**DATE:16/08/2022**

**ASSIGNMENT NO:02**

**TOPIC : DIFFERENT OPERATION’S COLLECTION DATA TYPES**

**List Operations:**

**1.Concatinating using +**

ls1=[1,2,3]

ls2=[4,5,6]

print(ls1+ls2)

**OUTPUT:**

[1, 2, 3, 4, 5, 6]

**2.Repeating using \***

ls1=['j','a','v','a']

print(ls1\*3)

**OUTPUT:**

['j', 'a', 'v', 'a', 'j', 'a', 'v', 'a', 'j', 'a', 'v', 'a']

**3.List Slicing**

**Extracting full list without using any index**

ls1=['a','b','c','d','e']

print(ls1[:])

**OUTPUT:**

['a', 'b', 'c', 'd', 'e']

**Selecting some middle elements**

ls1=['p','y','t','h','o','n']

print(ls1[2:5])

**OUTPUT:**

[1, 2, 3, 4]

**Using Negative Indexing**

ls1=[1,2,3,4,5,6]

print(ls1[:-2])

**OUTPUT:**

[1, 2, 3, 4]

**List Modifying**

ls1=['a','b','c','d','e']

ls1[1:3]=['p','q']

print(ls1)

**OUTPUT:**

['a', 'p', 'q', 'd', 'e']

**append():**

ls1=['p','y','t','h','o','n']

ls1.append('list')

print(ls1)

**OUTPUT:**

['p', 'y', 't', 'h', 'o', 'n', 'list']

**#extend():**

ls1=[1,2,3,4]

ls2=[5,6]

ls2.extend(ls1)

ls1.extend(ls2)

print(ls1)

print(ls2)

**OUTPUT:**

[1, 2, 3, 4, 5, 6, 1, 2, 3, 4]

[5, 6, 1, 2, 3, 4]

**Sort**

ls1=[88,67,45,78,17]

ls1.sort()

print(ls1)

**OUTPUT:**

[17, 45, 67, 78, 88]

**Reverse:**

ls1=['j','a','v','a']

ls1.reverse()

print(ls1)

**OUTPUT:**

['a', 'v', 'a', 'j']

**Count()**

ls1=[1,2,4,6,2,8,9]

ls1.count(2)

**OUTPUT:**

**2**

**Clear()**

ls=[1,2,3]

ls.clear()

print(ls)

**OUTPUT:**

[]

Insert():

ls=[1,2,3,4]

ls.insert(1,"hi")

print(ls)

OUTPUT:

[1, 'hi', 2, 3, 4]

**Index();**

ls=[1,2,3,4,5,6,7,8,9]

ls.index(2)

**OUTPUT:**

**1**

**Deleting An Elements**

**Using pop()**

ls=[1,2,3,-4]

x=ls.pop()

print(ls)

print(x)

**OUTPUT:**

[2, 3, -4]

1

**Remove():**

ls=[5,8,-9,-3]

ls.remove(-9)

print(ls)

**OUTPUT:**

[5, 8, -3]

**Del():**

ls=[1,2,6,7]

del ls[2]

print(ls)

**OUTPUT:**

[1, 2, 7]

**Deleting An All Odd Indexed Elements**

ls=[1,2,3,4,5,6,7,8,9]

del ls[1::2]

print(ls)

**OUTPUT:**

[1, 3, 5, 7, 9]

**Tuple Operations.:**

**Different types of tuples**

**Empty tuple:**

my\_tuple = ()

print(my\_tuple)

**Tuple having integers:**

my\_tuple = (1, 2, 3)

print(my\_tuple)

**Tuple with mixed datatypes:**

my\_tuple = (1, "Hello", 3.4)

print(my\_tuple)

**Nested tuple:**

my\_tuple = ("mouse", [8, 4, 6], (1, 2, 3))

print(my\_tuple)

**OUTPUT:**

()

(1, 2, 3)

(1, 'Hello', 3.4)

('mouse', [8, 4, 6], (1, 2, 3))

**Accessing tuple elements using indexing**

my\_tuple = ('p','e','r','m','i','t')

print(my\_tuple[0])

print(my\_tuple[5])

print(my\_tuple[6])

my\_tuple[2.0]

**nested tuple**

n\_tuple = ("mouse", [8, 4, 6], (1, 2, 3))

**nested index**

print(n\_tuple[0][3])

print(n\_tuple[1][1])

**Negative indexing for accessing tuple elements**

my\_tuple = ('p', 'e', 'r', 'm', 'i', 't')

print(my\_tuple[-1])

print(my\_tuple[-6])

**OUTPUT:**

p

t

s

4

t

p

**Slicing**

**Accessing tuple elements using slicing**

my\_tuple = ('p','r','o','g','r','a','m','i','z')

# elements 2nd to 4th

print(my\_tuple[1:4])

# elements beginning to 2nd

print(my\_tuple[:-7])

# elements 8th to end

print(my\_tuple[7:])

# elements beginning to end

print(my\_tuple[:])

**OUTPUT:**

('r', 'o', 'g')

('p', 'r')

('i', 'z')

('p', 'r', 'o', 'g', 'r', 'a', 'm', 'i', 'z')

