

Introduction to Computer and Programming

Lecture 1

Yue Zhang
Westlake University

August 1, 2023

Chapter 1.

The Big Picture

- 1 What Is a Computer?
- 2 The Course
- 3 Python Setup

- 1 What Is a Computer?
- 2 The Course
- 3 Python Setup



Figure: Desktop Computer



Figure: Laptop



Figure: Mobile phone



Figure: Tablet



Figure: Smartcar



Figure: Smartcar



Figure: Supercomputer - Dawning

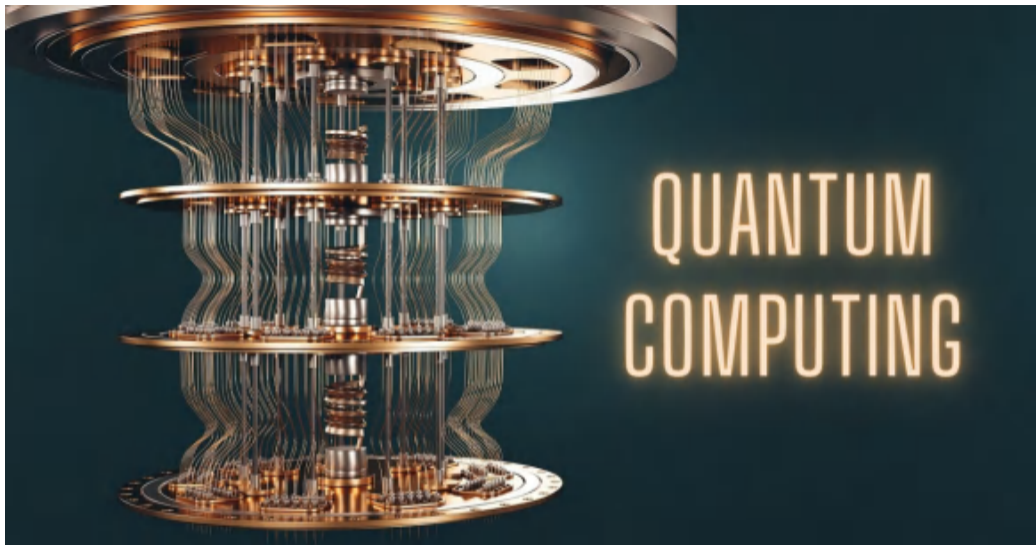


Figure: Quantum Computer

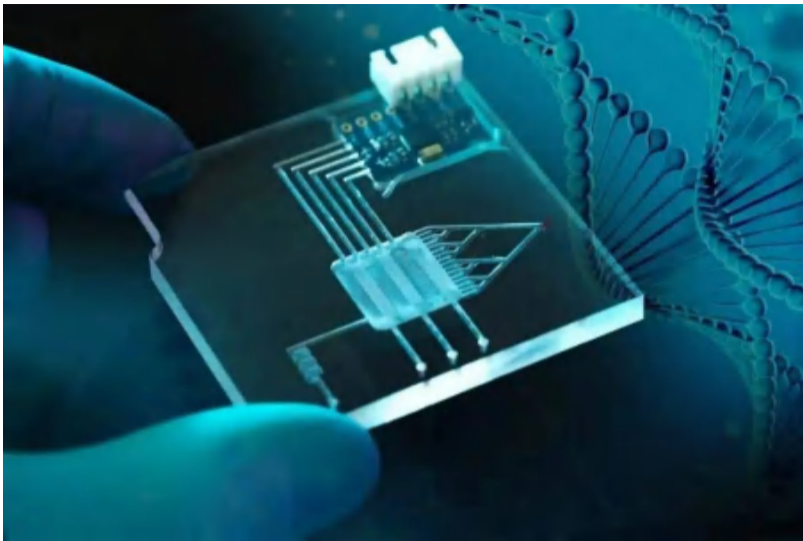
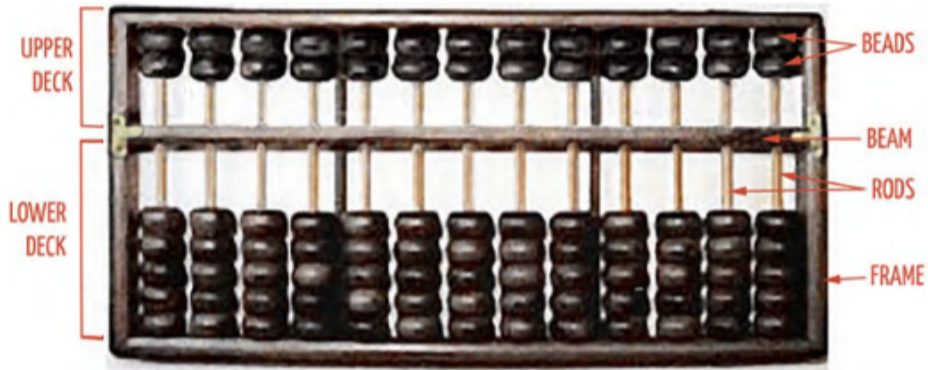


Figure: Biological Computer

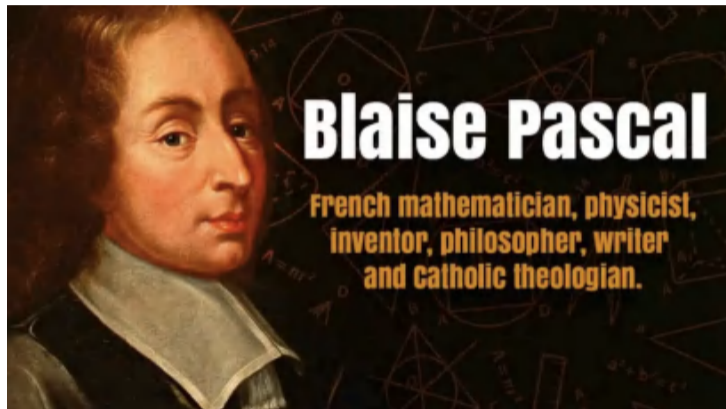
A computer is a programmable computing system,
which has two essential components:

