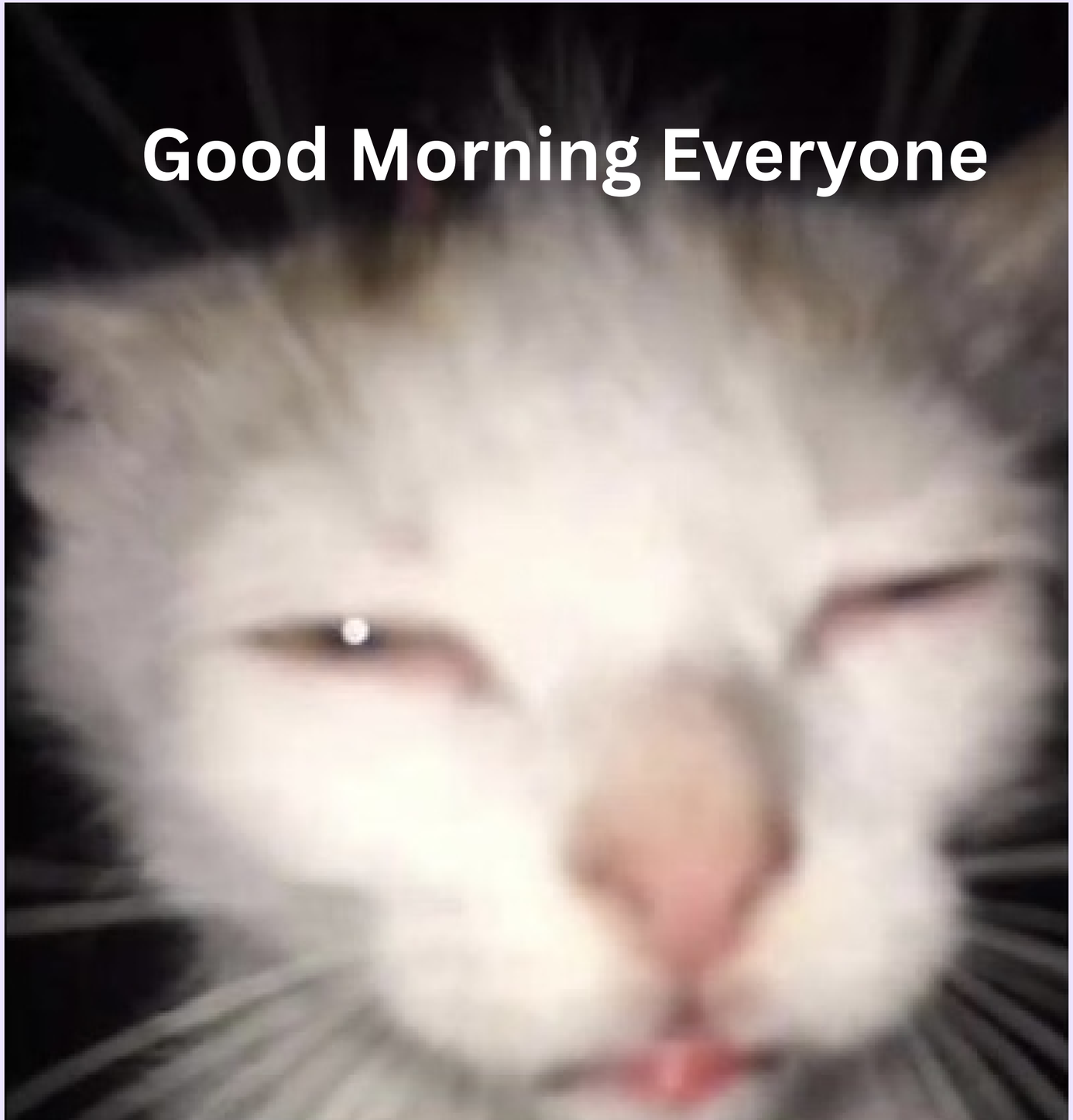
Intro to Python

FiML CLUB
National Forensic Sciences University


Good Morning Everyone

Who am I?

I am Bhaskar Bhar

I am in BTECH sem 7

I am an open source enthusiast, I have done contributions in nDPI and Zeek.

