

## List and Tuples

### Session Objectives

- Understand the meaning of data structures and why we use them.
- Common data structures in Python
- Understand what lists are.
- Understand common methods and operations associated with lists.
- Understand the meaning of list comprehension
- Understand what tuples are.
- Understand common methods and operations associated with tuples.
- Understand the Comparison between Lists and Tuples

0,0	0,1	0,2	0,3
1,0	1,1	1,2	1,3
2,0	2,1	2,2	2,3
3,0	3,1	3,2	3,3

[4X4] Matrix[row\*col]

i == j [diagonal Index]

### Memory

list1 = [1,2,3,4,5] X101  
 list2 → X101  
 Deep Copy

list3 = [1,2,3,4,5] X111

Shallow Copy

list1 = [1,2,3,4,5]

list2 = list1

list3 = list1.copy()

list2[-1] = 7  
 list1[0] = 9

```

# Lists are mutable [means we can change the List content]
country_list = ['India', 'Russia', 'America', 'China', 'Canada', 'Japan', 'Vietnam',
               'Norway', 'France', 'Finland', 'Sweden', 'Croatia', 'Australia',
               'India', 'China', 'Pakistan', 'Japan', 'Iran', 'Singapore', 'New-Zealand']
country_list[-3] = 'Spain' # 'Iran' -> 'Spain' [Mutable Container]
print(country_list)

['India', 'Russia', 'America', 'China', 'Canada', 'Japan', 'Vietnam', 'Norway', 'France', 'Finland', 'Sweden',
 'Croatia', 'Australia', 'India', 'China', 'Pakistan', 'Japan', 'Spain', 'Singapore', 'New-Zealand']

country_list[8:11] = ['France'] # ['France', 'Finland', 'Sweden'] -> ['France'] [Mutable Container]
print(country_list)

['India', 'Russia', 'America', 'China', 'Canada', 'Japan', 'Vietnam', 'Norway', 'France', 'Croatia', 'Australia',
 'India', 'China', 'Pakistan', 'Japan', 'Spain', 'Singapore', 'New-Zealand']

```

```

country_list[8:11] = 'France' # After Norway ['F', 'r', 'a', 'n', 'c', 'e']
print(country_list)

['India', 'Russia', 'America', 'China', 'Canada', 'Japan', 'Vietnam', 'Norway', 'F', 'r', 'a', 'n', 'c', 'e',
 'India', 'China', 'Pakistan', 'Japan', 'Spain', 'Singapore', 'New-Zealand']

# Hoping so, You understand the difference of above 2 block of code....
country_list[8:14] = ['France', 'Croatia', 'Australia']
print(country_list)

['India', 'Russia', 'America', 'China', 'Canada', 'Japan', 'Vietnam', 'Norway', 'France', 'Croatia', 'Australia',
 'India', 'China', 'Pakistan', 'Japan', 'Spain', 'Singapore', 'New-Zealand']

# ('France') <'String'>
# ('France',) <'Tuple'>
# ['France'] <'List'>

```

```

country_list[-7:-5] = ('Finland',) # ['India', 'China'] # ('Finland',)
print(country_list)

['India', 'Russia', 'America', 'China', 'Canada', 'Japan', 'Vietnam', 'Norway', 'France', 'Croatia', 'Australia',
 'Finland', 'Pakistan', 'Japan', 'Spain', 'Singapore', 'New-Zealand']

# Repeating a content of a list <*>
char_list = ['a', 'b', 'c', 'd', 'e', True, False]
print(char_list * 2) #
print(len(char_list)) # 7
print(len(char_list*2)) # 14

['a', 'b', 'c', 'd', 'e', True, False, 'a', 'b', 'c', 'd', 'e', True, False]
7
14