- Hardware
- Software

History of Computing – Early History



- Abacus



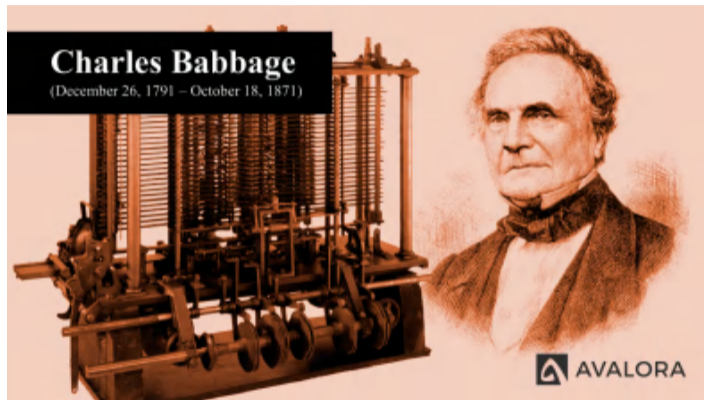
- Blaise Pascal (19 June 1623 – 19 August 1662)
Device to Add, Subtract, Multiply and Divide


History of Computing – Early History



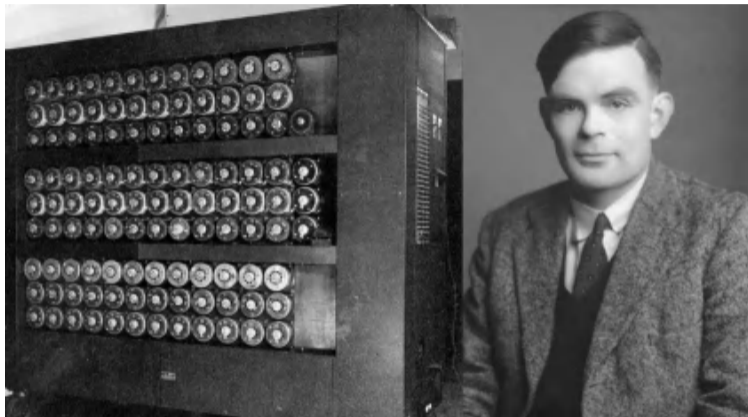
- Ada Lovelace (10 December 1815 – 27 November 1852)
The First Programmer

History of Computing – Early History



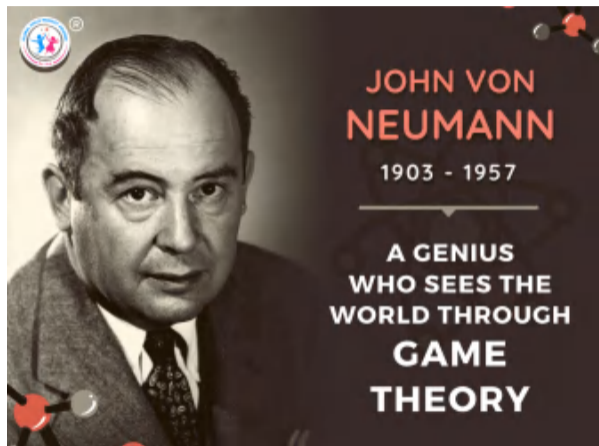
- Charles Babbage (26 December 1791 – 18 October 1871) Analytical Engine 

History of Computing – Early History



- Alan Turing (23 June 1912 – 7 June 1954)
The Turing Machine

History of Computing – Early History



- John von Neumann
(December 28, 1903 – February 8, 1957)

History of Computing – The Earliest Modern Computers

Harvard Mark I (February 1944)



ENIAC (February 1946)



UNIVAC (March 1953)



- Used for Scientific Computing

History of Computing – Early Computers (1951-1959)

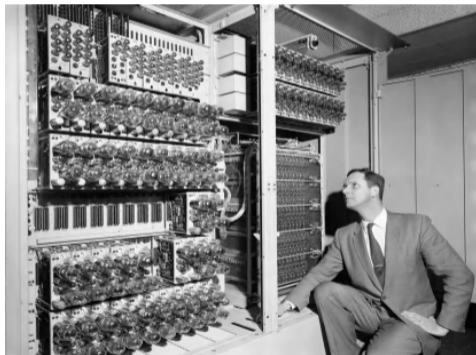
Vacuum Tube



Input and Output Device

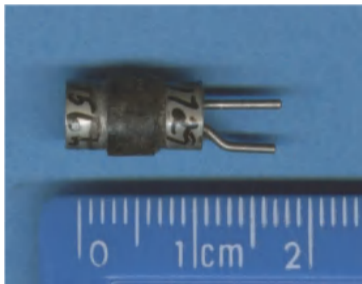


Huge Machine VS. Human



History of Computing – Early Computers (1959-1965)

Transistor



Disk



Machine

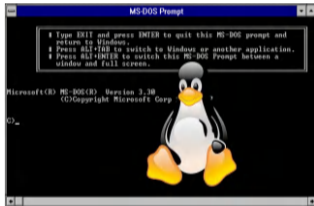


History of Computing – Early Computers (1965-1990s)

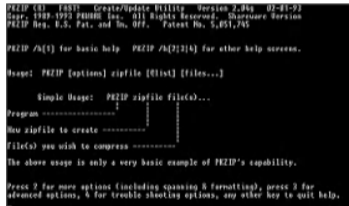
Keyboard and Mouse



DOS/Linux



Text Interface



- From Scientific Computing to General Computing

History of Computing – Modern Computers

Windows



MacOS



IOS



WWW



- Ubiquitous Computing
- Embedded Computing
- Cloud Computing

History of Computing – Early Programming (1951-1959)



- Machine Language (Punched Tape)

History of Computing – Early Programming (1951-1959)

Assembly Language

Machine Language

```
mov ecx, ebx
mov esp, edx
mov edx, r9d
mov rax, rdx
```

Assembler + Linker

```
100101011001
010011111011
111010101101
01010101010
```

Programmer

Processor

- Assembly Language

History of Computing – Early Programming (1959-1965)

Fortran

Example code - FORTRAN IV or 66

```
THE TPK ALGORITHM
FORTRAN IV STYLE
DIMENSION A(11)
FUN(T) = SQRT (ABS(T) + 5.)*T**3
READ (5,1) A
FORMAT(5F10.2)
DO 10 J = 1, 11
  I = 11 - J
  Y = FUN(A(I+1))
  IF (400.0-Y) 4, 8, 8
    WRITE (6,5) I
    FORMAT(I10, 10H TOO LARGE)
  GO TO 10
  WRITE (6,9) I, Y
  FORMAT(I10, F12.6)
CONTINUE
STOP
END
```