Linkedin: Just search Bhaskar Bhar(You will find a odd looking guy with specs wearing red tshirt)

Github username: bhaskarbhar

Sooo, What is Python??

Its just a programming language which is run by its own virtual environment, thats all.

What are the features of Python??

Its an interpreted programming language, that means line by line code is executed

Its Open Source, that is you can build your own libraries

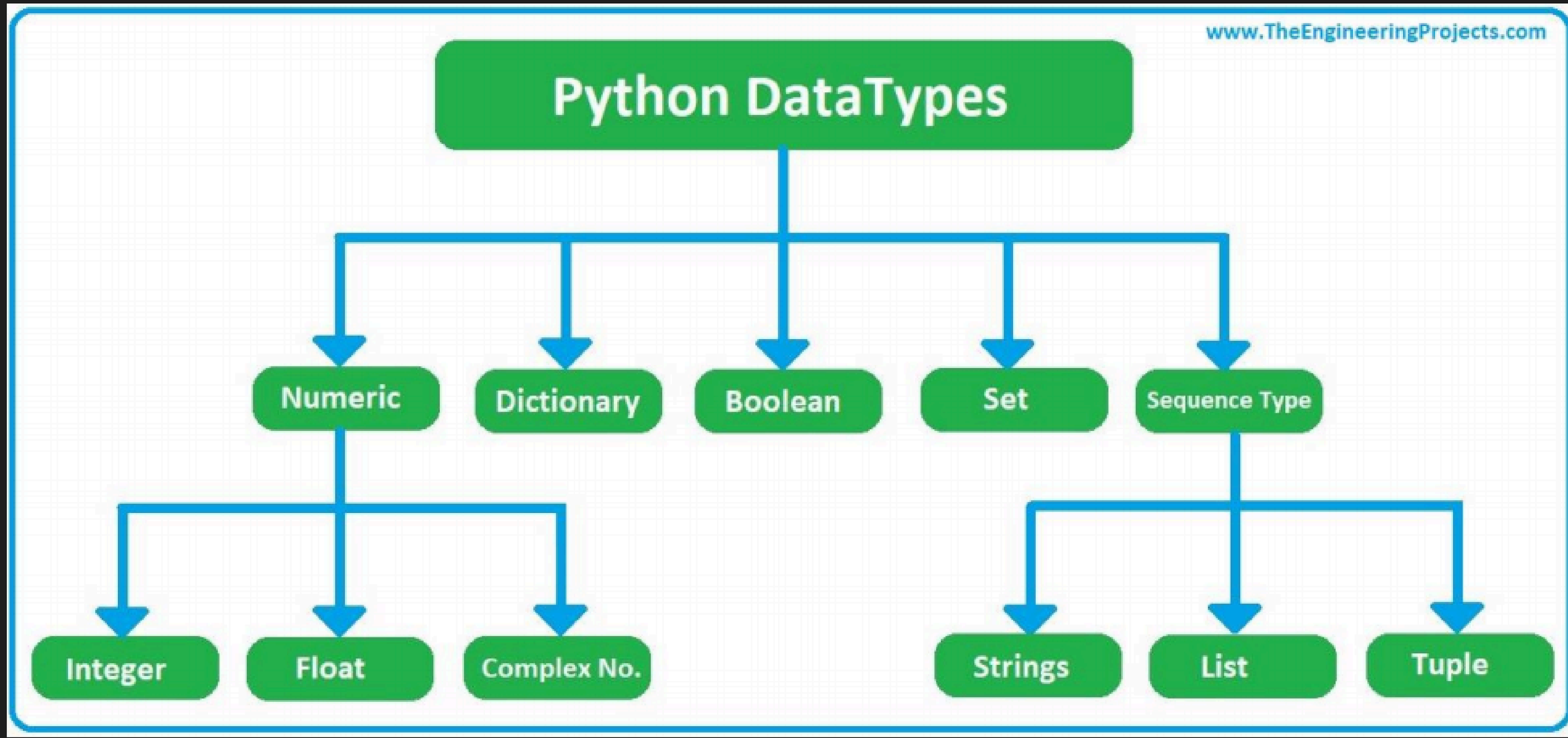
Huge number of libraries for nearly every purpose

