1. str = ("hi")

print(str[1])

output:

i

str = ("hi4wewffe")

print(str[0:4])

output:

hi4w

1. str = ("hi4wewffe")

print(str[-1])

output:

e

1. str = ("hi4wewffe")

str1=str+str

print(str1)

output:

hi4wewffehi4wewffe

1. str = ("he")

str1=str\*2

print(str1)

output:

hehe

1. x,y,z=2,4,5

print(x)

print(y)

print(z)

output:

2

4

5

1. x,y,z=2,4,5

print(x>z)

output:

false

1. print('%s and %d' % ('hriday', 1))

output:

hriday and 1

str = "abccba is mamm"

str =str.casefold()

#for case sensitive string

str1=reversed(str)

if list(str)==list(str1):

print("its palindrome string")

else:

print("not palindrome")

str =str.split()

str.sort()

for words in str :

print(words)

output :

not palindrome

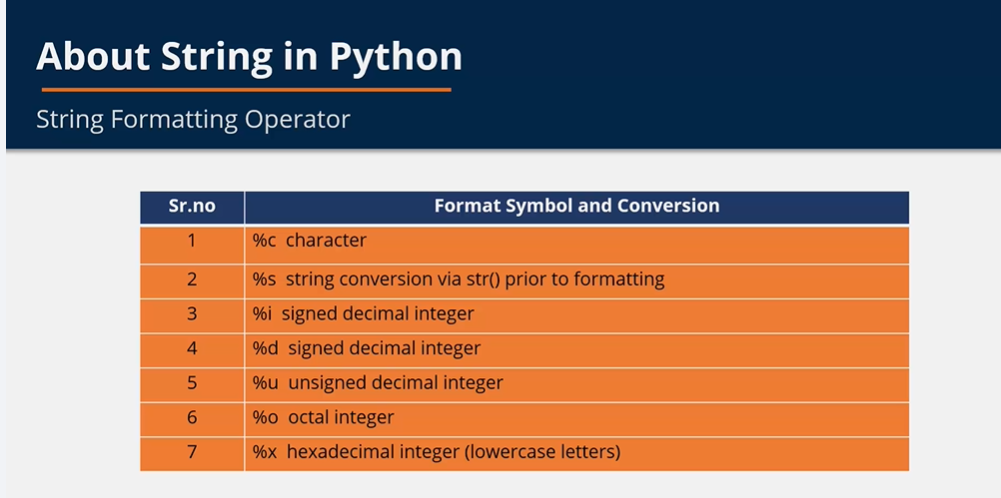
abccba

is

mamm

str=input(“”)#is used for inputing ur own value for string

while calling a particular element in list using[] , remember to start listing elements from zero(0).

  
strings are sequence of characters.

1. 
2. a=[['m'], 'd', 'd', 'a']
3. b=[ a ,'d', 'd', 'a']
4. print(a[2:3])#here ] and ( does not represent the actual meaning used in set theory [ means till that no.
5. print(b[0:3])
6. print(b[:1])
7. print(a[:])

output :

['d']

[[['m'], 'd', 'd', 'a'], 'd', 'd']

[[['m'], 'd', 'd', 'a']]

[['m'], 'd', 'd', 'a']

a=[1,2,3,4,5,'y','A','A']

a[1]=5

a[2:5]=['a','e','i','o','u']

print(a[:])

output:

[1, 5, 'a', 'e', 'i', 'o', 'u', 'y', 'A', 'A']

a=[1,2,3,4,5,'y','A']

del a[2:6]

print(a[:])

output:

[1, 2, 'A']

a=[1,2,3,4,5,'y','A']

a.remove(4)

print(a[:])

print(a.pop())

a.clear()

print(a[:])

output:

[1, 2, 3, 5, 'y', 'A']

A

[]

TUPLES IS SIMILAR TO LIST AND IT IS ORDERED & UNCHANGEABLE

CAN INSERT DUPLICATE ELEMENTS. LIKE 5,5

tuple=(“xxxx”)

tuple=(1,2,3)

print(tuple)

tuple\_1=('hello', 1 ,1, (1,2,3),[2,4,6])#here”hello” can also be used

print(tuple\_1)

tuple\_2= 23,5,6

print(tuple\_2)

output:

(1, 2, 3)

('hello', 1, 1, (1, 2, 3), [2, 4, 6])

(23, 5, 6)

tuple\_1=('hello')

print(type(tuple\_1))#here type means what type the element'hello' used by tuple is

#the result tells that tuple must contain more than 1 element to be a tuple

Output:

<class 'str'>

tuple\_1=('hello',)

print(type(tuple\_1))

# when comma is put it means that there is another element

Output:

<class 'tuple'>

\

tuple=('hello',[1,2,3],4,6)

print(tuple[0][1:3])

print(tuple[1][2])

print(tuple[3])

output:

el

3

6

tuple=('hello',[1,2,3],4,6)

tuple[1][2]=1

print(tuple)

output:

('hello', [1, 2, 1], 4, 6)

hi=(1,4,2,6,8,4,6,9,1,0,2,4,6,89,2,45,67,3,24,24,67,9)

print(hi.count(0))

print(hi.count(1))

print(hi.count(2))

print(hi.count(3))

print(hi.count(4))

print(hi.count(5))

print(hi.count(6))

print(hi.count(7))

print(hi.count(8))

print(hi.count(9))

output:

1

2

3

1

3

0

3

0

1

2

set={1,2,3,4,5,6,6,6}

print(set)

output:

{1,2,3,4,5,6}

[]=list

()=tuple

{}=set

List is not allowed inside set but tuple is allowed inside the set.

set={1,2,3,4,(5,6),6,6}

print(set)

set.add(2)

print(set)

set.update([2,3,4,5,2,34,454])

print(set)

set.remove(2)#can also use discard method like: set.discard(5)

print(set)

output:

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

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

{1, 2, 3, 4, 5, 6, 454, 34, (5, 6)}

{1, 3, 4, 5, 6, 454, 34, (5, 6)}

set = {1,2,3,4,(5,6),6,6}

set1 = {1,2,3,4,5,6,7,8}

set2 = set - set1

print(set2)

set3=set|set1

set4=set&set1

print(set3)

print(set4)

output:

{(5, 6)}

{1, 2, 3, 4, 5, 6, 7, 8, (5, 6)}

{1, 2, 3, 4, 6}

Dictionary:

1. it is a collection of elements.
2. It is unordered and changeable
3. It has key/value pair.

