

**Program Name** Write and execute the program  
demo1.py

```
values = [10, 20, 30]
```

```
for value in values:  
    print(value)
```

**output**

**Program Name** Write and execute the program  
demo2.py

```
values = [10, 20, 30]
```

```
for _ in values:  
    print(_)
```

**output**

**Program Name** Write and execute the program demo3.py

```
values = [10, 20, 30]
```

```
for value in values:  
    print(value+2)
```

**output**

**Program Name** Write and execute the program demo4.py

```
values = [10, 20, 30]  
new = []
```

```
for value in values:  
    new.append(value+2)
```

```
print(new)
```

**output**

**Program Name** Write and execute the program  
demo5.py

```
values = [10, 20, 30]
new = [value+2 for value in values]

print(new)
```

**output**

**Program Name** Write and execute the program  
demo6.py

```
values = [10, 20, 30]
new = [value+2 for value in values]

print(new)
```

**output**

**Program Name** Write and execute the program demo7.py

```
values = [10, "daniel", 20, "john", 30]
```

```
for value in values:  
    print(value)
```

**output**

**Program Name** Write and execute the program demo8.py

```
values = [10, "daniel", 20, "john", 30]
```

```
for value in values:  
    if isinstance(value, int):  
        print(value)
```

**output**

**Program Name** Write and execute the program  
demo9.py

```
values = [10, "daniel", 20, "john", 30]
```

```
for value in values:  
    if isinstance(value, str):  
        print(value)
```

**output**

**Program Name** Write and execute the program  
demo10.py

```
values = [10, "daniel", 20, "john", 30]  
new = [value for value in values if isinstance(value, int)]
```

```
print(new)
```

**output**

**Program Name** Write and execute the program  
demo11.py

```
values = [10, "daniel", 20, "john", 30]
new = [value for value in values if isinstance(value, str)]

print(new)
```

**output**

**Program Name** Write and execute the program  
demo12.py

```
values = [10, "daniel", 20, "john", 30]
new = [value for value in values if isinstance(value, float)]

print(new)
```

**output**

**Program Name** Write and execute the program demo13.py

```
values = [10, 20, 30]
result = min(values)
print(result)
```

**output**

**Program Name** Write and execute the program demo14.py

```
values = [10, 20, 30]
result = max(values)
print(result)
```

**output**

**Program Name** Write and execute the program demo15.py

```
values = [10, 20, 30]
result = sum(values)
print(result)
```

**output**

**Program Name** Write and execute the program demo16.py

```
def total(values):  
    t = 0  
    for value in values:  
        t = t + value  
    return t
```

```
a = [10, 20, 30]  
result = total(a)  
print(result)
```

**output**

**Program Name** Write and execute the program demo17.py

```
def multiply(values):  
    t = 1  
    for value in values:  
        t = t * value  
    return t
```

```
a = [2, 3, 4]  
result = multiply(a)  
print(result)
```

**output**



**Program Name** Write and execute the program demo18.py

```
def big_num_in_list(values):
    big = values[0]
    for value in values:
        if value > big:
            big = value
    return big

a = [10, 20, 30, 40]
result = big_num_in_list(a)
print(result)
```

**output**

**Program Name** Write and execute the program demo19.py

```
def small_num_in_list(values):
    small = values[0]
    for value in values:
        if value < small:
            small = value
    return small

a = [10, 20, 30, 40]
result = small_num_in_list(a)
print(result)
```

**output**

### Program

Count the number of strings from a given list of strings. The string length is 2 or more and the first and last characters are the same.