Easy to code, even chatgpt answers in python first if you dont mention to use other language

Variables

- There are a certain rules that we have to keep in mind while declaring a variable:
- The variable name cannot start with a number. It can only start with a character or an underscore.
- Variables in python are case sensitive.
- They can only contain alpha-numeric characters and underscores.
- No special characters are allowed. Python allows you to assign a single value to several variables simultaneously.
- For example – a = b = c =1
- Here, an integer object is created with the value 1, and all three variables are assigned to the same memory location.
- You can also assign multiple objects to multiple variables.
- For example – a,b,c = 1,2,"john".

Variables are symbolic names that act as containers or labels for values stored in the computer's memory.


www.TheEngineeringProjects.com
Python DataTypes
Numeric
Dictionary
Boolean
Set
Sequence Type
Integer
Float
Complex No.
Strings
List
Tuple

Integer

12

Integers are used to represent whole number values.

Ex, x = 10

Float

1.23

Float data type is used to represent decimal point values.

Ex, x = 2.5

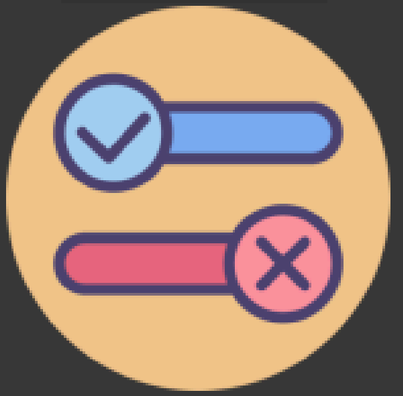
Complex

$a+ib$

Complex numbers are used to represent imaginary values.

Ex, x = 2 + 3j

Boolean



Boolean is used for categorical o/p, since the o/p of Boolean is either true or false.

Ex. X = 1> 2

Python Operators

- Arithmetic operators
- Assignment operators
- Comparison operators
- Logical operators
- Identity operators
- Membership operators
- Bitwise operators

Python Arithmetic Operators

X = 10 & Y = 6

Operator	Name	Example	Output
+	Addition	x + y	16
-	Subtraction	x - y	4
*	Multiplication	x * y	60
/	Division	x / y	1.66
%	Modulus	x % y	4
**	Exponentiation	x ** y	1000000
//	Floor division	x // y	1

Python Assignment Operators

Operator	Example	Same As
=	x = 5	x = 5
+=	x = +3	x = x + 3
-=	x -= 3	x = x - 3
*=	x *= 3	x = x * 3
/=	x /= 3	x = x / 3
%=	x %= 3	x = x % 3
//=	x //= 3	x = x // 3

Python Comparison Operators

Operator	Name	Example
==	Equal	x == y
!=	Not equal	x != y
>	Greater than	x > y
<	Less than	x < y
>=	Greater than or equal to	x >= y
<=	Less than or equal to	x <= y

Python Logical Operators

Operator	Description	Example
and	Returns True if both statements are true	x < 5 and x < 10
or	Returns True if one of the statements is true	x < 5 or x < 4
not	Reverse the result, returns False if the result is true	not(x < 5 and x < 10)

Python Identity Operators

Operator	Name	Example
is	Returns True if both variables are the same object	x is y
is not	Returns True if both variables are not the same object	x is not y

Python Membership Operators

Operator	Name	Example
in	Returns True if a sequence with the specified value is present in the object	x in y
not in	Returns True if a sequence with the specified value is not present in the object	x not in y

Python Bitwise Operators

Operator	Name	Example
&	AND	Sets each bit to 1 if both bits are 1
|	OR	Sets each bit to 1 if one of two bits is 1
^	XOR	Sets each bit to 1 if only one of two bits is 1
~	NOT	Inverts all the bits
<<	Zero fill left shift	Shift left by pushing zeros in from the right and let the leftmost bits fall off
>>	Signed right shift	Shift right by pushing copies of the leftmost bit in from the left, and let the rightmost bits fall off

Python Branching

- If statement
- Elif Statement
- Else statement

For this we will code!!!!

