Introduction

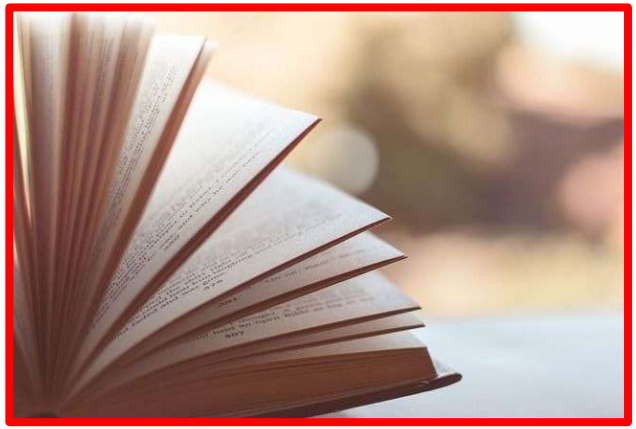
Introduction to Python and Data Science

Instructor: Dr. Sneha Singh, SCEE

Term : August 2025

Books

- Python Programming for the Absolute Beginner, Third Edition (Chapters 1-7), by Michael Dawson
- Think Like a Programmer - An Intro. to Creative Problem Solving - V. Spraul (No Starch, 2012)
- Probability, Random Variables and Stochastic processes by Papoulis and Pillai, 4th Edition, Tata McGraw Hill Edition.



Source: Image by Pexels from Pixabay

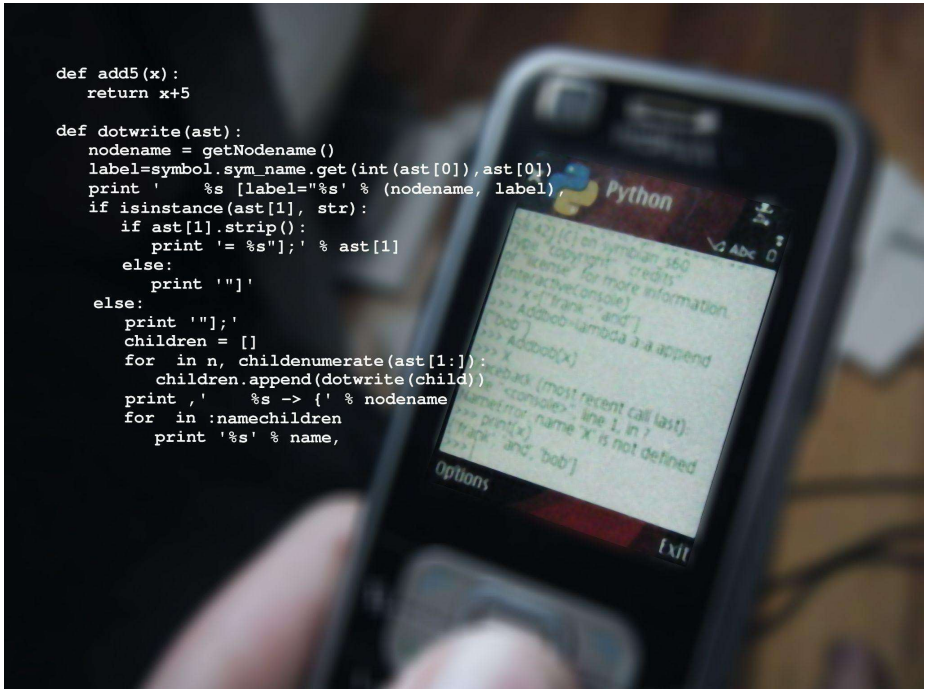
Marks Distributions

- ➔ Attendance- >=80% (To appear in the exams; Quiz, Mid Sem, End Sem)
- ➔ Two Subjective Exams - 50% (50)
- ➔ One/Two Quiz/s (MCQ) - 20% (25)
- ➔ Lab Assignment - 15% (15)
- ➔ Home Assignment - 10% (10)
- ➔ Attendance - 5% (5)

Course Content?

Python Basics:

- ➔ Variables and Data Types: Strings, Integers, Float
- ➔ Array: Lists, Numpy
- ➔ Operations, Operator Overloading
- ➔ Tuples and Dictionaries
- ➔ Functions: Inbuilt and User-defined
- ➔ Conditional Statements and Loops
- ➔ Errors: Syntax and Logical



Images by Gerd Altmann and 1315674 from Pixabay

Course Content?

Python Basics

- ➔ Reading and Writing Files
- ➔ Object Oriented Programming
- ➔ Command Line Arguments
- ➔ Handling Exceptions

Complexity Analysis

Stacks and Queues

```
class Bike:

    def handle(self):
        print("Used for acceleration.")

    def break(self):
        print("Used to stop bike.")
```

Slide 23

Course Content?

Plots: Visualizing Data

Statistics

Probability

Regression, interpolation

Machine learning

Clustering

Image by Colin Behrens from Pixabay

Labs

Days: Tuesday/Friday

Time: 2-4, 4-6 pm on Tuesdays

4-6 pm on Fridays

Venue: A10 and A11 PC Lab

Note:

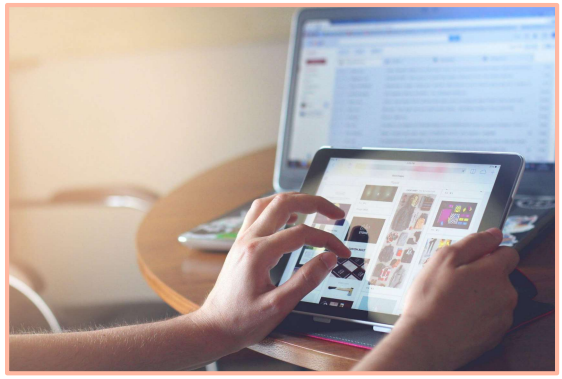
- Every lab will have MARKS for VIVA. So it is compulsory to attend labs
- 1-2 times “any kind of cheating” -> 0 marks in lab/exam
- 3 times “any kind of cheating” in lab -> F grade

History and Background

Computing

Process of using computer technology to complete a given goal-oriented task.

- design and development of software and hardware systems
- structuring, processing and managing information
- pursuit of scientific studies
- making intelligent systems
- creating and using different media for entertainment and communication



Source: https://www.techopedia.com/definition/6597/computing
Image by fancycrave1 from Pixabay

Computing

- Cloud Computing (Services through Internet)
  - Google Vision API
- High Performance Computing (High Speed)
  - https://iitmandi.ac.in/new/high-performance-cluster
- Edge Computing (Compute at Edges: Latency)

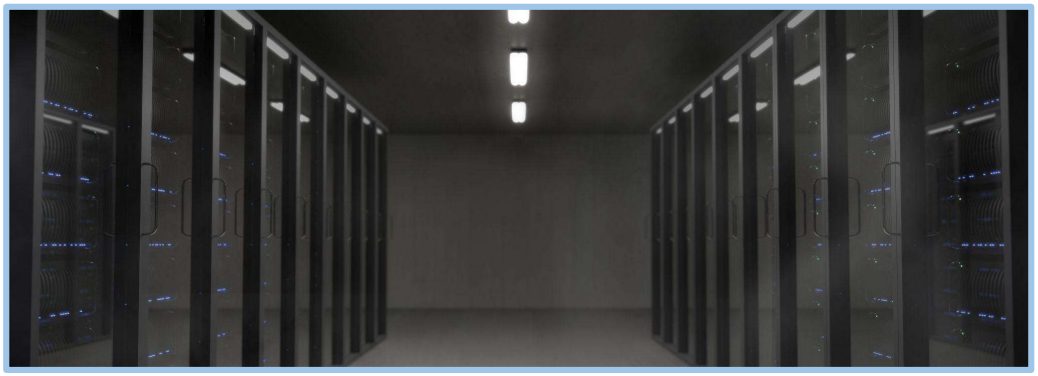


Image by Mudassar Iqbal and Elias from Pixabay

Questions

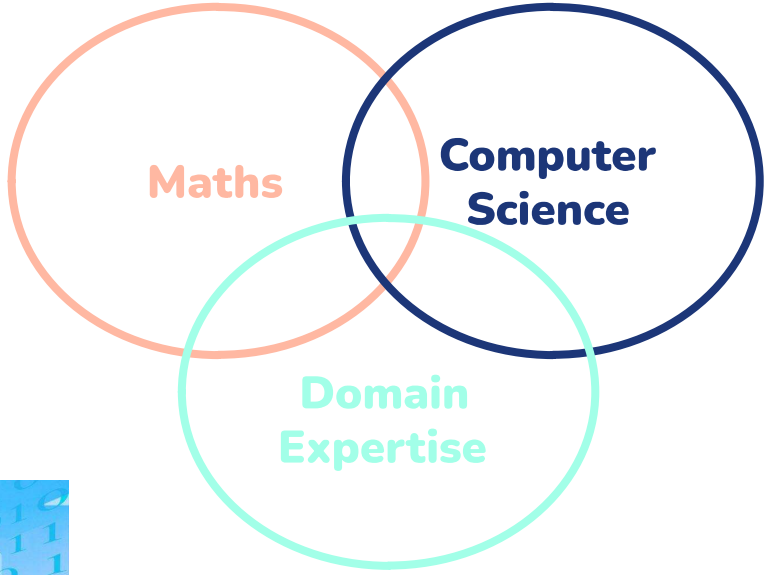
- Can creating music through computers be considered as computing task?
- What is difference between cloud computing and edge computing?
- Data science is a field of study that combines how many subjects?
- What are the 3 subjects that data science combines?
- What is the purpose of combining the 3 subjects?