### Name

demo20.py

```
def match_words(words):  
    ctr = 0  
  
    for word in words:  
        if len(word) > 1 and word[0] == word[-1]:  
            ctr = ctr + 1  
  
    return ctr  
  
x = ['abc', 'xyz', 'aba', '1221']  
result = match_words(x)  
print(result)
```

### output

**Program** Check list is empty or not

**Name** demo21.py

```
values = []  
if not values:  
    print("List is empty Boss")
```

### output

**Program Name** Clone or copy a list.  
demo22.py

```
original_list = [10, 22, 44, 23, 4]
new_list = list(original_list)

print(original_list)
print(new_list)
```

**output**

**Program Name** Find the list of words that are longer than n  
demo23.py

```
def long_words(n, str):
    word_len = []
    txt = str.split(" ")

    for x in txt:
        if len(x) > n:
            word_len.append(x)

    return word_len

a = "The quick brown fox jumps over the lazy dog"
result = long_words(3, a)
print(result)
```

**output**

**Program Name** Write and execute the program  
demo24.py

```
color = ['Red', 'Green', 'White', 'Black', 'Pink', 'Yellow']  
values = enumerate(color)
```

```
for value in values:  
    print(value)
```

**output**

**Program Name** Write and execute the program  
demo25.py

```
color = ['Red', 'Green', 'White', 'Black', 'Pink', 'Yellow']  
values = enumerate(color)
```

```
for k, v in values:  
    print(k, v)
```

**output**

**Program Name** Shuffle and print a specified list.  
demo26.py

```
from random import shuffle
```

```
color = ['Red', 'Green', 'White', 'Black', 'Pink', 'Yellow']  
print(color)
```

```
shuffle(color)  
print(color)
```

**output**

**Program Name** Write and execute the program  
demo27.py

```
a = [1, 3, 5, 7, 9]  
b = [1, 2, 4, 6, 7, 8]  
d1 = list(set(a) - set(b))
```

```
print(d1)
```

**output**

**Program Name** Write and execute the program demo28.py

```
s = ["a", "b", "c", "d"]  
result = "".join(s)  
print(result)
```

**output**

**Program Name** Write and execute the program demo29.py

```
num = [10, 30, 4, -6]  
print(num.index(30))
```

**output**

**Program Name** Flatten a shallow list demo30.py

```
import itertools  
  
original_list = [[2, 4, 3], [1, 5, 6], [9], [7, 9, 0]]  
new_merged_list = list(itertools.chain(*original_list))  
print(new_merged_list)
```

**output**

**Program**      append a list to the second list  
**Name**           demo31.py

```
list1 = [1, 2, 3, 0]
list2 = ['Red', 'Green', 'Black']
final_list = list1 + list2

print(final_list)
```

**output**

**Program**      Select an item randomly from a list  
**Name**           demo32.py

```
import random

color_list = ['Red', 'Blue', 'Green', 'White', 'Black']
result = random.choice(color_list)
print(result)
```

**output**

**Program Name**      Get unique values from a list  
demo34.py

```
my_list = [10, 20, 30, 40, 20, 50, 60, 40]
my_set = set(my_list)
my_new_list = list(my_set)

print("Original List : ", my_list)
print("List of unique numbers : ", my_new_list)
```

**output**

**Program Name**      Frequency of elements in a list  
demo35.py

```
import collections

my_list = [10, 10, 10, 10, 20, 20, 20, 20, 40, 40, 50, 50, 30]
ctr = collections.Counter(my_list)

print("Original List : ", my_list)
print("Frequency of the elements in the List : ", ctr)
```

**output**



**Program Name** Write and execute the program  
demo36.py

```
color1 = "Red", "Green", "Orange", "White"  
color2 = "Black", "Green", "White", "Pink"  
  
print(set(color1) & set(color2))
```

**output**

**Program Name** Compute the difference between two lists  
demo37.py

```
from collections import Counter  
  
color1 = ["red", "orange", "green", "blue", "white"]  
color2 = ["black", "yellow", "green", "blue"]  
  
counter1 = Counter(color1)  
counter2 = Counter(color2)  
  
print("Color1-Color2: ", list(counter1 - counter2))  
print("Color2-Color1: ", list(counter2 - counter1))
```

**output**

**Program Name** Concatenate elements of a list  
demo38.py

```
color = ['red', 'green', 'orange']
```

```
print('-'.join(color))
```

```
print("".join(color))
```

**output**

**Program Name** Write and execute the program  
demo39.py

```
values = [1, 2, 3, 4]
```

```
result = ['emp{0}'.format(value) for value in values]
```

```
print(result)
```