Please open your IDE

The if-else statement

- The if-else statement provides an else block combined with the if statement which is executed in the false case of the condition.
- If the condition is true, then the if-block is executed. Otherwise, the else-block is executed.

```
if condition:
   #block of statements
else:
   #another block of statements (else-block)
```

```
num = int(input("enter the number?"))
if num%2 == 0:
   print("Number is even...")
else:
   print("Number is odd...")
```

Output

Python Iteration

- **While loop**
- **For loop**
- **Continue and Break**

For this we will code!!!!
Please open your IDE

While Loop

- The Python while loop allows a part of the code to be executed until the given condition returns false. It is also known as a pre-tested loop.
- It can be viewed as a repeating if statement. When we don't know the number of iterations then the while loop is most effective to use.

```
while expression:
    statements
```

For Loop

- A for loop is used for iterating over a sequence (that is either a list, a tuple, a dictionary, a set, or a string).
- This is less like the for keyword in other programming languages, and works more like an iterator method as found in other object-orientated programming languages.
- With the for loop we can execute a set of statements, once for each item in a list, tuple, set etc.

- **1. Continue Statement -** When the continue statement is encountered, the control transfer to the beginning of the loop. Let's understand the following example.
- **2. Break Statement -** When the break statement is encountered, it brings control out of the loop.