C

My first program in C

```
#include <stdio.h>
int main()
{
    printf("Hello World!");
    return 0;
}
```

Lisp

Example of Lisp System Session, 3

```
(defun append (x y)
  (if (equal x nil) y
      (cons (car x) (append (cdr x) y) ) ) )

(defun reverse (x)
  (if (equal x nil) nil
      (append (reverse (cdr x))
              (list (car x) ) ) ) )

(defun reverse (x) (reverse2 x nil))
(defun reverse2 (x y)
  (if (equal x nil) y
      (reverse2 (cdr x) (cons (car x) y) ) ) )
```

- High-level Language

History of Computing – Early Programming (1971-1989)

Pascal

```
ELSE IF number < 0 THEN BEGIN
    writeln('*** ERROR: number < 0');
END
ELSE BEGIN
    sqrt := sqrt(number);
    writeln(number:12, sqrt:12:6);
    writeln;

    root := 1;
    REPEAT
        root := (number/root + root)/2;
        writeln(root:24:6,
            100*abs(root - sqrt)/sqrt:12:2,
            '%');
    UNTIL abs(number/sqr(root) - 1) < epsilon;
    END
UNTIL number = 0
END.
```

C++

```
#include <yarp/os/Network.h>
#include <yarp/os/Port.h>
#include <yarp/os/Bottle.h>
#include <yarp/os/Time.h>
#include <stdio.h>
using namespace yarp::os;
int main() {
    Network yarp;
    Port output;
    output.open("/sender");
    int top = 100;
    for (int i=1; i<=top; i++) {
        // prepare a message
        Bottle bot;
        bot.addString("testing");
        bot.addInt(i);
        bot.addString("of");
        bot.addInt(top);
        // send the message
        output.write(bot);
        printf("Sent message: %s\n", bot.toString().c_str());
        // wait a while
        Time::delay(1);
    }
    output.close();
    return 0;
}
```

- Structured Language

History of Computing – Modern Programming

Java

```
J AverageFinder.java x
1 package average.finder;
2
3 public class AverageFinder {
4     public static void main(String[] args) {
5         System.out.println("Average finder v0.1");
6         double avg = findAverage(args);
7         System.out.println("The average is " + avg);
8     }
9
10    1 usage
11    private static double findAverage(String[] input) {
12        double result = 0;
13        for (String s : input) {
14            result += Integer.parseInt(s);
15        }
16        return result;
17    }
```

C#

```
using System;
// namespace declaration
namespace HelloWorldApp {
    // Class declaration
    class sample {
        // Main Method
        static void Main(string[] args) {
            // statement
            // printing Hello World!
            Console.WriteLine("Hello World!");
            Console.ReadKey();
        }
    }
}
```

- Object Oriented Program
- Virtual Machine

History of Computing – Modern Programming

Perl

```
#!/usr/bin/perl
use strict;

#converts 24 hr argument to 12 hr value - returns the hr and AM or PM
sub convertToThe12hr {
    local $_ = $_[0];

    if (($_ < 0) || ($_ > 23))
    {
        return ("$_ is not a valid hour!");
    }

    return ($_, "AM") if $_ < 12;
    return ($_, "PM") if $_ == 12;
    return ($_ - 12, "PM");
}
```

Python

```
user_input = input("What is your name? ")

if user_input == "Python":
    print("Welcome to IDLE!")
else:
    print("Welcome to Python!")

print("This statement is an unsaved change!")
```

JavaScript

```
1
2 <html>
3 <head>
4 <script>
5     function HelloWorld() {
6         alert("Hello World");
7     }
8 </script>
9 </head>
10 <body onload="HelloWorld()">
11 |
12 </body>
13 </html>
```

Ruby

```
# Example comments
class FieldFormatter

  attr_reader :field_pattern

  def initialize
    @field_pattern = /\A(?:\w+)?\z/

    # Examples of syntax coloring
    some_string = "Hello.\nThis is a test string. #(Time.now) is the time.\n"
    i = 100
    i += 50 unless i < 10
  end

  def format_field(unformatted_field)
    unformatted_field.gsub!(/(?:\A|(?!\A)\z)/, '\1\2')
  end
end

field_formatter = FieldFormatter.new()
```

PHP

```
1 <?php
2 $str_isset = "";
3 $bol_isset = isset($str_isset);
4
5 If ($bol_isset){
6     echo "The variable is set";
7 }
8 else {
9     echo "The variable is not set";
10 }
11 ?>
12
```

Bash

```
#!/bin/bash

clear

echo "The script starts now."
echo "Hi, USER!"

echo "I will now fetch you a list of connected users:"
echo
echo

echo "I'm setting two variables now."
COLOUR="black"
VALUE="1"
echo "This is a string: $COLOUR"
echo "And this is a number: $VALUE"
echo

echo "I'm giving you back your prompt now."
echo
```

• Scripting Languages

History of Computing – Modern Programming

Default (GPT-3.5)

请编程实现二分查找算法。

当然可以！以下是一个使用Python编写的标准二分查找算法示例：

```
python
def binary_search(arr, target):
    left, right = 0, len(arr) - 1

    while left <= right:
        mid = left + (right - left) // 2

        if arr[mid] == target:
            return mid
        elif arr[mid] < target:
            left = mid + 1
        else:
            right = mid - 1

    return -1 # 如果目标元素不在数组中, 返回 -1

# 示例用法
sorted_array = [1, 3, 5, 7, 9, 11, 13, 15, 17]
```

Send a message

ChatGPT may produce inaccurate information about people, places, or facts. [ChatGPT August 3 Version](#)

GPT-4

请编程实现一个冒泡排序。

冒泡排序是一种简单的排序算法。它重复地遍历要排序的数列，一次比较两个元素，如果它们的顺序错误就把它们交换过来。遍历数列的工作是重复地进行直到没有再需要交换，也就是说该数列已经排序完成。