```

## Adding items in a lists:

- `.append()` : Add one item at the end
- `.extend()` : Adds all items from another list
- `.insert()` : Insert an item at specific position/Index.

```
country_list = ['India', 'Russia', 'America', 'China', 'Canada', 'Japan', 'Vietnam', 'Norway', 'France',
                'Croatia', 'Australia', 'Finland', 'Pakistan', 'Spain', 'Singapore', 'New-Zealand']
country_list.append('Nepal') # Item will be added at the end.
print(country_list)

['India', 'Russia', 'America', 'China', 'Canada', 'Japan', 'Vietnam', 'Norway', 'France', 'Croatia', 'Australia',
 'Finland', 'Pakistan', 'Spain', 'Singapore', 'New-Zealand', 'Nepal']
```

```
new_country_list = ['South Africa','UAE','Iran','Thailand','Mexico','Germany','UK']
country_list.extend(new_country_list)
print(country_list)

['India', 'Russia', 'America', 'China', 'Canada', 'Japan', 'Vietnam', 'Norway', 'France', 'Croatia', 'Australia',
 'Finland', 'Pakistan', 'Spain', 'Singapore', 'New-Zealand', 'Nepal', 'South Africa', 'UAE', 'Iran', 'Thailand',
 'Mexico', 'Germany', 'UK']

# Insert(Pos, Val)
country_list.insert(3,'Qatar')
print(country_list)

['India', 'Russia', 'America', 'Qatar', 'China', 'Canada', 'Japan', 'Vietnam', 'Norway', 'France', 'Croatia',
 'Australia', 'Finland', 'Pakistan', 'Spain', 'Singapore', 'New-Zealand', 'Nepal', 'South Africa', 'UAE', 'Iran',
 'Thailand', 'Mexico', 'Germany', 'UK']
```

```
# Insert(Pos, Val)
country_list.insert(3,['Brazil','Zimbia','Argentina','South Korea']) # Existing List Become Nested
print(country_list)

['India', 'Russia', 'America', ['Brazil', 'Zimbia', 'Argentina', 'South Korea'], 'Qatar', 'China', 'Canada',
 'Japan', 'Vietnam', 'Norway', 'France', 'Croatia', 'Australia', 'Finland', 'Pakistan', 'Spain', 'Singapore',
 'New-Zealand', 'Nepal', 'South Africa', 'UAE', 'Iran', 'Thailand', 'Mexico', 'Germany', 'UK']

# Insert(Pos, Val) # TypeError: insert expected 2 arguments, got 5
country_list.insert(3,'Brazil','Zimbia','Argentina','South Korea') #Error
print(country_list)
```

```
# Adding a List using '+' Operator [Concatenation]
weekday_list = ['Mon', 'Tue', 'Wed', 'Thur', 'Fri']
weekend_list = ['Sat', 'Sun']
week_list = weekday_list + weekend_list
print(week_list)
print(type(week_list)) # <'List'>

['Mon', 'Tue', 'Wed', 'Thur', 'Fri', 'Sat', 'Sun']
<class 'list'>
```

```
# Indexing in Nested List [2D Matrix] [row][col] [3*5]
nested_list = [
    [1,2,3,4,5],
    [0,2,4,6,8],
    [1,3,5,7,9],
]
print(nested_list[0][1]) # 2
print(nested_list[-1][-4]) # 3
print(nested_list[2][-1]) # 9
print(nested_list[-3][-3]) # 3
print(nested_list[2][-5]) # 1
print(nested_list[-2][4]) # 8
print(nested_list[-3][3]) # 4
```

2  
3  
9  
3  
1  
8  
4

```
# Slicing [Start:Stop:Step]
nested_list = [
    [1,2,3,4,5],
    [0,2,4,6,8],
    [1,3,5,7,9],
]
print(nested_list[0:2]) # [[1,2,3,4,5],[0,2,4,6,8]]
[[1, 2, 3, 4, 5], [0, 2, 4, 6, 8]]

print(nested_list[::-1]) # reverse all the rows
[[1, 3, 5, 7, 9], [0, 2, 4, 6, 8], [1, 2, 3, 4, 5]]

print(nested_list[::-1][0:2]) # [1, 3, 5, 7, 9], [0, 2, 4, 6, 8]
[[1, 3, 5, 7, 9], [0, 2, 4, 6, 8]]

print(nested_list[-1][1:4]) # [1,3,5,7,9] # [3,5,7]
[3, 5, 7]
```

```
print(nested_list[-1][::-1]) # [9,7,5,3,1]
[9, 7, 5, 3, 1]

# Comparison List
_list1 = [1,2,3,4,5]
_list2 = [1,2,3]
result = (_list1 == _list2) # Boolean Return[Data Compare]
print(result) # False
False

# Comparison List
_list1 = [1,2,3,4,5]
_list2 = [1,2,3,4,5]
result = (_list1 == _list2) # Boolean Return[Data Compare]
print(result) # True
True
```

```
# Comparison List
_list1 = [1,2,3,4,5,6,7]
_list2 = [1,2,3,4,5]
result = (_list1 != _list2) # Boolean Return[Data Compare]
print(result) # True
```