**output**

**Program Name** Iterate over two lists simultaneously  
demo40.py

```
num = [1, 2, 3]
color = ['red', 'white', 'black']
```

```
for (k, v) in zip(num, color):
    print(k, v)
```

**output**

**Program Name** Find items starting with a specific character from a list.  
demo41.py

```
def test(values, char):
    result = [value for value in values if value.startswith(char)]
    return result
```

```
text = ["abcd", "abc", "bcd", "bkie", "cder", "cdsw", "sdfsd",
        "dagfa", "acjd"]
```

```
print(text)
char = "a"
print("\nItems start with", char, "from the said list:")
result2 = test(text, char)
print(result2)
```

**output**

**Program** Flatten a given nested list structure  
**Name** demo42.py

```
n_list = [[20, 30], [60, 70, 80], [90, 100, 110, 120]]
```

```
flat_list = [item for sublist in n_list for item in sublist]  
print(flat_list)
```

**output**

**Program** Round each item in a list of floats to 2 decimal places?  
**Name** demo43.py

```
values = [5.3187645388, 0.587699, 88.467281467382647]  
rounded_values = [round(value, 2) for value in values]
```

```
print(values)  
print(rounded_values)
```

**output**

**Program Name** Convert list of float values into list of int values  
demo44.py

```
float_values = [5.318, 0.58, 88.4]
int_values = [int(value) for value in float_values]

print(float_values)
print(int_values)
```

**output**

**Program Name** Find the maximum and minimum values  
demo45.py

```
def max_min_val(values):
    a = [value for value in values if isinstance(value, int)]
    b = [value for value in values if isinstance(value, int)]
    big = max(a)
    small = min(b)
    return (big, small)

list_val = ['Python', 3, 2, 4, 5, 'version']
result = max_min_val(list_val)
print(result)
```

**output**

**Program Name** Find the maximum and minimum values  
demo46.py

```
def extract_string(values, l):  
    result = [value for value in values if len(value) == l]  
    return result
```

```
str_list1 = ['Python', 'list', 'exercises', 'practice', 'solution']  
l = 8  
print(str_list1)  
result2 = extract_string(str_list1, l)  
print(result2)
```

**output**

**Program Name** Extract specified size of strings from a give list of string values  
demo47.py

```
def extract_string(values, l):  
    result = [value for value in values if len(value) == l]  
    return result
```

```
str_list1 = ['Python', 'list', 'exercises', 'practice', 'solution']  
l = 8  
print(str_list1)  
result2 = extract_string(str_list1, l)  
print(result2)
```

**output**

**Program Name**      Average of list  
demo48.py

```
a = [1, 1, 3, 4, 4, 5, 6, 7, 9]
result = sum(a) / len(a)

print(result)
```

**output**

**Program Name**      Average of two lists  
demo49.py

```
a = [1, 1, 3, 4, 4, 5, 6, 7, 9]
b = [1, 2, 3, 4, 7, 5, 6, 7]
c = a + b

result = sum(c) / len(c)

print(result)
```

**output**

**Program Name** Count Integers in a given mixed list  
demo50.py

```
values = [1, 'abcd', 3, 1.2, 4, 'xyz', 5, 'pqr', 7, -5, -12.22]
```

```
ctr = 0
```

```
for value in values:
```

```
    if isinstance(value, int):
```

```
        ctr = ctr + 1
```

```
print(values)
```

```
print(ctr)
```

**output**

**Program Name** Count strings in a given mixed list  
demo51.py

```
values = [1, 'abcd', 3, 1.2, 4, 'xyz', 5, 'pqr', 7, -5, -12.22]
```

```
ctr = 0
```

```
for value in values:
```

```
    if isinstance(value, str):
```

```
        ctr = ctr + 1
```

```
print(values)
```

```
print(ctr)
```

**output**



### Program

Remove all elements from a given list that are present in another list

### Name

demo52.py

```
list1 = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

```
list2 = [2, 4, 6, 8]
```

```
result = [value for value in list1 if value not in list2]
```

```
print(result)
```

### output

### Program

Reverse strings in a given list of string values

### Name

demo53.py

```
def reverse_strings_list(values):  
    a = [value[::-1] for value in values]  
    return a
```

