CN-Python-Lec06-Data Structures-I

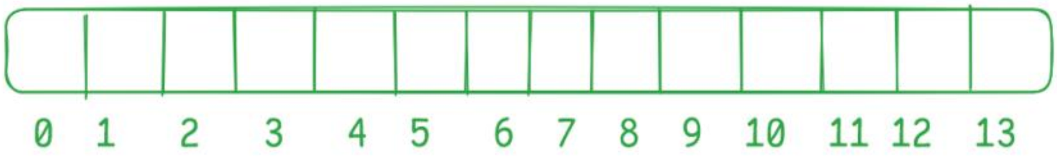
Data Structures-I

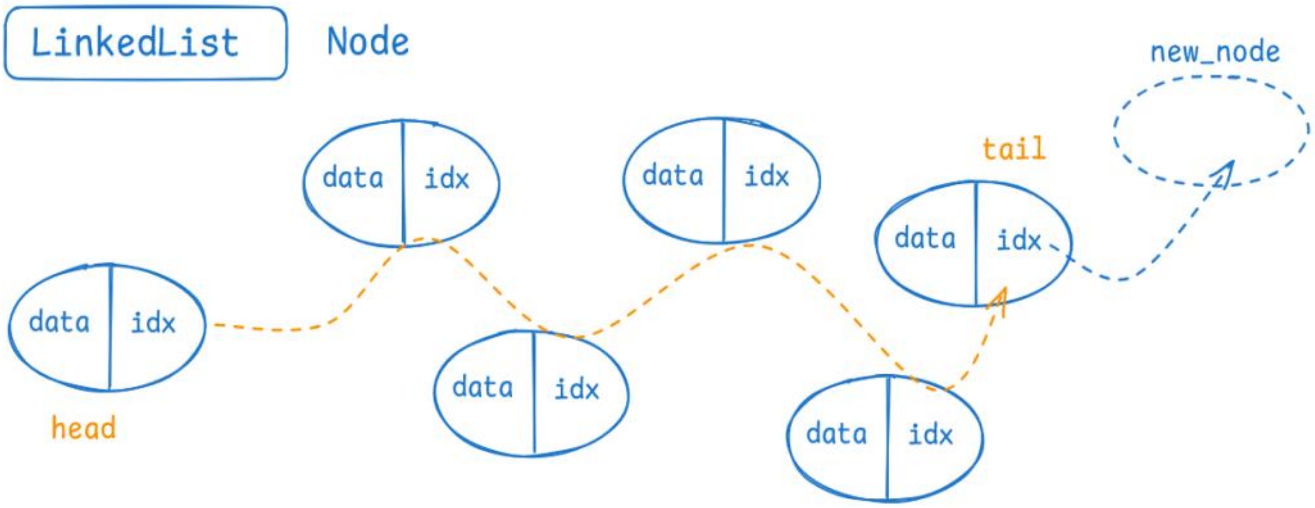
🎯 Session Objectives

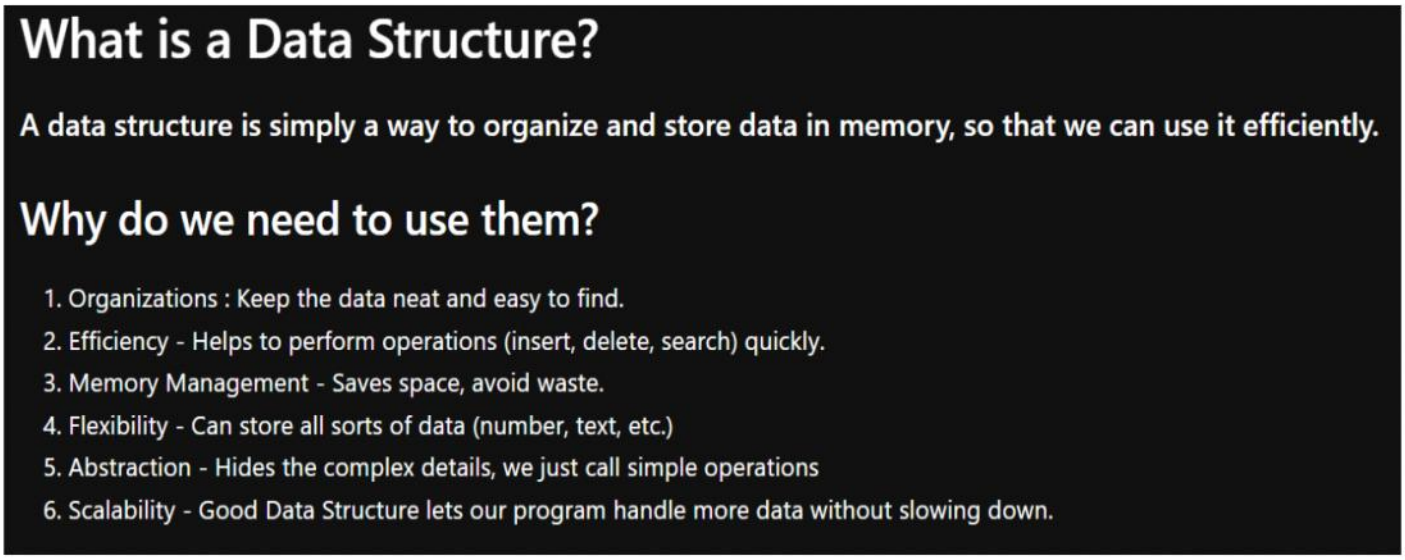
- 📚 Understand the meaning of data structures and why we use them.
- 📁 Common data structures in Python
- 📋 Understand what lists are.
- ⚙️ Understand common methods and operations associated with lists.