以下是一个简单的冒泡排序的 Python 实现：

```
python
def bubble_sort(arr):
    n = len(arr)
    # 遍历所有数组元素
    for i in range(n):
        # 最后一个元素已经有序, 不需要再进行比较
        for j in range(0, n-i-1):
            # 遍历从0到n-i-1, 交换如果发现元素大于它后面的元素
            if arr[j] > arr[j+1]:
                arr[j], arr[j+1] = arr[j+1], arr[j]

    return arr

# 示例
arr = [64, 34, 25, 12, 22, 16, 00]
```

Send a message

ChatGPT may produce inaccurate information about people, places, or facts. [ChatGPT August 3 Version](#)

- Natural Language Processing (ChatGPT)

The Human Computer Interface

Abacus



Punched Tape



Monitor, Command Line and Keyboard



Touchscreen and Voice



Scientific Computing
for only computer engineers

General Applications
for everyone

- 1 What Is a Computer?
- 2 The Course
- 3 Python Setup

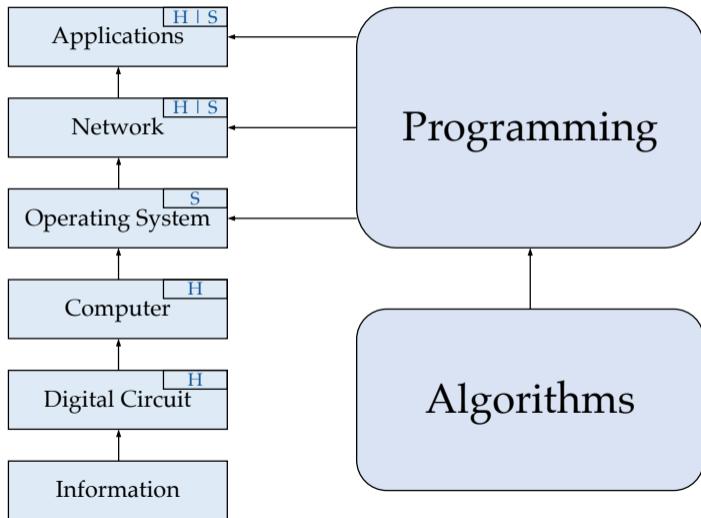
Why do you need to learn computing?

- Thinking
- Designing
- Career
- Leisure

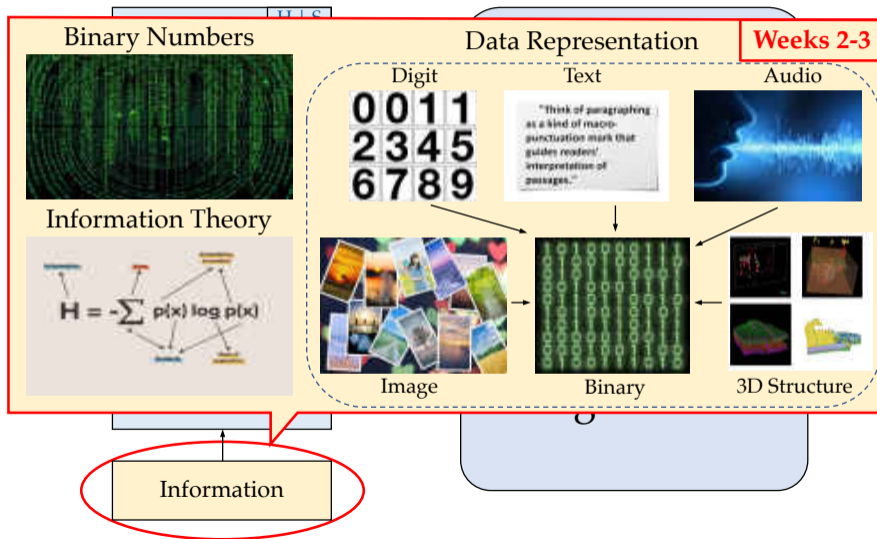
Two Important Components

- Hardware
 - The physical elements of a computing system (printer, circuit boards, wires, keyboard, etc.)
- Software
 - The programs that provide the instructions for a computer to execute

