



<https://apssdc.in>

# APSSDC

Andhra Pradesh State Skill Development Corporation



## Day 09 Lists and Tuples in Python

### Recap

- Strings
- String Methods
- List Intro..

### Todays Objectives

- Accessing Elements of List
  - Indexing
  - Slicing
- List Methods
- Tuple
  - Accessing Elements of Tuple
    - Indexing
    - Slicing
  - Tuple Methods

```
In [1]: ▶ 1 li = [1, 2, 5, 4, 8, 85]
```

```
In [2]: ▶ 1 print(len(li), max(li), min(li), sum(li))
```

```
6 85 1 105
```

```
In [3]: ▶ 1 li2 = [1, 2, 5, 4, 8, 85, 'Python']
          2
          3
          4 print(len(li2))
          5 print(min(li2))
```

7

```
-----
TypeError                                 Traceback (most recent call last)
<ipython-input-3-831c9faba063> in <module>
      4 print(len(li2))
      5
----> 6 print(min(li2))
```

**TypeError:** '<' not supported between instances of 'str' and 'int'

## Accessing Elements from the list

- Indexing
  - +ve
  - -ve
- Slicing
  - +ve
  - -ve

```
In [4]: ▶ 1 print(li[0])
```

1

```
In [6]: ▶ 1 print(li[5])
```

85

```
In [7]: ▶ 1 print(li[-2])
```

8

```
In [8]: ▶ 1 print(li[-6])
```

1

```
In [9]: 1 print(li[7])
```

```
-----  
IndexError                                Traceback (most recent call last)  
<ipython-input-9-c96b4b304720> in <module>  
----> 1 print(li[7])
```

**IndexError**: list index out of range

```
In [10]: 1 print(li[1.2])
```

```
-----  
TypeError                                Traceback (most recent call last)  
<ipython-input-10-f165e6fb7646> in <module>  
----> 1 print(li[1.2])
```

**TypeError**: list indices must be integers or slices, not float

## Syntax

list\_var[SI: EI]

```
In [20]: 1 print(li[0:1], li[0: 2], li[: 2], li[0:])
```

```
[1] [1, 2] [1, 2] [1, 2, 5, 4, 8, 85]
```

```
In [12]: 1 print(li[:-2], li[-1:-3], li[4:], li[len(li) - 2:], li[:-3])
```

```
[1, 2, 5, 4] [] [8, 85] [8, 85] [1, 2, 5]
```

```
In [21]: 1 print(li[-2: ], li[-3: ], li[-3: -1])
```

```
[8, 85] [4, 8, 85] [4, 8]
```

```
In [17]: 1 print(li[ : 2: -1])
```

```
[85, 8, 4]
```

```
In [18]: 1 print(li[::-1])
```

```
[85, 8, 4, 5, 2, 1]
```

```
In [19]: 1 print(li[0::2])
```

```
[1, 5, 8]
```

```
In [23]: 1 print(li[:2:-1])
```

```
[85, 8, 4]
```

```
In [24]: 1 li
```

```
Out[24]: [1, 2, 5, 4, 8, 85]
```

```
In [25]: 1 print(li[:2:-1])
```

```
[85, 8, 4]
```

```
In [26]: 1 print(li[: :-1])
```

```
[85, 8, 4, 5, 2, 1]
```

```
In [27]: 1 print(li[: :2])
```

```
[1, 5, 8]
```

```
In [28]: 1 print(li[: :-2])
```

```
[85, 4, 2]
```

```
In [29]: 1 print(li)
```

```
[1, 2, 5, 4, 8, 85]
```

```
In [30]: 1 print(li2)
```

```
[1, 2, 5, 4, 8, 85, 'Python']
```

```
In [31]: 1 print(len(li2))
```

```
7
```

```
In [33]: 1 print(li2[0], li2[5])
```

```
1 85
```

```
In [34]: 1 print(li2[-1], li[-2])
```

```
Python 8
```

```
In [35]: 1 li2[-1].upper()
```

```
Out[35]: 'PYTHON'
```

```
In [36]: 1 li2[-1].strip()
```

```
Out[36]: 'Python'
```

```
In [37]: 1 print(li2[0:5], li2[ :5])
```

