Creating a list

In [1]:
```
L1=[1,2,3,"Ram",True]
```

In [2]:
```
print(L1)
```
```
[1, 2, 3, 'Ram', True]
```

In [3]:
```
type(L1)
```
Out[3]:
```
list
```

In [4]:
```
#There are various type of list like 2D and 3D too means two and three dimension lists
l2=[1,2,3,["ram","shyam"]]
print(l2) #it is list inside list but it is 2D
```
```
[1, 2, 3, ['ram', 'shyam']]
```

In [5]:
```
#similarly for 3D= 3 bracket
l3=[[["krishna","hari",3,4,5]]]
```

In [8]:
```
print(l3)
```
```
[[['krishna', 'hari', 3, 4, 5]]]
```

Accessing item from list

In [1]:
```
#lets create list
l=["animal","fruits","food","people", 1, 2.5,[6,7,8]]
```

In [2]:
```
#indexing
l[0]
```
Out[2]:
```
'animal'
```

In [5]:
```
#if I need 6 as a output
l[6]
```
Out[5]:
```
[6, 7, 8]
```

In [6]:
```
l[6][0]
```
Out[6]:
```
6
```

In [7]:
```
#Negative indexing
l[-2]
```
Out[7]:
```
2.5
```

In [8]:
```
l[-1][-3]
```
Out[8]:
```
6
```

In [9]:
```
#Now lets see slicing too
l[0:3]
```
Out[9]:
```
['animal', 'fruits', 'food']
```

In [10]:
```
#if we want to reverse the list
l[::-1]
```
Out[10]:
```
[[6, 7, 8], 2.5, 1, 'people', 'food', 'fruits', 'animal']
```

Adding item to list: insert, extend, append

In [5]:
```
a=[1,2,3,4]
#append
a.append(7)
```

In [6]:
```
print(a)
```
```
[1, 2, 3, 4, 7]
```

In [7]:
```
#insert
a[0]=11  #in zero position, I am replacing 11
#but, lets look insert
a.insert(0,555)  # in zero psition, I am inserting 555
print(a)
```
```
[555, 11, 2, 3, 4, 7]
```

In [8]:
```
print(a)
```

```
In [8]: print(a)
```

```
[555, 11, 2, 3, 4, 7]
```

```
In [9]: #extend
        l2=[9,10,132]
```

```
In [10]: a.extend(l2)
```

```
In [11]: print(a)
```

```
[555, 11, 2, 3, 4, 7, 9, 10, 132]
```

Editing item in list: we can edit, it is not like string, sring is immutable but lists is mutable

```
In [12]: print(a)
```

```
[555, 11, 2, 3, 4, 7, 9, 10, 132]
```

```
In [13]: #lets change 555 to 444
         a[0]=444
         print(a)
```

```
[444, 11, 2, 3, 4, 7, 9, 10, 132]
```

```
In [15]: #Or lets change using negative indexing
         a[-1]=2222
         print(a)
```

```
[444, 11, 2, 3, 4, 7, 9, 10, 2222]
```

```
In [16]: #lets change 2,3,4 to A, B and C
         a[2:5]=("A","B","C")
```

```
In [17]: print(a)
```

```
[444, 11, 'A', 'B', 'C', 7, 9, 10, 2222]
```

Deleting in list - del, remove, pop, clear

```
In [18]: #lets del 444
         del a[0]
```

```
In [19]: print(a)
```

```
[11, 'A', 'B', 'C', 7, 9, 10, 2222]
```

```
In [20]: #or we can use remove
         a.remove("B")
```

```
In [21]: print(a)
```

```
[11, 'A', 'C', 7, 9, 10, 2222]
```

```
In [22]: #pop: it will remove last item
         a.pop()
```

```
Out[22]: 2222
```

```
In [23]: print(a)
```

```
[11, 'A', 'C', 7, 9, 10]
```

```
In [24]: #clear will cleared the list
         a.clear()
         print(a)
```

```
[]
```

```
In [25]: #we can totally delete list too
         del a
```

```
In [26]: print(a)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[26], line 1
----> 1 print(a)

NameError: name 'a' is not defined
```

List Comprehension

```
In [3]: #suppose we have number [2,3,4] provide multilied by 2
        #lets use lets comprehension:
        a=[2,3,4]
        output=[i*2 for i in a]
```

In [4]:

```
print(output)
```

```
[4, 6, 8]
```

In [5]:

```
#if we did not use list comprehension here, how we can do
a=[2,3,4]
output=[]
for i in a:
    output.append(i*2)

print(output)
```

```
[4, 6, 8]
```

In [1]:

```
#Second Example
#lets say we have string ["a","b","c"] now we need all this in uppercase
#lets use traditional way first
```

In [3]:

```
letter=["a","b","c"]
upperl=[]
for i in letter:
    i=i.upper()
    upperl.append(i)

print(upperl)
```

```
['A', 'B', 'C']
```

In [7]:

```
#now lets use list comprehension
upperl=[i.upper() for i in letter]
print(upperl)
```

```
['A', 'B', 'C']
```

In [ ]:

```
#Third Example
#lets separate odd and even number [1,2,3,4,5,6,7,8,9]
```