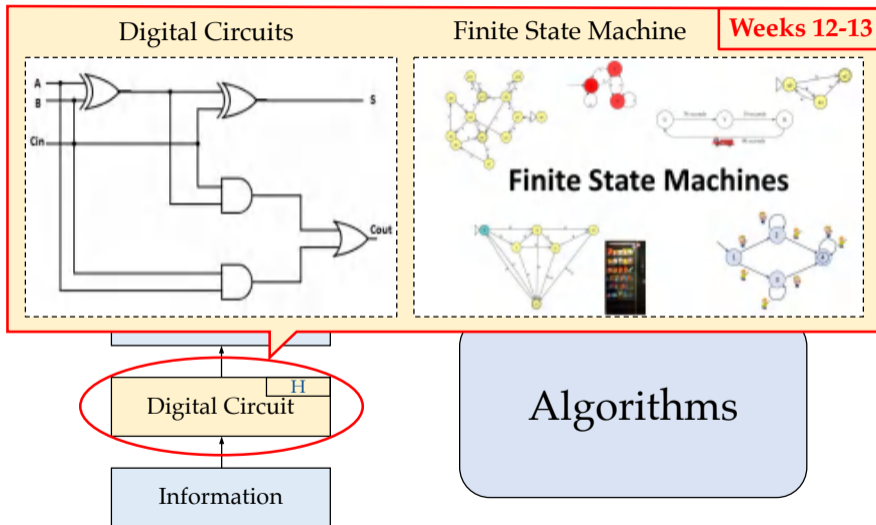
Layers of a Computing System



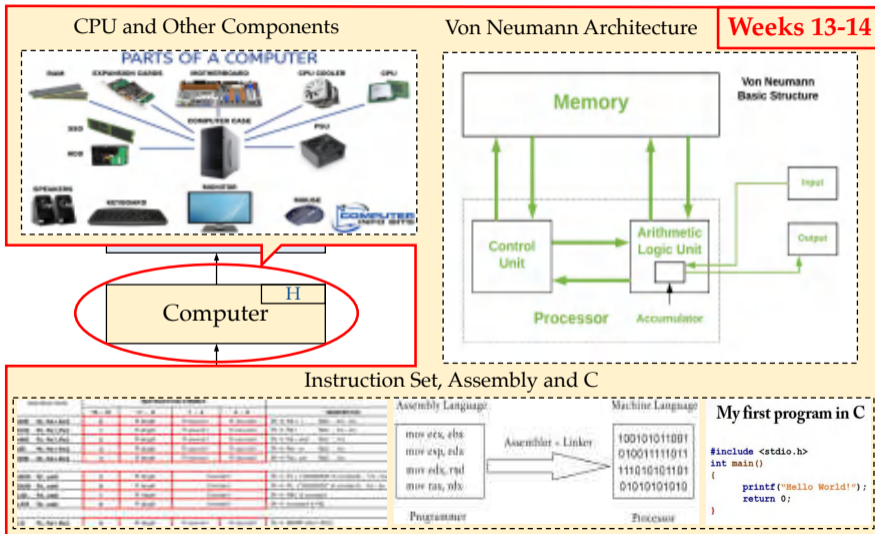
Layers of a Computing System



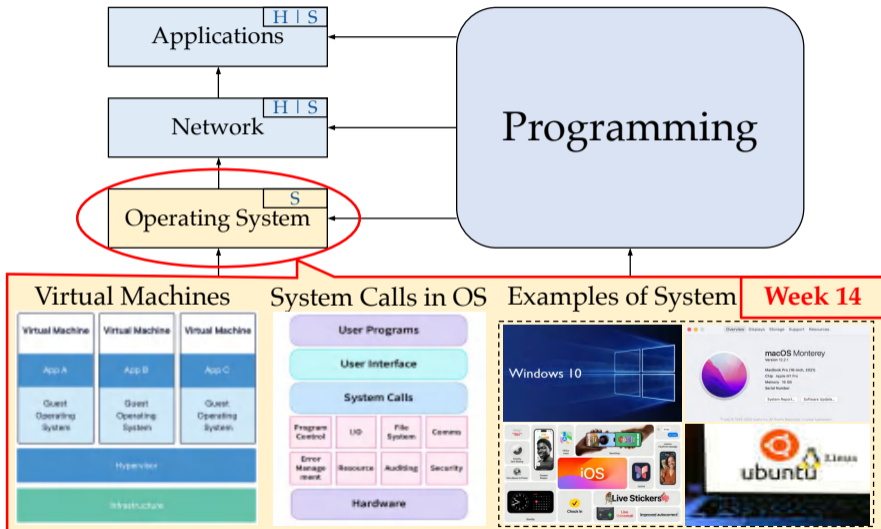
Layers of a Computing System



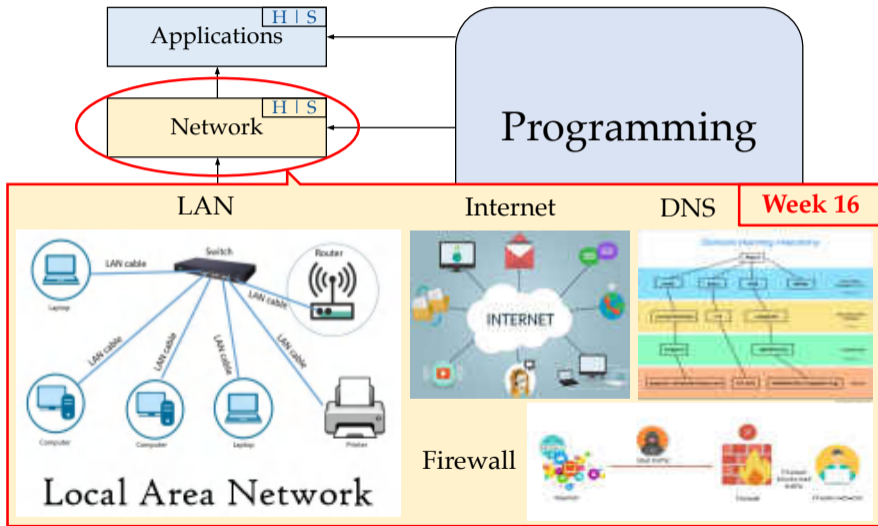
Layers of a Computing System



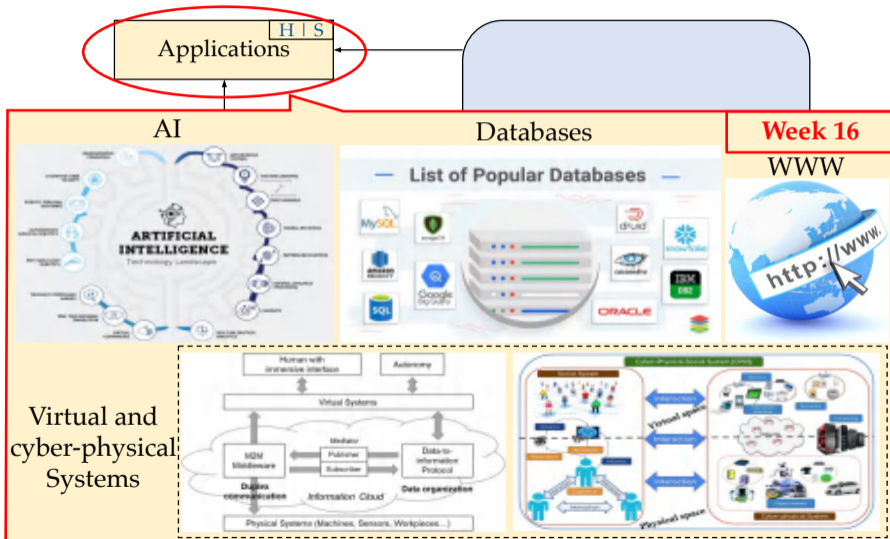
Layers of a Computing System



Layers of a Computing System




Layers of a Computing System




Layers of a Computing System

Searching



Graph Algorithms




Sorting

Sorting Algorithms	Time Complexity			Space Complexity
	Best Case	Average Case	Worst Case	Worst Case
Bubble Sort	$O(N^2)$	$O(N^2)$	$O(N^2)$	$O(1)$
Selection Sort	$O(N^2)$	$O(N^2)$	$O(N^2)$	$O(1)$
Insertion Sort	$O(N^2)$	$O(N^2)$	$O(N^2)$	$O(1)$
Quick Sort	$O(N \log N)$	$O(N \log N)$	$O(N^2)$	$O(N)$
Merge Sort	$O(N \log N)$	$O(N \log N)$	$O(N \log N)$	$O(N)$
Heap Sort	$O(N \log N)$	$O(N \log N)$	$O(N \log N)$	$O(1)$