True

```
# Comparison List
_list1 = [1,2,3,4,5,6,7]
_list2 = [1,2,3,4,5]
result = (_list1 >= _list2) # Boolean Return[Data Compare]
print(result) # True
```

True

```
# Comparison List
_list1 = [1,2,3,4,5,6,7]
_list2 = [1,2,3,4,5]
result = (_list1 <= _list2) # Boolean Return[Data Compare]
print(result) # False
```

False

```
# Comparison List
_list1 = ['1','2','3','4','5']
_list2 = [1,2,3,4,5]
result = (_list1 != _list2) # Boolean Return[Data Compare]
print(result) # True
```

True

```
# Comparison List
_list1 = ['1','2','3','4','5']
_list2 = [1,2,3,4,5]
result = (_list1 >= _list2) # Boolean Return[Data Compare]
print(result) # TypeError: '>=' not supported between instances of 'str' and 'int'
```

```
# Comparison List [ASCII Compare] ['a'>'A'>'0']
_list1 = ['a','b','c']
_list2 = ['1','2','3','4','5']
result = (_list1 >= _list2) # Boolean Return[Data Compare]
print(result) # True
```

True

## **Removing items from a list:**

1. `.remove()` - Remove the First Occurrence of an element.
2. `.pop()` - Removes by index, return value can be stored back to variable.
3. `del()` - Delete elements by index or slicing.
4. `.clear()` - Empties the List.

```

country_list = ['India', 'Russia', 'America', 'China', 'Canada', 'Japan', 'Vietnam', 'Norway', 'France',
               'Croatia', 'Australia', 'Finland', 'Pakistan', 'Spain', 'Singapore', 'New-Zealand']
country_list.remove('Spain') # Element Value
print(country_list)

['India', 'Russia', 'America', 'China', 'Canada', 'Japan', 'Vietnam', 'Norway', 'France', 'Croatia', 'Australia',
 'Finland', 'Pakistan', 'Singapore', 'New-Zealand']

country_list.remove('Finland') # Element Value
print(country_list)

['India', 'Russia', 'America', 'China', 'Canada', 'Japan', 'Vietnam', 'Norway', 'France', 'Croatia', 'Australia',
 'Pakistan', 'Singapore', 'New-Zealand']

country_list.remove('Brazil') # Element Value
print(country_list) # ValueError: list.remove(x): x not in list

```

```

# pop() -> While popping out the element you pass an index, you can take a backup of popped element
pop_country = country_list.pop(-1) # 'New-Zealand'
print(pop_country) # 'New-Zealand'
print(country_list) # Full List Except 'New-Zealand'

New-Zealand
['India', 'Russia', 'America', 'China', 'Canada', 'Japan', 'Vietnam', 'Norway', 'France', 'Croatia', 'Australia',
 'Pakistan', 'Singapore']

# append() # Insert at the end.
country_list.append(pop_country)
print(country_list)

['India', 'Russia', 'America', 'China', 'Canada', 'Japan', 'Vietnam', 'Norway', 'France', 'Croatia', 'Australia',
 'Pakistan', 'Singapore', 'New-Zealand']

```

```

country_replacement = country_list.pop(9) # 'Croatia'
print(country_replacement) # 'Croatia'
print(country_list)

Croatia
['India', 'Russia', 'America', 'China', 'Canada', 'Japan', 'Vietnam', 'Norway', 'France', 'Australia', 'Pakistan',
 'Singapore', 'New-Zealand']

# insert(pos , val)
country_list.insert(4 , country_replacement)
print(country_list)

['India', 'Russia', 'America', 'China', 'Croatia', 'Canada', 'Japan', 'Vietnam', 'Norway', 'France', 'Australia',
 'Pakistan', 'Singapore', 'New-Zealand']

# del, We can pass an indexing or slicing to del an element of a list.
del country_list[3] # 'China'
print(country_list)

['India', 'Russia', 'America', 'Croatia', 'Canada', 'Japan', 'Vietnam', 'Norway', 'France', 'Australia', 'Pakistan',
 'Singapore', 'New-Zealand']

