

Q 1. index() :-

a. `index(1)` will return the index of first occurrence of 1 in a

a. `index(7) = 1`

a. `index(1) = 0`

list / Tuple

Q 2. Write a program to store seven fruits in a list entered by the user ?

```

→ F1 = input("Enter fruit Number 1 : ")
F2 = input("Enter fruit Number 2 : ")
F3 = input("Enter fruit Number 3 : ")
F4 = input("Enter fruit Number 4 : ")
F5 = input("Enter fruit Number 5 : ")
F6 = input("Enter fruit Number 6 : ")
F7 = input("Enter fruit Number 7 : ")
myfruitlist = [F1, F2, F3, F4, F5, F6, F7]
print(myfruitlist)
    
```

Output :-

Enter fruit Number 1 : Pineapple

Enter fruit Number 2 : Watermelon

Enter fruit Number 3 : Green Apple

Enter fruit Number 4 : Orange

Enter fruit Number 5 : Mango

Enter fruit Number 6 : Grapes

Enter fruit Number 7 : Pear

['Pineapple', 'Watermelon', 'Green Apple', 'Orange',
 'Mango', 'Grapes', 'Pear']

Q 3. WAP to arr them

```

→ M1 = int()
M2 = int()
M3 = int()
M4 = int()
M5 = int()
M6 = int()
    
```

Sorted arr

Sorted print

Output

3

2 WAP to accept marks of six student & display them and sort manner ?

→ $M_1 = \text{int}(\text{input}("Marks of 1st student :"))$

$M_2 = \text{int}(\text{input}("Marks of 2nd student :"))$

$M_3 = \text{int}(\text{input}("Marks of 3rd student :"))$

$M_4 = \text{int}(\text{input}("Marks of 4th student :"))$

$M_5 = \text{int}(\text{input}("Marks of 5th student :"))$

$M_6 = \text{int}(\text{input}("Marks of 6th student :"))$

$\text{Sortedmarks} = [M_1, M_2, M_3, M_4, M_5, M_6]$

$\text{Sortedmarks} = \text{sort}()$

$\text{print}(\text{Sortedmarks})$

Output :-

Marks of 1st student : 86

Marks of 2nd student : 76

Marks of 3rd student : 56

Marks of 4th student : 67

Marks of 5th student : 89

Marks of 6th student : 65

[56, 65, 67, 76, 86, 89]

3 Check that a tuple can not be change in python ?

4 WAP to sum a list with four Number ?

→ $l = [23, 45, 18, 14]$

print ("list = ", l)

print ("sum of list number are = ", l[0] + l[1] + l[2] + l[3])

Output :-

list = [23, 45, 18, 14]

sum of list number are = 94

5 WAP to Count the number of 0 (zero) in the following tuples ?

$a = (7, 0, 8, 0, 0, 9)$

→ $a = (7, 0, 8, 0, 0, 9)$

print ("Count of zero = ", a.count(0))

Output :-

Count of zero = 3

13-12-28

* Dictionary & Sets :-

Dictionary is a Collection of

Key - value pairs.

key value

Syntax :- $a = \{ "key" : "Value" \}$

"Meera" : "python",

"Marks" : 70,

"list" : [1, 2, 3]

→ We can add list in dictionary

How to print :-

`print(a["key"])` output : Value

`print(a["Marks"])` output : 70

`print(a["list"])` output : [1,2,3]

* Notes :- property of python dictionary

- It is unorder. (Check कोई dictionary में नहीं है)
- It is mutable (Dictionary सिर्फ)
- Cannot Contain duplicate key.

Like

`a["Marks"] = 100`

It can change in dictionary value:- Replace 70 by 100

* Dictionary Method :-

Consider the Following dictionary :-

```
a = { "name" : "mca",
      "form" : "Mdaipur",
      "marks" : [60,70,80] }
```

1. `Items()` :-

Returns a list of (key, values) tuples

`print(a.items())`

```
[ name : mca
  form : Mdaipur
  marks : [60,70,80] ]
```

- dict - `Items()`

3. `a[0]` :- Returns a list containing elements
at index 0.
Value :- [1, 2, 3, 4, 5]

4. `a[0:2]` :-
Returns the value in list from index
0 to 2 (including 0 and 2) and
excluding index 3 (given).
Value :- [1, 2]

Output :-
[name : 'Ranu'
 age : 18
 marks : (60, 70, 80)
 friends : 'Abhi']

We also add list

`a + b` (L = a + b [2, 3, 4, 5])

5. `a.get("key")` :-
Returns the value of key
in list.

