

1. What is a List?

```
1 A list is a data type in Python that is a mutable, or changeable, ordered sequence of
  elements. Each element or value that is inside of a list is called an item. Just as
  strings are defined as characters between quotes, lists are defined by having values
  between square brackets [ ] .
2 List Properties:
3 - mutable
4 - ordered
5 - duplicates are allowed
6 - enclosed by []
7 - used to store any datatype >> (heterogeneous data type)
8 - it requires more memory
9 - lists are slower than tuple
```

2. What is a Tuple?

```
1 Tuples are used to store multiple items in a single variable. Tuple is one of 4
  built-in data types in Python used to store collections of data. A tuple is a
  collection which is ordered and immutable.
2 Tuple Properties:
3 - Immutable(unchangeable)
4 - ordered
5 - duplicates are allowed
6 - enclosed by ()
7 - used to store any datatype >> (heterogeneous data type)
8 - it requires less memory
9 - tuple are faster than list
```

3. What is the difference between List and Tuple?

```
1 The key difference between the tuples and lists is that while the tuples are
  immutable objects the lists are mutable. This means that tuples cannot be changed
  while the lists can be modified.
2 Tuples are more memory efficient than the lists.
3 When it comes to the time efficiency, again tuples have a slight advantage over the
  lists especially when lookup to a value is considered.
```

4. Python Program to find the largest element in the list

In [1]:

```
1 l1 = [1,2,5,8,6,4,5,21,564,1,84,65,4,614,856,514]
2 print(max(l1))
```

856



5. Python program to interchange first and last elements in a list.

In [3]:

```

1 l1 = [1,2,5,8,6,4,5,21,564,1,84,65,4,614,856,514]
2 l2 = l1[0]
3 print(f"before interchange first value is {l1[0]} and last value is {l1[-1]}")
4 l1[0] = l1[-1]
5 l1[-1] = l2
6 print(f"after interchange first value is {l1[0]} and last value is {l1[-1]}")

```

before interchange first value is 1 and last value is 514

after interchange first value is 514 and last value is 1



6. Python program to swap two elements in a list

In [13]:

```

1 l1 = [1,2,5,8,6,4,5,21,564,1,84,65,4,614,856,514]
2 a = int(input("Enter index first number in list to swap - "))
3 b = int(input("Enter index of second number in list to swap - "))
4 print(f"before interchange first value is {l1[a]} and last value is {l1[b]}")
5 c = a
6 a = b
7 b = c
8 print(f"after interchange first value is {l1[a]} and last value is {l1[b]}")
9

```

Enter index first number in list to swap1

Enter index of second number in list to swap3

before interchange first value is 2 and last value is 8

after interchange first value is 8 and last value is 2



7. Python program to Reverse a List

In [25]:

```

1 l1 = [1,2,5,8,6,4,5,21,564,1,84,65,4,614,856,514]
2 l2 = l1[::-1]
3 print(l2)

```

[514, 856, 614, 4, 65, 84, 1, 564, 21, 5, 4, 6, 8, 5, 2, 1]



8. Python program to count occurrences of an element in a list

In [31]:

```
1 l1 = [1,2,5,8,6,4,5,21,564,1,84,65,4,614,856,514]
2 a = int(input("Enter the number to find count -"))
3 b=0
4 for i in l1:
5     if i == a:
6         b = l1.count(a)
7 print("Count -",b)
```

Enter the number to find count -1

Count - 2



9. Python program to find the sum of elements in a list

In [32]:

```
1 l1 = [1,2,5,8,6,4,5,21,564,1,84,65,4,614,856,514]
2 l2 = sum(l1)
3 print(l2)
```

2754



10. Python program to Multiply all numbers in the list

In [33]:

```
1 l1 = [1,2,5,8,6,4,5,21,564,1,84,65,4,614,856,514]
2 for i in l1:
3     i*= i
4 print(i)
```

264196



11. What are the ways to find the length of a list

In [36]:

```
1 l1 = [1,2,5,8,6,4,5,21,564,1,84,65,4,614,856,514]
2 #fisrt
3 print(len(l1))
4 #second
5 count=0
6 for i in l1:
7     count +=1
8 print(count)
9
10
```

16

16



12. Python program to find the smallest and largest number in a list (Without min-max function)

In [41]:

```

1 l1 = [1,2,5,8,6,4,5,21,564,1,84,65,4,614,856,514]
2 l2 = sorted(l1)
3 print(f"min - {l2[0]}")
4 print(f"max - {l2[-1]}")

```

```

min - 1
max - 856

```



13. Python Program to find the area of a circle

In [42]:

```

1 from math import pi
2 r = eval(input("Enter radius"))
3 area = pi*r**2
4 print(area)

```

```

Enter radius2
12.566370614359172

```



14. Take inputs from the user to make a list. Again take one input from the user and search it in the list and delete that element, if found. Iterate over a list using for loop.

