In [ ]:
1. Is a list mutable?

Ans: yes, lists are mutable

In [1]:
```
l1 = ["pritam", 10, 5]
l1[1] = "soni"
print(l1)
```

```
['pritam', 'soni', 5]
```

In [ ]:
2. Does a list need to be homogeneous?

Ans: No, list doen not need to be homogenous.

In [2]:
```
l2 = ["hello", 10, 2.4]
print(l2)
```

```
['hello', 10, 2.4]
```

In [ ]:
3. What is the difference between a list and a tuple.

Ans: Lists are mutable while tuples are immutable.

In [3]:
```
l1 = ["pritam", 10, 5]
l1[1] = "soni"
print(l1)
```

```
['pritam', 'soni', 5]
```

In [4]:
```
t1 = (12, 5, "pritam")
t1[0] = "hello"
print(t1)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-4-dbd976dcf901> in <module>
      1 t1 = (12, 5, "pritam")
----> 2 t1[0] = "hello"
      3 print(t1)

TypeError: 'tuple' object does not support item assignment
```

In [ ]:
4. How to find the number of elements in the list?

Ans: By len() function.

In [5]:
```
len(l1)
```

Out[5]:
```
3
```

In [ ]:
5. How to check whether the list is empty or not?

In [6]:
```
l2 = []
if len(l2) == 0:
    print("Above list is empty!")
else:
    print("Above list is not empty!")
```

```
Above list is empty!
```

In [ ]:
6. How to find the first and last element of the list?

In [12]:
```
l3 = [10, 5, 4.6, "india", 89, 13, 4, 1, 9, 9]
print(f"first element in the above list is {l3[0]}")
print(f"last element in the above list is {l3[-1]}")
```

```
first element in the above list is 10
last element in the above list is 9
```

In [ ]:
7. How to find the largest and lowest value in the list?

Ans: By min() and max() funtion.

In [17]:
```
l4 = [4,12,0,6,3,89,45]
print(f"maximum value: {max(l4)}")
print(f"minimum value: {min(l4)}")
```

```
maximum value: 89
minimum value: 0
```

In [ ]:
8. How to access elements of the list?
Ans: Elements of a list can be accessed by slicing the list with corresponding indices.

In [23]:
```
print(f" first element: {l3[0]}")
print(f" last element: {l3[-1]}")
print(f" first fifth elements: {l3[:5]}")
print(f" last fifth elements: {l3[-5:]}")
```

```
 first element: 10
 last element: 9
 first fifth elements: [10, 5, 4.6, 'india', 89]
 last fifth elements: [13, 4, 1, 0, 9]
```

In [ ]:
9. Remove elements in a list before a specific index.

In [26]:
```
l3.remove(l3[2])
print(l3)
l3.pop(3)
print(l3)
```

```
[10, 5, 89, 13, 4, 1, 0, 9]
[10, 5, 89, 4, 1, 0, 9]
```

In [ ]:
10. Remove elements in a list between 2 indices.

In [28]:
```
l5 = [10, "hello", 23, 2.5, 210]
del l5[1:2]
print("After removing element between indices 0 and 1: ", l5)
```

```
After removing element between indices 0 and 1:  [10, 23, 2.5, 210]
```

In [ ]:
11. Return every 2nd element in a list between 2 indices.

In [46]:
```
l6 = [10, 52,1,0,3,9,56,448,101,9]
print(f"Every 2nd element in above list: {l6[1:10:2]}")
```

```
Every 2nd element in above list: [52, 0, 9, 448, 9]
```

In [ ]:
12. Get the first element from each nested list in a list.

In [70]:
```
l7 = [[1,2,3],[4,5,6],[7,8,9]]
for i in l7:
    print(i[0], end="\n")
```

```
1
4
7
```

In [ ]:
13. How to modify elements of the list?

Ans: By using assignment operator to corresponding indices.

In [74]:
```
l8 = ["pritam", 15, 2.3, 5]
l8[0]="india"
print(l8)
```

```
['india', 15, 2.3, 5]
```

In [ ]:
14. How to concatenate two lists?

Ans: By '+' operator

In [75]:
```
l9 = [1,2,3]
l10 = [8,9,10]
print(f"After concatenate l9 and l10: {l9+l10}")
```

```
After concatenate l9 and l10: [1, 2, 3, 8, 9, 10]
```

In [ ]:
15. How to add two lists element-wise in python?

