**Data Type:**

**String**

'''

#fname = input('Enter Your Name: ')

#print(fname)

#Stirng....

print("Hello")

print("Let's Play")

print('Let\'s Play')

#Indexing

name = "jerin"

print(name[3])

print(name[3])

print(len(name))

#Slicing

city = 'bangaluru'

name = 'Sohanur'

print(city[3:])

print(city[3:8])

#name

print(name[0:5])

print(name[:5])

##concate two/multiple string

fname = 'Md. Sohanur'

mname = 'Rahman'

lname = 'Sohan'

print(fname+ ' '+mname+ ' '+lname)

##string is immutable that means it can not change any word specific character. we can't change or replace the word. let's see a example:

nam = 'Sohan'

#nam[0] = 'R'

#It's does not work, beacuse a string is a immutable. we can use it on the other ways we can take a temporary variable such as temp

temp = nam[:0] + 'R' + nam[1:] #Here we use the string Slicing 0 index is replaced by the 'R' character

print(temp)

#here we use now some buil-in function of string

emp\_nam = 'sohan'

print(emp\_nam)

print(emp\_nam.capitalize())

print(emp\_nam.find('h'))

print(emp\_nam.upper())

print(emp\_nam.lower())

sentence = 'he is running'

print(sentence.split(' '))

print(sentence.split(' ')[2])

'''

#List Data Tpye: List is a collection of object type Data where containing a sequence of sort Data

'''num = [1,2,3,4,5]

emp\_nam1 = ['sohan','rohan','mohan']

print(num)

print(emp\_nam1)

#Marged list is used to concate two List

marg\_list = [1,2,3,'soahn',[1,2,3,'rohan']]

fruits = ['Apple', 'Banana','Orange']

print(fruits[1])

print(fruits[-2])

#Blank-list

Blank\_list = []

print(Blank\_list)

#list is a mutable so that we can easily change the value of the list elements such

fruits[2] = 'Mango' #list is mutable(Changeble), it can change the elementsby accessing the Indexing

print(fruits)

print('Apple' in fruits)

print('Orange' in fruits)

print(len(fruits))

for fruit in fruits: #using for loop to print the list elements

print(fruits)

for i in range(len(fruits)):

print('at index',i,'element is ',fruits[i])

#for i in range(len(fruits)):

# print('at index '+ str(i) +' element is '+fruits[i])

animl = ['cow','hourse','cat']

both = fruits+animl;

print(both)

print(animl\*3) # print three times

#list Slicing

print(fruits[1:3])

fruits[1:3] = ['strobary','Greps']

print(fruits)

#matrix

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

print(matrix[0][0])

print(matrix[0][1])

print(matrix[0][2])

#some bult-in function

fruits.append('Mango') #append() function add a new elementin the list

print(fruits)

#sort

fruits.sort()

print(fruits)

print('The Index is: ',fruits.index('Mango'))

print('The Index is: ',fruits.index('Greps'))

#insert a new element

print('The Inserted value is: ',fruits.insert(2,'Banana'))

print(fruits)

print(fruits.count('Apple'))

fruits.append('Apple')

print(fruits)

print(fruits.count('Apple'))

#deletion

fruits.pop()

print('The Previous List is: ',fruits)

fruits.pop(1)

print('After deletion List is: ',fruits)

#remove the specific element

fruits.remove('Banana')

print('The remove the list is :',fruits)

#The string is converted into a list

sentence1 = "Sohanur Rahman Sohan"

list\_to\_str = list(sentence1)

print('The string is converted into a list is: ',list\_to\_str)

print('The split list is: ',sentence1.split(" "))

#Max and Min function

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

print('The Max element is: ', max(numbr))

print('The Min element is: ', min(numbr))'''

'''#Tuple:to identify the tuple here we see that the first braket or small braket

t = 1,2,3.5,'sohan'

print(t)

Blank\_tuple = ()

print(Blank\_tuple)

num = (1,2,3,4,5)

print(num[2])

print(num[-3])

print('The Slicing tuple is: ',num[2:])

add = (3+5)

print(add\*2)

add = (3+5,)

print(add\*3)

#email\_id = srsohan014@gmail. split two parts

email\_id = 'srsohan014@gmail.'