In [62]:

```

1 l = input("Enter the input -").split(" ")
2 print(l)
3 inp2 = input("Enter value to delete - ")
4 for i in l:
5     if i == inp2:
6         l.remove(inp2)
7         print("deleted")
8         break
9     else:
10        print("Value not found")
11 for i in l:
12     print(i)

```

```

Enter the input -1 2 3 4
['1', '2', '3', '4']
Enter value to delete - 1
deleted
2
3
4

```



15. You are given a list of integer elements. Make a new list that will store a square of elements of the previous list. (With and without list comprehension) i. Input_list = [2,5,6,12] ii. Output_list = [4,25,36,144]

In [2]:

```

1 l1 = [2,5,6,12]
2 a = 1
3 l2=[]
4 for i in l1:
5     a = i**2
6     l2.append(a)
7 print("Without list comprehension",l2)
8 l3 =[i**2 for i in l1]
9 print("list comprehension",l3)

```

Without list comprehension [4, 25, 36, 144]

list comprehension [4, 25, 36, 144]

WAP to create two lists, one containing all even numbers and the other containing all odd numbers between 0 to 151

In [95]:

```

1 l1 =[]
2 l2 =[]
3 for i in range(0,152):
4     if i % 2 == 0:
5         l1.append(i)
6     else:
7         l2.append(i)
8 print("even list - ",l1)
9 print("odd list - ",l2)
10

```

even list - [0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96, 98, 100, 102, 104, 106, 108, 110, 112, 114, 116, 118, 120, 122, 124, 126, 128, 130, 132, 134, 136, 138, 140, 142, 144, 146, 148, 150]

odd list - [1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67, 69, 71, 73, 75, 77, 79, 81, 83, 85, 87, 89, 91, 93, 95, 97, 99, 101, 103, 105, 107, 109, 111, 113, 115, 117, 119, 121, 123, 125, 127, 129, 131, 133, 135, 137, 139, 141, 143, 145, 147, 149, 151]

17. Python program to Count Even and Odd numbers in a List

In [96]:

```
1 l1 = [1,2,5,8,6,4,5,21,564,1,84,65,4,614,856,514]
2 count = 0
3 count1 = 0
4 for i in l1:
5     if i % 2 == 0:
6         count+=1
7     else:
8         count1 +=1
9 print("even count ",count)
10 print("odd count ",count1)
```

even count 10
odd count 6



18. WAP to make new lists, containing only numbers which are divisible by 4, 6, 8, 10, 3, 5, 7, and 9 in separate lists for range(0,151)

In [115]:

```
1 l1 =[]
2 l2 =[]
3 for i in range(0,151):
4     if i % 4==0 or i%6==0 or i%8==0 or i%10 ==0 or i%3==0 or i%5==0 or i%7==0 or i%9 ==0:
5         l1.append(i)
6 print("list - ",l1)
```

list - [0, 3, 4, 5, 6, 7, 8, 9, 10, 12, 14, 15, 16, 18, 20, 21, 24, 25, 27, 28, 30, 32, 33, 35, 36, 39, 40, 42, 44, 45, 48, 49, 50, 51, 52, 54, 55, 56, 57, 60, 63, 64, 65, 66, 68, 69, 70, 72, 75, 76, 77, 78, 80, 81, 84, 85, 87, 88, 90, 91, 92, 93, 95, 96, 98, 99, 100, 102, 104, 105, 108, 110, 111, 112, 114, 115, 116, 117, 119, 120, 123, 124, 125, 126, 128, 129, 130, 132, 133, 135, 136, 138, 140, 141, 144, 145, 147, 148, 150]



