may not know what directory means.

## Python

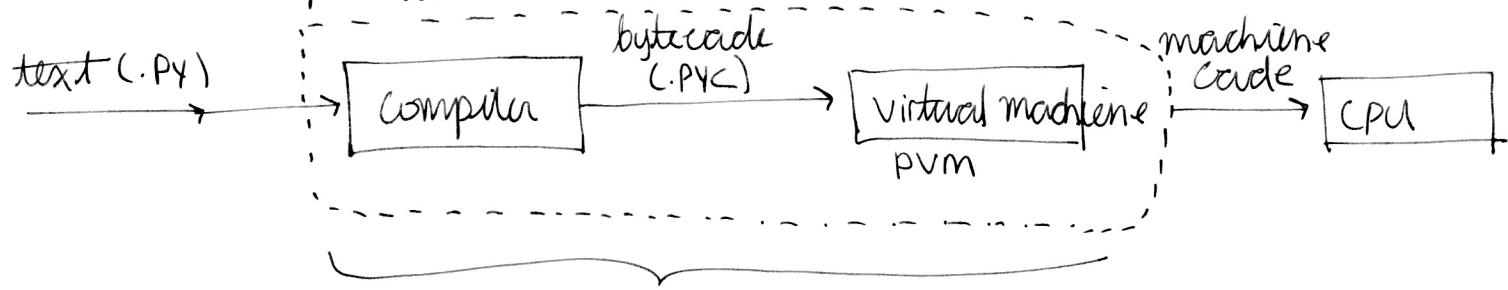
High-level programming language with human-readable syntax:

- very popular in research (esp data analysis)

### How python works:



More detailed picture



Interpreter  
(when you download python, this is what you're downloading)

Demo: creating first program and run from command line

Pre-lecture: ~~download python, gitbash (Windows)~~  
Bash ~~Macosx~~