Output :- [name : 'Ranu'

6. `b[0:2]`,

'name' : 'Ranu',
'age' : 18, 'marks' :
'L' : [1, 2, 3, 4]]

a[0] :- value L

print a

print list of

value L

* print La, Lb
value Od/
value La

print La, La
value La

* Seta :-

else :-
if add
else add
print

else in

n

4. $\text{a.pop}()$ -

Returns a list containing elements

from begining.

Time Complexity :-

Linear time complexity

5. $\text{a.append}(x)$ -

Adds the value to list from after
the last element with a additional value
by using pass.

a.append([1, friend * 2, 3])

Output :-

Output

[Name is even

Don't like you

Marks is [60, 70, 80]

friend is child]

We also add list

a.append([1, "a", [2, 3, 4, 5]])

6. $\text{a.get}("from")$ -

Returns the value of object

key.

Output : Name

a = [1, 2,

3, Harry's pollen],

"area": [1, 2, 3, 4]

"L": [1, 2, 3, 4]]

2 a.keys () :-

Returns a list containing dictionary keys.
point (a.keys ())
[name, from, marks]

3 a.update () :-

Add the value in list form update the dictionary with supplied value key-value() pairs.

a.update ({ "friend" : 8999 "3 })
point (a)

Output

[Name : MCA
from : Udaipur
marks : [60, 70, 80]
friend : 8999]

We also add list

a.update ({ "ds" : [2, 4, 6, 8] })

4 a.get ("from") :-

Returns the value of specific keys.

Output : Udaipur

a = { 1 : 2 ,

{ "Harry" : "potter" } ,

'MCA' : 1 st sem '

{ L : [1, 2, 3, 4] }

a (" point C -

Output - cl

Type Cast of

point L

* point (a.get

point Output

point (a[

point (a.get

point (a[

* Sets :-

S = {

S.add

S.add

point (

→ Sets in

`a["point"] = type(a.keys) ("mca")`

Output - dictionary

Type Cast of Dictionary in list

`point (point (list(a)))`

* `point (a.get ("mca"))`
`point Output ← 1st Sem`

`point (a ["mca"]) → 1st Sem`

`point (a.get ("mca []")) → Returns none`
`point (a.get [mca]) → Error → module`

* Sets :-

Set is a Collection of non-repetitive
Element

`s = Set()`

`s.add (1)`

`s.add (2)`

`point (s)`

Output `s {1, 2, 3}`

of specified

→ Sets in python are data type containing unique values

* Properties of Sets :-

- Sets are unordered :- Element are unorderd
- Sets are unindexed :-
- There is no way to change item in sets.
- Sets can not contain duplicate values.

* Operation on Sets :-

Consider the following sets :-

$$S = \{ 1, 8, 2, 3 \}$$

1. len () :-

Returns the length of the set.

point (len(s))

Output - 4

2. Remove () :-

Update the set & removes the element
in the function.

point (s.remove(8))

Output { 1, 2, 3 }

3. Pop () :-

Removes one an arbitrary Element from
the set & returns the element removed.

point (s.pop(1))

Output →

PAGE NO.

4

Clear () :-

Empty

point (s.clear())

Output →

5

Union () :-

Ret both

point (s.union({1, 8, 2}))

6. Intersection () :-

Item or
point (s.

Output :- { }

* Important Function

Ex:- S = { 1,

point (type)

point (s

 = a = { 3
 (type)

 S = Set U
 point (type)

4 Clear () :-

Empty is the sets

point (S. Clear ())

Output →

5 Union () :-

Returns a new set with all items from both sets.

point (S. Union (E 8, 113))
E 1, 8, 2, 3, 113

6 Intersection () :-

Returns a set which contains only item in both sets.

point (S. Intersection (E 8, 113))