- 🚀 Understand the meaning of list comprehension
- 🔒 Understand what tuples are.
- 🛠️ Understand common methods and operations associated with tuples.
- ⚖️ Understand the Comparison between Lists and Tuples

What is an Array:

Time Complexity
Space Complexity





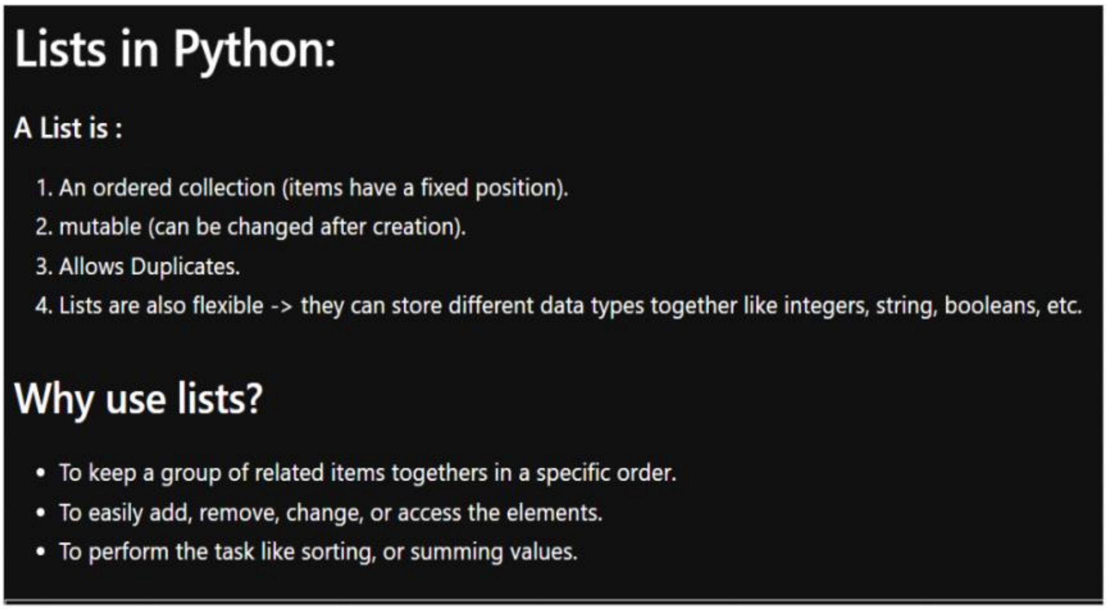


What is a Data Structure?