Week 10

String



```
graph BT; Information --> DigitalCircuit[Digital Circuit]; DigitalCircuit --> Computer; style Computer fill:#add8e6,stroke:#333,stroke-width:1px; style DigitalCircuit fill:#add8e6,stroke:#333,stroke-width:1px; style Information fill:#add8e6,stroke:#333,stroke-width:1px;
```

```
graph TD; subgraph Algorithms; direction TB; A[Algorithms]; end; A --- Searching; A --- Sorting;
```

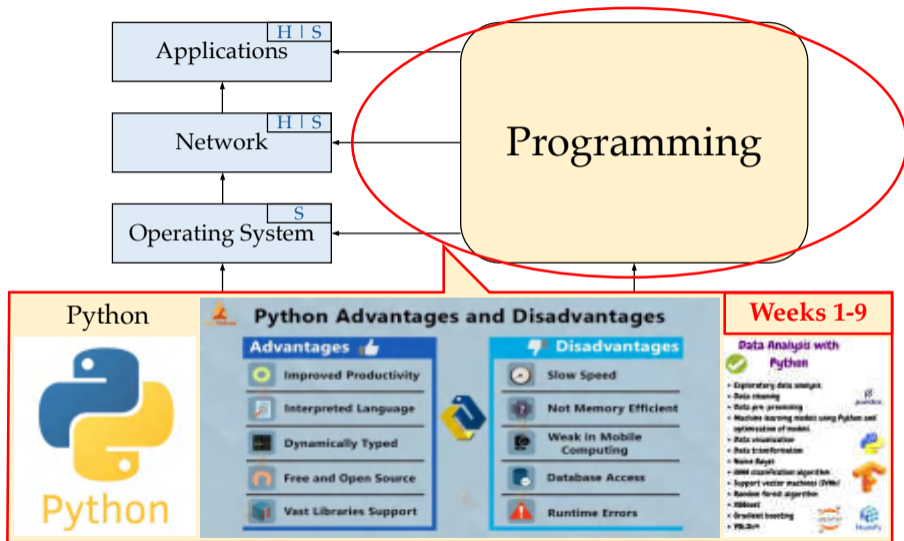
Yue Zhang

Westlake University

August 1, 2023

41 / 69

Layers of a Computing System



Semester Plan

Week		Content	Week		Content
Week 1	HS	Introduction	Week 9	S	Classes and Objects
Week 2	HS	Binary Numbers; IDLE	Week 10	S	Algorithms
Week 3	HS	Digitization; Python Program	Week 11	H	Gates and Circuits
Week 4	S	Branching and Looping	Week 12	H	Computation Theory
Week 5	S	Problem Solving	Week 13	H	Computer Architecture
Week 6	S	Functions and Modules	Week 14	H	Operating System
Week 7	P	Project Proposal Presentation	Week 15	HS	Computers and Programming
Week 8	S	Lists and Maps	Week 16	HS	Network and Applications

Week 17 (Week 18) — Project Presentations

Tuesday – Lecture (Yue)

Friday – Lab (TAs)

- Check off assignments
 - Programming (S)
 - Quiz (H)
- Discuss Project
- QA

Programming check-off and quizzes take **50%** of your final grade.

Project Assignment

- Project Teamup – 5 students each team, self teamup.
- Project Ideas
 - A mini MatLab program, which achieves some of the essential functionalities, such as calculation and chart drawing.
 - A Westlake BBS system, which allows faculties and students to chat and exchange data.
 - Improving the Website of a college, a lab or our university by putting more functionalities.
 - Helping some research team to build efficiency tools.
- The project idea is one important first step to your project design. You can talk to university offices, labs or other users for making design specifications.

Project Assignment

- Project takes 50% of your score.
 - **Contribution** – solves a meaningful problem
e.g, crawl and maintain financial data
enhance a website of a college
solve a problem for a lab
 - **Significance** – the amount of work
1000 lines of code is significant
 - **Completeness** – everything functions
 - **Novelty** – nice design, algorithm etc.
 - **Teamwork** – everyone works together

Project Assignment

- Team up by week4
 - First project proposal by the end of week 4 (draft)
 - Second project proposal by the end of week 6
Project proposal presentation on week 7
 - Mid-term report by the end of week 12
 - Final report submission by the end of week 16
 - Final presentation week 17
- Grading
 - instructor, teaching assistants, fellow students.

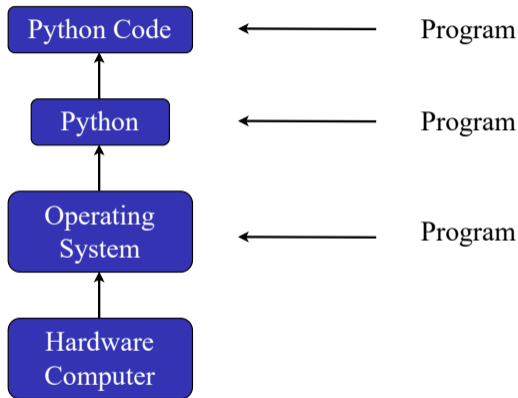
- 1 What Is a Computer?
- 2 The Course
- 3 Python Setup**

What is a program?

- A set of instructions to achieve a goal using a computer.
- Examples?



What is a program?

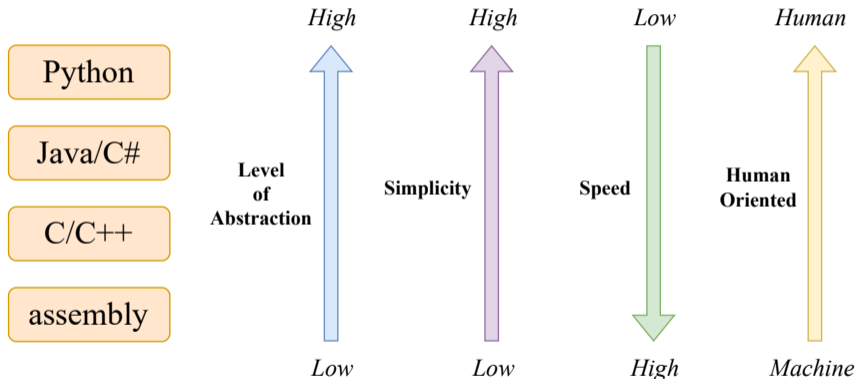


- Programs operates computers

What is programming?