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

```
In [38]: 1 print(li2[-3:])
```

```
[8, 85, 'Python']
```

```
In [39]: 1 print(li2[0: :2])
```

```
[1, 5, 8, 'Python']
```

```
In [40]: 1 print(li2[: :-2])
```

```
['Python', 8, 5, 1]
```

```
In [41]: 1 print(li2[::-1])
```

```
['Python', 85, 8, 4, 5, 2, 1]
```

```
In [42]: 1 matrix = [[1,2,3],[4,5,6], [6,7,8]]  
2  
3 print(matrix)
```

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

```
In [43]: 1 print(matrix[0])
```

```
[1, 2, 3]
```

```
In [44]: 1 print(matrix[0][0])
```

```
1
```

```
In [45]: 1 print(matrix[1][1])
```

```
5
```

```
In [47]: 1 print(matrix[0][1:], matrix[0][1:2])
```

```
[2, 3] [2]
```

```
In [48]: 1 print(matrix[1][::-1])
```

```
[6, 5, 4]
```

```
In [50]: ▶ 1 print(len(matrix[2]), max(matrix[2]))
```

3 8

```
In [52]: ▶ 1 print(matrix[0][1:2])
```

[2]

```
In [53]: ▶ 1 for ele in matrix:  
2     print(ele)
```

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

```
In [54]: ▶ 1 for ele in li:  
2     print(ele)
```

1  
2  
5  
4  
8  
85

```
In [57]: ▶ 1 for i in range(3):  
2     for j in range(3):  
3         if j==1:  
4             print(matrix[i][j])
```

2  
5  
7

```
In [58]: ▶ 1 for ele in matrix:  
2     print(ele[1])
```

2  
5  
7

```
In [64]: ▶ 1 print(matrix[:, matrix[:,2]])
```

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

```
In [66]: ▶ 1 for ele in matrix:
           2     print(ele[1:])
           3
           4 for ele in matrix:
           5     print(ele[1], ele[2])
```

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

## List Methods

```
In [67]: ▶ 1 empty = []
```

```
In [68]: ▶ 1 empty.append(5)
           2 empty.append(55)
           3 empty.append([1,2,3])
           4
           5 print(empty)
```

```
[5, 55, [1, 2, 3]]
```

```
In [69]: ▶ 1 for i in range(5):
           2     empty.append(input("Enter elements to add to list"))
```

```
Enter elements to add to list53
Enter elements to add to listgdf
Enter elements to add to listdgd
Enter elements to add to list343
Enter elements to add to list5
```

```
In [70]: ▶ 1 print(empty)
```

```
[5, 55, [1, 2, 3], '53', 'gdf', 'dgd', '343', '5']
```

```
In [71]: ▶ 1 a=['a']
          2 for i in a:
          3     a.append(i.upper())
          4
          5 print(a)
```

ERROR:root:Internal Python error in the inspect module.  
Below is the traceback from this internal error.

Traceback (most recent call last):  
File "C:\Users\Jesus\anaconda3\lib\site-packages\IPython\core\interacti  
veshell.py", line 3343, in run\_code  
exec(code\_obj, self.user\_global\_ns, self.user\_ns)  
File "<ipython-input-71-d7c67c993e5e>", line 3, in <module>  
a.append(i.upper())  
KeyboardInterrupt

During handling of the above exception, another exception occurred:

Traceback (most recent call last):  
File "C:\Users\Jesus\anaconda3\lib\site-packages\IPython\core\interacti  
veshell.py", line 2044, in showtraceback  
stb = value.\_render\_traceback\_()  
AttributeError: 'KeyboardInterrupt' object has no attribute '\_render\_trac  
eback '

```
In [ ]: ▶ 1 print(a)
```

IOPub data rate exceeded.  
The notebook server will temporarily stop sending output  
to the client in order to avoid crashing it.  
To change this limit, set the config variable  
`--NotebookApp.iopub\_data\_rate\_limit`.

