

| 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 |





















      

























**Code:**

print("Lists: \n")

l = list()

for i in range(5):

a = int(input('Enter a number: '))

l.append(a)

print(f'Original List: {l}')

l.reverse()

print(f'Reversed List: {l}')

print("\n")

b = int(input("Element to add in the list: "))

print(f"Before adding: {l}")

l.append(b)

print(f'After Adding: {l}')

print("\n")

c=int(input("Position to insert an element: "))

d = int(input("Enter the element to insert: "))

if c > len(l):

print("Invalid input")

else:

print(f'Before: {l}')

l[c] = d

print(f"After: {l}")

print("\n")

e = int(input("Enter the element to remove from the list: "))

if e not in l:

print("Invalid input")

else:

print(f"Before removing: {l}")

l.remove(e)

print(f'After Removing: {l}')

print("\n")

l1 = [2,3,4]

print(f"Before extending: {l}")

l.extend(l1)

print(f"AFter Extending: {l}")

print("\n")

f = int(input("Enter the index to pop: "))

if f > len(l):

print("Invalid input")

else:

print(f"Before: {l}")

l.pop(f)

print(f"After: {l}")

print("\n")

print(f"Before sorting: {l}")

l.sort()

print(f"After sorting: {l}")

print("\n")

print(f"Before clearing: {l}")

l.clear()

print(f"After clearing: {l}")

print("\n")

print("\n")

print("Sets: \n")

s = set()

for i in range(5):

a = int(input("Enter number: "))

s.add(a)

print("\n")

print(f"The Set is :{s}")

b = int(input("Enter a number to remove: "))

if b in s:

print(f"Before: {s}")

s.remove(b)

print(f"After: {s}")

else:

print("Number not in set")

print("\n")

c = int(input("Enter a number to discard: "))

s.discard(c)

print(f"Before: {s}")

s1 = {2,36,32,3}

s.update(s1)

print(f"After: {s}")

print("\n")

print("Union: ")

res = s.union(s1)

print(res)

print("\n")

print(f"Before difference update: {s}")

s.difference\_update(s1)

print(f"After difference update: {s}")

print("\n")

print("\n")

print("Dictionary: \n")

d = dict()

for i in range(5):

k = int(input("Key: "))

v = int(input("Value: "))

d[k] = v

print(f"Dictionary: {d}")

print(f"Before update: {d}")

d.update({4:"A", 7:"B", 8:"C",9:"D"})

print(f"After update: {d}")

print("\n")

a = int(input("Enter a element to get: "))

res = d.get(a)

print(res)

b = int(input("To pop: "))

if b in d:

print(f"Before Popping: {d}")

d.pop(b)

print(f"After popping: {d}")

else:

print("Invalid Input")

print("\n")

print(f"Before Clearing: {d}")

d.clear()

print(f"After Clearing: {d}")

print("\n")

print("\n")

print("Tuple: \n")

t = (1,23,4,5,6,7)

print(t)

a = int(input("Enter a number: "))

print(t.count(a))

print("\n")

print("\n")

b = int(input('Enter a number whose index to find: '))

print(t.index(b))

**Output:**

Lists:

Enter a number: 5

Enter a number: 12

Enter a number: 23

Enter a number: 1

Enter a number: 0

Original List: [5, 12, 23, 1, 0]

Reversed List: [0, 1, 23, 12, 5]

Element to add in the list: 4

Before adding: [0, 1, 23, 12, 5]

After Adding: [0, 1, 23, 12, 5, 4]

Position to insert an element: 2

Enter the element to insert: 45

Before: [0, 1, 23, 12, 5, 4]

After: [0, 1, 45, 12, 5, 4]

Enter the element to remove from the list: 45

Before removing: [0, 1, 45, 12, 5, 4]

After Removing: [0, 1, 12, 5, 4]

Before extending: [0, 1, 12, 5, 4]

After Extending: [0, 1, 12, 5, 4, 2, 3, 4]

Enter the index to pop: 3

Before: [0, 1, 12, 5, 4, 2, 3, 4]

After: [0, 1, 12, 4, 2, 3, 4]

Before sorting: [0, 1, 12, 4, 2, 3, 4]

After sorting: [0, 1, 2, 3, 4, 4, 12]

Before clearing: [0, 1, 2, 3, 4, 4, 12]

After clearing: []

Sets:

Enter number: 2

Enter number: 3

Enter number: 1

Enter number: 5

Enter number: 6

The Set is :{1, 2, 3, 5, 6}

Enter a number to remove: 2

Before: {1, 2, 3, 5, 6}

After: {1, 3, 5, 6}

Enter a number to discard: 7

Before: {1, 3, 5, 6}

After: {32, 1, 2, 3, 36, 5, 6}

Union:

{32, 1, 2, 3, 36, 5, 6}

Before difference update: {32, 1, 2, 3, 36, 5, 6}

After difference update: {1, 5, 6}

Dictionary:

Key: 1

Value: 45

Key: 2

Value: 34

Key: 3

Value: 56

Key: 4

Value: 89

Key: 5

Value: 67

Dictionary: {1: 45, 2: 34, 3: 56, 4: 89, 5: 67}

Before update: {1: 45, 2: 34, 3: 56, 4: 89, 5: 67}

After update: {1: 45, 2: 34, 3: 56, 4: 'A', 5: 67, 7: 'B', 8: 'C', 9: 'D'}

Enter a element to get: 2

34

To pop: 0

Invalid Input

Before Clearing: {1: 45, 2: 34, 3: 56, 4: 'A', 5: 67, 7: 'B', 8: 'C', 9: 'D'}

After Clearing: {}

Tuple:

(1, 23, 4, 5, 6, 7)

Enter a number: 23

1

Enter a number whose index to find: 6

4

**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.