```

```

# Slicing[Start:Stop:Step]
del country_list[3:6] # ['Croatia', 'Canada', 'Japan']
print(country_list)

['India', 'Russia', 'America', 'Vietnam', 'Norway', 'France', 'Australia', 'Pakistan', 'Singapore', 'New-Zealand']

del country_list[-2:] # ['Singapore', 'New-Zealand']
print(country_list)

['India', 'Russia', 'America', 'Vietnam', 'Norway', 'France', 'Australia', 'Pakistan']

# .clear() -> Empties the List
country_list.clear()
print(country_list) # []

[]

# del <listName> : Remove list from the memory
del country_list
print(country_list) # NameError: name 'country_list' is not defined

```

```

# Sorting [Ascending - Default] & Desc[Reverse()] [::-1] # 'ASCII'
country_list = ['India', 'Russia', 'America', 'china', 'Canada', 'japan', 'Vietnam', 'Norway', 'France',
               'Croatia', 'Australia', 'Finland', 'Pakistan', 'spain', 'Singapore', 'New-Zealand']
country_list.sort() # ['A'-'Z', 'a'-'z']

print(country_list)

['America', 'Australia', 'Canada', 'Croatia', 'Finland', 'France', 'India', 'New-Zealand', 'Norway', 'Pakistan',
 'Russia', 'Singapore', 'Vietnam', 'china', 'japan', 'spain']

# country_list.sort()
print(country_list[::-1])

['spain', 'japan', 'china', 'Vietnam', 'Singapore', 'Russia', 'Pakistan', 'Norway', 'New-Zealand', 'India', 'France',
 'Finland', 'Croatia', 'Canada', 'Australia', 'America']

```

```

country_list = ['India', 'Russia', 'America', 'china', 'Canada', 'japan', 'Vietnam', 'Norway', 'France',
               'Croatia', 'Australia', 'Finland', 'Pakistan', 'spain', 'Singapore', 'New-Zealand']
country_list.sort(reverse = True) # Descending
print(country_list)

['spain', 'japan', 'china', 'Vietnam', 'Singapore', 'Russia', 'Pakistan', 'Norway', 'New-Zealand', 'India', 'France',
 'Finland', 'Croatia', 'Canada', 'Australia', 'America']

# ASCII Role ✕
country_list = ['India', 'Russia', 'America', 'china', 'Canada', 'japan', 'Vietnam', 'Norway', 'France',
               'Croatia', 'Australia', 'Finland', 'Pakistan', 'spain', 'Singapore', 'New-Zealand']
country_list.sort(key = str.lower) # Truly Sorted
print(country_list)

['America', 'Australia', 'Canada', 'china', 'Croatia', 'Finland', 'France', 'India', 'japan', 'New-Zealand',
 'Norway', 'Pakistan', 'Russia', 'Singapore', 'spain', 'Vietnam']

```

```

country_list = ['India', 'Russia', 'America', 'china', 'Canada', 'japan', 'Vietnam', 'Norway', 'France',
               'Croatia', 'Australia', 'Finland', 'Pakistan', 'spain', 'Singapore', 'New-Zealand']
country_list.sort(key = str.upper) # Truly Sorted
print(country_list) # 'INDIA', 'RUSSIA', 'AMERICA'

['America', 'Australia', 'Canada', 'china', 'Croatia', 'Finland', 'France', 'India', 'japan', 'New-Zealand',
 'Norway', 'Pakistan', 'Russia', 'Singapore', 'spain', 'Vietnam']

# ASCII Rule ✕
country_list = ['India', 'Russia', 'America', 'china', 'Canada', 'japan', 'Vietnam', 'Norway', 'France',
               'Croatia', 'Australia', 'Finland', 'Pakistan', 'spain', 'Singapore', 'New-Zealand']
country_list.sort(key = str.lower , reverse = True) # Reverse the Truly Sorted List
print(country_list)

['Vietnam', 'spain', 'Singapore', 'Russia', 'Pakistan', 'Norway', 'New-Zealand', 'japan', 'India', 'France',
 'Finland', 'Croatia', 'china', 'Canada', 'Australia', 'America']