A data structure is simply a way to organize and store data in memory, so that we can use it efficiently.

Why do we need to use them?

1. Organizations : Keep the data neat and easy to find.
2. Efficiency - Helps to perform operations (insert, delete, search) quickly.
3. Memory Management - Saves space, avoid waste.
4. Flexibility - Can store all sorts of data (number, text, etc.)
5. Abstraction - Hides the complex details, we just call simple operations
6. Scalability - Good Data Structure lets our program handle more data without slowing down.



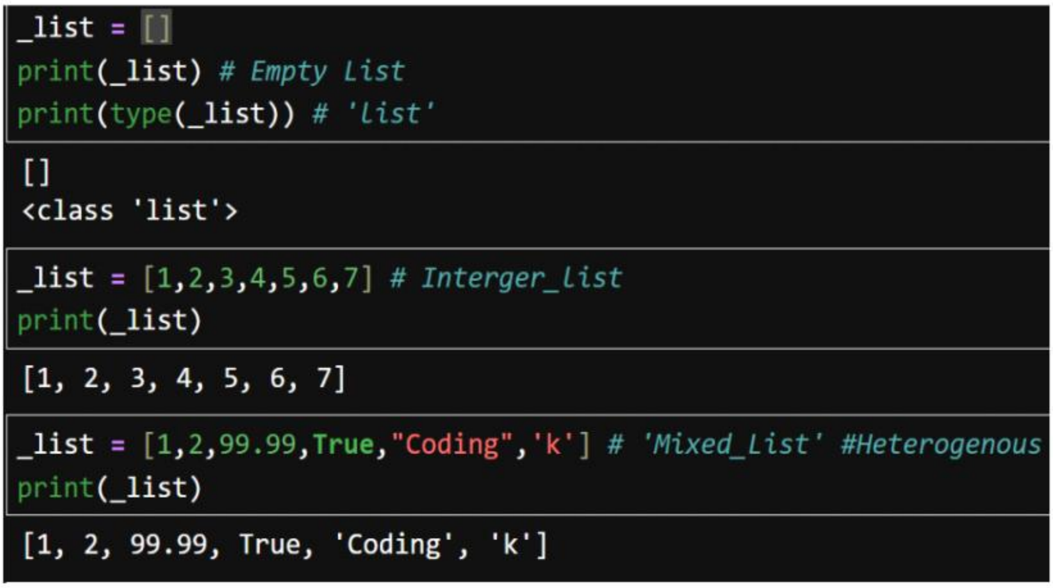
Lists in Python:

A List is :

1. An ordered collection (items have a fixed position).
2. mutable (can be changed after creation).
3. Allows Duplicates.
4. Lists are also flexible -> they can store different data types together like integers, string, booleans, etc.

Why use lists?

- To keep a group of related items togethers in a specific order.
- To easily add, remove, change, or access the elements.
- To perform the task like sorting, or summing values.



```
_list = []
print(_list) # Empty List
print(type(_list)) # 'list'
```

