

Vidyavardhini’s College of Engineering and Technology

Department of Computer Engineering

|  |
| --- |
| Experiment No. 3 |
| To explore basic data types of Python like strings, lists, dictionaries and tuples |
| Date of Performance: 31/01/2024 |
| Date of Submission: 14/02/2024 |



Vidyavardhini’s College of Engineering and Technology

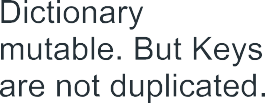
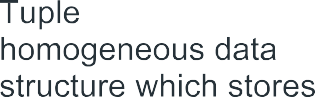
Department of Computer Engineering



























































**Code:**

#list

l=["vcet",41,5.6,"vaidehi"]

print(l)

l.append(45)

print(l)

l.insert(2,34)

print(l)

l.pop()

print(l)

l.reverse()

print(l)

l.remove(34)

print(l)

l.clear()

print(l)

output:

['vcet', 41, 5.6, 'vaidehi']

['vcet', 41, 5.6, 'vaidehi', 45]

['vcet', 41, 34, 5.6, 'vaidehi', 45]

['vcet', 41, 34, 5.6, 'vaidehi']

['vaidehi', 5.6, 34, 41, 'vcet']

['vaidehi', 5.6, 41, 'vcet']

[]

l1=[23,16,78,34,65]

l1.sort()

print(l1)

output:

[16, 23, 34, 65, 78]

#tuple

t=(2,3,6,8,4,8,)

print(t.count(8))

print(t.index(4))

print(sorted(t))

print(max(t))

print(min(t))

print(sum(t))

print(len(t))

output:

2

4

[2, 3, 4, 6, 8, 8]

8

2

31

6

#set

s={23,16,78,34,65}

s.add(20)

print(s)

print(sorted(s))

print(max(s))

print(min(s))

print(sum(s))

print(len(s))

print(all(s))

**Output:**

{16, 65, 34, 20, 23, 78}

[16, 20, 23, 34, 65, 78]

78

16

236

6

True

#dictionary

d={'Name':'Vaidehi','Roll No.':'41','Age':'19'}

print(d)

print(d.get('Name'))

print(list(d.keys()))

print(list(d.values()))

print(d.pop('Roll No.'))

**Output:**

{'Name': 'Vaidehi', 'Roll No.': '41', 'Age': '19'}

Vaidehi

['Name', 'Roll No.', 'Age']

['Vaidehi', '41', '19']

41

**Conclusion:**

Python offers a rich variety of data types, each serving different purposes and providing flexibility for programmers to manipulate data effectively. Python provides several built-in data types, including integers, floats, strings, booleans, lists, tuples, dictionaries, sets, and NoneType. Python is dynamically typed, meaning you don't need to declare the type of a variable explicitly. The type of a variable is inferred at runtime based on the assigned value. Some data types in Python are mutable, meaning their values can be changed after creation (e.g., lists, dictionaries, sets), while others are immutable, meaning their values cannot be changed after creation (e.g., integers, floats, tuples, strings). You can check the type of an object using the type( ) function and perform type casting using constructors of different data types. ach data type in Python comes with its own set of methods and functions that can be used to manipulate and operate on data of that type efficiently.