Current values:  
NotebookApp.iopub\_data\_rate\_limit=1000000.0 (bytes/sec)  
NotebookApp.rate\_limit\_window=3.0 (secs)

```
In [3]: ▶ 1 li = [1,2,3,4]
          2
          3 empty = [5, 55, [1, 2, 3], '53', 'gdf', 'dgd', '343', '5']
```

```
In [4]: ▶ 1 empty.extend(li)
          2
          3 print(empty)
```

```
[5, 55, [1, 2, 3], '53', 'gdf', 'dgd', '343', '5', 1, 2, 3, 4]
```

```
In [5]: ▶ 1 print(empty + li)
```

```
[5, 55, [1, 2, 3], '53', 'gdf', 'dgd', '343', '5', 1, 2, 3, 4, 1, 2, 3, 4]
```



```
In [6]: ▶ 1 emp2 = empty + li
          2
          3 print(emp2)
```

[5, 55, [1, 2, 3], '53', 'gdf', 'dgd', '343', '5', 1, 2, 3, 4, 1, 2, 3, 4]

```
In [7]: ▶ 1 print(emp2[2])
```

[1, 2, 3]

```
In [9]: ▶ 1 emp2[2] = li
          2
          3 print(emp2)
```

[5, 55, [1, 2, 3, 4], '53', 'gdf', 'dgd', '343', '5', 1, 2, 3, 4, 1, 2, 3, 4]

```
In [10]: ▶ 1 emp2[2:4] = li
           2
           3 print(emp2)
```

[5, 55, 1, 2, 3, 4, 'gdf', 'dgd', '343', '5', 1, 2, 3, 4, 1, 2, 3, 4]

```
In [11]: ▶ 1 emp2.insert(1, 10)
           2
           3 print(emp2)
```

[5, 10, 55, 1, 2, 3, 4, 'gdf', 'dgd', '343', '5', 1, 2, 3, 4, 1, 2, 3, 4]

```
In [12]: ▶ 1 emp2.pop()
           2
           3 print(emp2) # LIFO -> Last In First Out
```

[5, 10, 55, 1, 2, 3, 4, 'gdf', 'dgd', '343', '5', 1, 2, 3, 4, 1, 2, 3]

```
In [13]: ▶ 1 emp2.pop(2)
           2
           3 print(emp2)
```