19. From a list containing ints, strings, and floats, make three lists to store them separately.

In [130]:

```

1 l1 = [1,2.3,5,8,6,4.5,5,21,564,1,84,65,4,"py","hi",856,514]
2 l2 =[]
3 l3 =[]
4 l4 =[]
5 for i in l1:
6     if isinstance(i,int):
7         l2.append(i)
8     elif isinstance(i,float):
9         l3.append(i)
10    elif isinstance(i,str):
11        l4.append(i)
12 print(l2)
13 print(l3)
14 print(l4)

```

```

[1, 5, 8, 6, 5, 21, 564, 1, 84, 65, 4, 856, 514]
[2.3, 4.5]
['py', 'hi']

```



20. What's The Difference Between The Python append() and extend() Methods?

1 append() adds a single element to the end of the list while . extend() can add multiple individual elements to the end of the list.

21. Write a Python program to append a list to the second list

In [149]:

```

1 l1 = ["a", "b", 1, 2, 3]
2 l2 = ["c", "d", 4, 5, 6]
3 l1.append(l2)
4 print(l1)

```

```
['a', 'b', 1, 2, 3, ['c', 'd', 4, 5, 6]]
```



22. Write a Python program to find the third-largest number in a list

In [151]:

```

1 l1 = [1,2,5,8,6,4,5,21,564,1,84,65,4,614,856,514]
2 l1.sort(reverse = True)
3 print(l1[2])

```

564



23. Write a Python program to get the frequency of the elements in a list.

In [86]:

```

1 l1 = [11,21,11,21,31,21,41]
2 for i in l1:
3
4     if l1.count(i) > 1:
5         l2 = l1.count(i)
6         print(l2, end=" ")
7     else:
8         l3 = l1.count(i)
9         print(l3, end=" ")
10    print("frequency of ",i)
11

```

```

2 frequency of 11
3 frequency of 21
2 frequency of 11
3 frequency of 21
1 frequency of 31
3 frequency of 21
1 frequency of 41

```



24. Write a Python program to check whether a list contains a sublist

In [156]:

```

1 l1 = [1,2.3,5,8,[6,4.5],5,21,564,1,84,65,4,["py","hi"],856,514]
2 for i in l1:
3     if isinstance(i,list):
4         print("subList present")
5         break
6 else:
7     print("Sublist not present")
8
9

```

subList present



25. Write a Python program to generate all sublists of a list

In [55]:

```

1 l1 = [1,2,[3,4]]
2 lists = [[]]
3 for i in range(len(l1) + 1):
4     for j in range(i):
5         lists.append(l1[j: i])
6 print(lists)
7
8

```

```

[[], [1], [1, 2], [2], [1, 2, [3, 4]], [2, [3, 4]], [[3, 4]]]

```



26. Write a Python program to find common items from two

lists

In [47]:

```
1 l1 =[1,2,5,4,6,8,56,12,54,36]
2 l2 =[65,54,32,8,4,5,1,56]
3 for i in l1:
4     for j in l2:
5         if j == i:
6             print(j,end=" ")
7
```

1 5 4 8 56 54



27. How to flatten a list in python?

In [168]:

```
1 l1 = [[1,2,4],[5,5,6,4],[5,6,4,8,2,]]
2 l2=[]
3 for i in l1:
4     for j in i:
5         l2.append(j)
6 print(l2)
7
```

[1, 2, 4, 5, 5, 6, 4, 5, 6, 4, 8, 2]



28. How to sort a list in ascending and descending order without using the sort function?

In [181]:

```
1 l1 =[1,2,5,4,6,8,56,12,54,36]
2 #l2 = reversed(l1)
3 #l1.reverse()
4 #print(l1)
5 for i in range(len(l1)):
6     for j in range(i+1, len(l1)):
7         if l1[i] > l1[j]:
8             l1[i],l1[j] = l1[j],l1[i]
9 print(l1)
```

[1, 2, 4, 5, 6, 8, 12, 36, 54, 56]



29. How to sort a tuple?

In [183]:

```

1 #using sorted function
2 t1 = (1,2,5,4,6,8,56,12,54,36)
3 t2 = sorted(t1)
4 t2

```

Out[183]:

[1, 2, 4, 5, 6, 8, 12, 36, 54, 56]