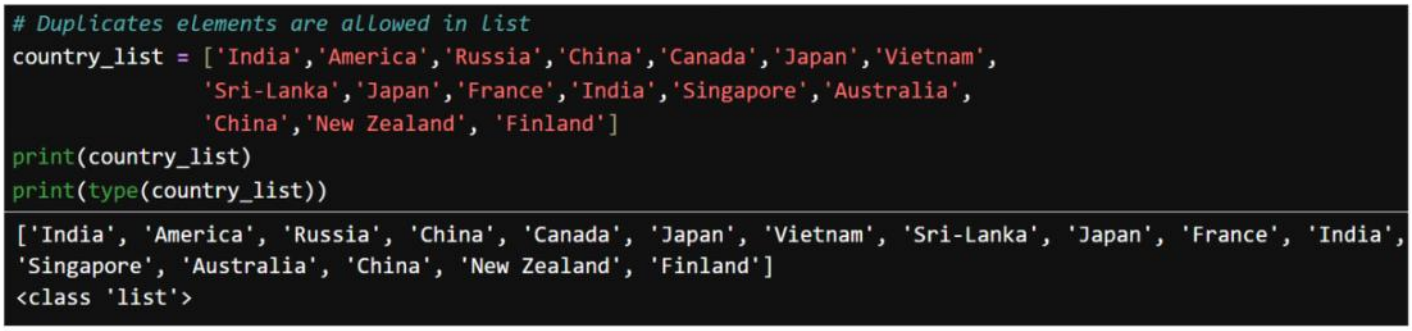
```
[]
<class 'list'>
```

```
_list = [1,2,3,4,5,6,7] # Interger_list
print(_list)
```

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

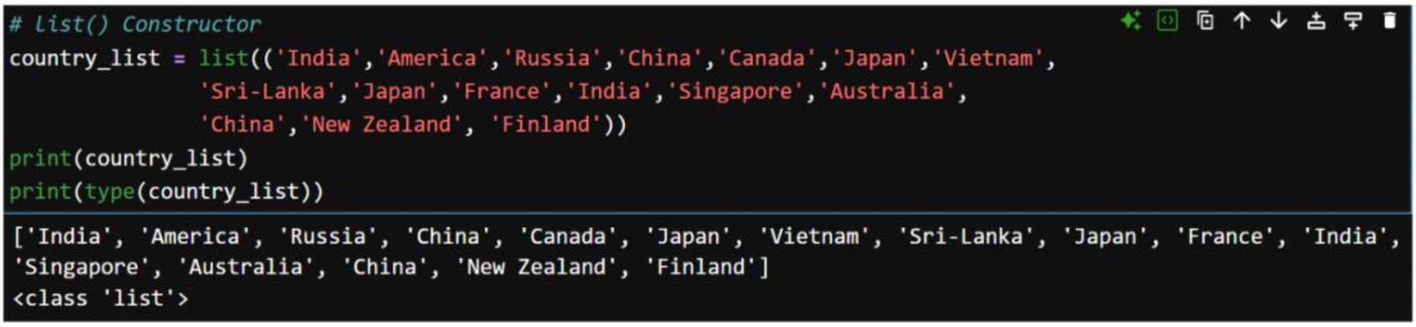
```
_list = [1,2,99.99,True,"Coding",'k'] # 'Mixed_List' #Heterogenous
print(_list)
```

```
[1, 2, 99.99, True, 'Coding', 'k']
```



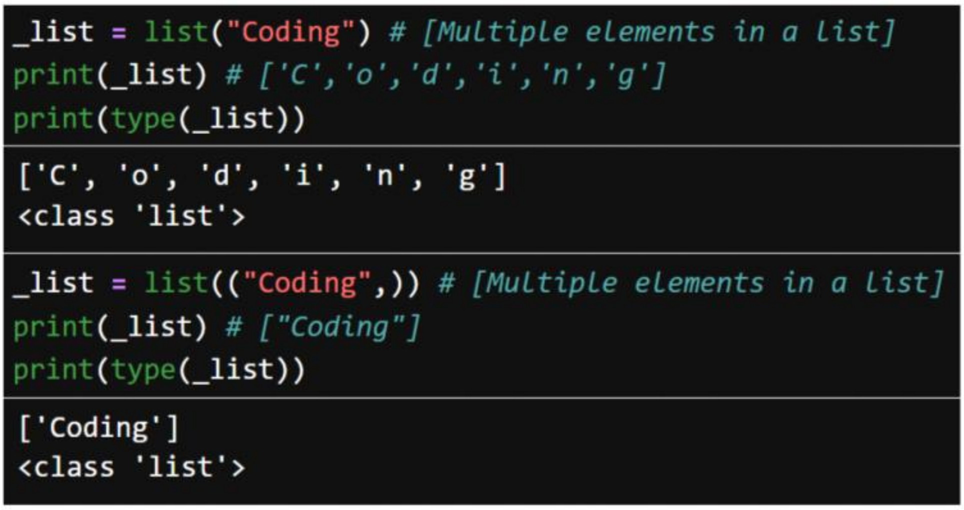
```
# Duplicates elements are allowed in List
country_list = ['India','America','Russia','China','Canada','Japan','Vietnam',
               'Sri-Lanka','Japan','France','India','Singapore','Australia',
               'China','New Zealand', 'Finland']
print(country_list)
print(type(country_list))
```

```
['India', 'America', 'Russia', 'China', 'Canada', 'Japan', 'Vietnam', 'Sri-Lanka', 'Japan', 'France', 'India',
'Singapore', 'Australia', 'China', 'New Zealand', 'Finland']
<class 'list'>
```



```
# List() Constructor
country_list = list(('India','America','Russia','China','Canada','Japan','Vietnam',
               'Sri-Lanka','Japan','France','India','Singapore','Australia',
               'China','New Zealand', 'Finland'))
print(country_list)
print(type(country_list))
```

```
['India', 'America', 'Russia', 'China', 'Canada', 'Japan', 'Vietnam', 'Sri-Lanka', 'Japan', 'France', 'India',
'Singapore', 'Australia', 'China', 'New Zealand', 'Finland']
<class 'list'>
```