user,domain = email\_id.split('@')

print(user)

print(domain)

#Unpacking Of tuple

(x,y,z) = (1,2,3)

print(x)

print(y)

print(z)

#Using function

def min\_max(t):

return min(t),max(t)

t = (1,23,45,5,7)

minimum,maximum = min\_max(t)

print(minimum)

print(maximum)

#List and Tuple using both at a times

nums = [1,2,3]

names = ['Sohan','Rohan','Mohan']

print(zip(nums,names))

#for loop using to print tuple

t = (1,2,3,4,5)

for i in t:

print(i)

#marg\_list

marg\_listt = [(1,2,3,),('sohan',4,9)]

for first,second,last in marg\_listt:

print(first)

print(second)

print(last)'''

#Dictionary Data Type: 1st way to create Dictionary

'''Student\_name\_Age = {'Sohan':23,'Rohan':24,'Mohan':25}

print(Student\_name\_Age)

#Dictionary Data Type: 2nd way to create Dictionary

details = dict([('name','Sohan'),('Age','23')])

print(details)

#Dictionary Data Type: 3rd way to create Dictionary

d = dict(name = 'Sohan',Age = 23)

print(d)

#length and key check

print(len(d))

print('Age' in d) #to ckeck the key value, it's print the by defualt value..

print('Sohan' in d.values()) #to ckeck the just value

'''

'''

#print built in and print function

#print 12,13 #print built-in function

print (12,13) # print function

#print 'Sohan',

#print 'Rohan'

print ('Sohan',)

print ('Rohan')

print(12,'Sohan',[1,2,3],(4,5,6,7,8,9)) #print function allow all data type, so that we can use the to print different types of data using only one function

#Assignment Logic means that put the value in a variable,

x = 4

y = 6

z = 8

print(x,y,z)

# Assignment Logic can put the value at a single line without using multiple lines let's see that how it work

x,y,z = 4,6,8 # here the L.H.S and R.H.S values are must be same, Only the same value can print easily. If we don't follow this rules a error will accuare that is called 'UNPACK values'

print(x,y,z)

#Augmented Assignment is reduce the code of lines. we are reducing the statement of codes let's have a look,,

a = 7

a = a+1 #normal manner

print(a)

#now Augmented Assignment impementation

a = 7

a += 1 #Augmented Assignment

print(a)

'''

'''

#Condition

print(True+2)

print(True+0)

print(False+2)

print(False+0)

#Boolean function

print(bool('Sohan'))

print(bool(''))

print(bool((1,2,3,4,5)))

#In the same way the other data type will be execute print the Boolean function according the values

#Blank value,list[],Tuple(), Dictionary{} all have the Boolean value is 0.....

#Condition

name = 'Sohan'

Age = 23

Country = 'Bangladesh'

#if statement

#if name == 'Sohan' or Age ==23: # Or Operator to check multiple Condition at a time.. Here the any statement is true then the statement will be execute.

if name == 'Sohan' and Age ==23 and Country == 'Bangladesh': # and Operator to check multiple Condition at a time, It execute when the both statements are True, If anyone Condition is false it will be execute the false value

print('Hello ' +name + ' Your Age is ' +str(Age) + ' and Country is '+ Country)

else: #it's called else closed

print('Invalid Data')

#List in and Operator

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

if 3 in list: #here it check the membership of the list variable. it check the container of the list variable.

print('Available')

else:

print('Not Available')

num = int(input('Enter the number to ckeck positive, negatiive or zero: \n'))

if num > 0: #if is the normal if statement

print('positive')

elif num < 0: #elif like as else if in c language

print('negatiive')

else: #else is the normal else statement

print('Zero')

'''

'''

#ton ckeck the Dictionary Operator in Operator

price = { 'Apple':'120',

'Banana' : '150'

}

#if 'Apple' in price: # in Operator

if 'Apple' not in price: # not in Operator

print("cost of Apple is " + price['Apple'])

else:

print('Your fruit is not Available in the Dictionary')

print(price.get('Banana',''))

print(price.get('Banana',''))

'''

#Assignment Solution: which is the biggest number

'''a,b,c = 12,56,23

if a > b and a > c:

print('a is grater')

elif b > a and b > c:

print('b is grater')

else:

print('c is greater')