```

```

country_list = ['India', 'Russia', 'America', 'china', 'Canada', 'japan', 'Vietnam', 'Norway', 'France',
               'Croatia', 'Australia', 'Finland', 'Pakistan', 'spain', 'Singapore', 'New-Zealand']
country_list.sort(key = str.lower) #Truely Sorted List
print(country_list[::-1])

['Vietnam', 'spain', 'Singapore', 'Russia', 'Pakistan', 'Norway', 'New-Zealand', 'japan', 'India', 'France',
 'Finland', 'Croatia', 'china', 'Canada', 'Australia', 'America']

List =['1','2','3','4', 'India', 'usa', 'Japan', 'Austria', 'italy', '7','8','9'] # Homogenous
List.sort(key = str.lower) # ['1', '2', '3', '4', '7', '8', '9', 'Austria', 'India', 'italy', 'Japan', 'usa']

print(List)

['1', '2', '3', '4', '7', '8', '9', 'Austria', 'India', 'italy', 'Japan', 'usa']

```

```

List =[1,2,3,4, 'India', 'usa', 'Japan', 'Austria', 'italy', 7,8,9] # Heterogenous
new_list = []
for x in List:
    new_list.append(str(x))

print(new_list)

['1', '2', '3', '4', 'India', 'usa', 'Japan', 'Austria', 'italy', '7', '8', '9']

new_list.sort(key = str.lower)
print(new_list)

['1', '2', '3', '4', '7', '8', '9', 'Austria', 'India', 'italy', 'Japan', 'usa']

# 'Joining' strings from a list
word_list = ['Python', 'is', 'an', 'awesome', 'programming', 'language']
final_statement = ' '.join(word_list)
print(final_statement)

Python is an awesome programming language

```

```

char_list = ['P', 'y', 't', 'h', 'o', 'n', ' ', 'P', 'r', 'o', 'g', 'r', 'a', 'm', 'm', 'i', 'n', 'g']
result = ''.join(char_list)
print(result)

Python Programming

# .count() in List # Count the repeated value in a List.
country_list = ['India', 'Russia', 'America', 'china', 'Canada', 'japan', 'Vietnam', 'Norway', 'France',
                'Croatia', 'Australia', 'Finland', 'Pakistan', 'spain', 'Singapore', 'New-Zealand',
                'India', 'india', 'China']
count_india = country_list.count('India') # 2
print(count_india) # 'India' != 'india' # 2

2

num_list = [1, 2, 3, 1, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 12, 2, 2, 2, 121, 1, 12, 12, 2]
print(num_list.count(2))
print(num_list.count(1))

12
9

```

```

# DeepCopy [Pointer points to the same object] VS Shallow Copy [Independent Object]
country_list = ['India', 'Russia', 'America', 'china', 'Canada', 'japan', 'Vietnam', 'Norway', 'France',
                'Croatia', 'Australia', 'Finland', 'Pakistan', 'spain', 'Singapore', 'New-Zealand']
new_country_list = country_list # Deep Copy
print(id(country_list)) # 'Memory Address'
print(id(new_country_list)) # 'Memory Address'

1958077271232
1958077271232

new_country_list[-5:] = ['Brazil']
# new_country_list and country_list both will be affected [Deep Copy]
print(new_country_list)

['India', 'Russia', 'America', 'china', 'Canada', 'japan', 'Vietnam', 'Norway', 'France', 'Croatia', 'Australia',
 'Brazil']

print(country_list)

['India', 'Russia', 'America', 'china', 'Canada', 'japan', 'Vietnam', 'Norway', 'France', 'Croatia', 'Australia',
 'Brazil']

```

```

# Shallow Copy #.copy()
country_list = ['India', 'Russia', 'America', 'china', 'Canada', 'japan', 'Vietnam', 'Norway', 'France',
                'Croatia', 'Australia', 'Finland', 'Pakistan', 'spain', 'Singapore', 'New-Zealand']
new_country_list = country_list.copy() # Shallow Copy
print(id(country_list)) # Memory Address
print(id(new_country_list)) # Memory Address
# Different Memory Address as they both are independent data
1958117325568
1958117316608

new_country_list[-3:] = ['Qatar']
print(new_country_list)

['India', 'Russia', 'America', 'china', 'Canada', 'japan', 'Vietnam', 'Norway', 'France', 'Croatia', 'Australia',
 'Finland', 'Pakistan', 'Qatar']

print(country_list) # No Changes Reflected on my main List
['India', 'Russia', 'America', 'china', 'Canada', 'japan', 'Vietnam', 'Norway', 'France', 'Croatia', 'Australia',
 'Finland', 'Pakistan', 'spain', 'Singapore', 'New-Zealand']