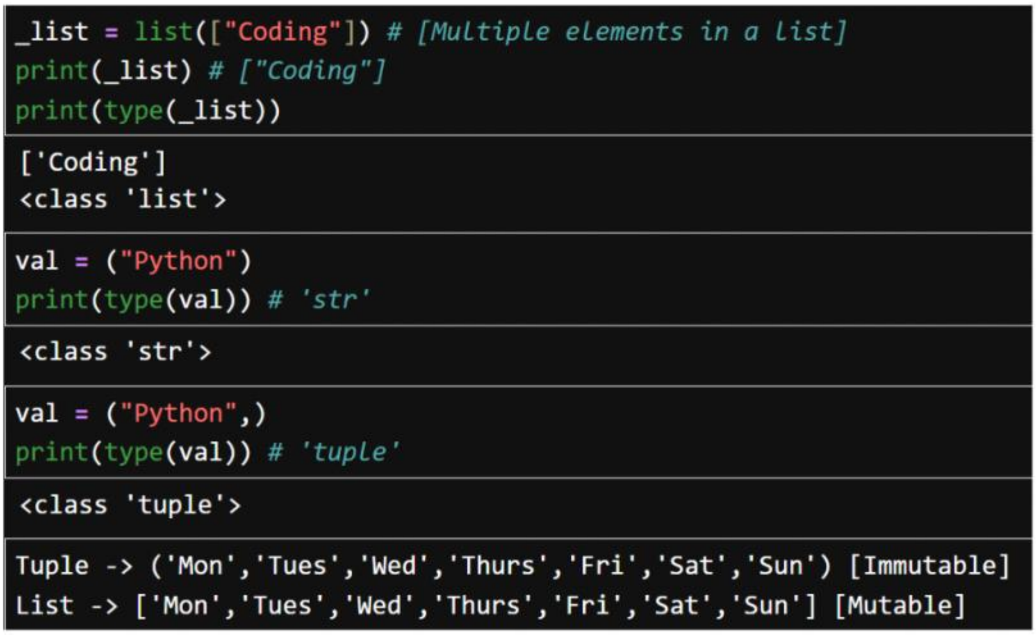


```
_list = list("Coding") # [Multiple elements in a List]
print(_list) # ['C','o','d','i','n','g']
print(type(_list))
```

```
['C', 'o', 'd', 'i', 'n', 'g']
<class 'list'>
```

```
_list = list(("Coding",)) # [Multiple elements in a List]
print(_list) # ["Coding"]
print(type(_list))
```

```
['Coding']
<class 'list'>
```



```
_list = list(["Coding"]) # [Multiple elements in a List]
print(_list) # ["Coding"]
print(type(_list))
```

```
['Coding']
<class 'list'>
```