Data Science

Field of study that combines:

- Mathematics,
- Domain Expertise,
- and Programming

To extract meaningful insights from data.



Source: https://www.datarobot.com/wiki/data-science/
Image by Gerd Altmann from Pixabay
https://medium.com/analytics-vidhya/introduction-to-data-science-28deb32878e7

Some Real Life Applications

Tree Counting: https://tinyurl.com/nd4msyf2

GUI to annotate object boxes: https://github.com/heartexlabs/labelImg

- The above GUI can be modified to annotate Leaf Tips

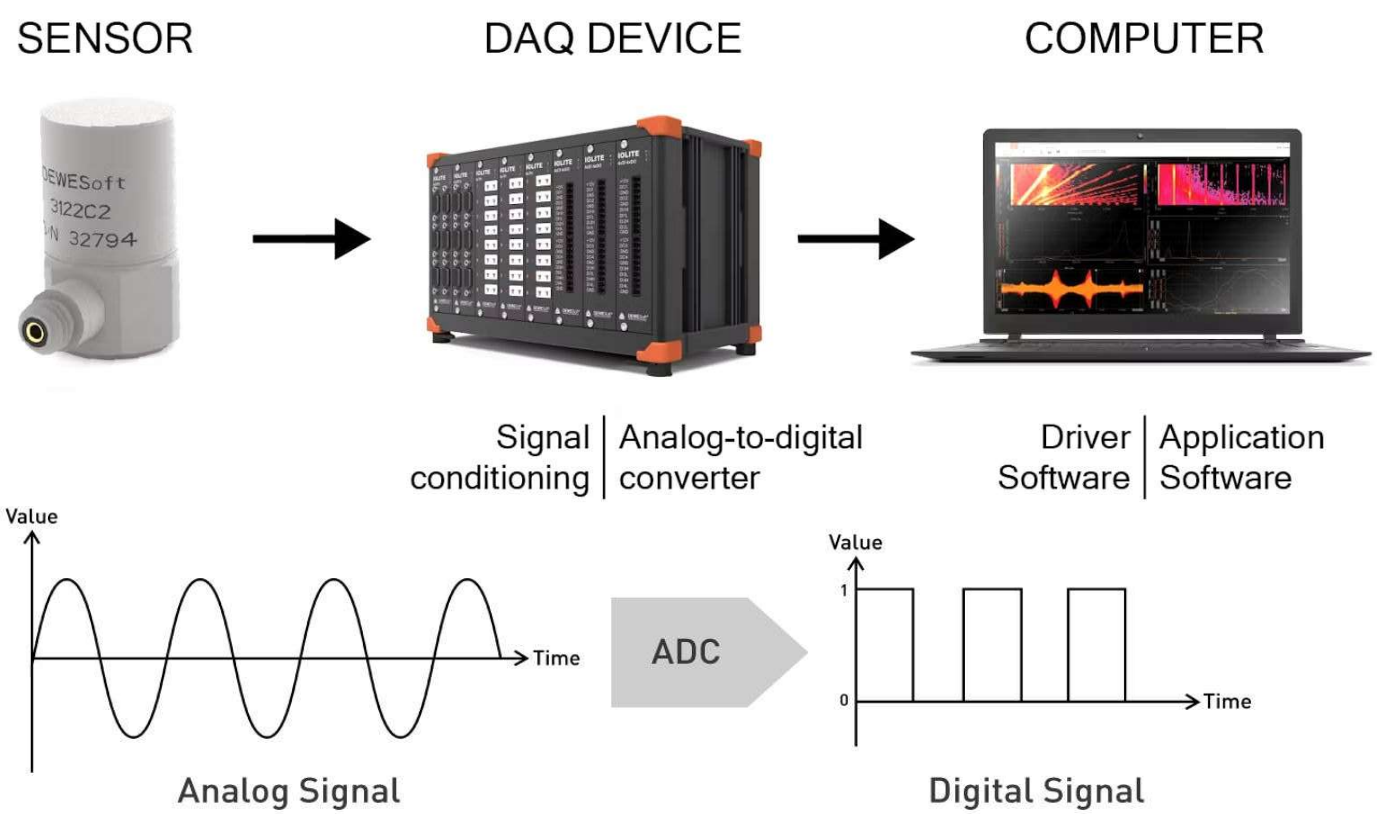
Motorcycle Violations: https://tinyurl.com/3x5hhtb3

Character Control in Games: https://github.com/sreyafrancis/PFNN

- Video: https://tinyurl.com/2d5a5k92

OpenOCRCorrect: https://tinyurl.com/47kabh28

Analog and Digital Systems

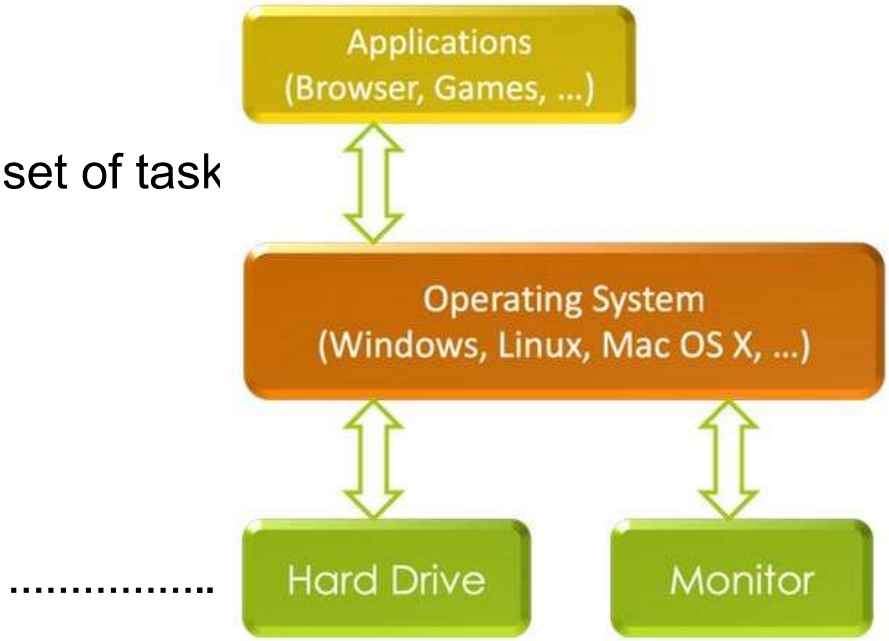


Images by https://dewesoft.com/blog/what-is-data-acquisition

Computers: Basic Structures

Computer System Categories:

- *Software:* Enables hardware to perform set of task
  - Programming
  - Algorithms
  - Instructions
- *Hardware:* Physical Components.



Computers: Softwares

System Software: Operation Coordination

Operating System

Utility Programs

Compiler, Interpreter

Images by barek2marcin, Memed_Nurrohmad and 1315674 from Pixabay

Computers: Softwares

System Software: Performing Specific Task/Tasks

Text Processing

Data Operations

DBMS

Computers: Functional Diagram

Registers, Hold :