```
colors_list = ["Red", "Green", "Blue", "White", "Black"]
```

```
result = reverse_strings_list(colors_list)
```

```
print(colors_list)
```

```
print(result)
```

### output

**Program Name**      Sort a given list of strings (numbers) numerically  
demo54.py

```
values = ['4', '12', '100', '200', '-12', '-500']  
result = [int(value)      for value in values]  
result.sort()
```

```
print(values)  
print(result)
```

**output**

**Program Name** Remove specific words from a given list  
demo55.py

```
def remove_words(list1, remove_words):  
    for word in list(list1):  
        if word in remove_words:  
            list1.remove(word)  
    return list1  
  
colors = ['red', 'green', 'blue', 'white', 'black', 'orange']  
print(colors)  
  
remove_colors = ['white', 'orange']  
result = remove_words(colors, remove_colors)  
  
print(remove_colors)  
print(result)
```

**output**

**Program Name** Reverse a given list of lists  
demo56.py

```
def reverse_list_of_lists(list1):  
    return list1[::-1]  
  
colors = [['orange', 'red'], ['green', 'blue'], ['white', 'black', 'pink']]  
print(colors)  
print(reverse_list_of_lists(colors))  
  
nums = [[1,2,3,4], [0,2,4,5], [2,3,4,2,4]]  
print(nums)  
print(reverse_list_of_lists(nums))
```

**output**

**Program Name** Remove the **None** value from a given list  
demo57.py

```
def remove_none(nums):  
    result = [x for x in nums if x is not None]  
    return result  
  
nums = [12, 0, None, 23, None, -55, 234, 89, None, 0, 6, -12]  
print(nums)  
print(remove_none(nums))
```

**output**

**Program Name** List of strings and characters to a single list of characters  
demo58.py

```
def l_strs_to_l_chars(lst):  
    result = [i for element in lst for i in element]  
    return result
```

```
colors = ["red", "white", "a", "b", "black", "f"]  
print(colors)  
print(l_strs_to_l_chars(colors))
```

**output**

**Program Name** Adding value to list  
demo59.py

```
def define(a):  
    j = "@gmail.com"  
    return [i + j for i in a]
```

```
values = ['john', 'matthew', 'peter', 'daniel', 'samuel']
```

```
print(values)  
print(define(values))
```

**output**

**Program  
Name**

Access keys  
demo60.py

```
def keys_only(students):  
    return list(students.keys())  
  
students = { 'Laura': 10, 'Spencer': 11, 'Bridget': 9, 'Howard ': 10}  
print(students)  
print(keys_only(students))
```

**output**

**Program  
Name**

Access keys  
demo61.py

```
def keys_only(students):  
    return list(students.values())  
  
students = { 'Laura': 10, 'Spencer': 11, 'Bridget': 9, 'Howard ': 10}  
print(students)  
print(keys_only(students))
```

**output**

### Explain about isinstance(p1, p2)

- ✓ isinstance(p1, p2) is a predefined function in python.
- ✓ By using this function we can check the type of object and it returns boolean value.

**Program Name**      isinstance(p1, p2) function  
demo62.py

```
a = 10
b = "Daniel"

print(isinstance(a, int))
print(isinstance(a, str))

print(isinstance(b, int))
print(isinstance(b, str))
```

**output**

```
True
False
False
True
```

**Program Name**      list comprehension  
demo63.py

```
values = ['daniel', 10, 'mohan', 20, 'veeru', 'mouli', 30]
result = [value for value in values if isinstance(value, int)]

print(values)
print(result)
```

**output**

```
['daniel', 10, 'mohan', 20, 'veeru', 'mouli', 30]
[10, 20, 30]
```

**Program** list comprehension  
**Name** demo64.py

```
values = ['daniel', 10, 'naresh', 20, 'veeru', 'prasad', 30]
result = [value.upper() for value in values if isinstance(value,
str)]

print(values)
print(result)
```

**output**

```
['daniel', 10, 'naresh', 20, 'veeru', 'prasad', 30]
['DANIEL', 'NARESH', 'VEERU', 'PRASAD']
```