In [8]:

```
#using old method
num=[1,2,3,4,5,6,7,8,9]
even=[]
odd=[]

for i in num:
    if i%2==0:
        even.append(i)
    else:
        odd.append(i)

print(even)
print(odd)
```

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

In [10]:

```
#using list comprehension

num=[1,2,3,4,5,6,7,8,9]

even=[i%2==0 for i in num]
odd=[i%2!=0 for i in num]
```

In [11]:

```
print(even)
print(odd)
```

```
[False, True, False, True, False, True, False, True, False]
[True, False, True, False, True, False, True, False, True]
```

In [12]:

```
#we did not want True and false or boolean values, we want number
#lets try second way
```

In [16]:

```
num=[1,2,3,4,5,6,7,8,9]
odd=[i for i in num if i%2!=0]
even=[i for i in num if i%2==0]
print(even)
print(odd)
```

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

List Comprehension with if else

In [1]:

```
#first lets look example in tradition, suppose I have number [1,2,3] then i want output [1,2,15] if i=3 then mu
num=[1,3,15]
for i in num:
    if i==3:
        i*5
print(num)
```

```
[1, 3, 15]
```

In [6]:
```
#Now lets use list comprehension step by step
#first step
#nums=[i for i in num]
#where we keep if else?
#in the left middle

nums=[i*5 if i==3 else i for i in num]
```

In [7]:
```
print(nums)
```

```
[1, 15, 15]
```

In [8]:
```
#what happen if we keep if else in right side? #this will act as filter
nums=[i*5 for i in num if i==3]
```

In [9]:
```
print(nums)
```

```
[15]
```

In [10]:
```
#lets use previous example odd and even and lets print even only filtering odd
num=[6,7,8,9,19,11,12]

even=[i for i in num if i%2==0]
print(even)
```

```
[6, 8, 12]
```

Lets practice few program

1.Create 2 lists from a given list where

- 1st list will contain all the odd numbers from the original list and
- the 2nd one will contain all the even numbers

L = [1,2,3,4,5,6]

In [1]:
```
L=[1,2,3,4,5,6]
even=[]
odd=[]

even=[i for i in L if i%2==0]
odd=[i for i in L if i%2!=0]

print(even)
print(odd)
```

```
[2, 4, 6]
[1, 3, 5]
```

1. How to take list as input from user

In [5]:
```
num=list(input("Enter a list: "))
```

```
Enter a list: 1 2 3 4 5
```

In [8]:
```
print(num)
```

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

In [9]:
```
#we get comma there because we put space.
```

1. Write a program to merge 2 list without using the + operator *L1 = [1,2,3,4]*L2 = [5,6,7,8]

In [10]:
```
L1 = [1,2,3,4]
L2 = [5,6,7,8]
```

In [11]:
```
# I can merge in various ways
L1.extend(L2)
```

In [12]:
```
print(L1)
```

```
[1, 2, 3, 4, 5, 6, 7, 8]
```

In [15]:
```
L1=["a","b","c"]
L2=["d","e","f"]
L1.append(L2)
```

In [16]:
```
print(L1)
```

```
['a', 'b', 'c', ['d', 'e', 'f']]
```

In [17]:
```
#append is taking L2 as a whole so best way will be extend
```

```
#but lets see how + operator would have been worked
l1=[1,2,3]
l2=[4,5,6]
l3=l1+l2
print(l3)
```

```
[1, 2, 3, 4, 5, 6]
```

In [18]:

```
#yes + and extend is same.
```

1. Write a program to replace an item with a different item if found in the list

L = [1,2,3,4,5,3] replace 3 with 300

In [19]:

```
L = [1,2,3,4,5,3]
#it can be easily replace
L[5]=300
print(L)
```

```
[1, 2, 3, 4, 5, 300]
```

In [20]:

```
#it can be solved using function def too, but , we will see it later during function
```

1. L = [1,2,1,2,3,4,5,3,4] Remove duplicate value from list

In [22]:

```
#we know that list accepts duplicates values but not set so let change it in sets
L = [1,2,1,2,3,4,5,3,4]
LS=set(L)
```

In [23]:

```
print(LS)
```

```
{1, 2, 3, 4, 5}
```

In [24]:

```
#Now we can change it to its original value
L=list(LS)
print(L)
```

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

In [25]:

```
print(type(L))
```

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

1. Write a program to check if a list is in ascending order or not

In [33]:

```
L=["mango","orange","apple"]
l1=L.sort()
if L==l1:
    print("True")
else:
    print("false")
    print("if you need I can sort L in ascending order")
    enter=input("Please enter yes or no: ")
    if enter=="yes":
        L.sort()
        print(L)
    else:
        print("Thank you")
```

```
false
if you need I can sort L in ascending order
Please enter yes or no: yes
['apple', 'mango', 'orange']
```

1. Merge two list using zip function

In [35]:

```
a=[1,2,3,4]
b=["a","b","c"]

merged=[]

for i,j in zip(a,b):
    merged.append((i,j))
print(merged)
```

```
[(1, 'a'), (2, 'b'), (3, 'c')]
```

In [ ]:

```
#this is different type of merging by element wise.
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js