**# Changing tuple values**

my\_tuple = (4, 2, 3, [6, 5])

 TypeError: 'tuple' object does not support item assignment

my\_tuple[1] = 9

However, item of mutable element can be changed

my\_tuple[3][0] = 9    # Output: (4, 2, 3, [9, 5])

print(my\_tuple)

**Tuples can be reassigned**

my\_tuple = ('p', 'r', 'o', 'g', 'r', 'a', 'm', 'i', 'z')

print(my\_tuple)

**OUTPUT:**

(4, 2, 3, [9, 5])

('p', 'r', 'o', 'g', 'r', 'a', 'm', 'i', 'z')

**Concatenation**

print((1, 2, 3) + (4, 5, 6))

**Repeat**

print(("Repeat",) \* 3)

**OUTPUT:**

(1, 2, 3, 4, 5, 6)

('Repeat', 'Repeat', 'Repeat')

**Deleting tuples**

my\_tuple = ('p', 'r', 'o', 'g', 'r', 'a', 'm', 'i', 'z')

# can't delete items

# TypeError: 'tuple' object doesn't support item deletion

# del my\_tuple[3]

# Can delete an entire tuple

del my\_tuple

# NameError: name 'my\_tuple' is not defined

print(my\_tuple)

**OUTPUT:**

NameError Traceback (most recent call last)

<ipython-input-50-6d502a6648b0> in <module>

**10**

**11** # NameError: name 'my\_tuple' is not defined

---> 12 print(my\_tuple)

NameError: name 'my\_tuple' is not defined

**Membership test in tuple**

my\_tuple = ('a', 'p', 'p', 'l', 'e',)

**# In operation**

print('a' in my\_tuple)

print('b' in my\_tuple)

**# Not in operation**

print('g' not in my\_tuple)

**OUTPUT:**

True

False

True

**# Using a for loop to iterate through a tuple**

for name in ('John', 'Kate'):

    print("Hello", name)

**OUTPUT:**

Hello John

Hello Kate

**Dictionary Operations**.

**Creating Dictionary.**

# empty dictionary

my\_dict = {}

# dictionary with integer keys

my\_dict = {1: 'apple', 2: 'ball'}

# dictionary with mixed keys

my\_dict = {'name': 'John', 1: [2, 4, 3]}

# using dict()

my\_dict = dict({1:'apple', 2:'ball'})

# from sequence having each item as a pair

my\_dict = dict([(1,'apple'), (2,'ball')])

# get vs [] for retrieving elements

my\_dict = {'name': 'Jack', 'age': 26}

# Output: Jack

print(my\_dict['name'])

# Output: 26

print(my\_dict.get('age'))

# Trying to access keys which doesn't exist throws error

# Output None

print(my\_dict.get('address'))

# KeyError

print(my\_dict['address'])

**OUTPUT:**

Jack

26

None

---------------------------------------------------------------------------

KeyError Traceback (most recent call last)

[<ipython-input-9-2ec6dc01caba>](https://localhost:8080/) in <module>

**13**

**14** # KeyError

---> 15 print(my\_dict['address'])

KeyError: 'address'

**Changing and adding Dictionary Elements**

my\_dict = {'name': 'Jack', 'age': 26}

# update value

my\_dict['age'] = 27

#Output: {'age': 27, 'name': 'Jack'}

print(my\_dict)

# add item

my\_dict['address'] = 'Downtown'

# Output: {'address': 'Downtown', 'age': 27, 'name': 'Jack'}

print(my\_dict)

**OUTPUT**

{'name': 'Jack', 'age': 27}

{'name': 'Jack', 'age': 27, 'address': 'Downtown'}

# **Removing elements from a dictionary**

# create a dictionary

squares = {1: 1, 2: 4, 3: 9, 4: 16, 5: 25}

# remove a particular item, returns its value

print(squares.pop(4))

print(squares)

# remove an arbitrary item, return (key,value)

print(squares.popitem())

print(squares)

# remove all items

squares.clear()

print(squares)

# delete the dictionary itself

del squares

# Throws Error

print(squares)

**OUTPUT:**

16

{1: 1, 2: 4, 3: 9, 5: 25}

(5, 25)

{1: 1, 2: 4, 3: 9}

{}

---------------------------------------------------------------------------

NameError Traceback (most recent call last)

[<ipython-input-11-d119a0a5b684>](https://localhost:8080/) in <module>

**28**

**29** # Throws Error

---> 30 print(squares)

NameError: name 'squares' is not defined

**#Set Operations.**

# Program to perform different set operations like in mathematics

# define  sets

E = {0, 2, 4, 6, 8};

N = {1, 2, 3, 4, 5};

# set union

print("Union of E and N is",E | N)

# set intersection

print("Intersection of E and N is",E & N)

# set difference

print("Difference of E and N is",E - N)

# set symmetric difference

print("Symmetric difference of E and N is",E ^ N)

**OUTPUT:**

Union of E and N is {0, 1, 2, 3, 4, 5, 6, 8}

Intersection of E and N is {2, 4}

Difference of E and N is {0, 8, 6}

Symmetric difference of E and N is {0, 1, 3, 5, 6, 8}