```

```

# Shallow Copy # slicing
country_list = ['India', 'Russia', 'America', 'china', 'Canada', 'japan', 'Vietnam', 'Norway', 'France',
                'Croatia', 'Australia', 'Finland', 'Pakistan', 'spain', 'Singapore', 'New-Zealand']
new_country_list = country_list[:] # Shallow Copy
print(id(country_list)) # Memory Address
print(id(new_country_list)) # Memory Address
# Different Memory Address as they both are independent data
1958131855936
1958077271232

new_country_list[0:4] = ['Brazil', 'India']
print(new_country_list)

['Brazil', 'India', 'Canada', 'japan', 'Vietnam', 'Norway', 'France', 'Croatia', 'Australia', 'Finland', 'Pakistan',
 'spain', 'Singapore', 'New-Zealand']

print(country_list)
['India', 'Russia', 'America', 'china', 'Canada', 'japan', 'Vietnam', 'Norway', 'France', 'Croatia', 'Australia',
 'Finland', 'Pakistan', 'spain', 'Singapore', 'New-Zealand']

```

```

# Shallow Copy # list() Constructor
country_list = ['India', 'Russia', 'America', 'china', 'Canada', 'japan', 'Vietnam', 'Norway', 'France',
                'Croatia', 'Australia', 'Finland', 'Pakistan', 'spain', 'Singapore', 'New-Zealand']
new_country_list = list(country_list) # Shallow Copy
print(id(country_list)) # Memory Address
print(id(new_country_list)) # Memory Address
# Different Memory Address as they both are independent data
1958117279168
1958122910208

new_country_list[4:7] = ['Switzerland', 'Malaysia']
print(new_country_list)

['India', 'Russia', 'America', 'china', 'Switzerland', 'Malaysia', 'Norway', 'France', 'Croatia', 'Australia',
 'Finland', 'Pakistan', 'spain', 'Singapore', 'New-Zealand']

print(country_list)
['India', 'Russia', 'America', 'china', 'Canada', 'japan', 'Vietnam', 'Norway', 'France', 'Croatia', 'Australia',
 'Finland', 'Pakistan', 'spain', 'Singapore', 'New-Zealand']

```

```

# Adding a List using '+' Operator [Concatenation]
weekday_list = ['Mon', 'Tue', 'Wed', 'Thur', 'Fri']
weekend_list = ['Sat', 'Sun']
week_list = weekday_list + weekend_list
new_week_list = week_list # Deep Copy
print(id(week_list))
print(id(new_week_list)) # 'Pointing to the Same Memory Address'

1958117310656
1958117310656

# Before
print(week_list)
print(new_week_list)

['Mon', 'Tue', 'Wed', 'Thur', 'Fri', 'Sat', 'Sun']
['Mon', 'Tue', 'Wed', 'Thur', 'Fri', 'Sat', 'Sun']

```

```

# After
del new_week_list[-3:] # Removing 'Fri', 'Sat', 'Sun'
print(week_list)
print(new_week_list)

['Mon', 'Tue', 'Wed', 'Thur']
['Mon', 'Tue', 'Wed', 'Thur']

# List Comprehensions [Shorthand Property] # List Create
squared = [i**2 for i in range(1,6)] # range(start,stop,step) # [1,2,3,4,5]
print(squared) # [1,4,9,16,25]

[1, 4, 9, 16, 25]

add_100 = [i+100 for i in range(11,111,11)] # range(start,stop,step) # [11,22,33,44,55,...110]
print(add_100) # [111,122,133,144,155]

[111, 122, 133, 144, 155, 166, 177, 188, 199, 210]

```

## What Exactly is a Tuple?

- Ordered Collection (items maintains its positions)
- It can store different data-types together.
- But the Only Big Difference here is : Immutability (Once Created, their elements can't be modified)

```

_tuple = ()
print(_tuple)
print(type(_tuple)) # ()

()

<class 'tuple'>

_tuple = (1,2,'3','4',False,True,19.99,'9+10j','k','Coding')
print(_tuple)

(1, 2, '3', '4', False, True, 19.99, '9+10j', 'k', 'Coding')

