**Python**

Introduction to python:

* It is an object oriented programming language and high level & scripting language.
* It is introduced by Guido Van Rossum in 1991.
* Python uses **.py** extension
* Now we are using **3.13.0 version** to do python
* It will have libraries, packages, modules, frameworks etc.
* It will execute the code by line by line. And it will give fast o/p and it will show the error in which line there is an error
* Python is used in various domains like
* Web Development
* Data Science
* Machine Learning
* Automation
* Game Development

KEY FEATURES OF PYTHON

* Easy to learn: The python has a clean syntax and the language we using in the python is simple
* Interpreted language: The code executes line-by-line and there is no need to compile before running the code
* High-level-language: While using python we focus on problem solving rather than technical details
* Dynamic Typing: There is no need to declare variable while using python, python determines type automatically
* Cross-Platform: If we write code once in python then it can run anywhere
* Open-source: Python is free to use and easy to modify data

Rules in python:

* Starting letter should be in alphabets or underscore
* The first letter of a variable, function or class must be one of letters like(A-Z) or (a-z)
* Don’t start with digit or symbols
* Variables are case sensitive

**DATATYPES**

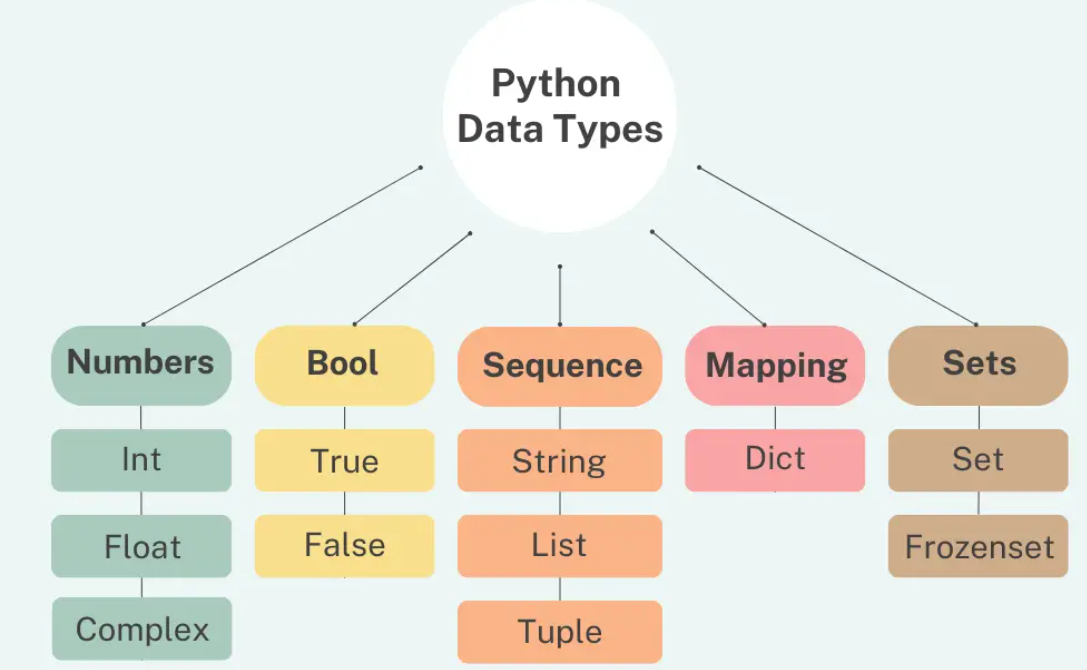
**Datatypes:** It is used to specify what type of data has to be stored into the variable.

* Without datatype we cannot store the data into the variable
* Python support the dynamic datatype i.e. at the time of execution of program datatype of a variable will be decided based on the data which is assigned to that variable.

Ex: X=10

There many types in datatypes:

1. Numeric datatype 🡪 Int, Float, Complex
2. Sequence datatype 🡪 String, List, Tuple
3. Boolean datatype 🡪 True (or) False
4. Mapping datatype 🡪 Dictionary
5. Set datatype 🡪 Set, Frozen set
6. Binary datatype 🡪 Bytes, Memory View, Byte array
7. None
8. User defined datatypes



**Numeric datatype 🡪 Int, Float, Complex**

Numerical Datatype: Numerical datatypes represent values that can be used in mathematical operation.

There are three types:

1. Integer Datatype
2. Float Datatype
3. Complex Datatype

Integer Datatype(int):

* It is a whole number that can be positive values, negative values, or zero
* It does not take fractional or decimal components
* Integers are used in counting, mathematical operation

For example: …..-2,-1,0,1,2…..

* It will take (-infinity to +infinity) including with Zero

Eg: Where a=20 b=7

a=20  
b=7  
print(a+b)

output: 27

Float Datatype(float):

* It is used to represent the real numbers with decimal points or in fractional numbers.