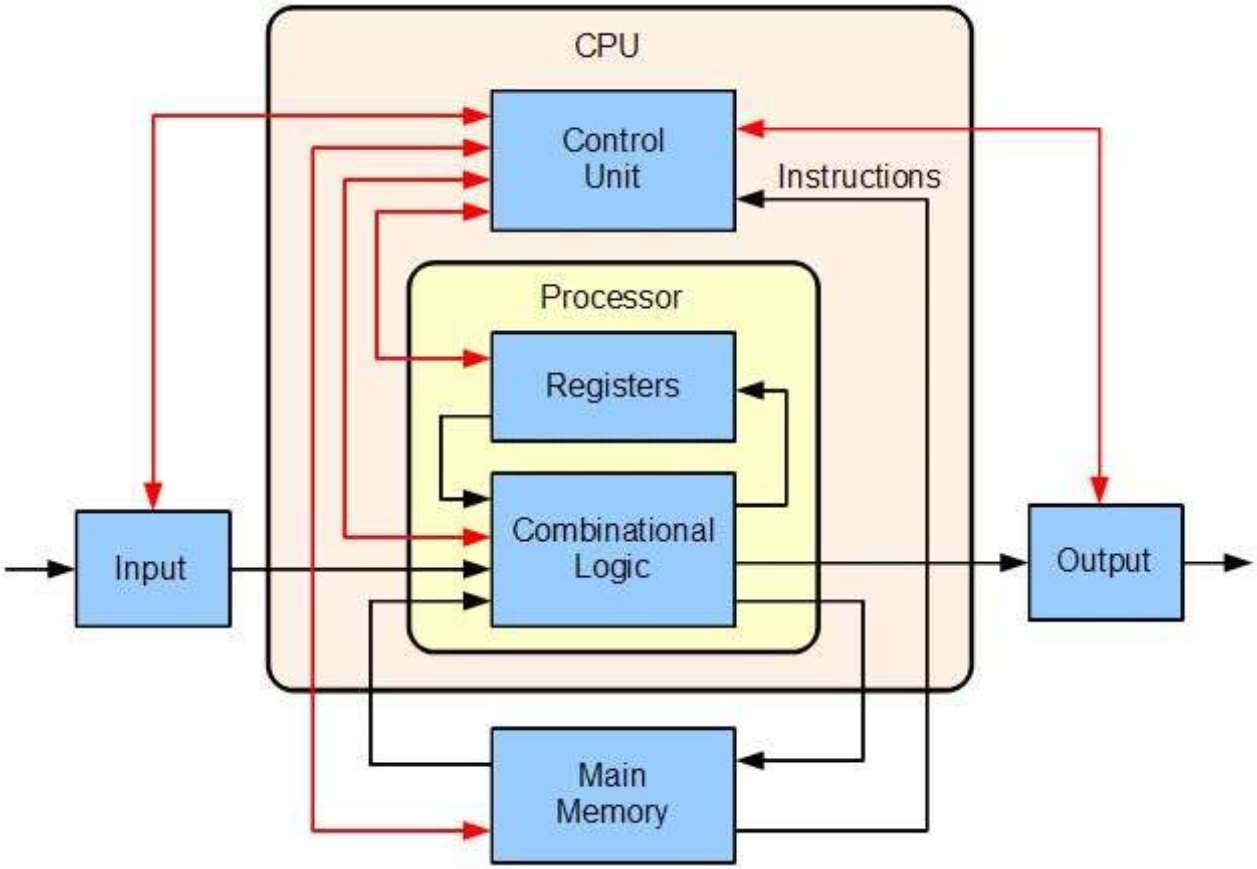
- Data
- Instructions
- Storage Address

Control Unit:

- Read instructions
- Send signals to other parts

Combinational Logic:

- Performs Arithmetic +/-
- Logical: AND/OR/XOR



Computers (History): Hardwares

Abacus (2700–2300 BC)

Principle: Counting (beads)

Operations:

- Addition
- Subtraction
- Multiplication
- Division
- Square Root
- Cubic Root

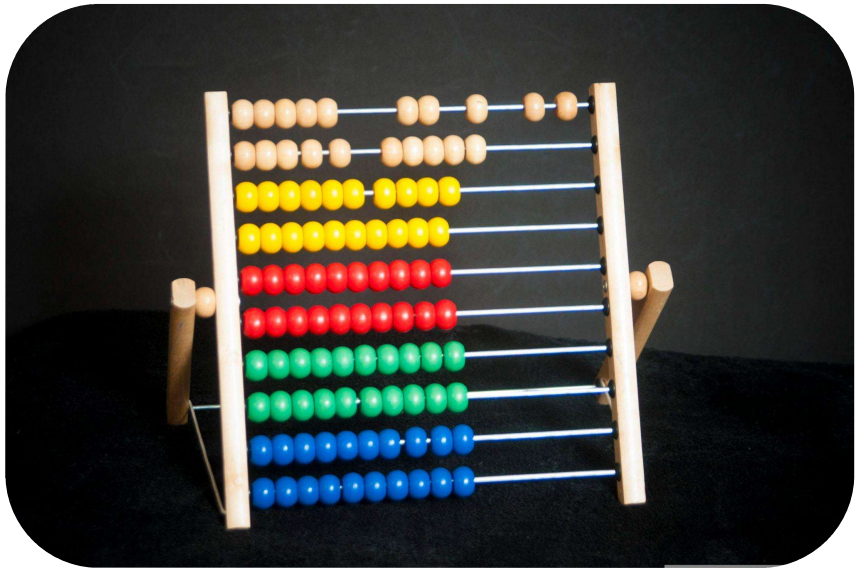


Image by Michelle Scott from Pixabay

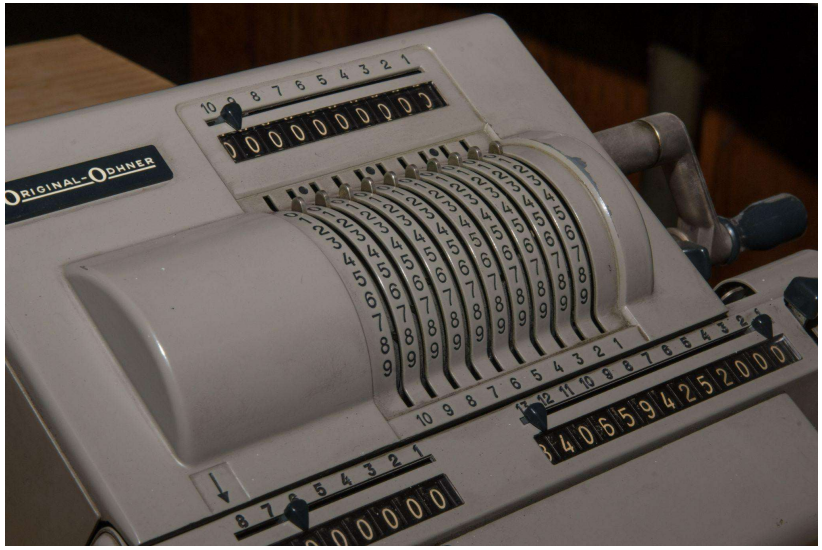
Computers (History): Hardwares

Mechanical Calculators (1820)

Principle: Counting (gears)

Operations:

- Addition
- Subtraction
- Multiplication
- Division



Images by Michael Kauer and MasterTux from Pixabay

Computers (History): Hardwares

Slide Rule (1814)

Principle: Log scales

- Operations:
- Multiplication
- Division
- Square
- Square Root

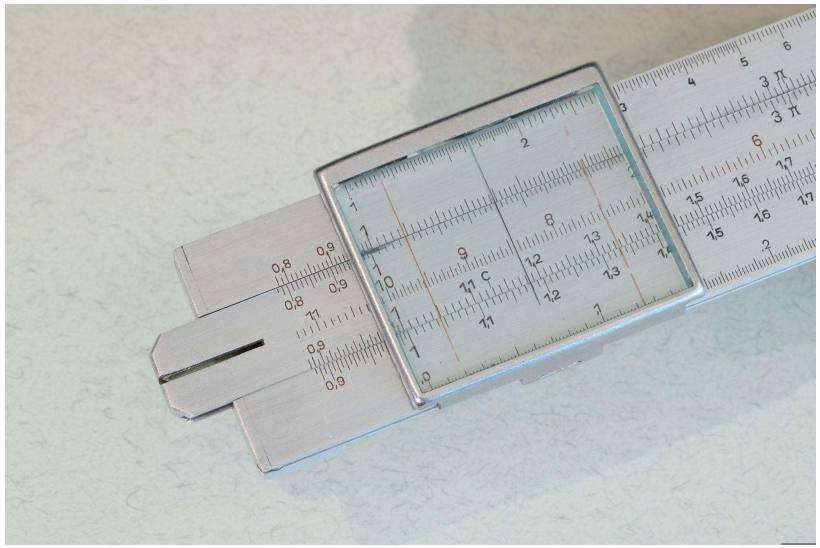
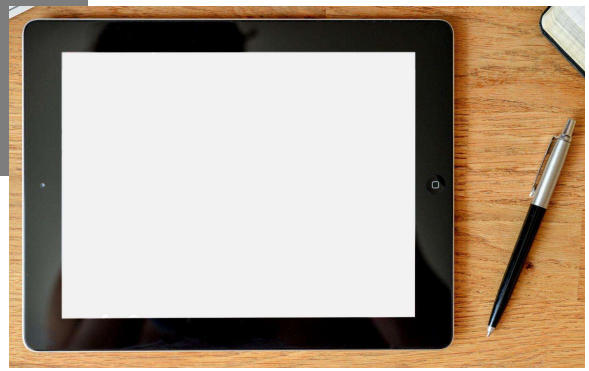


Image by Olaf from Pixabay

Functional Units: Input Devices

Function: Read data/coded information from



Images by Thorsten Frenzel, kalhh and stokpic from Pixabay

Functional Units: Output

Function: Send/write processed data to display on

Images by PublicDomainPictures, PublicDomainPictures and Maria from Pixabay

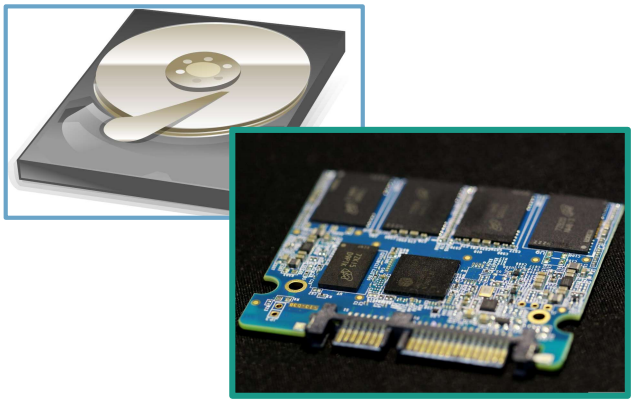
Functional Units: Memory

Function: Store, transfer, and extract, program and data

Fix, reliable, and easy to find/fetch data

Floppy Disk: 1.44MB
Super Disk: 120/200/240 MB
Zip disk: 250 MB

Compact Disk (CD) 700MB
Digital Versatile Disk (DVD) 4.7GB



Hard Disk (HDD): 16GB-20TB
Solid State (SSD): 128GB-100TB

Images by Erwin, PublicDomainPictures, Clker-Free-Vector-Images and manseok Kim from Pixabay