BASIC SEQUENTIAL DATA TYPES

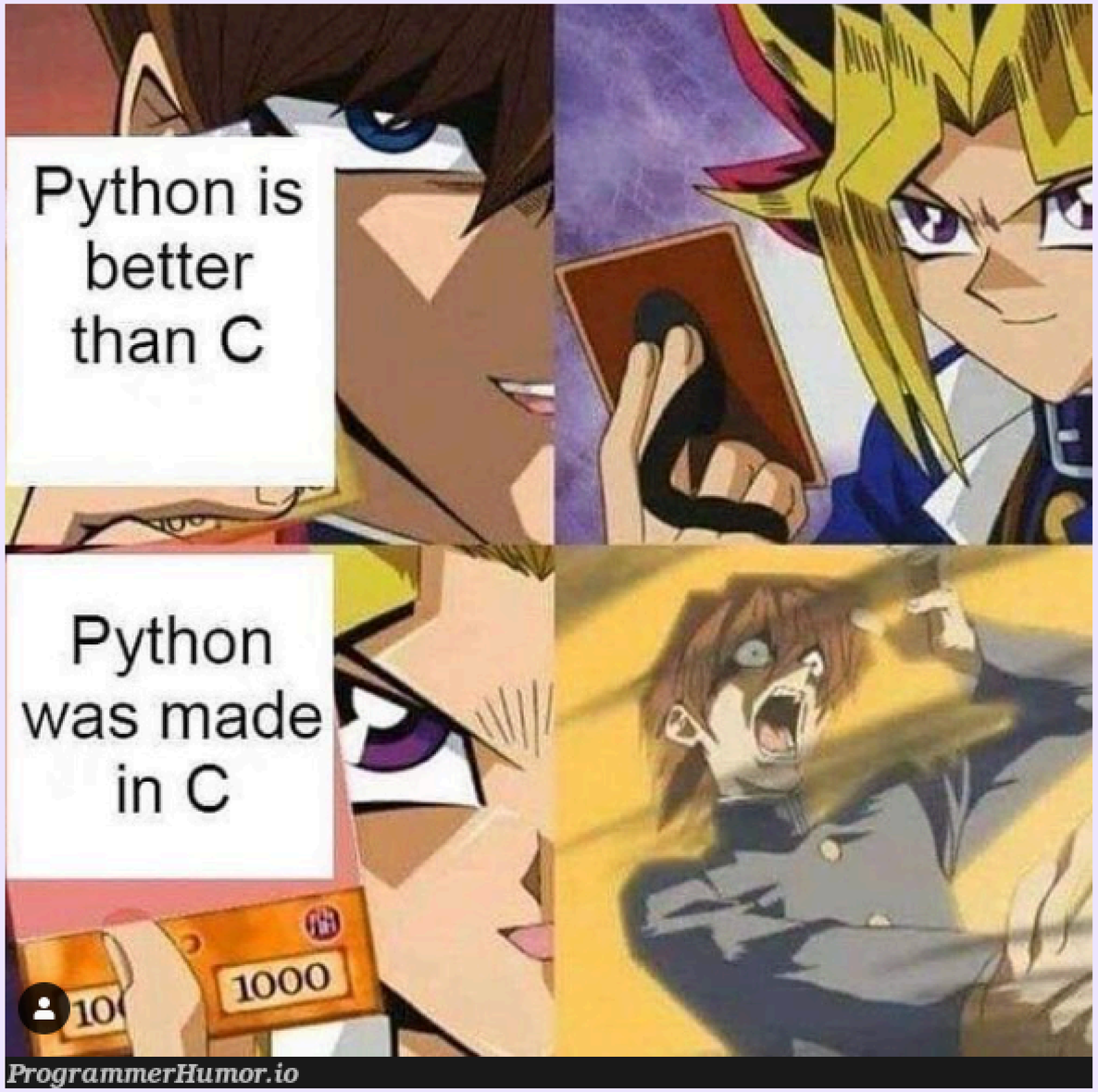
Strings

List

Tuple

Dictionary

For this we will code!!!!
Please open your IDE



Strings

- String is a sequence of characters.
- String may contain alphabets, numbers and special characters.
- Usually strings are enclosed within a single quotes and double quotes.
- Strings is immutable in nature.

- List is a sequence of values called items or elements.
- The elements can be of any data type.
- The list is a most versatile data type available in Python which can be written as a list of comma-separated values (items) between square brackets.
- List are mutable, meaning, their elements can be changed.

- **Tuples** are very similar to lists, except that they are immutable (they cannot be changed).
- They are created using **parentheses**, rather than square brackets.

- Python dictionary is an unordered collection of items.
- While other compound data types have only value as an element, a dictionary has a **key: value** pair.
- Dictionaries are optimized to retrieve values when the key is known.

Libraries

for ML and data Analysis

FML CLUB
National Forensic Sciences University

random

The **random** module in Python is part of the standard library.

It is used to generate **pseudo-random(We can give seeds)** numbers and perform random operations such as shuffling, selecting random items, or generating distributions.

random.random()
Returns a float between 0.0 and 1.0.

random.randint(a, b)
Returns a random integer between a and b (inclusive).

random.choice(seq)
Returns a random element from a list, tuple, or string.

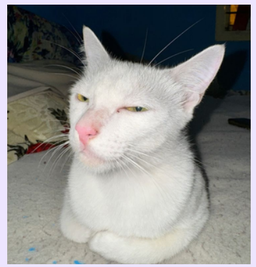
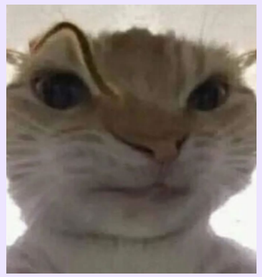
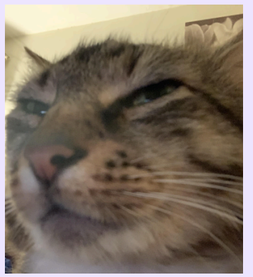
Numpy(Numerical Python)

To understand this module lets understand what is an array? And why its different from list?

Array is a data type allowing only homogenous data or data of same data type

[ ,  ,  ,  , ]