Eg.:

Mydic={1:’a’, 2:’b’}

Mydict=dict{[(1,’apple’),(2,’ball’)]}

mydict={1:'sunday', 2:'monday',3:'tuesday'}

print(mydict[1])

output:

Sunday

mydict={1:'sunday', 2:'monday',3:'tuesday'}

mydict[1]='first day of week sunday not'

mydict[4]='wednesday'

print(mydict)

output:

{1: 'first day of week sunday not', 2: 'monday', 3: 'tuesday', 4: 'wednesday'}

dict={1:1,2:2,3:3,4:4,5:5,6:6}

print(dict.pop(4))

print(dict)

print(dict.popitem())

del dict[2]

print(dict)

dict.clear()

print(dict)

output:

4

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

(6, 6)

{1: 1, 3: 3, 5: 5}

{}

dict={8:1,2:2,3:3,4:4,5:5,6:6}

print(len(dict))

print(sorted(dict))

output:

6

[2, 3, 4, 5, 6, 8]

dict={8:1,2:2,3:3,4:4,5:5,6:6}

for x in dict:

print(dict[x])

output:

1

2

3

4

5

6

a=int(input("enter a number "))

if (a>10):

print( a,"is greater than 10")

elif (a == 10):

print(a,"is equal to 10")

else:

print(a, "is lesser than 10")

a=int(input("enter a number "))#this is an example of nested loop meaning loop inside loop

if (a<=0):

if(a==0):

print(a, "is a zero no.")

else:

print(a, "is a -ve no.")

else:

print(a,"is +ve no.")

l=[1,2,3,4,5,6,7,8,9,10]

sum = 0

for i in l:

sum=sum+i

print("sum is",sum)

output:

sum is 1

sum is 3

sum is 6

sum is 10

sum is 15

sum is 21

sum is 28

sum is 36

sum is 45

sum is 55

l=list(input("print ur list: "))

sum = 0

for i in l:

print(i)

else:

print("there ain't any more elements in list")

print(range(3,10))

print(list(range(3,10,2)))#here 2 is step size

output:

range(3, 10)

[3, 5, 7, 9]

l=[1,2,3,4,5,6,7]

for a in range(len(l)-1):

print(l[a])

output:

1

2

3

4

5

6

a=10

i=1

sum=0

while i<=a:

sum=sum+i

i=i+1

print(sum)

output:

1

3

6

10

15

21

28

36

45

55

i=1

over= "we have completed ten"

while i<=a:

print("hello")

i=i+1

else:

print(over)

output:

hello

hello

hello

hello

hello

hello

hello

hello

hello

hello

we have completed ten

for val in "abced":

if val=="d":

break

print(val)

print("we are outside")

output:

a

b

c

e

we are outside

for val in "abced":

if val=="b":

break

print(val)

if val=="c":

break

print(val)

print("we are outside")

output:

a

a

we are outside

for val in "abced":

if val=="b":

continue

print(val)

print("outside loop")

output:

a

c

e

d

outside loop

def greet (name, place):

print(name+','+place)

greet("hi","bye")

output:

hi,bye

def greet(y):

return y\*y

y = int(input("enter no: "))

print(greet(y))

output:

enter no:

def greet(x,y):

print("my name is " , x , "and I am a",y)

x= str(input("what is ur name: "))

y = str(input("what is your profession: "))

print(greet(x,y))

def stu\_name(\*name):

for a in name:

print("my name is " , a)

stu\_name("hi","bye","tue")

output:

my name is hi

my name is bye

my name is tue

def stu\_name(\*name):

for a in name:

print("my name is " , a)

name=input("1: "), input("2 :"), input("3 :")

print(stu\_name(\*name))

output:

1: hriday

2 :spandan

3 :nilesh

my name is hriday

my name is spandan

my name is nilesh

None

def factorial(a):

if a == 1:

return 1

else:

return(a\*factorial(a-1))

a=int(input("enter the no. for factorial product: "))

print(factorial(a))

b = input("do you want to do it again : YES/NO -")

if (b=="YES"):

a=int(input("enter the no. for factorial product: "))

print(factorial(a))

else:

print("end")

##HERE FACTORIAL =

FOR X= X\* X-1 \* X-2 \* X-3……. \* 1

mul=lambda x:x\*x

print(mul(5))

output:25

import sys

random=[input("enter : " )]

for entry in random:

try:

print("the entry is", entry)

d=1/int(entry)

break

except:

print("oops!", sys.exc\_info(),"occurred")

print("next entry")

print("the reciprocal of", entry , "is",d)

class a:

c=100

def func(b):

print("hi")

print(a.c)

print(a.func(1))

output:

100

hi

class a:

def mul(x):

print(x\*x)

print(a.mul(5))

output:

25

None

class a:

def func(k):

print("hello")

m=a()

m.func()#here m is object

output:

hello

class a:

def \_\_init\_\_(self):#default constructor

self.a="geeks"

def print\_geek(self):

print(self.geek)

x=a()

x.print\_geek()

output:

geeks