Ans: By mapping.

In [81]:
```
from operator import add
add_list = list(map(add, l9, l10))
print("After adding lists l9 and l10 element wise: ", add_list)
```

```
After adding lists l9 and l10 element wise:  [9, 11, 13]
```

In [ ]:
16. Difference between del and clear?

Ans: The del keyword is used to delete objects. In Python everything is an object, so
    the del keyword can also be used to delete variables, lists, or parts of a list etc.
    The clear is used to clear the entire list.

In [ ]:
17. Difference between remove and pop?

Ans: Remove is used to remove a specific item from the list
    whereas pop delete the item from a particular index.

In [ ]:
18. Difference between append and extend?

Ans: Append is used to add a single element in the list to the last position while
    extend is used to add a series of elements in the list by assigning them in a single li
ne.

In [ ]:
19. Difference between indexing and Slicing?

Ans: Indexing is used to access a single element from the list while slicing is used to acce
ss series of elements
    or we can say range of elements from the given list.

In [ ]:
20. Difference between sort and sorted?

Ans: sort function will modify the list it is called on whereas the sorted function will cre
ate
    a new list containing a sorted version of the given list.

In [ ]:
21. Difference between reverse and reversed?

Ans: Reverse function returns the reverse version of a list while reversed function not actu
ally
    reverse the list, it returns an object that can be used to iterate over the container's
elements in reverse order.

In [ ]:
22. Difference between copy and deepcopy?

Ans: A shallow copy constructs a new compound object and then inserts references into it to
the objects found in the original.
    A deep copy constructs a new compound object and then
    recursively, inserts copies into it of the objects found in the original.

In [ ]:
23. How to remove duplicate elements in the list?

Ans: By using set function.

In [89]:
```
l11 = [1,2,4,1,5,4,2,3,4,2,1,5,4,2,5]
list(set(l11))
```

Out[89]:
```
[1, 2, 3, 4, 5]
```

In [ ]:
24. How to find an index of an element in the python list?

Ans: By using index function.

In [90]:
```
l11.index(3)
```

Out[90]:
```
7
```

In [ ]:
25. How to find the occurrences of an element in the python list?

Ans: By count() function.

In [91]:
```
l11.count(1)
```

Out[91]:
```
3
```

In [ ]:
26. How to insert an item at a given position?

Ans: By insert() function.

In [95]:
```
l12 = [1,2,4,5]
l12.insert(2,3)
print(l12)
```

```
[1, 2, 3, 4, 5]
```

In [ ]:
27. How to check if an item is in the list?

Ans: By simple in and not operators.

In [96]:
```
if 2 in l12:
    print("2 is in list.")
if 2 not in l12:
    print("2 is not in list.")
```

```
2 is in list.
```

In [ ]:
28. How to flatten a list in python?

Ans: By using nested loops, we can flatten a nested list.

In [99]:
```
a = [[1,2,3],[4,5,6],[7,8,9]]
for i in a:
    for item in i:
        print(item, end=" ")
```

```
1 2 3 4 5 6 7 8 9
```

In [ ]:
29. How to convert python list to other data structures like set, tuple, dictionary?

Ans: By the concept of typecasting.

In [104]:
```
b = [1,2,3,4,5]
print(type(b))
type(set(b))
type(tuple(b))
```

```
<class 'list'>
```

Out[104]:
```
tuple
```

In [ ]:
30. How to apply a function to all items in the list?

In [110]:
```
def double(integer):
    return integer*2
integer_list = [1, 2, 3]

output_list = list(map(double, integer_list))

print(output_list)
```

```
[2, 4, 6]
```

In [ ]:
31. How to filter the elements based on a function in a python list?

In [114]:
```
d = [1,2,3,4,5,6,7,8,9]
d_filtered = filter(lambda item: item>6, d)
print("Filtered list: ", list(d_filtered))
```

```
Filtered list:  [7, 8, 9]
```

In [ ]:
32. How python lists are stored in memory?

Ans: lists are stored in distinct chunks of memory which are linked together with
    pointers, which enables efficient use of memory generally and
    doesnt require resizing.