# Online Python compiler (interpreter) to run Python online.

# Write Python 3 code in this online editor and run it.

print("Hello this is a revesion program")

print("This is a second program of the Python")

fruit = "Orange"

veg = 'flower'

print("Fruits:", fruit)

print("Vegitable:", veg)

formats = 'Hi, \'this is written under the single quto\', "Hope you get some idea"!'

formatbslh = "Hi this is \"Ravi Pathak\" "

print("String Formatting:", formats)

print("Backslash:", formatbslh)

#showing the all seperate character of the fruit(Orange).

print("First char accessing of the fruit:", fruit[0])

print("Second char accessing of the fruit:", fruit[1])

print("Third char accessing of the fruit:", fruit[2])

print("Fourth char accessing of the fruit:", fruit[3])

print("Fifth char accessing of the fruit:", fruit[4])

print("sixth char accessing of the fruit:", fruit[5])

#different built-in function

print("Length of the Fruit:", len(fruit))

#string function

print("Lower function:", fruit.lower())

print("Upper function:", fruit.upper())

#concatination

print('I ' + 'Love '+'Python ' + 'Programming')

#repeating

print('-' \* 20)

print("Ravi "\*10)

#format function

print("I, {} Love Python Program".format("Ravi"))

print("{}, Ravi {} {} {}".format("I", "Love", "Python", "Programming"))

print("{0:8} | {1:<8}".format("fruit", "Quantity"))

print("{0:8} | {1:>8}".format("fruit", "Quantity"))

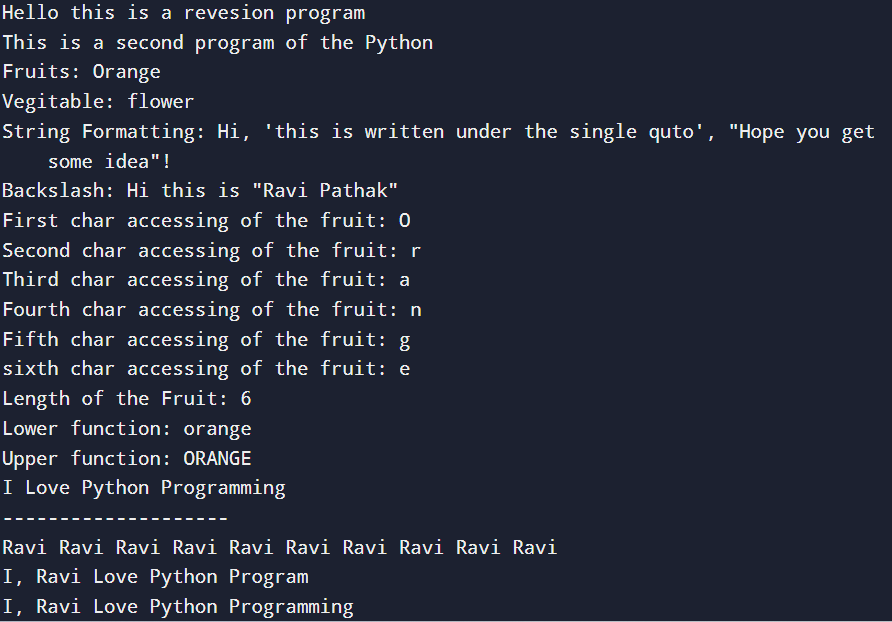
print("{0:8} | {1:^8}".format("fruit", "Quantity"))

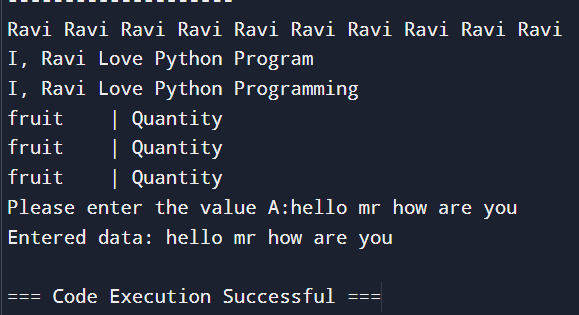
#input from the user

data = input("Please enter the value A:")

print("Entered data:", data)

Output:





# for concatinating need same type data

value = '3'

add = int(value) + 2

print("Addition:", add)

#float type conversion

print("Float type:", float(add))

stringdata = """ this is a comment

ok """

print("string data:", stringdata)

print(8/4)

#boolean data type

booldatatrue = True

booldatafalse = False

print("True Data:", booldatatrue)

print("False Data:", booldatafalse)

#comparator

print("4>10:", 4>10)

print("5<10:", 5<10)

print("5==3:", 5==3)

print("5==5:", 5==5)

#conditional

if(4>2):