```

```

_tuple = 1,2,3,4,5
print(_tuple)
print(type(_tuple))
(1, 2, 3, 4, 5)
<class 'tuple'>

# tuple() Constructor -> TypeCasting
country_list = ['India', 'Russia', 'America', 'china', 'Canada', 'japan', 'Vietnam', 'Norway', 'France',
                 'Croatia', 'Australia', 'Finland', 'Pakistan', 'spain', 'Singapore', 'New-Zealand']
country_tuple = tuple(country_list)
print(country_tuple)
print(type(country_tuple))

('India', 'Russia', 'America', 'china', 'Canada', 'japan', 'Vietnam', 'Norway', 'France', 'Croatia', 'Australia',
 'Finland', 'Pakistan', 'spain', 'Singapore', 'New-Zealand')
<class 'tuple'>

char_tuple = tuple('Python') # ('P', 'y', 't', 'h', 'o', 'n') # int('77') # 77
print(char_tuple)

('P', 'y', 't', 'h', 'o', 'n')

```

```

_tuple = tuple(('Python',))
print(_tuple)
print(type(_tuple))

('Python',)
<class 'tuple'>

nested_list = [
    [1,2,3,4,5],
    [0,2,4,6,8],
    [1,3,5,7,9],
]
nested_tuple = tuple(nested_list)
nested_tuple

([1, 2, 3, 4, 5], [0, 2, 4, 6, 8], [1, 3, 5, 7, 9])

# List as an element is mutable even if inside the immutable container
nested_tuple[0] # [1, 2, 3, 4, 5]

[1, 2, 3, 4, 5]

```

```

nested_tuple[:2] # ([1, 2, 3, 4, 5], [0, 2, 4, 6, 8])
([1, 2, 3, 4, 5], [0, 2, 4, 6, 8])

nested_tuple[0].append(6)
nested_tuple

([1, 2, 3, 4, 5, 6], [0, 2, 4, 6, 8], [1, 3, 5, 7, 9])

# nested_tuple : Element_tuple
nested_tuple = (('Mon', 'Tue', 'Wed'), ('Thurs', 'Fri', 'Sat', 'Sun'))
print(nested_tuple)
print(type(nested_tuple)) # 'tuple'
print(type(nested_tuple[0])) # 'tuple'
print(type(nested_tuple[1])) # 'tuple'

(('Mon', 'Tue', 'Wed'), ('Thurs', 'Fri', 'Sat', 'Sun'))
<class 'tuple'>
<class 'tuple'>
<class 'tuple'>

```

```

nested_tuple[0] = ('Monday', 'Tuesday', 'Wednesday')
print(nested_tuple) # TypeError: 'tuple' object does not support item assignment

nested_tuple[0].append('Monday') # AttributeError: 'tuple' object has no attribute 'append'

```

```

# Indexing & Slicing in Tuples
country_list = ['India', 'Russia', 'America', 'china', 'Canada', 'japan', 'Vietnam', 'Norway', 'France',
                'Croatia', 'Australia', 'Finland', 'Pakistan', 'spain', 'Singapore', 'New-Zealand']
country_tuple = tuple(country_list)
print(country_tuple[-7]) # 'Croatia'
print(country_tuple[5]) # 'Japan'
print(country_tuple[-16]) # 'India'
print(country_tuple[4:9:2]) # ('Canada', 'Vietnam', 'France')
print(country_tuple[-2:-11:-2]) # ('Singapore', 'Pakistan', 'Aus', 'France', 'Vietnam')
print(country_tuple[-7:15:-3]) # ()
print(country_tuple[-14:12:4]) # ('America', 'Vietnam', 'Aus')

Croatia
japan
India
('Canada', 'Vietnam', 'France')
('Singapore', 'Pakistan', 'Australia', 'France', 'Vietnam')
()
('America', 'Vietnam', 'Australia')

```

```
# Len() -> Finding the tuple Length
country_list = ['India', 'Russia', 'America', 'china', 'Canada', 'japan', 'Vietnam', 'Norway', 'France',
               'Croatia', 'Australia', 'Finland', 'Pakistan', 'spain', 'Singapore', 'New-Zealand']
country_tuple = tuple(country_list)
print(len(country_tuple)) # 16
```

```
16
```

```
# min() , max() , sum()
_num_list = (11,22,33,44,55,66,77,88,99,110,False,True) # Mixed Type
print(len(_num_list)) # 12
print(min(_num_list)) # 0[False]
print(max(_num_list)) # 110
print(sum(_num_list)) # 605 + True[1] + False[0] = 606
```

```
12
```

```
False
```

```
110
```

```
606
```