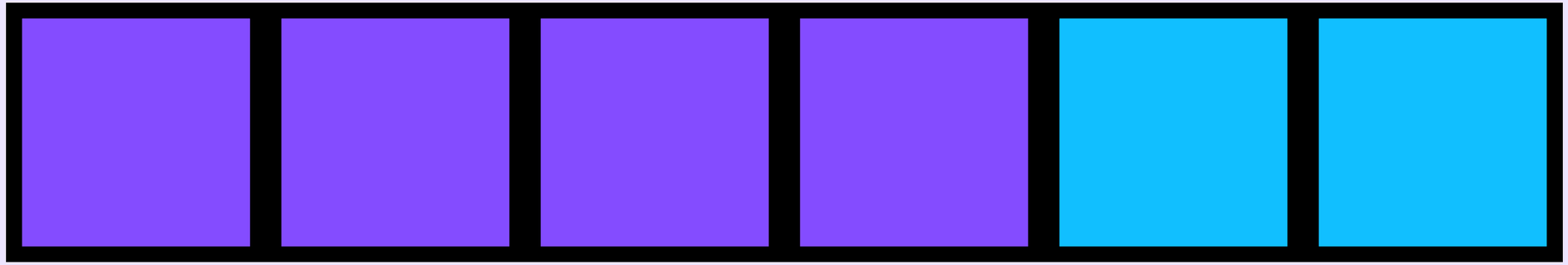
Whereas List allows heterogenous data

[ ,  ,  ,  , ]

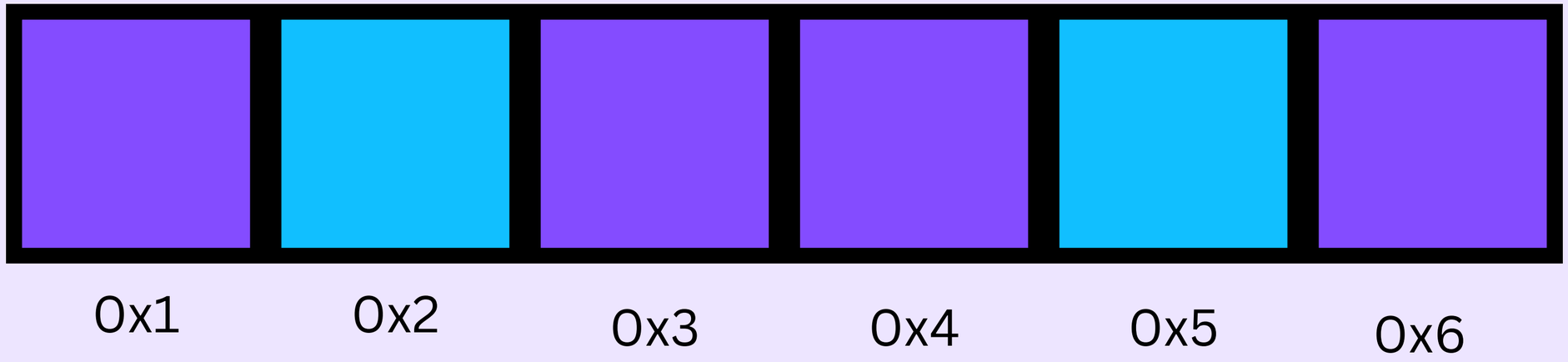
Why numpy arrays are faster??

Reason 1:

Contiguous memory allocation

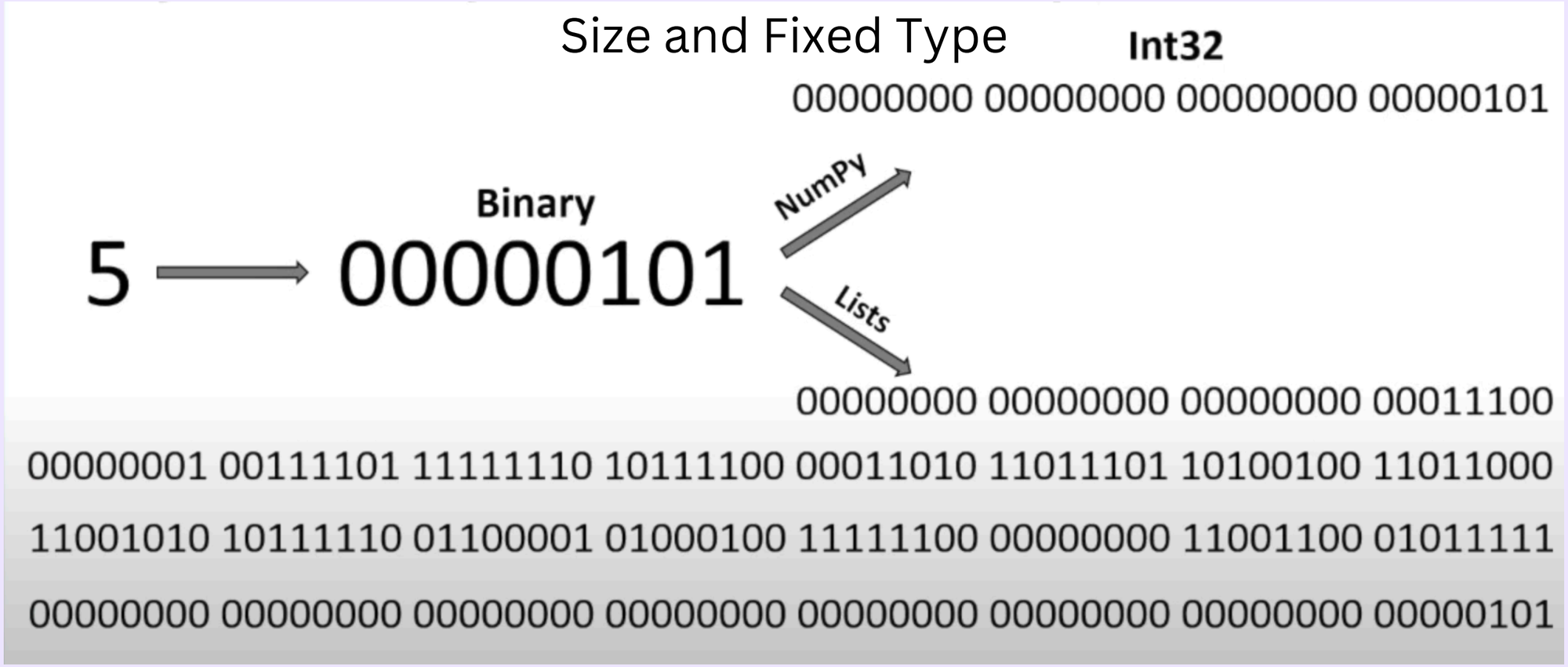


In lIst:



Reason 2:

Size and Fixed Type



List actually maintains 4 things:

Size

Reference Count

Object Type

Object Value

numpy.array() → Create an array
Used to create a NumPy array from a list or tuple.

ndim → Number of dimensions
Tells how many dimensions (axes) an array has.

shape → Dimensions of the array
Returns a tuple showing the rows × columns (for 2D) or sizes along each axis.

size → Total number of elements
Gives the count of all elements in the array.

pandas

Pandas is a powerful open-source Python library used for data analysis and manipulation.

It is built on top of NumPy, and provides high-level data structures and functions designed to make working with structured/tabular data easy.

The two main data structures in Pandas are:

- **Series** → one-dimensional labeled array (like a column).
- **DataFrame** → two-dimensional labeled data (like an Excel table).

Why Pandas is Important in Data Analysis / DS?????

- Data Cleaning → handle missing values, duplicates, inconsistent data.
- Exploration → summary statistics (.mean(), .describe()).
- Manipulation → filtering, grouping, merging datasets.
- Integration → works with CSV, Excel, SQL, JSON, etc.

pd.Series() → Create a Series
A one-dimensional labeled array.

pd.DataFrame() → Create a DataFrame
A two-dimensional labeled table.

df.head() → View first rows of data
Shows the first 5 rows (default), useful for quickly inspecting data.

df.shape → Shape of the DataFrame
Returns a tuple (rows, columns).

df.columns → List of column labels
Gives the names of all columns.

matplotlib

Matplotlib is the most widely used data visualization library in Python.

