Content Covered

- List
- List Methods
- Tuple

List

- Multiple data separated by comma & enclosed in []
- Data Type can be different
- Can be used as stack
- Function like index, len, slicing, repetition, count valid in list too

List Example

- string_list = ['ram', 'rabi', 'rabin', 'rabindra']
- number_list = [5, 6, 1, 45, 7.7]
- mixed_list = ['rabi', 45, 'rabin', 34, False]
- two_d_list = [['rabi', 25], ['rabin',22]]

List Methods

Indexing

- Extract element in given index
- Similar to string indexing
- Supports negative index
- Example:
  - my_list = [10, 'rabi', 2.9, False]
  - second_elem = my_list[1]
  - print(second_elem)
  - my_list[2] = 15 # Doesn't work in string
  - print(my_list)
  - list_2d = [['python', 'back'], ['java', 'front']]
  - print(list_2d[-1][0])

List Methods

slicing

- Similar to string slicing
- Returns list
- Supports negative index
- Example:
  - my_list = [10, ‘rabi’, 2.9, False]
  - sub_list = my_list[1:3]
  - print(sub_list)
  - print(my_list[:3])
  - print(my_list[1:])

List Methods

Repetition

- Similar to string repetition
- Repeat element of list given number of times
- Example:
  - my_list = [10, 'rabi', 2.9, False]
  - new_list = my_list * 2
  - print(new_list)
  - print(['0'] * 5)

List Methods

len

- Calculates number of item in list
- Example:
  - my_list = [10, ‘rabi’, 2.9, False]
  - total_items = len(my_list)
  - print(total_items)

List Methods

join

- Similar to string join
- Joins each element of list with mentioned delimiter
- Result of join is string
- Example:
  - my_list = ['b','u','s']
  - joined_string = '.'.join(my_list)
  - print(joined_string)

List Methods

index

- Similar to string index
- Used to find where element occurred in list
- Returns first occurrent position
- NOTE: .find not applicable in list
- Example:
  - my_list = [‘b’,’u’,’s’]
  - print(my_list.index(‘u’))

List Methods

count

- Similar to string count
- Give count of number of occurrence of element in list
- Example:
  - my_list = [‘b’,’u’,’s’, ‘e’, ‘s’]
  - print(my_list.count(‘u’))
  - print(my_list.count(‘s’))

List Methods

Append

- Adds element at the end of list
- In place operation
- Example:
  - my_list = [‘b’,’u’,’s’]
  - my_list.append(‘e’)
  - print(my_list)
  - my_list.append(‘s’)
  - print(my_list)

List Methods

pop

- Removes and return element from mentioned position
- Default position is last
- In place operation
- Example:
  - my_list = ['b','u','s']
  - elem = my_list.pop()
  - print(my_list)
  - print(elem)
  - first_elem = my_list.pop(0)
  - print(my_list, elem)

List Methods

insert

- Insert element at specified index of list
- Shifts another elements by right
- Example:
  - my_list = [‘python’, ‘math’, ‘Data Science’]
  - my_list.insert(1, ‘Database’)
  - print(my_list)

List Methods

remove

- Deletes first occurrence of specified value from list
- Shifts another elements by left
- Example:
  - my_list = [‘python’, ‘math’, ‘Data Science’]
  - my_list.remove(‘math’)
  - print(my_list)

List Methods

del

- Delete element from mention index
- Index can be slice
- Shifts another elements by left
- Example:
  - my_list = [‘python’, ‘math’, 12, 16, True]
  - del(my_list[1])
  - print(my_list)
  - del(my_list[:2])
  - print(my_list)

List Methods

extend

- Expand list with element from another iterable
- Elements are added at last
- Example
  - lis_1= [1, 2]
  - lis_2 = [3, 4]
  - lis_1.extend(lis_2)
  - print(lis_1)
  - tup_1 = (5, 6)
  - lis_1.extend(tup_1)
  - print(lis_1)

List Methods

min / max

- Display min / max from element of list
- All items has to be same data type for it to work
- For string its lexical sort (same as dictionary)
- We can specify key as well
- Example
  - num_lis = [1, 2, 3.0]
  - print(min(num_lis))
  - str_list = ['apple', 'Cat', 'ball']
  - print(max(str_list))
  - print(max(str_list, key=str.lower))

List Methods

sum

- Display sum of elements in list
- All data must be numeric
- Example
  - num_lis = [1, 2.5, 0.5]
  - total = sum(num_lis)
  - print(total)

List Methods

clear

- Used for emptying list
- Example
  - num_lis = [1, 2.5, 0.5]
  - print(num_lis)
  - num_lis.clear()
  - print(num_lis)

List Methods

Copy List

- list_y = list_x doesn't copy list but point same value
- To copy list, we can use either of the following
  - list_y = list(list_x)
  - list_y = list_x[:]

List Methods

Sorting List

- sort & sorted sorts items in the list
- sort modifies existing list but sorted preserves
- Example:
  - lis_a = ['apple', 'Cat', 'ball']
  - sort_lis = lis_a.sort()
  - print(lis_a)
  - print(sort_lis)

List Methods

Reverse

- For reversing items in the list
- We can use .reverse or slice with –ve jump
- Example:
  - lis_a = ['apple', 'Cat', 'ball']
  - print(lis_a[::-1])
  - lis_a.reverse()
  - print(lis_a)

Tuple

- Multiple data separated by comma & enclosed in ()
- Immutable list. i.e. can't be modified once assigned
- Used for storing protected data
- Example:
  - str_tuple = ('rabi', 'rabin', 'rabindra')
  - num_tuple = (1, 2, 3)
  - mixed_tuple = ('rabi', {'age': 24}, ['ok', 'bye'], 14)
  - single_tuple = 1,

Tuple Unpacking

- Example:
  - str_tuple = (‘Python’, 3, ‘BoardWay’)
  - course, duration, institute = str_tuple
  - print(course)
  - print(duration)
  - print(institute)

Tuple Methods

- Supports all the methods of list that doesn't modify tuple data in-place
- Example:
  - indexing, count, len etc.
  - sort is not supported but sorted is as sort modifies in-place

Exercise