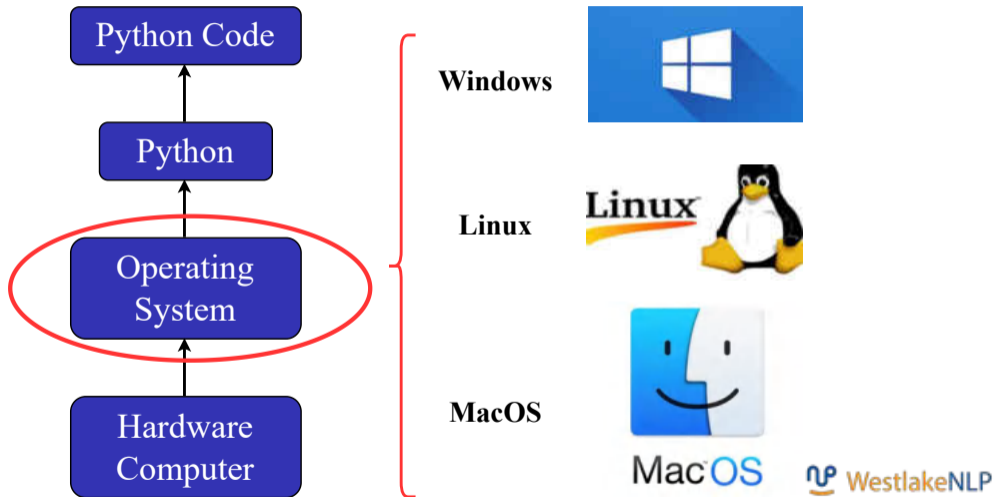
- Write a program.
- Talk to a computer.
 - Tell it what to do.

Different Types of Programs

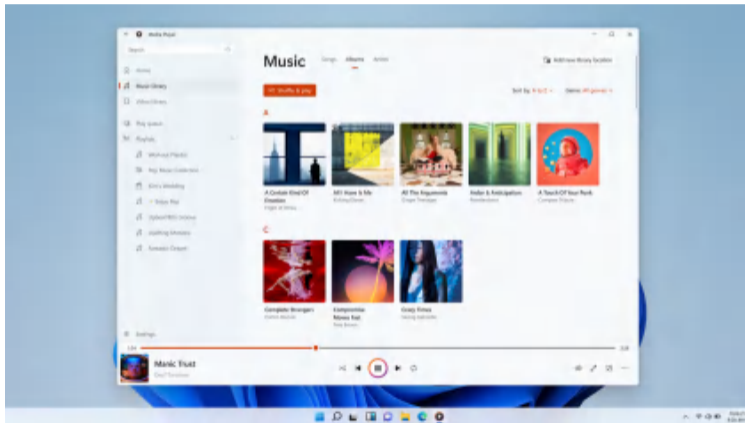


- Invented in the 1990s.
- Named after the comedy show “Monty Python’s Flying Circus”.
- Gain popularity and adopted by Google Inc.
- Used frequently in AI.
- Different Versions.

Python Background



User Interfaces of OS - Graph User Interface(GUI)



Python Background

User Interfaces of OS - Text User Interface(Text Console)

```
Command Prompt - C:\Users\ericm\AppData\Loca...
Microsoft Windows [Version 10.0.19042.928]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ericm>C:\Users\ericm\AppData\Local\Programs\Python\Python39\python.exe
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hello, World!")
Hello, World!
>>>
```

Windows Command Line



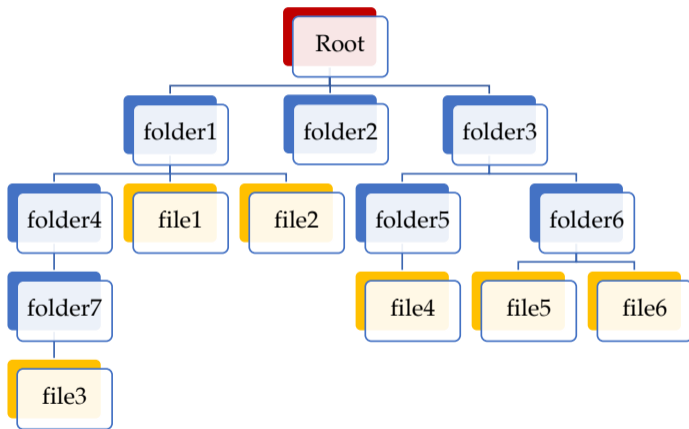
```
Adityas-MacBook-Pro:~ adityasharma$ python3
Python 3.7.2 (default, Feb 12 2019, 08:16:38)
[Clang 10.0.0 (clang-1000.11.45.5)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy as np
>>> arr1 = np.array([[1,-2],[0,2],[10,4],[6,4]])
>>> arr1.shape
(4, 2)
>>> arr2 = np.array([[11,2],[10,-2],[1,1],[0,-4]])
>>> arr2.shape
(4, 2)
>>> sum = np.add(arr1,arr2)
>>> sum
array([[12,  0],
       [10,  0],
       [11,  5],
       [ 6,  0]])
>>> sum.shape
(4, 2)
>>> multiply = np.multiply(arr1,arr2)
>>> multiply
array([[ 11, -4],
       [  0, -4],
       [10,  4],
       [  0, -16]])
>>> multiply.shape
(4, 2)
>>> dot_product = np.dot(arr1,arr2)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ValueError: shapes (4,2) and (4,2) not aligned: 2 (dim 1) != 4 (dim 0)
>>>
```