Functional Units: Processor

Function: Process data/instructions

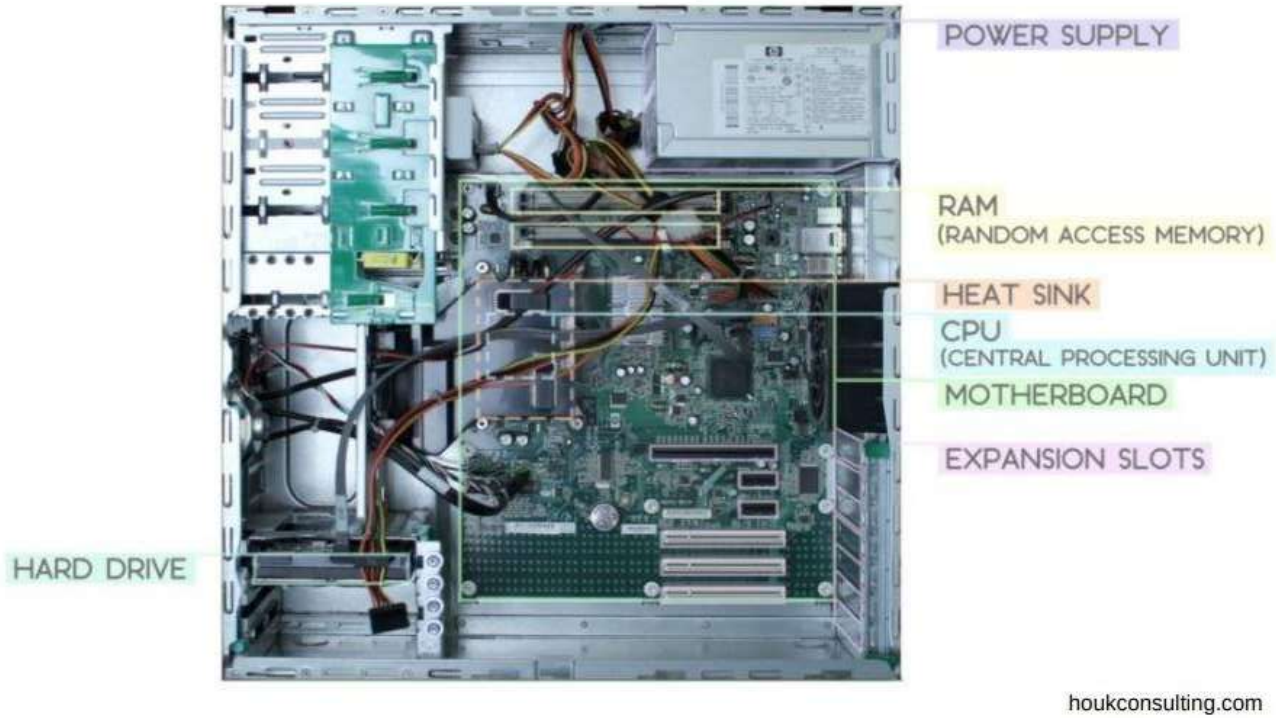


Image from https://students.iitmandi.ac.in/moodle/pluginfile.php/64385/mod_resource/content/1/lec3_computersProgramming.pdf

Functional Units: Registers

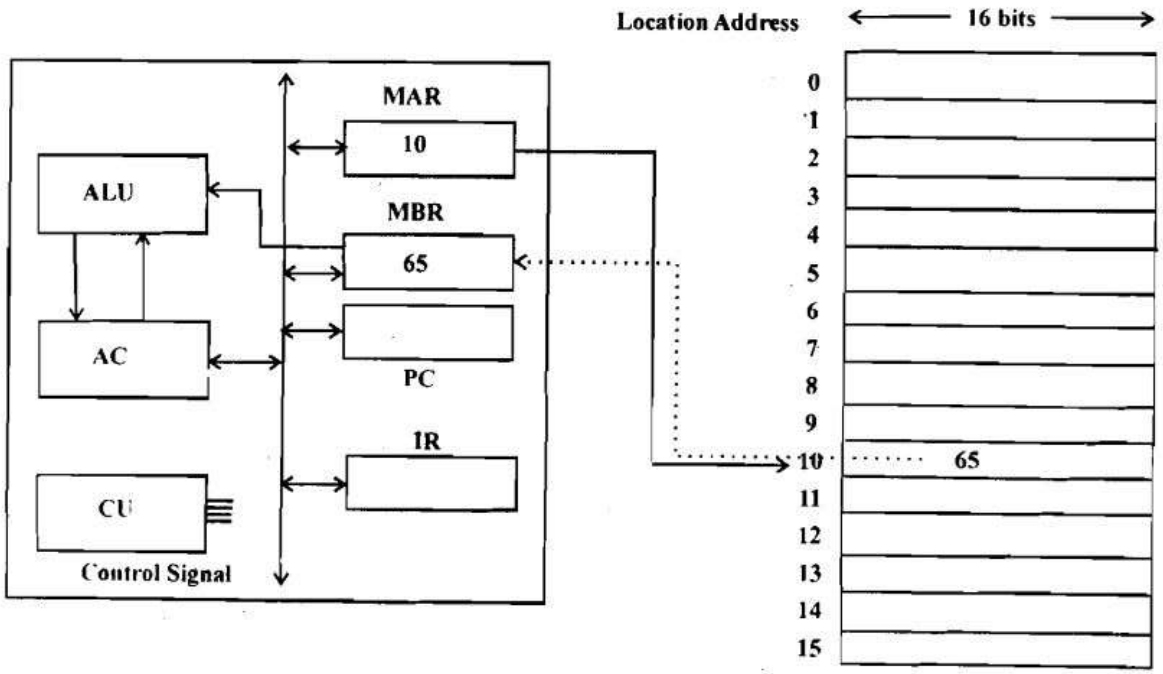
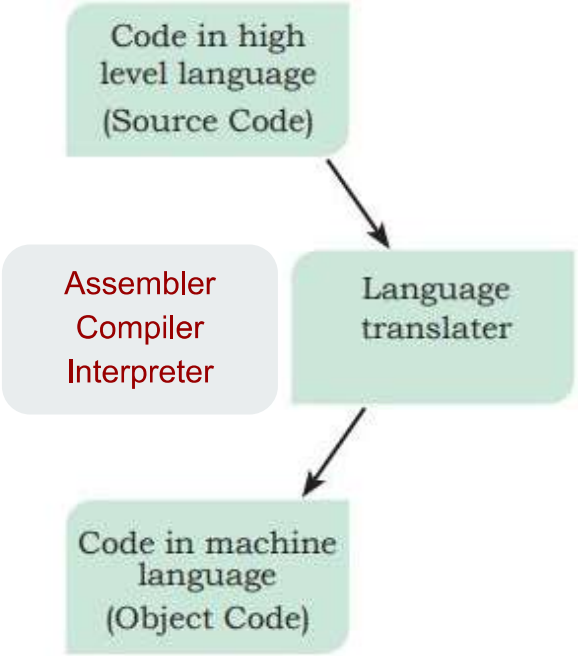


Image by Wikipedia



Questions

- What is the principle used in slide rule?
- What are the 4 important units of Computer?
- Which of the unit in CPU is connected to all other CPU units?
- Why do we need registers if we have main memory?
- What do registers hold in CPU?

Information System: Decimal V/S Binary Numbers

0	0
1	1
2	10
3	11
.	100
.	.
.	.
9	.
10	

Image by Gerd Altmann from Pixabay

Binary Numbers

In Decimal Number System (dec ->10, i.e. base is 10):

- We have single digit numbers: 0, 1, 2 , 3, 4, 5, 6, 7, 8, 9
- We have units place, tens place, hundreds place etc.
- 123 means 3X1 + 2X10 + 1X100

Similarly in Binary Number System (binary ->2, i.e. base is 2):

- We have single digit numbers: 0, 1
- You can say: We have units place, twos place, fours place etc.
- So 101 means 1X1 + 0X2 + 1X4 (= 5 in decimal number system)

Just like 9+1 = 10 in decimal number system, we have 1+1 = 10 in binary number system! (since 1 is the largest one digit in binary number system like 9 in decimal number system). Similarly: 1+0 = 1 = 0+1, 10 + 01 = 11, 011 + 010 = 101.