Output :- E 8 }

* Important Facts of sets :-

Ex:- S = { 1, 2, 3, 4, 5, 6, 7, 8 }

point (type (S)) → Class set

point (S) → { 1, 2, 3, 4, 5, 6, 7 }

1 a = { 3 }

(type (a)) → Dictionary < class 'dict' >

S = Set()

point (type (S)) → Set (Class 'Set')

3 $a = \{1, 2, [4, 5]\}$

It will generate an error
Set & list are immutable

Point (a) → Unhashable type : "List"

↓
List is mutable

4 $a = \{1, 2, 3, 4\}$ { Tuple & set are immutable so it
Point (a) not change }

Output → $\{1, 2, 3, 4\}$

5 $a = \{1: 2\}$ → It is dictionary
Point (Type(a)) → Dictionary < (class 'dict')

$a = \{1, 2, 3\}$

$a.add(\{5, 6\})$

→ Error Unhashable type : 'set'

6 $a.add((7, 8, 9))$

Point (a)

Output → ~~(1, 2, 3, (7, 8, 9))~~
 $\{(7, 8, 9), 1, 2, 3\}$

PAGE NO.

19-12-22

C

→ Write a program
words with
provide us
mydict = {}

point ("op:
a=input()
print("the")

Output :-
Option one :
Enter the
The meaning

→ Write a program
user and

→ $s1 = int()$

$s2 = int()$

$s3 = int()$

$s4 = int()$

$s5 = int()$

$s6 = int()$

$s7 = int()$

$s8 = int()$

$s = \{s1, s2, s3, s4, s5, s6, s7, s8\}$

point

Chapter : 5 Practice Set

1 Write a program to create a dictionary of Hindi words with values as their English translation provide user with an option to look it up!

→ mydict = { "kitab" : "book",
 "Kalam" : "pen",
 "hathee" : "Elephant",
 "ghoda" : "Horse" }

print ("option are : ", mydict.keys())

a = input("Enter the Hindi word : \n")

print ("the meaning of your words is : ", mydict.get(a))

Output:-

Option are : dict_keys(['kitab', 'Kalam', 'hathee', 'ghoda'])

Enter the Hindi word : Kitab

The meaning of your words is : book

2 Write a program to input Eight numbers from the user and display all the unique numbers ().

→ s1 = int(input("Enter no 1 : "))

s2 = int(input("Enter no 2 : "))

s3 = int(input("Enter no 3 : "))

s4 = int(input("Enter no 4 : "))

s5 = int(input("Enter no 5 : "))

s6 = int(input("Enter no 6 : "))

s7 = int(input("Enter no 7 : "))

s8 = int(input("Enter no 8 : "))

s = {s1, s2, s3, s4, s5, s6, s7, s8}

print(s)

Output :-

Enter no 1 : 2

Enter no 2 : 4

Enter no 3 : 5

Enter no 4 : 6

Enter no 5 : 7

Enter no 6 : 8

Enter no 7 : 9

Enter no 8 : 4

Output :- {2, 4, 5, 6, 7, 8, 9}

Q.3 Can we have a set with (18 int) and "18" (string) as a value in it?

→ $s = \{18, "18"\}$

point (s)

Output : (18, '18')

Q.4 What will be the length of following set s :

$s = \{1\}$

$s.add(80)$

$s.add(80.0)$

$s.add("80") \Rightarrow$ Length of s after these operations

→ $s = \{80, 80.0, "80"\}$

point (len(s))

$a = 3$

Output :- 3.

Q.5. $s = \{3\}$

What is the type of s ?

→ $a = \{3\}$

point (type(a))

Output :- {3}

Q.6 Create an em
center their
use keys
names are

→ $s = \{3\}$

$s.update("E")$

$s.update("E")$

$s.update("E")$

$s.update("E")$

$s.update("E")$

point (s)

Output :- E

Q.7 If name c
paragra

→ $s.update$
 $point(s)$

Output :-

Q.8 If language

→ $s.update$
 $point(s)$

Output :-

<class 'dict'>

Q.6 Create an empty dictionary. Allow 4 friends to enter their favourite language as values and use keys as their names. Assume that the names are unique.

→ $s = \{ \}$

$s.\text{update}(\{ "Ruchi" : "Python" \})$

$s.\text{update}(\{ "Sonu" : "Python" \})$

$s.\text{update}(\{ "Kriti" : "Python" \})$

$s.\text{update}(\{ "Maya" : "HTML" \})$

print(s)

Output :-

{'Ruchi': 'python', 'Sonu': 'python'}

'Kriti': 'python', 'Maya': 'HTML' }

Q.7 If name of 2 friends are same, what will happen to the program in problem 6.