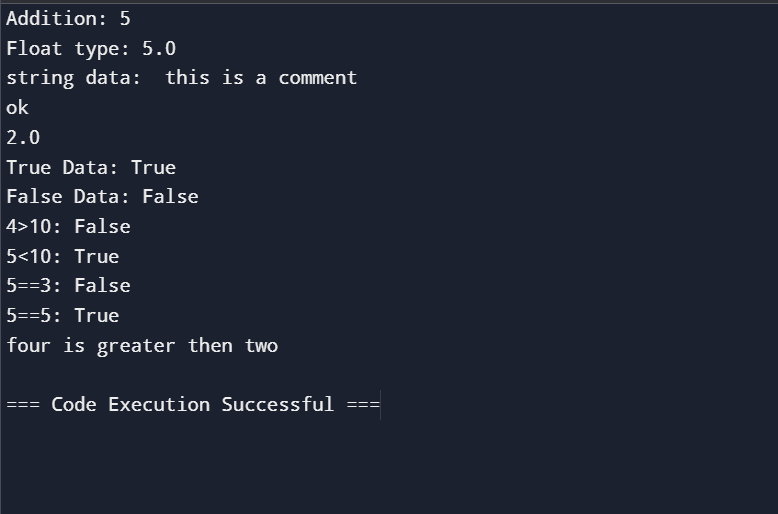
print("four is greater then two")

elif 6>3:

print("Six is greater then three")

else:

print("Four is less then two")



#function

def greetings(name):

print("Hi, Mr/Mrs. {} ".format(name))

#same function can use two times

greetings("Ravi")

greetings("Bhawana")

#passing the value to the function

def say\_hi(name = "Ravi Pathak"):

print("Hi, {}!".format(name))

say\_hi()

say\_hi("RaviKumar Pathak")

def multiple\_argument(first, second, third):

print("first: {}, second: {}, third: {}!".format(first, second, third))

multiple\_argument("Ravi", "Kumar", "Pathak")

#with this we can have the full information ofthe function

help(say\_hi)

#function on condition

def odd\_even\_check(number):

""" below code will check the odd and even number"""

if number % 2 == 0:

return 'Even Number'

else:

return 'Odd Number'

print(odd\_even\_check(7))

print(odd\_even\_check(10))

A computer screen shot of a computer code

Description automatically generated

#Lists

animals = ['Dog', 'Cow', 'Goat']

print("Animals:", animals)

#accessing one by one data

print("First animal:", animals[0])

print("Second animal:", animals[1])

print("Third animal:", animals[2])

print("Number of animal:", len(animals))

#manipulating the data

animals[0] = "Pig"

print(animals)

animals[0] = " "

print(animals)

animals.append("Bear")

print(animals)

#for multiple data

more\_animals = ["horse", "Zebra", "Mouse"]

animals.extend(more\_animals)

print(animals)

#can insert at the specific position

animals.insert(1, "Dog")

print(animals)

#Slices

print("---------------slice concept------------")

slicedatafirst = animals[0:3]

print(slicedatafirst)

slicedatasecond = animals[:3]

print(slicedatasecond)

slicedatathird = animals[2:]

print(slicedatathird)

#extracting part from the string

part\_of\_string = 'home'[1:3]

print(part\_of\_string)

#finding thr index of the animal

index\_animal = animals.index('Cow')

print(index\_animal)

#exception handling

try:

cow\_index = animals.index('cow')

except:

cow\_index = "cow not found"

print(cow\_index)

#Loops

for animal in animals:

print(animal.upper())

for anim in animals:

print(anim.lower())

#while

num = 3

while num > 0:

print("Hello")

num = num-1

#sorted

sort\_animals = sorted(animals)

print(sort\_animals)