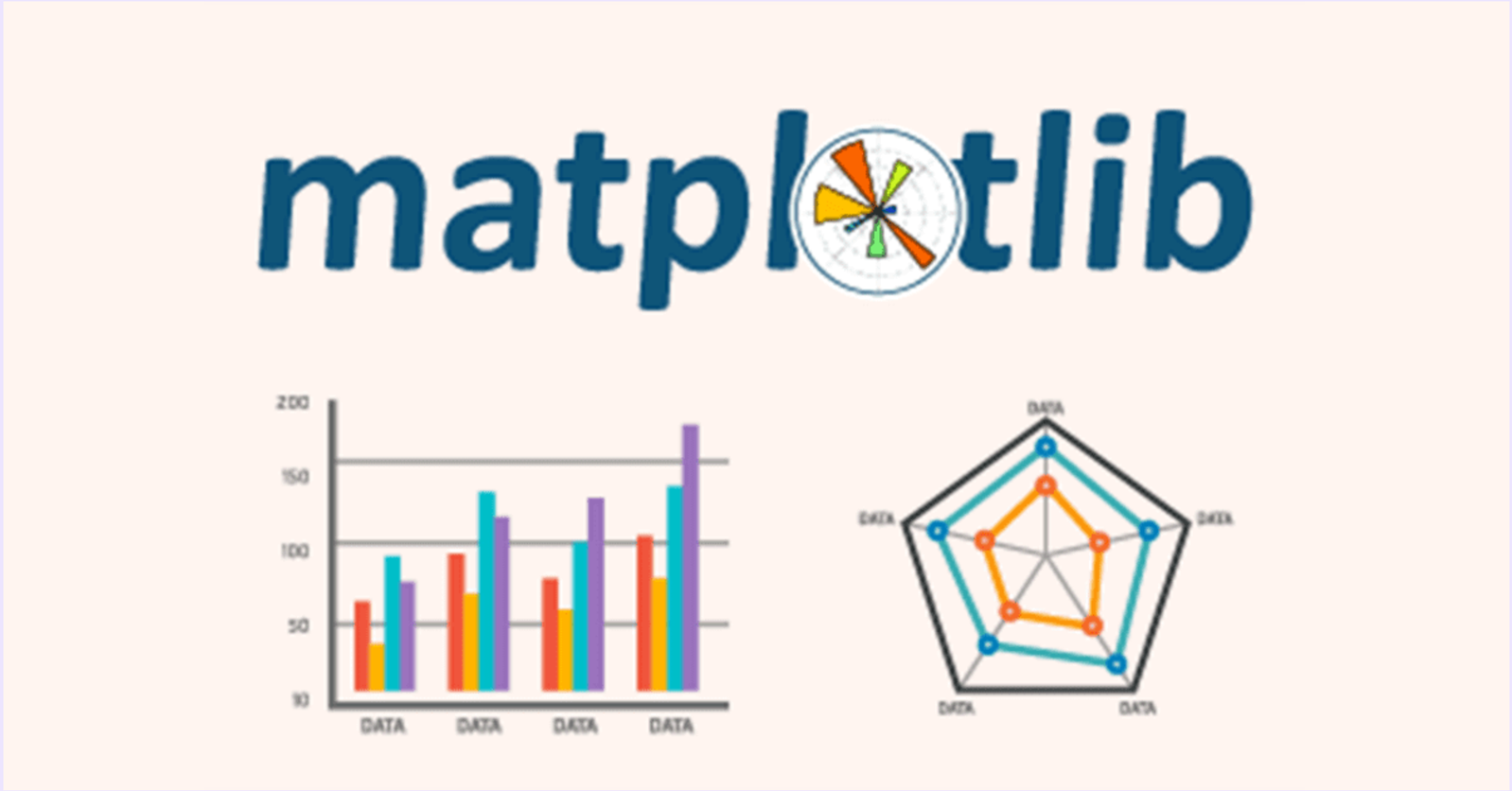
We will loook examples for the submodule for pyplot

Basic Pyplot Usage

```
import matplotlib.pyplot as plt

# Sample data
x = [1, 2, 3, 4, 5]
y = [2, 4, 6, 8, 10]

# Line plot
plt.plot(x, y, label="Line")
plt.xlabel("X-axis")
plt.ylabel("Y-axis")
plt.title("Basic Line Plot")
plt.legend()
plt.show()
```



Google Colab Notebook Link:

https://colab.research.google.com/drive/1Bf0MB_0mHspX2hHPcH583A5AggrquJyW?usp=sharing

Or

https://shorturl.at/3pYgn


THE END
OF THE WORLD WIDE PRESENTATION
makeameme.org