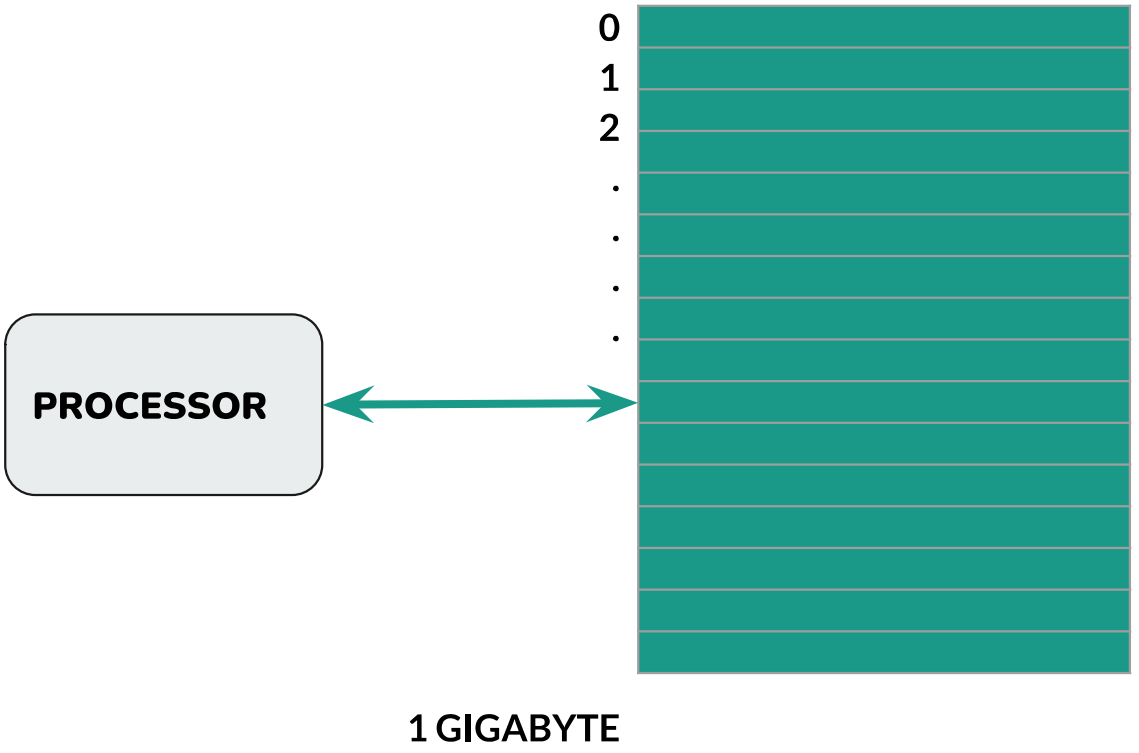
Questions

- What is the difference between 10 in binary number system and 10 in decimal number system?
- Convert the decimal number 20 to binary.
- Convert the binary number 10101 to decimal.
- When will a binary number be odd?
- How will you multiply a binary number by 2?

The Computing Machine

Memory:

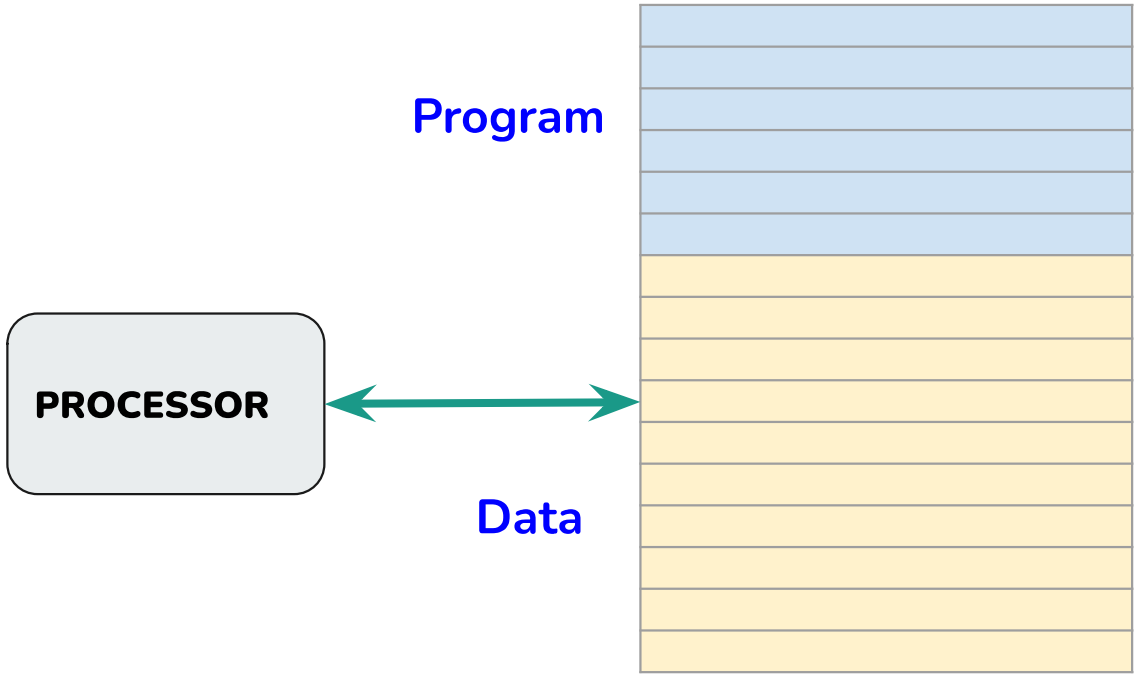
- Series of locations
- Store Information



The Computing Machine

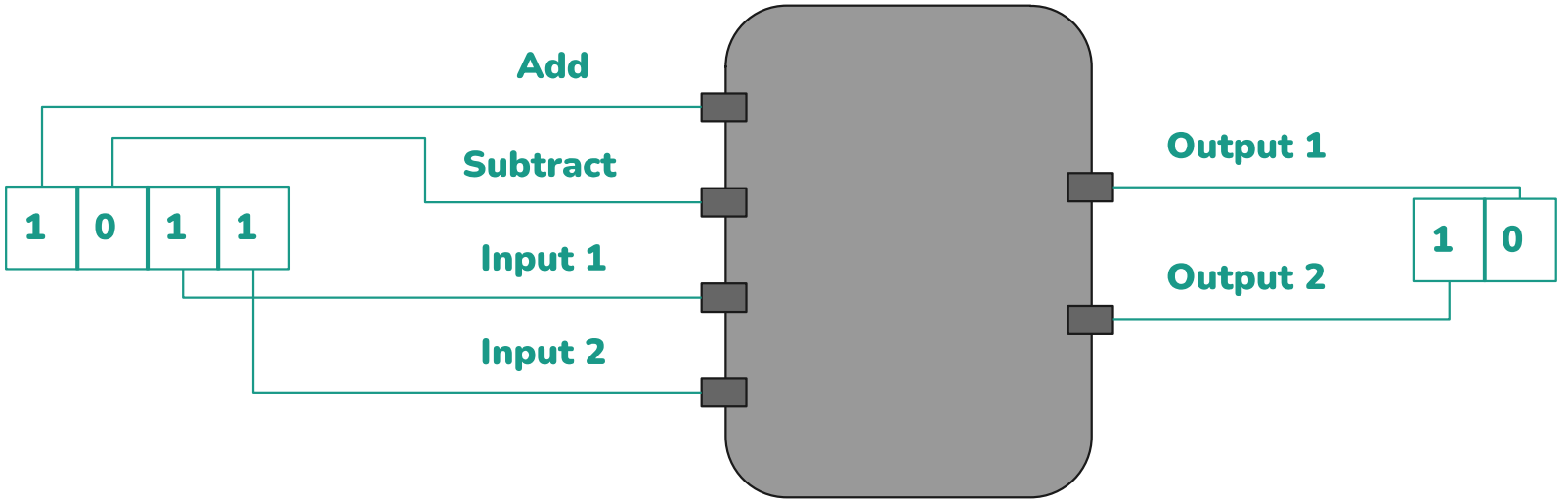
Memory:

- Series of locations
- Store Information



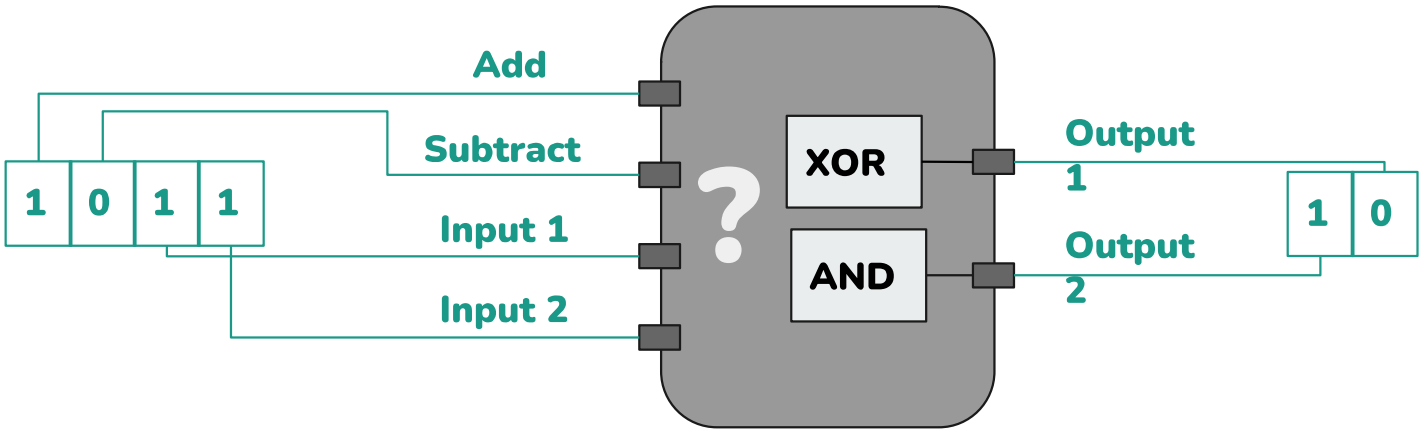
Machine Code

How a computer adds two one bit binary numbers?