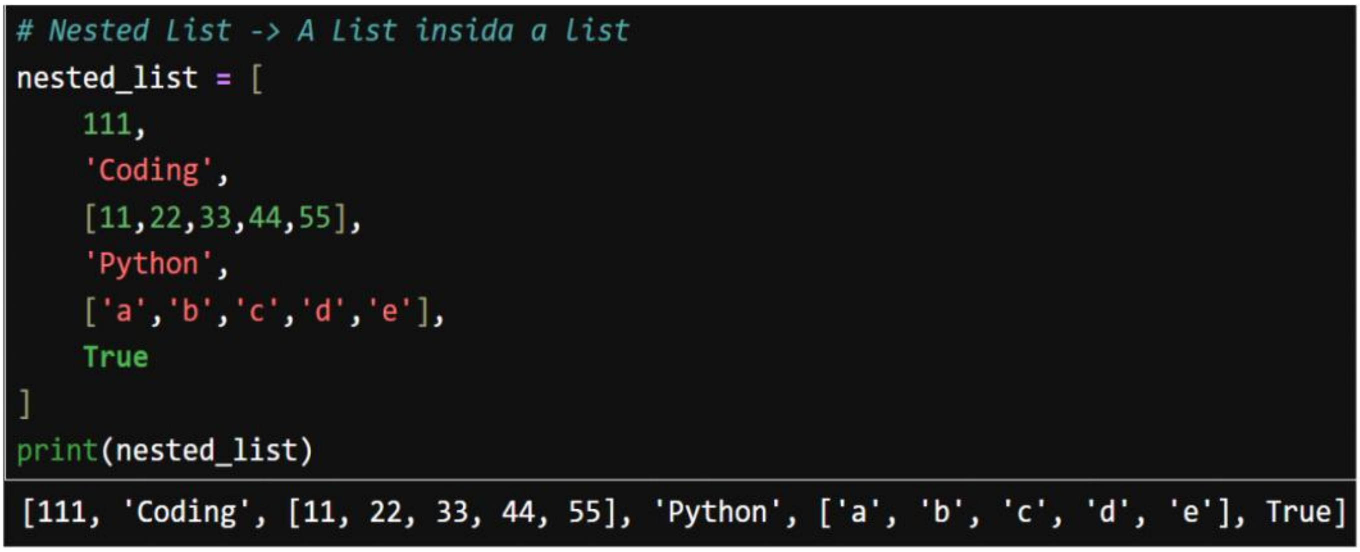
```
val = ("Python")
print(type(val)) # 'str'
```

```
<class 'str'>
```

```
val = ("Python",)
print(type(val)) # 'tuple'
```

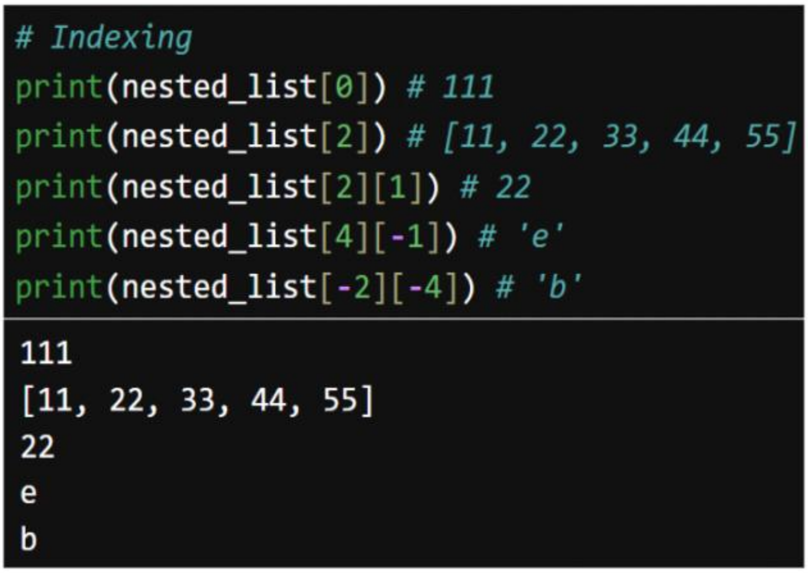
```
<class 'tuple'>
```

```
Tuple -> ('Mon','Tues','Wed','Thurs','Fri','Sat','Sun') [Immutable]
List -> ['Mon','Tues','Wed','Thurs','Fri','Sat','Sun'] [Mutable]
```