→ $s.\text{update}(\{ "Ruchi" : "Java" \})$

print(s)

Output :-

{'Ruchi': 'Java', 'Sonu': 'python', 'Kriti': 'python',
'Maya': 'HTML' }

Q.8 If language of two friends are same, what will happen to the program in problem 6?

→ $s.\text{update}(\{ "Sonu" : "Java" \})$

print(s)

Output :—

```
{'Ruchi': 'Java', 'Sonu': 'Java', 'Kriti': 'Python',  
'maya': 'HTML'}
```

- Q. 2 Can you change the value inside list which
Command in set S
 $S = \{8, 7, 12, "Harry", 1, 2\}$

→

* Conditional Expression :—

In python programming we
must be able to execute instruction on a
Condition (S) being met.

→ This is what conditions are for.

If else or elif in python :—

If else and elif statement
are mostly multiway decision taken by our program
due to create condition in our code.

Syntax :—

```
If (Condition 1) : True  
    print ("yes")      → yes  
elif (Condition 2) : True  
    print ("No")       → No  
else :
```

* Relational

to eval

```
a = 22  
if (a > 10)  
    print ("Greater")  
elif (a > 7)  
    print ("Greater")  
else:  
    print ("Less")
```

Output :—

```
a = 22  
if (a > 10)  
    print ("Greater")  
if (a > 50)  
    print ("Greater")  
if (a > 8):  
    print ("Greater")
```

Print ("maybe") → maybe

If else and elif

a = 22

if (~~a < 10~~) :

print ("Greater") → Greater.

elif (a > 7) :

print ("Greater than 7")

else :

print ("Lesser")

Output :— Greater than 7

a = 22

if (~~a < 10~~) :

print ("Greater")

if (a > 50) :

print ("Greater than 50")

if (a > 8) :

print ("Greater than 8")

elif statement

programmer

use.

True

→ yes

True

No

* Relational Operator :—

Relational Operator are used
to evaluate the Condition inside the if statement.

>, <, >=, <=, ==, !=

Logical Operator :-

logical operator in python logical operator Condition statement AND, OR, NOT

AND →

True if both operands are true. Else false.

OR → True if at least one operand is true. Else false.

NOT →

Inverse true to false & false to true.

Example :-

```
1 Age = input ("Enter your age")
if (age > 34 and age < 56) :
    print ("you can work with us") → True
else :
    print ("you can't work with us")
```

```
2 Age = input ("Enter your age")
if (age > 34 OR age < 56) :
    print ("you can work with us")
else
    print ("you can't work with us")
```

Output :-

Note :-
IS :-

point two ok
a = None
if (a is None) :
print ("None")
else :
print ("Not None")

IN :- It is use of
1. Whether
2. To iterate

⇒ a =
print

Important Note

- 1 Where can be
- 2 Last else is inside a

9-1-23

Loops :-

for
instruction

Types of
Loops

Note :-

IS :- point two object is (`= =`)

`a = None`

`if (a is None) :`

`print ("yes")`

`else :`

`print ("No") .`

In - Check

IN :- It is use for above purpose

1 Whether item is in list or not.

2 To iterate we use in among List.

$\Rightarrow a = [45, 60, 90]$

`print (550 in a) —> False`

Important Note :-

1 Where can be any no. of elif statement.

2 Last else is Executed only if all the condition inside elif's fail.

9-1-23

Loops :-

Loops make it easy for a programmer to tell the Computer which set of instruction to repeat & how.

Types of loops in python :-

Primarily there are two

Type loops in python ..

1. While loop
2. for loop.

White loops :-

The syntax of while loops like this

While Condition :-

Body of the loop :-

The block keeps executing until the condition is true.

Ex:- $i = 0$

while $i > 10$:
print ("yes")



$i = 0$

while $i < 10$:

print ("yes") → yes infinite

$i = i + 1$ $0 \rightarrow 1$ yes,



$i = 0$

while $i < 10$:

print ("yes" + str(i)) → yes 0
 yes 1

$i = i + 1$

In while loops the condition is checked if it evaluates to true, the body of loop is executed otherwise.