Machine Code

How a computer adds two one bit binary numbers?



How about just saying this to your computer:-

Output = Input1 + Input2

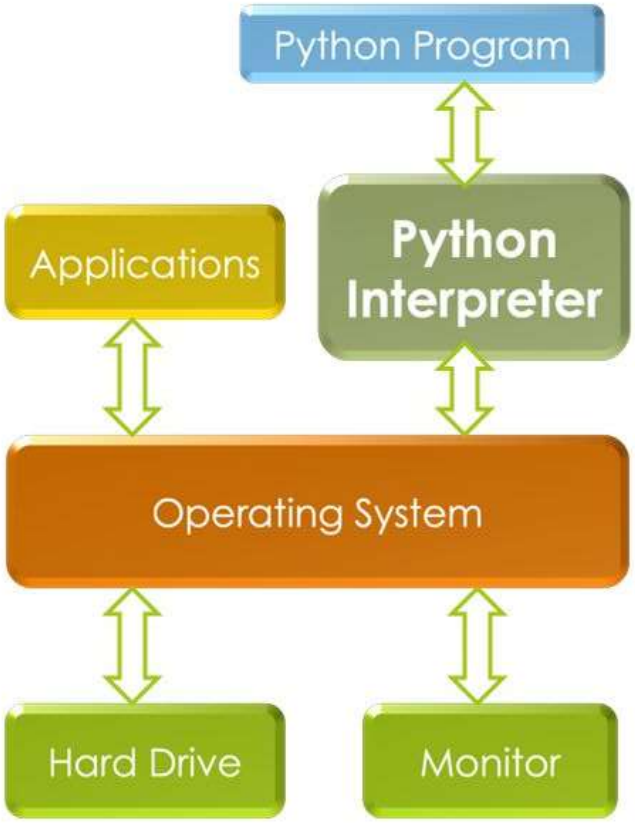
Programming Language

Computer Speak its language: When you write a program in Python, Java etc. It does not run directly on the OS.

Interpreter or virtual machine takes it and runs it for you translating your commands into the language of the OS

- High-level: Python, C++, Java
- Low-level: Machine language, computers can only execute these
- High-level languages have to be processed into low-level before the computer can run them

But high-level languages can run on different kinds of computers and are easier for humans to write and read, so most programs are written in high-level



Programming Language: Python

- Python is one of many programming languages and Free, open sourced.
- Easy to use and developed by Guido van Rossum in 1991.
- It is Object-oriented programming but is optional.
- We will use Python 3, there are 2 widely used versions.
- Running Python on the command line:

  $ python3

  or

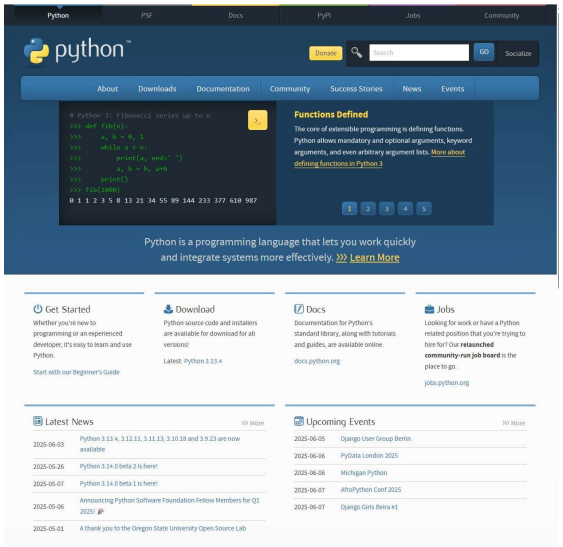
  $ python3 –i filename.py

Python: Interaction and Setting Up

There are two ways to interact with a Python interpreter:

1. Tell it to execute a program that is saved in a file with a *.py extension*

2. Interact with it in a program called a shell



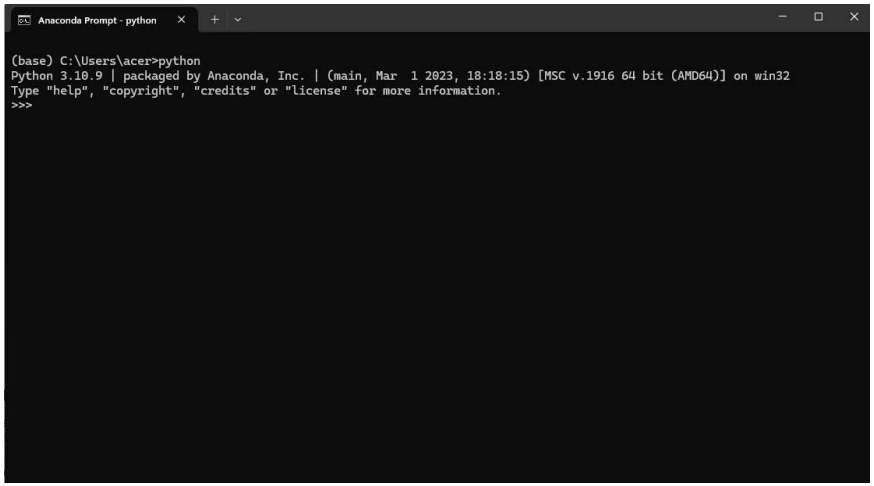
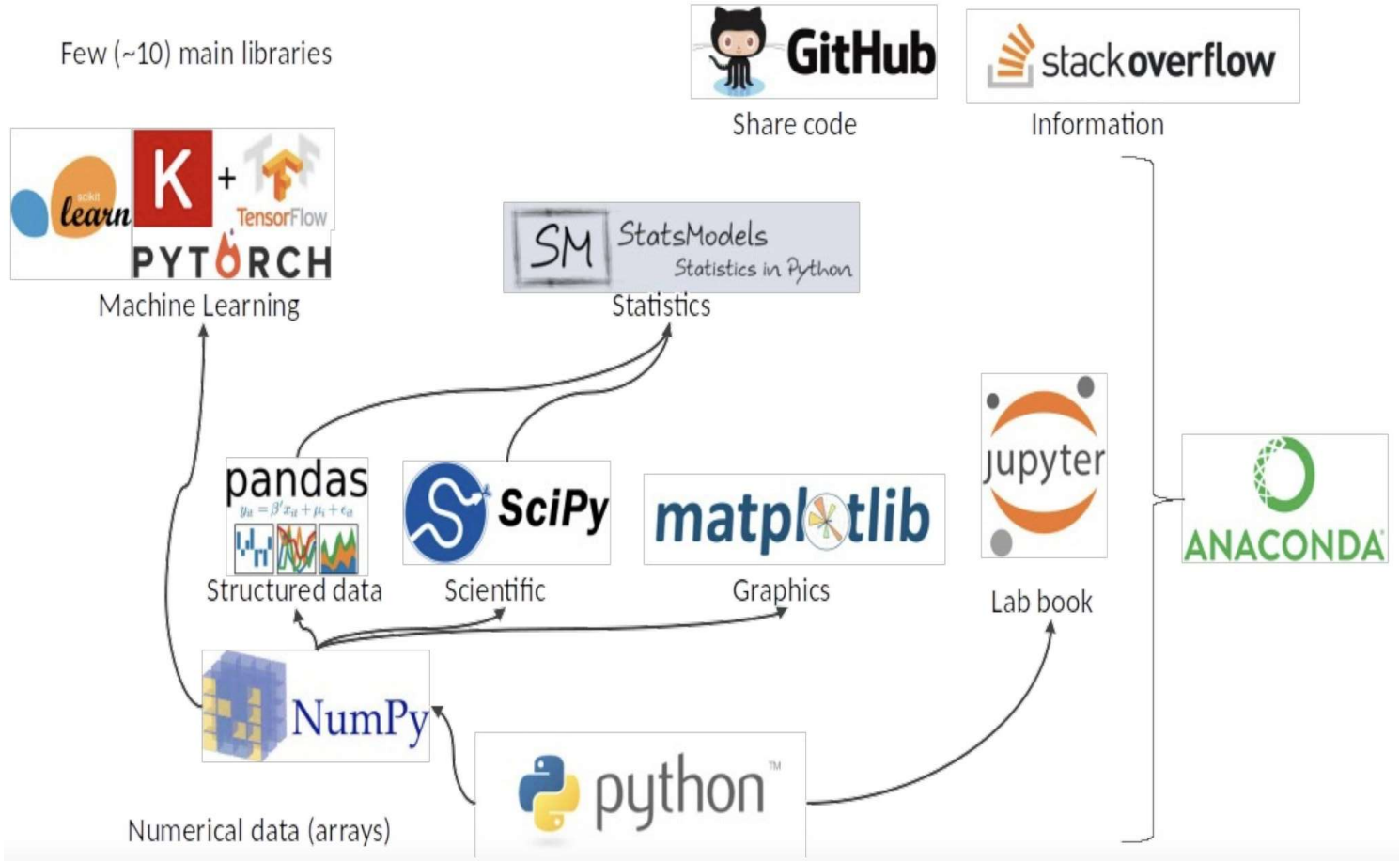


