[**https://drive.google.com/drive/folders/1W3\_DnBwTnA84HfTdKLkbJ79VUDHJYeus?usp=sharing**](https://drive.google.com/drive/folders/1W3_DnBwTnA84HfTdKLkbJ79VUDHJYeus?usp=sharing)

**MYSQL PASSWORD: GIRIJA@123**

**DAY 1&2:**

**Features of string**

* inmutable
* order
* slicing/indexing
* allow duplicates

Indexing:

 a = 'Nutella is tasty'

a[3]

‘e’

Slicing:

a[0:3]

Jumping:

A[0:10:2]

**IMPORTANT CODE OF PYTHON:**

To know the code of the specified string r int r something:

10/06/2025

**List []**

* li=[objects]
* list((objects))
* list is the mutable
* In list we can add, delete, edit the objects in it

**To find the length of the variable:**

Len(var)

List indexing:

**Tuple()**

* Tup=(objects)
* Tuple((objects))
* Tuple cannot be edit r delete
* In the place where we want it should not be changeable, we can use tuple
* We can do indexing slicing
* Tuple is a immutable
* Eg: product id, employee id which should not be edited r deleted
* To edit tuple we need to type caste to list {a=list(a)} and then can be edited

To get the data of the list which is in another tuple **[Double indexing]**

tup3[5][2]

**.count**

To count how many object in a tuple {mytuple.count(‘one’)}

**.index**

To find the position of the object {mytuple.index(‘three’)}

**Sorted**

To sort the object in ascending order{sorted(tuple1)}

In descending order {sorted(tuple1,reverse=true)}

**Sets{}**

* Myset={1,11,1,2,1,5}

Myset =set((1,11,1,2,1,5))

Output = {1, 2, 5, 11}

* Won’t allow duplicates
* Same value can be taken once
* We cannot index the set
* Result wont come in the same order of the data we gave
* We cant use append here
* We can use pop here but won’t delete the final one alone. It may delete the any object in the set
* We can use remove myset.remove(1)r(‘dragon’)
* We can use discard myset.discard(1)r(‘dragon’) we can even remove the object which are not actually in the set. In case if we don’t know whether that object is there r not but have to delete
* We can use add to add one data myset.add(100)
* To add multiple data use update myset.update([10,20,30])
* To clear all the data in the set myset.clear()
* To delete the set del myset
* To add two set use union A.union(B)
* To add more than two set use , in () A.union(B,C)
* To know the common data in two set A.intersection(B)
* To know the value of the one set which doesn’t have common data A.difference(B)

**Dictionary{}**

* dic = {'a':'one','b':'two','c':'three'}
* Key value pair
* In dic no indexing
* We can pick the value by its key dic['c']
* To edit the value of the id or add new mydic1.update({'ID':5000,'loc':'comibatore'}) (we can add multiple value here)
* To edit r add we can use this too mydic1["DOB"]=2005 (for the nonexisting we can add new id inside[] and value after =(we can add only one value here
* To delete any item mydic1.pop('loc') use pop
* To delete the last item mydic1.popitem()
* To delete we can use del mydic1['DOB']
* To clear the all the values of the dic mydic1.clear()
* To delete the variable del mydic1

**Function:**

* Print()
* Len()
* Type()

These the default r build in function

**User defined function**

Use def def xyz(a,b):

c=a+b

print(c)

* This is called user defined function
* This will run only in that page and not in other
* Once the function is defined you can use it in at any time
* This type of function is used for the multiple use of the single work
* To call the function xyz(200,300) the output will be 500
* To save the output in a variable don’t use print() instead use return

def xyz(a,b):

c=a+b

return c

* Return is not the replacement of print. Return can be used only in the function and not in any other place

**LAMBDA:**

q=lambda a,b:a+b

* Lambda will convert many line code to single line
* Lambda can be used inside a function
* To compress the long code into simple code use lambda
* Lambda can be used for the place where function is not named but yet used
* Eg: list(map(lambda n:n+9,seq))
* Output will be done according to the seq value

**MAP:**

* Map will combine the function and seq

def square(n):

return n+n

seq = [1,3,4,5,6,7]

list(map(square,seq))

* In map function should be there
* We can use list r tuple to view the output

**FILTER:**

* Filter will show the value
* Eg: list(filter(lambda x:x%2==0,seq))

In this we can get the value where it is true

Output will be [4,6]

**Outliers**

To find the outliers use the formula

[Q1-1.5(IQR), Q3+1.5(IQR)]