MacOS Terminal

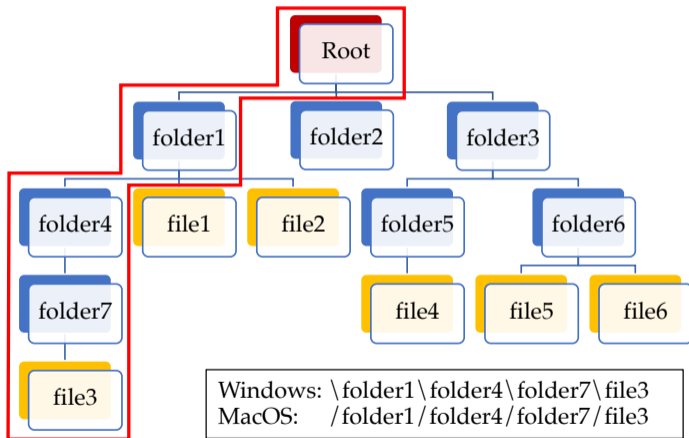


- Files System
 - Data Organization in Harddrive
- Files
 - Naming
 - myfile.py* (*.py* is **Extension Name**)
 - readme.txt*
 - video.mpeg*
 - audio.mp3*
- Folders
 - Tree Structure

Python Background (File System)



Python Background (File System)



Python Background (File System)

- Windows Disk Partition

```
# Windows  
C:\   D:\   E:\   ...  
C:\Documents\readme.txt
```

- MacOS does not have logical partition.

```
# MacOS  
/Users/yue_zhang/readme.txt
```

Python Background (File System)

- Home Folder

```
# Windows  
C:\Users\yue_zhang  
  
# MacOS  
Yues-MacBookPro:~ yue_zhang
```

- Current Working Folder

```
# Windows (cd)  
C:\Users\yue_zhang>cd  
C:\Users\yue_zhang  
  
# MacOS (pwd)  
Yues-MacBookPro:~ yue_zhang$ pwd  
/Users/yue_zhang
```

Python Background (File System)

- Listing the files and folders contained in the current working folder.

```
# Windows (dir)
```

```
C:\Users\yue_zhang>dir
```

```
Volume in drive has no label.  
Volume Serial Number is 5CFE-F491
```

```
Directory of C:\Users\yue_zhang
```

```
04/08/2023 12:02 PM <DIR> .  
04/08/2023 12:02 PM <DIR> ..  
04/08/2023 12:02 PM <DIR> Desktop  
... ..
```

```
# MacOS (ls)
```

```
Yues-MacBookPro:~ yue_zhang$ ls
```

```
Applications      Library           Desktop           Movies            Pictures  
Documents         Music            Public            Downloads         miniconda3
```

Python Background (File System)

- Making a new folder. (`mkdir <folder_name>`)

```
# Windows
```

```
C:\Users\yue_zhang>mkdir new_folder
```

```
C:\Users\yue_zhang>dir
```

```
... ..
```

```
Directory of C:\Users\yue_zhang
```

```
04/08/2023 12:02 PM <DIR> .
```

```
04/08/2023 12:02 PM <DIR> ..
```

```
04/08/2023 12:02 PM <DIR> new_folder
```

```
... ..
```

```
# MacOS
```

```
Yues-MacBookPro:~ yue_zhang$ mkdir new_folder
```

```
Yues-MacBookPro:~ yue_zhang$ ls
```

```
Applications      Library           Desktop          Movies           Pictures  
Documents          Music            Public           Downloads        miniconda3  
new_folder
```

Python Background (File System)

- Removing an empty folder. (`rmdir <folder_name>`)

```
# Windows
```

```
C:\Users\yue_zhang>rmdir new_folder
```

```
C:\Users\yue_zhang>dir
```

```
... ..
```

```
Directory of C:\Users\yue_zhang
```

```
04/08/2023 12:02 PM <DIR> .
```

```
04/08/2023 12:02 PM <DIR> ..
```

```
04/08/2023 12:02 PM <DIR> Desktop
```

```
... ..
```

```
# MacOS
```

```
Yues-MacBookPro:~ yue_zhang$ rmdir new_folder
```

```
Yues-MacBookPro:~ yue_zhang$ ls
```

```
Applications      Library           Desktop          Movies           Pictures
Documents          Music            Public           Downloads        miniconda3
```


Python Background (File System)

- Entering a folder by relative path from current working path.
(`cd <folder_name>`)

```
# Windows
```

```
C:\Users\yue_zhang>cd folder1
```

```
C:\Users\yue_zhang\folder1>cd
```

```
C:\Users\yue_zhang\folder1
```

```
# MacOS
```

```
Yues-MacBookPro:~ yue_zhang$ cd folder1
```

```
Yues-MacBookPro:~ yue_zhang/folder1$ pwd
```

```
/Users/yue_zhang/folder1
```

Python Background (File System)

- Entering the parent folder. (`cd ..`)

```
# Windows
```

```
C:\Users\yue_zhang>cd ..
```

```
C:\Users>cd
```

```
C:\Users
```

```
# MacOS
```

```
Yues-MacBookPro:~ yue_zhang$ cd ..
```

```
Yues-MacBookPro:~ Users$ pwd
```

```
/Users
```

Python Background (File System)

- Entering a folder by absolute path. (`cd <folder_name>`)

```
# Windows
```

```
C:\Users\yue_zhang>cd D:\python_totutorial
```

```
D:\python_totutorial>cd
```

```
D:\python_totutorial
```

```
# MacOS
```

```
Yues-MacBookPro:~ yue_zhang$ cd /Users/yue_zhang/python_totutorial
```

```
Yues-MacBookPro:~ yue_zhang$ pwd
```

```
/Users/yue_zhang/python_totutorial
```

Python Background (File System)

- Where is Python?

```
# Windows
C:\python3.9\python.exe
# MacOS
/usr/bin/python3.9
```

- Go there and execute python.

```
# Windows
C:\Users\yue_zhang>cd C:\python3.9
C:\Users\yue_zhang\python3.9>python.exe
Python 3.9.12 <v3.9.12:7ds3ahj512, Aug 22, 2023, 12:34:32> [MSC v.1914 64 bit<AMD64>] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>

# MacOS
Yues-MacBookPro:~ yue_zhang$ cd /usr/bin
Yues-MacBookPro:~ bin$ ./python3.9
Python 3.9.12 (main, Apr 5 2022, 01:52:34)
[Clang 12.0.0 ] :: Anaconda, Inc. on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

This week check-off:

Install Python