1) import keyword

print (keyword.kwlist)

2) print(bool(x))

3) count = 0

while (count < 3):

count = count+1

print("Hello programmer")

4) l = ["today", "is", "holiday"] #this is list

for i in l:

print(i)

5)t = ("we", "have", "unity")

for i in t:

print(i)

6)s = "Geeks"

for i in s :

print(i)

7)from \_\_future\_\_ import print\_function

for i in range(1, 5): #you have to write two nested for loops using range

for j in range(i):

print(i, end=' ')

print()

8)print("\nDictionary Iteration")

d = dict()

d['xyz'] = 123

d['abc'] = 345

for i in d :

print("%s %d" %(i, d[i]))

9)for letter in 'allisgeeksforgeeks':

# break the loop as soon it sees 'e'

# or 's'

if letter == 'e' or letter == 's':

break

print ('Current Letter :', letter)

10)for letter in 'Python':

if letter h found break

for letter in 'Python': # First Example

if letter == 'h':

break

print('Current Letter :', letter)

for letter in 'geeksforgeeks' use \*\*Break\*\* when e or s comes

11)

for num in range(10,20): #to iterate between 10 to 20

for i in range(2,num): #to iterate on the factors of the number

if num%i == 0: #to determine the first factor

j=num/i #to calculate the second factor

print('%d equals %d \* %d' % (num,i,j))

break #to move to the next number, the #first FOR

else: # else part of the loop

print(num, 'is a prime number')

12)infinite loop

use infinite loop & print the entered input

infinite loop use infinite loop & print the entered input

var = 1First

while var == 1 : # This constructs an infinite loop

num = input("Enter a number :")

print ("You entered: ", num)

print ("Good bye!")

13) To find the prime numbers from 2 to 100 The use a nested for loop

for i in range(101):

for j in range(2,i-1):

if i%j==0:

break

else:

print(i)

14)

password= input("Enter the password\t")

if password=="secret":

print ("..............................")

print ("welcome")

else :

print ("Access Denied";)

15)print A All grades above 89 B All grades above 79 and below 90 C All grades above 69 and below 80 D All grades below 70

num =int(raw\_input("Enter the number:"))

if num > 89:

letter = 'A'

elif num > 79:

letter = 'B'

elif nuFirstm > 69:

letter = 'C'

else num > 89:

letter = 'D'

print "The Grade is " , letter

16) import collections

myList = [1,1,2,3,4,5,3,2,3,4,2,1,2,3]

print(collections. Counter(myList))

num\_int = 123

num\_str = "456"

print(type(num\_int))

print(type(num\_str))

num\_str = int(num\_str)

num\_sum = num\_int + num\_str

print(num\_sum)

print(num\_sum)

17) Create a list add physics chemistry 1997 2000 in list1

Add 1 2 3 4 5 6 7 in list 2

Print list1[0] & list2[1:5]

18)Create a list add physics chemistry 1997 2000 in list1 Add 1 2 3 4 5 6 7 in list 2 Print list1[0] & list2[1:5]

list1 = ['physics', 'chemistry', 1997, 2000]

list2 = [1, 2, 3, 4, 5, 6, 7 ]

print ("list1[0]: ", list1[0])print ("list2[1:5]: ", list2[1:5])

19) Create a list add physics chemistry 1997 2000 in list1 extends it with list two]

list1 = ['physics', 'chemistry', 'maths']

list2 = list(range(5)) #creates list of numbers between 0-4

list1.extend(list2)print ('Extended List :', list1)

Result

Extended List : ['physics', 'chemistry', 'maths', 0, 1, 2, 3, 4]

## 20) Count Control with a Python while Loop

## sum = 0

## for count in xrange(1,10):

## sum = sum+count

## print(sum)

## sum = 0

## count = 1

## while (count < 10):

## sum = sum + count

## count= count + 1

## print(sum, count)

## 21)Sorting the list

## list1 = ['physics', 'Biology', 'chemistry', 'maths']

## list1.sort()print ("list now : ", list1)

22)

tuple1, tuple2 = (123, 'xyz'), (456, 'abc')

print cmp(tuple1, tuple2)

print cmp(tuple2, tuple1)

tuple3 = tuple2 + (786,);

print cmp(tuple2, tuple3)

## Result

When we run above program, it produces the following result −

-1

1

-1

23) wing example shows the usage of tuple() method.

list1 = ['maths', 'che', 'phy', 'bio']

tuple1 = tuple(list1)print ("tuple elements : ", tuple1)

## Result

When we run above program, it produces the following result −

tuple elements : ('maths', 'che', 'phy', 'bio')

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

# Count

# Output: 2

print(my\_tuple.count('p'))

25) # Index

# Output: 3

print(my\_tuple.index('l'))

26)Here is an example to make a dictionary with each item being a pair of a number and its square.

squares = {x: x\*x for x in range(6)}

# Output: {0: 0, 1: 1, 2: 4, 3: 9, 4: 16, 5: 25}

print(squares)

27) membership test is for keys only, not for values.

squares = {1: 1, 3: 9, 5: 25, 7: 49, 9: 81}

# Output: True

print(1 in squares)

# Output: True

print(2 not in squares)

# membership tests for key only not value

# Output: False

print(49 in squares)

Here are some examples that uses built-in functions to work with dictionary.

29) # Creating a Nested Dictionary

# as shown in the below image

Dict = {1: 'Geeks', 2: 'For',

print(dic.get("A"))

print(dic.get("C"))

print(dic.get("C","Not Found ! "))

1 None Not Found !//op

# Python code to demonstrate working of

# str() and items()

# Initializing dictionary

dic = { 'Name' : 'Nandini', 'Age' : 19 }

# using str() to display dic as string

print ("The constituents of dictionary as string are : ")

print (str(dic))

# using str() to display dic as list

print ("The constituents of dictionary as list are : ")

print (dic.items())

The size of dic is : 3

The data type of dic is :

The data type of li is :

3:{'A' : 'Welcome', 'B' : 'To', 'C' : 'Geeks'}}

print(Dict)

{1: 'Geeks', 2: 'For', 3: {'A': 'Welcome', 'B': 'To', 'C': 'Geeks'}}

30) Get method

dic = {"A":1, "B":2}

print(dic.get("A"))

print(dic.get("C"))

print(dic.get("C","Not Found ! "))

1 None Not Found !//op

# Python code to demonstrate working of

# str() and items()

# Initializing dictionary

dic = { 'Name' : 'Nandini', 'Age' : 19 }

# using str() to display dic as string

print ("The constituents of dictionary as string are : ")

print (str(dic))

# using str() to display dic as list

print ("The constituents of dictionary as list are : ")

print (dic.items())

The size of dic is : 3

The data type of dic is :

The data type of li is :