[5, 10, 1, 2, 3, 4, 'gdf', 'dgd', '343', '5', 1, 2, 3, 4, 1, 2, 3]

```
In [14]: ▶ 1 emp2.pop(100)
          2
          3 print(emp2)
```

```
-----
IndexError                                Traceback (most recent call last)
<ipython-input-14-bfab930aa297> in <module>
----> 1 emp2.pop(100)
      2
      3 print(emp2)

IndexError: pop index out of range
```

```
In [15]: ▶ 1 emp2.remove('gdf')
```

```
In [16]: ▶ 1 print(emp2)
```

```
[5, 10, 1, 2, 3, 4, 'dgd', '343', '5', 1, 2, 3, 4, 1, 2, 3]
```

```
In [18]: ▶ 1 print(emp2.remove('gdf'))
```

```
-----
ValueError                                Traceback (most recent call last)
<ipython-input-18-d29e36b460c0> in <module>
----> 1 print(emp2.remove('gdf'))

ValueError: list.remove(x): x not in list
```

```
In [20]: ▶ 1 print(emp2.pop())
          2
          3
          4 print(emp2)
```

```
2
[5, 10, 1, 2, 3, 4, 'dgd', '343', '5', 1, 2, 3, 4, 1]
```

```
In [21]: ▶ 1 emp3 = emp2
          2
          3 print(emp3, emp2)
```

```
[5, 10, 1, 2, 3, 4, 'dgd', '343', '5', 1, 2, 3, 4, 1] [5, 10, 1, 2, 3, 4, 'dgd', '343', '5', 1, 2, 3, 4, 1]
```

```
In [22]: ▶ 1 print(id(emp2), id(emp3))
```

```
1944735317312 1944735317312
```

```
In [23]: ▶ 1 print(emp3.pop())
          2
          3 print(emp2, emp3)

1
[5, 10, 1, 2, 3, 4, 'dgd', '343', '5', 1, 2, 3, 4] [5, 10, 1, 2, 3, 4, 'dgd', '343', '5', 1, 2, 3, 4]
```

```
In [24]: ▶ 1 emp3 = emp2.copy()
          2
          3
          4 print(id(emp2), id(emp3))
          5 print(emp2, emp3)

1944735317312 1944736306368
[5, 10, 1, 2, 3, 4, 'dgd', '343', '5', 1, 2, 3, 4] [5, 10, 1, 2, 3, 4, 'dgd', '343', '5', 1, 2, 3, 4]
```

```
In [25]: ▶ 1 print(emp3.pop())
          2
          3 print(emp2, emp3)

4
[5, 10, 1, 2, 3, 4, 'dgd', '343', '5', 1, 2, 3, 4] [5, 10, 1, 2, 3, 4, 'dgd', '343', '5', 1, 2, 3]
```

```
In [26]: ▶ 1 print(emp2.count(1))

2
```

```
In [27]: ▶ 1 print(emp2.index('dgd'))

6
```

```
In [28]: ▶ 1 print(emp2.index('dgda'))

-----
ValueError                                Traceback (most recent call last)
<ipython-input-28-072296294b17> in <module>
----> 1 print(emp2.index('dgda'))

ValueError: 'dgda' is not in list
```

```
In [29]: ▶ 1 emp2.reverse()
          2
          3 print(emp2)

[4, 3, 2, 1, '5', '343', 'dgd', 4, 3, 2, 1, 10, 5]
```

```
In [30]: ▶ 1 emp3.sort()
          2
          3
          4 print(emp3)
```

```
-----
TypeError                                 Traceback (most recent call last)
<ipython-input-30-d9096d7c8955> in <module>
----> 1 emp3.sort()
      2
      3
      4 print(emp3)

TypeError: '<' not supported between instances of 'str' and 'int'
```

```
In [31]: ▶ 1 li = [5,3,6,22,77,4,66,9,345]
          2 li.sort()
          3
          4 print(li)
```

```
[3, 4, 5, 6, 9, 22, 66, 77, 345]
```

```
In [32]: ▶ 1 li = [5,3,6,22,77,4,66,9,345]
          2 li.sort(reverse=True)
          3
          4 print(li)
```

```
[345, 77, 66, 22, 9, 6, 5, 4, 3]
```

```
In [33]: ▶ 1 matrix = [[1,20,3],[4,5,6], [6,75,8]]
          2
          3 print(matrix)
```

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

```
In [34]: ▶ 1 matrix = [[40,20,3],[35,5,6],[6,75,8]]
          2
          3
          4 matrix.sort()
          5
          6
          7 print(matrix)
```

```
[[6, 75, 8], [35, 5, 6], [40, 20, 3]]
```

```
In [36]: ▶ 1 matrix = [[40,20,3],[35,5,6], [6,75,8]]
          2
          3
          4 matrix.sort(key = lambda x :x[1])
          5
          6
          7 print(matrix)
```

```
[[35, 5, 6], [40, 20, 3], [6, 75, 8]]
```

```
In [37]: ▶ 1 print(sorted(matrix))
          2
          3
          4 print(matrix)
```

```
[[6, 75, 8], [35, 5, 6], [40, 20, 3]]
[[35, 5, 6], [40, 20, 3], [6, 75, 8]]
```

```
In [40]: ▶ 1 print(list(reversed(matrix)))
```

```
[[6, 75, 8], [40, 20, 3], [35, 5, 6]]
```

```
In [41]: ▶ 1 emp3.clear()
          2
          3
          4 print(emp3)
```

```
[]
```

```
In [42]: ▶ 1 del emp3
          2
          3 print(emp3)
```

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-42-70167493e5a1> in <module>
      1 del emp3
      2
----> 3 print(emp3)