The loop is checked and the condition

Q. 1 Write a program loop?



$i = 1$
while $i < 10$
print (i)
 $i = i + 1$

Note :-
if i,
the loop

Q. 2

Write a program



$i = ["a"]$
while $i < 1$
print (i)
 $i = i + 1$



$b = ["abcs"]$
Count = 0
while Count < len(b):
print (b[Count])
Count = Count + 1

The loop is entered, the process (Condition Check, and Execution) is Continue until the Condition is true.

Q.1 Write a program to print 1 to 10 using while loop?

→ $i = 1$

while $i <= 10$:

print(i)

$i = i + 1$

Note :-

If the Condition Never becomes false
the loop keeps getting Executed.

Q.2. Write a program to print list using while loop?

→ $i = ["a", "b", "c", "d"]$

while $i < 4$:

print(str(i))

$i = i + 1$

→ $l = ["Sakshi", "Ruchi", "Sonu", "Kriti", "Maya"]$

Count = 0

while Count $<= \text{len}(l)$:

print(l[Count])

Count = Count + 1

for loop :-

for loop as we do iterate thru
a sequence like like, tuple or string
[Streets]

Syntax of for loop look like this.

Ex :- $l = [1, 7, 8,]$

for item in l
print(item) $\rightarrow 1, 7, 8, 4.0$

\rightarrow fruit ["Mango", "Apple", "Orange", "Watermelon"]
for item in fruit
print(item) \rightarrow mango, Apple, Orange, Watermelon

Range function in python

\rightarrow The range function in python is used to generate
sequence of numbers.

\rightarrow for i in range(8): / (1, 8):
print(i) \rightarrow

0, 1, 2, 3, 4, 5, 6, 7

We can also specify the start, stop, and step
- size as for loop.

Syntax :- Range [Start, Stop, Step - size]

Ex :-

for i
print

Output :-

for i
print

Output :-

for i
print

Output :-

for loop with

with a
of loop
the

\rightarrow

I
for

or

Output :-

\rightarrow

for
print
if

Ex:-

```
for i in range (8) :
    print (i)
```

Output :- 0, 1, 2, 3, 4, 5, 6, 7

```
for i in range (1,8) :
    print (i)
```

Output :- 1, 2, 3, 4, 5, 6, 7

```
for i in range (1,8,2) :
    print (i)
```

Output :- 1, 3, 5, 7

for loop with else :-

An optional *else* can be used with a *for* loop if the code with the *for* loop is to be executed when the loop is exhausted.



```
l = [1, 7, 8]
```

```
for item in l :
```

```
    print (item)
```

```
else :
```

```
    print ("Done")
```

Output :- 1, 7, 8

Done.



```
for i in range (0,80) :
```

```
    print (i)
```

```
if i == 3 :
```

PAGE NO. 
break ;
else :
print ("This is inside else")

Output :-
0 This is inside else
1 This is inside else
2 This is inside else
3 This is inside else

When the for loop is not successfully else will terminate.
not be executed to check for natural termination of loop for else can be used.

* Break Statement :-
Break is used to come out of the loop when encounter it.
→ If it instructs the programme to exit the loop now.

* Continue Statement :-
Continue is used to skip the current iteration of the loop and continue with the next one.
→ ... 11. Programme to skip this iteration

→ for i :
print
if i =
 Con
print

Output :-

* Pass Statement

in the pyth
→ l =
for
pas

* Pass &

* Function and

def marks
as = l
print
return

A function is
task of c

→ for i in range (4)
 print (" printing ")
 if i == 2:
 Continue
 print (i)

Output :- 0 printing
 1 printing
 2 printing
 4 printing

* Pass Statement :-

else will

l termination

to come out

Pass is the Null statement
 in the python it instruct to "do Nothing"

→ l = [1, 7, 8]
 for item in l
 pass

* Pass ko Comment Try karna *

To exit the

* Function and Recursion :-

```
def marks (a):
  a1 = (a [0], a [1], a [2], a [3], a [4]) / 5
  # print (a)
  return a1
```

used to stop
 loop or Continue

in this iteration

A function is a group of statement performing a specific task it can be reuse by the programming in