30. Write a Python program to convert a list of multiple integers into a single integer a. [11, 33, 50] >>> 113350

In [218]:

```

1 l1 = [11,33,50]
2
3 for i in l1:
4     print(i,end="")
5 print()

```

113350



31. Difference between del and clear?

```

1 clear function is used to delete all the items in the list
2 del function is used to delete whole list

```

In [220]:

```

1
2 l1 = [11,33,50]
3 l1.clear()
4 print(l1)
5 del l1
6 print(l1)

```

[]

**NameError**

Traceback (most recent call last)

Input In [220], in <cell line: 5>():

```

3 print(l1)
4 del l1
----> 5 print(l1)

```

NameError: name 'l1' is not defined

32. Difference between remove and pop?

In []:

```
1 remove function is used to remove specific item in list
2 pop function is used to remove specific index item in list
```

In [224]:

```
1 l1 = [11,33,50]
2 l1.pop()
3 print(l1)
4 l1.remove(11)
5 print(l1)
```

```
[11, 33]
[33]
```



33. Difference between indexing and Slicing?

In [226]:

```
1 #Indexing is used to obtain individual elements.
2 l1 = [1,2,3,4]
3 print("indexing = ",l1[0])
4 #Slicing is used to obtain a sequence of elements.
5 print("Slicing = ",l1[0:3])
```

```
indexing = 1
Slicing = [1, 2, 3]
```



34. Difference between sort and sorted?

```
1 The primary difference between the two is that list.sort() will sort the list in-
  place, mutating its indexes and returning None , whereas sorted() will return a new
  sorted list leaving the original list unchanged.
```

In [234]:

```
1 l1 = [1,22,65,45,25,5,13,51,3,4]
2 l1.sort()
3 print (l1)
4 l2 = sorted(l1)
5 print(l2)
```

```
[1, 3, 4, 5, 13, 22, 25, 45, 51, 65]
[1, 3, 4, 5, 13, 22, 25, 45, 51, 65]
```



35. Difference between reverse and reversed?

```
1 reverse() actually reverses the elements in the container.
2 reversed() doesn't actually reverse anything, it merely returns an object that can be
  used to iterate over the container's elements in reverse order.
```

In [239]:

```
1 l1 = [1,22,65,45,25,5,13,51,3,4]
2 l1.reverse()
3 print(l1)
4 l2 =reversed(l1)
5 print(l2)
```

```
[4, 3, 51, 13, 5, 25, 45, 65, 22, 1]
<list_reverseiterator object at 0x000002B31B25C700>
```



36. Difference between copy and deep copy?

- 1 Deep copy doesn't reflect changes made to the new/copied object in the original object. Shallow Copy stores the copy of the original object and points the references to the objects.

In [243]:

```
1 from copy import copy, deepcopy
2 a = ["a","b","c",1,2,3,4]
3 b =copy(a)
4 a.append("d")
5 print(a)
6 print(b)
```

```
['a', 'b', 'c', 1, 2, 3, 4, 'd']
['a', 'b', 'c', 1, 2, 3, 4]
```



37. How to check whether the list is empty or not?

In [247]:

```
1 l1 = [1]
2 if len(l1) < 1:
3     print("List is Empty")
4 else:
5     print("List is not Empty")
```

List is not Empty



38. How to concatenate two lists?

In [249]:

```
1 l1 = ["Akshay"]
2 l2 = ["Patil"]
3 print(l1+l2)
```

['Akshay', 'Patil']



39. How to find the occurrences of an element in the python list?

In [253]:

```
1 l1 = [1,2,3,8,5,4,1,8,6,8,9,7,6,2,1,3,3,6,5,2,8,4,7,9]
2 inp = eval(input("Enter number - "))
3 count =0
4 for i in l1:
5     if inp == i:
6         count +=1
7 print(count)
```

Enter number - 1

3



40. How to flatten a list in python?

In [255]:

```
1 l1 = [[1,2,5,6,4],[5,7,8,9,5,6,4],[5,2,5,6,6,4,8,2,]]
2 l2=[]
3 for i in l1:
4     for j in i:
5         l2.append(j)
6 print(l2)
7
```

[1, 2, 5, 6, 4, 5, 7, 8, 9, 5, 6, 4, 5, 2, 5, 6, 6, 4, 8, 2]



In []:

1