NameError: name 'emp3' is not defined
```

## Tuples

Storing the Group of non-homogenous group of data

- Is created using ()
- It is immutable
- it is iterable
- it is ordered

```
In [43]: 1 t1 = ()
        2 t2 = tuple()
```

```
In [45]: 1 t1 = (1,2,3,4, 'Python', (1,2,3), [1,2,3])
```

```
In [46]: 1 print(t1[0], t1[-1], t1[2], t1[-2])
```

```
1 [1, 2, 3] 3 (1, 2, 3)
```

```
In [47]: 1 print(t1[0:2])
```

```
(1, 2)
```

```
In [48]: 1 t1[0] = 55
```

```
-----
TypeError                                Traceback (most recent call last)
<ipython-input-48-f8552766aa0a> in <module>
----> 1 t1[0] = 55
```

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

```
In [56]: 1 t1 = (1,2,3,4, 'Python', (1,2,3), [1,2,3])
        2
        3 t1[-1].append(5)
```

```
In [57]: 1 t1
```

```
Out[57]: (1, 2, 3, 4, 'Python', (1, 2, 3), [1, 2, 3, 5])
```

```
In [49]: 1 for ele in t1:
        2     print(ele)
```

```
1
2
3
4
Python
(1, 2, 3)
[1, 2, 3]
```

```
In [50]: 1 print(len(t1))
```

```
7
```

## Tuple Methods

```
In [51]: 1 print(t1.count(1))
```

```
1
```

```
In [52]: 1  
2 print(t1.index('Python'))
```

```
4
```

```
In [55]: 1 print('y' + 'o')  
2  
3 print('y' * 5)
```

```
yo  
yyyyy
```

## Tasks

s = 'Python is an interpreted high-level general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation. Wikipedia Developer: Python Software Foundation Stable release: 3.9.5 / 3 May 2021; 19 days ago Preview release: 3.10.0b1 / 3 May 2021; 19 days ago Typing discipline: Duck, dynamic, strong typing; gradual (since 3.5, but ignored in CPython) First appeared: February 1991; 30 years ago Paradigm: Multi-paradigm: object-oriented, procedural (imperative), functional, structured, reflective'

```
In [53]: 1 s = '''Python is an interpreted high-level general-purpose programming language.  
2 Developer: Python Software Foundation  
3 Stable release: 3.9.5 / 3 May 2021; 19 days ago  
4 Preview release: 3.10.0b1 / 3 May 2021; 19 days ago  
5 Typing discipline: Duck, dynamic, strong typing; gradual (since 3.5, but ignored in CPython)  
6 First appeared: February 1991; 30 years ago  
7 Paradigm: Multi-paradigm: object-oriented, procedural (imperative), functional, structured,  
8 reflective'''
```

```
In [54]: 1 print(s.split())
```

```
['Python', 'is', 'an', 'interpreted', 'high-level', 'general-purpose', 'pro  
gramming', 'language.', "Python's", 'design', 'philosophy', 'emphasizes',  
'code', 'readability', 'with', 'its', 'notable', 'use', 'of', 'significan  
t', 'indentation.', 'Wikipedia', 'Developer:', 'Python', 'Software', 'Found  
ation', 'Stable', 'release:', '3.9.5', '/', '3', 'May', '2021;', '19', 'day  
s', 'ago', 'Preview', 'release:', '3.10.0b1', '/', '3', 'May', '2021;', '1  
9', 'days', 'ago', 'Typing', 'discipline:', 'Duck,', 'dynamic,', 'strong',  
'typing;', 'gradual', '(since', '3.5,', 'but', 'ignored', 'in', 'CPython)',  
'First', 'appeared:', 'February', '1991;', '30', 'years', 'ago', 'Paradig  
m:', 'Multi-paradigm:', 'object-oriented,', 'procedural', '(imperative),',  
'functional,', 'structured,', 'reflective']
```

## Tasks

1. Reverse every element in list and update them in another list
2. extract lower-case alpha from each element and update in another list
3. remove special characters from the element and update in another list
4. get the middle character of every element and update in another list