Image by python.org

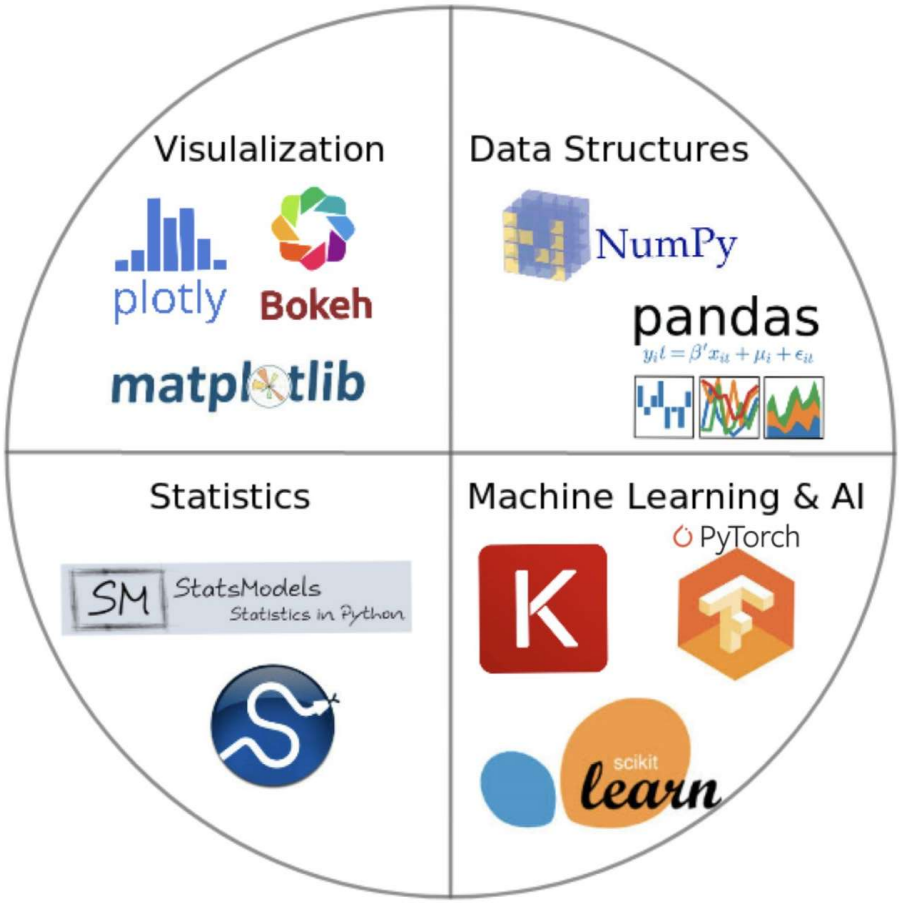
Python: Ecosystem

Image by medium.com



Python: Ecosystem

Image by duchesnay.github.io



Python: IDLE

- Python comes with integrated development environment, IDLE.
- It is a set of tools that make writing easier.
- There are two modes for working on IDLE:

1. Interactive
2. Script

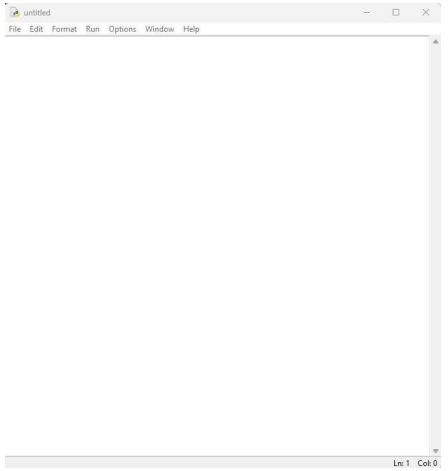
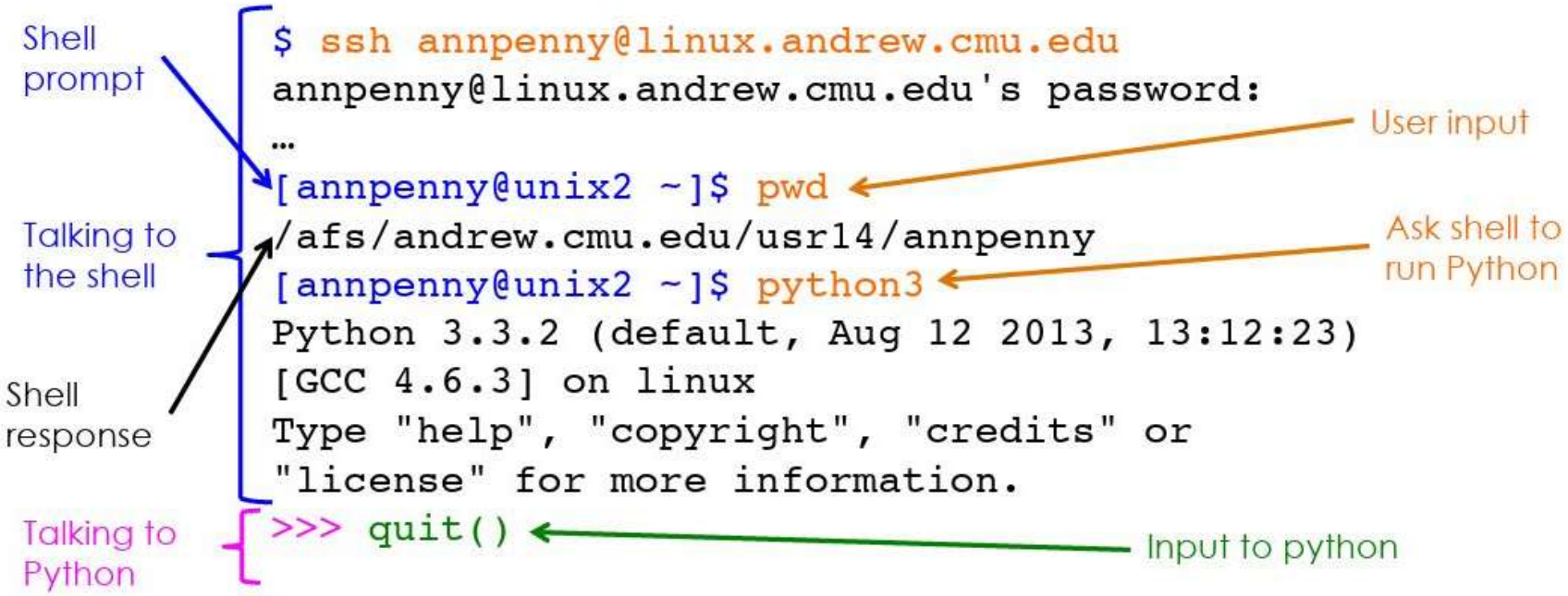


Image by python.org

Python: Command Line Interface

Be aware of the difference between *"talking to the shell"* and *"talking to Python"*



Python: Generating Error

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
<ipython-input-1-9d0462a5b07c> in <module>()
      1 from errors_01 import favorite_ice_cream
----> 2 favorite_ice_cream()

/Users/jhamrick/project/swc/novice/python/errors_01.pyc in favorite_ice_cream()
      5         "strawberry"
      6     ]
----> 7     print ice_creams[3]