**Eg: 0.1, 2.4, ½ = 0.5…..etc.**

* It is used in real life to see the measurement like distance, temp, calculations requiring dynamic scaling of numbers & values
* In Python floats use 64-bit double precision (IEEE 754 standard)
* IEEE stand for Institute of Electrical and Electronics Engineers.
* 1 bit for the sign
* 11 bits for the exponents
* 52 bits for the fraction

Eg: where a = 4.7

b = 1.7

a=4.7  
b=1.9  
print(a+b)

Output: 6.6

Complex Datatype:

* In complex datatype it represents complex numbers it contains like one real number and one imaginary number

Eg: a + bj 🡺 8+3j (or) 8-3j

Where ‘a’ is real part and ‘b’ is imaginary part and

‘j’ it represents the square root of “-1” (imaginary unit)

* It is mostly used in maths, signal etc.

Where a=3+4j b=5+3j

a=3+4j  
b=5+3j  
print(a+b)

Output: 8+7j

Where a=7+9j b=9-4j

a=7+9j  
b=9-4j  
print(a-b)

Output: -2+13j

**Sequence datatype 🡪 String, List, Tuple**

Sequence datatype means: In sequence datatypes we are used to store multiple values in an ordered manner. Sequence will support indexing, slicing and iteration.

**String(str):** It is an array of bytes representing Unicode characters

It is a collection of one or more characters putting them in a single (‘ ’) , double(“ ”), triple (‘’’ ‘’’) quotes.

* (‘ ’), (” ”) are used for the single line character
* (‘’’ ‘’’) are used for the multiple line character

Ex: #string

Single quotes

print('sravya')

Output: sravya

Double quotes

print("chintu")

Output: chintu

triple quotes

print('''chintu''')

Output: chintu

**List:** It is most commonly used datatype and it allows us to store the data in multiple items in a single variable and it is a **mutable**. We can modify the data.

It’s just like an array, It can add the data or delete the data in list and we use square brackets [ ] in list.

a=[1,2,3,4]  
result=a  
print(result)

Output: [1, 2, 3, 4]

a=['sravya','LP','kallu','sageetha']  
result=a  
print(result)

Output: ['sravya', 'LP', 'kallu', 'sageetha']

a=['sravya',7,"8.58"]  
result=a  
print(result)

Output: ['sravya', 7, '8.58']

**Tuple:** It is same like list but List is a mutable and Tuple is an **Immutable** it is used to store group of elements in single entity.

For tuple we use parentheses brackets ().

a=(6,"sravya",15.25)  
result=a  
print(result)

Output: (6, 'sravya', 15.25)

**Boolean datatype 🡪 True (or) False**

Boolean datatype: In Boolean function it will returns True (or) False, yes (or) no, 0 (or) 1. It is used to check the statement whether it is correct (or) wrong.

a=9>5  
result=a  
print(result)

Output: True

**Mapping datatype 🡪** **Dictionary**

Mapping datatype: It is also a collection of elements. It can store the values like key values and it will store as a pair value. And it is written in curly brackets. It is mutable.

Syntax: key : value

a={"rollno":58,"name":'sravya'}  
result=a  
print(result)

Output: {'roll no': 58, 'name': 'sravya'}

**Set datatype 🡪 Set, Frozen set**

Sets: It can define group of unordered elements. We use curly brackets in it. In set there is a drawback the output will come in unsequence order. We can add, remove the values. It is mutable.

a={10,20,30,40,50}  
result=a  
print(result)

Output: {50, 20, 40, 10, 30}

**Union (|):** The union combines all the elements from set1 and set2, but it removes duplicates (since sets only store unique elements).

**Syntax**: set1 | set2 or set1.union(set2)

**Intersection (**&**)**: The intersection finds the common elements between set1 and set2

**Syntax**: set1 & set2 or set1.intersection(set2)

Frozen set: It is same like set in set we can change the values but in frozen set we can’t add, remove or discard the values it is immutable.

* In this we will use the **“#function”**, Frozen set it is used for fixed values.
* When we give values for the values they take some unique I’d with that I’d it will save the values. By using this #function we can find the values.
* It has high speed & it will not allow the duplicates values.

a={5,6,7,8,9}  
result=a  
print(result)

Output: {5, 6, 7, 8, 9}

NONE: It is a special in python it indicates an object that will not contain any kind of values.

Example: X

**Binary datatype 🡪 Bytes, Byte array, Memory View**

Binary datatype: It is a data type used to store data as a sequence of bytes like images, files, audios, videos.

**Bytes:**

* It is immutable sequence in byte datatype
* In byte datatype it stores the values from 0 to 256
* If we take more then 256 we get the values error

**Byte array:**

* It is mutable sequence
* To change the values from the byte we will use byte array

a=[1,2,3,4,5]  
b=bytes(a)  
print(type(b))  
forain(b):  
 print(a)

Output: 1

2

3

4

5

**Memory View:** It is an object provides a view of a memory buffer without duplicating the underlying data. This view allows you to manipulate the content of the buffer directly while keeping the original data object intact.

A memory efficient way to access binary data without copying it.

Allow slicing & manipulation of the underlying binary data.

**NONE:** It is a null value

Ex: none = null

x=null

Output: null