'''

'''

#Loop Concept: while Loop:

#Number

x = 1

while x<=100: #to print 1 to 100 using while loop:

print(x)

x += 1

#string

name = 'Sohan'

i = 0

while i < len(name):

print(name[i])

i += 1

#Tuple

fruits = ('Apple','Banana','Orange')

i = 0;

while i<len(fruits):

print(fruits[i])

i += 1

#list

list = [1,2,3,4.5,'Sohan','Rohan']

i = 0;

while i<len(list):

print(list[i])

i +=1

'''

#Dictionary

#Dictionary = {'Apple':'140','Banana':'10','Orange':'100'}

#i = 0;

#while i<len(Dictionary):

# print(Dictionary[i])

# i +=1

'''

#Assignment Solution

data = {'user\_name' : 'Sohan',

'password' : '12345'

}

usernam = input('Enter Your User Name: ')

passwd = input('Enter Your User Password: ')

while usernam != data['user\_name'] and passwd != data['password']:

print('Invalid Data..Please Input The Correct Data.')

usernam = input('Enter Your User Name: ')

passwd = input('Enter Your User Password: ')

print('Welcome...')

'''

'''

#For loop: For Loop Can minimize the number of statements rather than while Loop such as a example

#print 1 to 100 using for Loop

#Number

for i in range(1,101): # Range Indexing start at 0 index so that the here using 1 to the start point and 101 is the ending point.

print(i)

#

print(range(1,10))

for i in range(1,51,2):

print(i) # to print odd numbers

print(i+1) # to print even numbers

#string

print('The Name Is :')

name = 'Sohan'

for i in range(len(name)):

print(name[i])

#List

print('The List Is :')

list = [1,'soahn',2,3.5,4,5,6,7]

for i in range(len(list)):

print(list[i])

#Tuple = ()

print('The Tuple Is :')

Tuple = (1,2,3,4,5)

for i in range(len(Tuple)):

print(Tuple[i])

'''

'''

#Matrix using while Loop and for Loop:

#while Loop:

print('The Matrix is :')

raw = 1

while raw<=4:

column = 1

while column<=4:

print (column, end = " ")

column += 1

print("")

raw += 1

#floyed Triangle Problem Solution

print('The Floyed Triangle is :')

raw = 1

while raw<=4:

column = 1

while column<=raw:

print(column, end = " ")

column += 1

print("")

raw += 1

print('The Floyed Triangle is :')

raw = 1

while raw<=4:

column = raw

while column<=4:

print (column, end = " ")

column += 1

print("")

raw += 1

#for loop

data = [(1,2,3),(4,5,6),(7,8,9)]

for i in range(len(data)):

for j in range(len(data)):

print(data[i][j], end = " ")

#Control statements Break, pass, continue

#Break statement

for i in range(10):

print(i)

if i == 5 :

break

#continue statement

for i in range(10):

print(i)

if i == 5 :

print('Before the continue statement')

continue

print('After the continue statement') # it deos not execute

#pass statement

a = 90

if a == 90:

pass

print("Good Result")

elif a > 80:

print('Just do your Work...')

'''

#function is a set of program where the function can be used multiple times the same statements. It a group of statement that perform specific task

''' print('Hello S R!')

print('How are You?')

print('How is your day is going?')

'''

#if we want to print these lines in 12 times, we can copy and paste these lines 12 times

#But avoiding the copy paste we can print these lines 12 times in using in a function such as

'''

def f1():

print('Hello S R!')

print('How are You?')

print('How is your day is going?')

f1() #here the function call 1st time

f1() #here the function call 2nd time

f1() #here the function call 3rd time

f1() #here the function call 4th time

'''

#why use the function???

#1. Reuse the function

#2. Minizing the redundancy

#3. Proceduaral decomposition : we decompose the whole program that why it is easy to understandable.

#How to create own function: There are two to create function such as using "def" keyword and "lemda" keyword

#def keyword

def func():

print('Welcome to the function world')

func() #here calling the function

#Global And Local Variable:

num = 100 #Global Variable

print(num)

def localvar():

num = 50 #local variable

print(num) # here print the local Variable

localvar()

print(num) #here access the Global Variable (' ')[2])