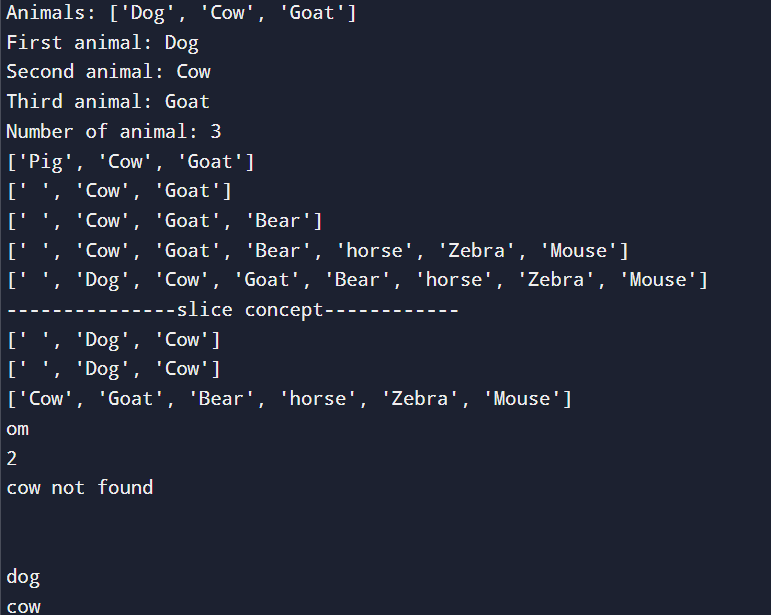
#range

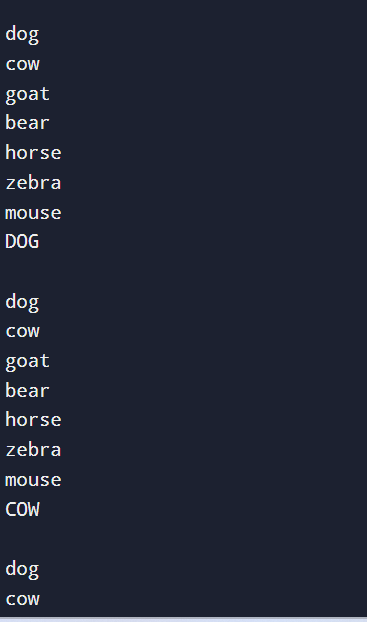
for number in range(5):

print(number)

for num1 in range(1,5):

print(num1)





A computer screen shot of a computer screen

Description automatically generated

#Dictonaries

dist\_data = {'num1' : '1', 'num2':'2', 'num3':'3'}

dist\_key1 = dist\_data['num1']

dist\_key2 = dist\_data['num3']

print("First Dictonarie value:",dist\_key1)

print("Second Dictonarie value:", dist\_key2)

#deleting

del dist\_data['num1']

print("After deleting the element:", dist\_data)

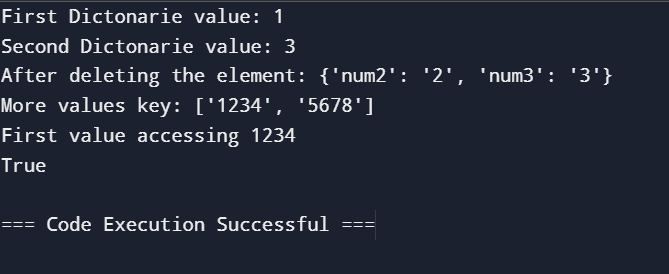
#more value dictionaries

contacts = {'Ravi': ['1234', '5678'], 'Kumar':'1111'}

print("More values key:", contacts['Ravi'])

print("First value accessing", contacts['Ravi'][0])

print('1111' in contacts.values())



#tuples

days = ('Sunday', 'Monday', 'Tuesday', 'Wednesday')

print("Tuples data:", days)

#converting tuples into the list and also checking the type of the object

listconvertdata = list(days)

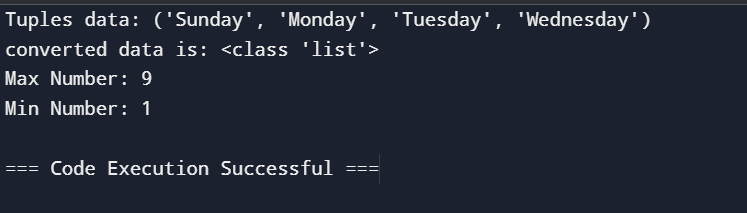
print("converted data is:", type(listconvertdata))

#max min number

numbers = (1,2,3,4,5,7,8,9)

print("Max Number:", max(numbers))

print("Min Number:", min(numbers))



#file system

hosts = open('/etc/hosts')

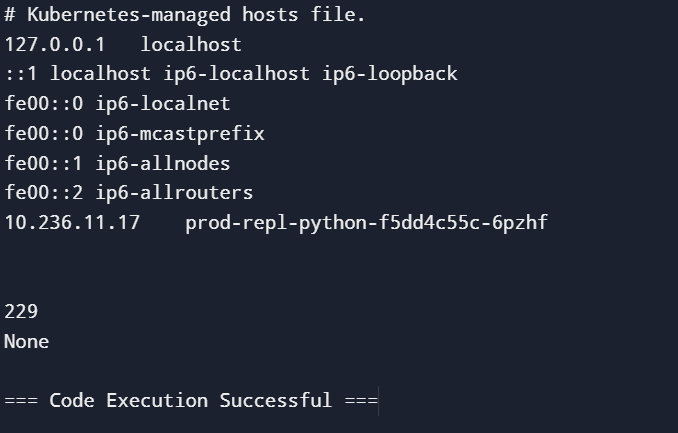
print(hosts.read())

print(hosts.read(10))

print(hosts.tell())

#close the file

print(hosts.close())



#modules (built-in functions)

import time

print(time.asctime())

print(time.timezone)