IndexError: list index out of range
```

Errors in Python have a very specific form, called a traceback.

Reading error messages: Identify the following pieces of information about it:

1. How many **levels** does the traceback have?
2. What is the **file name** where the error occurred?
3. What is the **function name** where the error occurred?
4. On which **line number** in this function did the error occur?
5. What is the **type of error**?
6. What is the **error message**?

Python: Syntax Highlighting

Text Editors:

Sublim

VS Code

Notepad

Atom

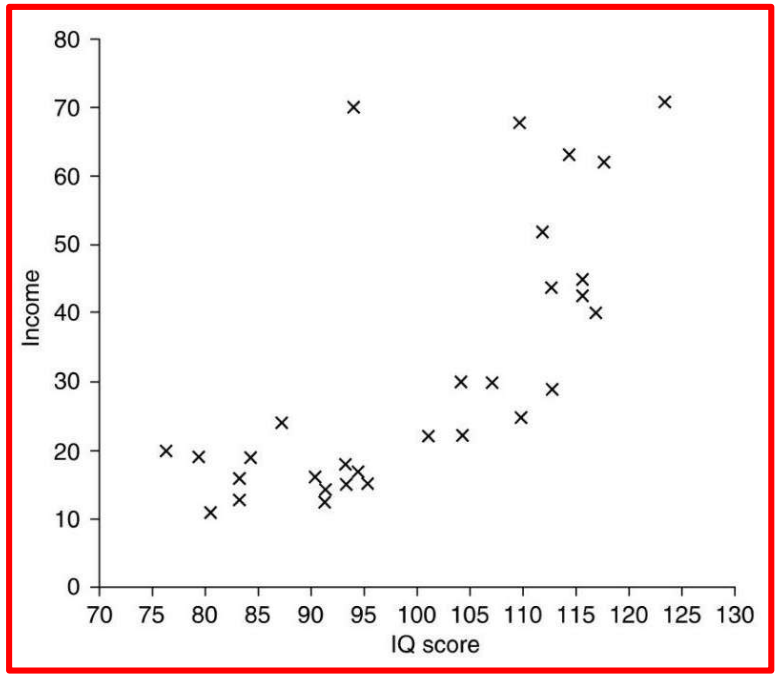
IDLE, etc

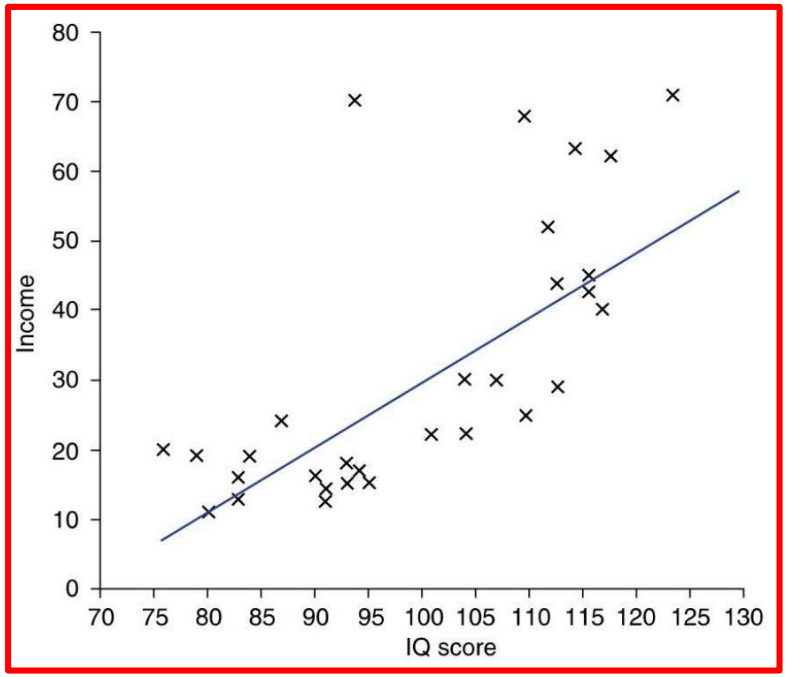
- Read/Identifies the file
- Interpreter will quickly understands the words
- Based on different word and interpretations, in-built functions, strings, numbers, library etc different color will be displayed for different editor settings.

```
def __init__(self, parent: QtGui.QTextDocument) -> None:
    super().__init__(parent)

    # Multi-line strings (expression, flag, style)
    self.tri_single = (QtCore.QRegExp("'''"), 1, STYLES['string2'])
    self.tri_double = (QtCore.QRegExp('"""'), 2, STYLES['string2'])

    rules = []

    # Keyword, operator, and brace rules
    rules += [(r'\b%s\b' % w, 0, STYLES['keyword'])
        for w in PythonHighlighter.keywords]
    rules += [(r'%s' % o, 0, STYLES['operator'])
        for o in PythonHighlighter.operators]
    rules += [(r'%s' % b, 0, STYLES['brace'])
        for b in PythonHighlighter.braces]

    # All other rules
    rules += [
        # 'self'
        (r'\bself\b', 0, STYLES['self']),

        # 'def' followed by an identifier
        (r'\bdef\b\s*(\w+)', 1, STYLES['defclass']),
        # 'class' followed by an identifier
        (r'\bclass\b\s*(\w+)', 1, STYLES['defclass']),

        # Numeric literals
        (r'\b[+-]?[0-9]+[lL]?\b', 0, STYLES['numbers']),
        (r'\b[+-]?0[xX][0-9A-Fa-f]+[lL]?\b', 0, STYLES['numbers']),
        (r'\b[+-]?[0-9]+(?:\.[0-9]+)?(?:[eE][+-]?[0-9]+)?\b', 0, STYLES['numbers']),

        # Double-quoted string, possibly containing escape sequences
        (r'"[^"\\]*(\\.[^"\\]*)*"', 0, STYLES['string']),
        # Single-quoted string, possibly containing escape sequences
        (r"'[^'\\]*(\\.[^'\\]*)*'", 0, STYLES['string']),

        # From '#' until a newline
        (r'#[^\n]*', 0, STYLES['comment']),
    ]

    # Build a QRegExp for each pattern
    self.rules = [(QtCore.QRegExp(pat), index, fmt)
        for (pat, index, fmt) in rules]
```

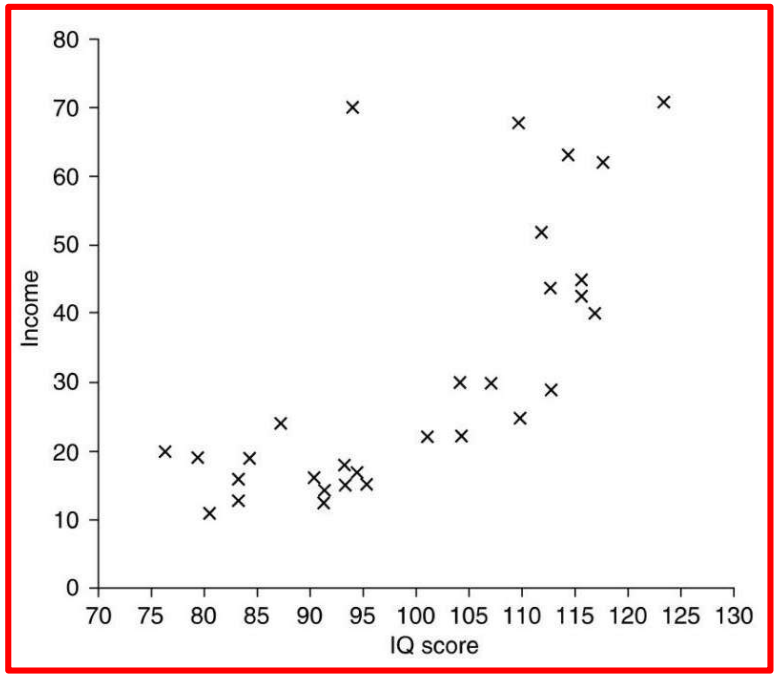
Applications: Machine Learning

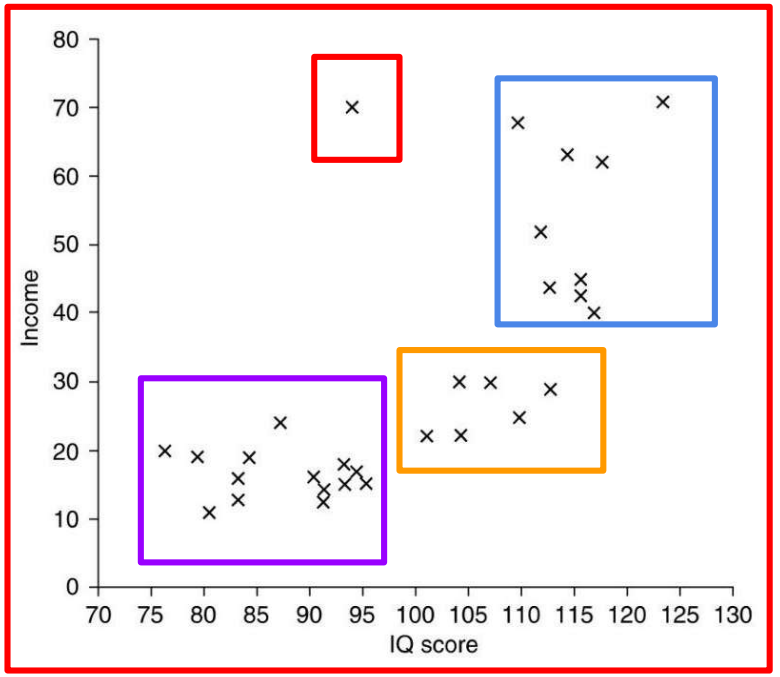
Regression


Income
IQ score
0
10
20
30
40
50
60
70
80
70 75 80 85 90 95 100 105 110 115 120 125 130


Income
IQ score
0
10
20
30
40
50
60
70
80
70 75 80 85 90 95 100 105 110 115 120 125 130

Clustering


Income
IQ score
0
10
20
30
40
50
60
70
80
70 75 80 85 90 95 100 105 110 115 120 125 130


Income
IQ score
0
10
20
30
40
50
60
70
80
70 75 80 85 90 95 100 105 110 115 120 125 130

Questions

- What is the difference regression and clustering?
- What is the difference between regression and interpolation?
- Given the points in 2D, how will you fit a line for interpolation?
- Given the points in 2D, how will you fit a line for regression?
- How will you make computers understand to cluster the negative numbers [-2, -1, 1, 2] into groups of -ve and positive numbers?