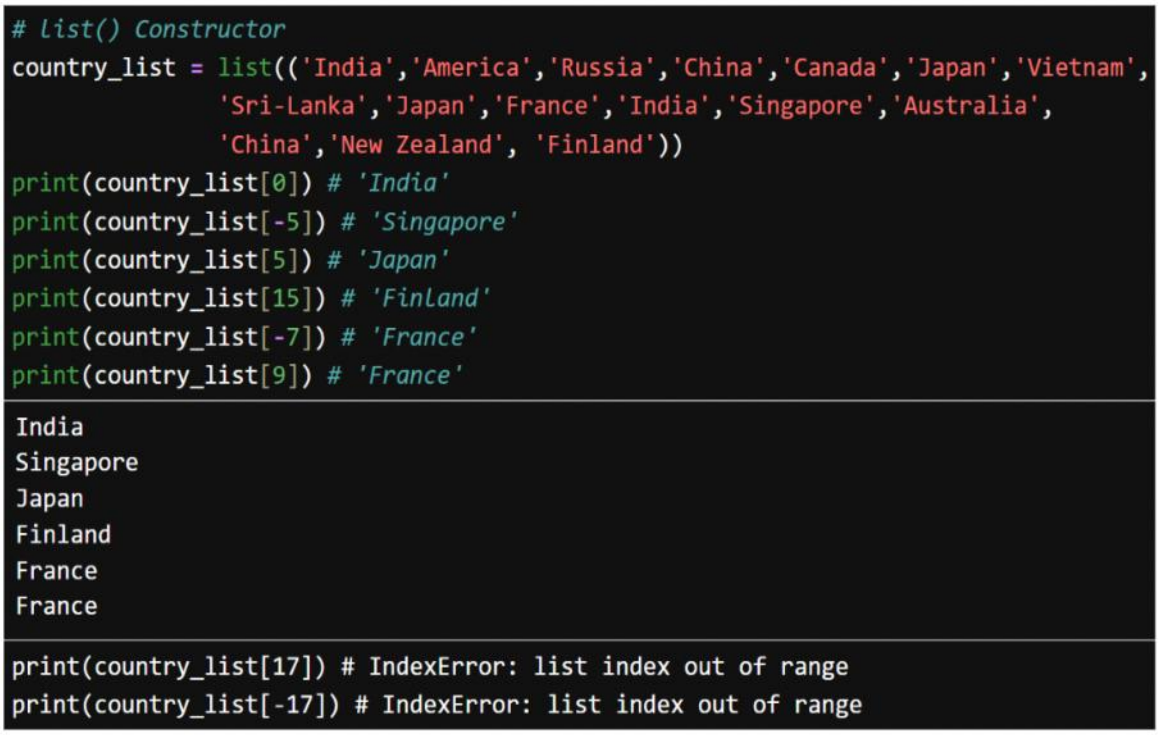
```
# Nested List -> A List insida a list
nested_list = [
    111,
    'Coding',
    [11,22,33,44,55],
    'Python',
    ['a','b','c','d','e'],
    True
]
print(nested_list)
```

```
[111, 'Coding', [11, 22, 33, 44, 55], 'Python', ['a', 'b', 'c', 'd', 'e'], True]
```



```
# Indexing
print(nested_list[0]) # 111
print(nested_list[2]) # [11, 22, 33, 44, 55]
print(nested_list[2][1]) # 22
print(nested_list[4][-1]) # 'e'
print(nested_list[-2][-4]) # 'b'
```

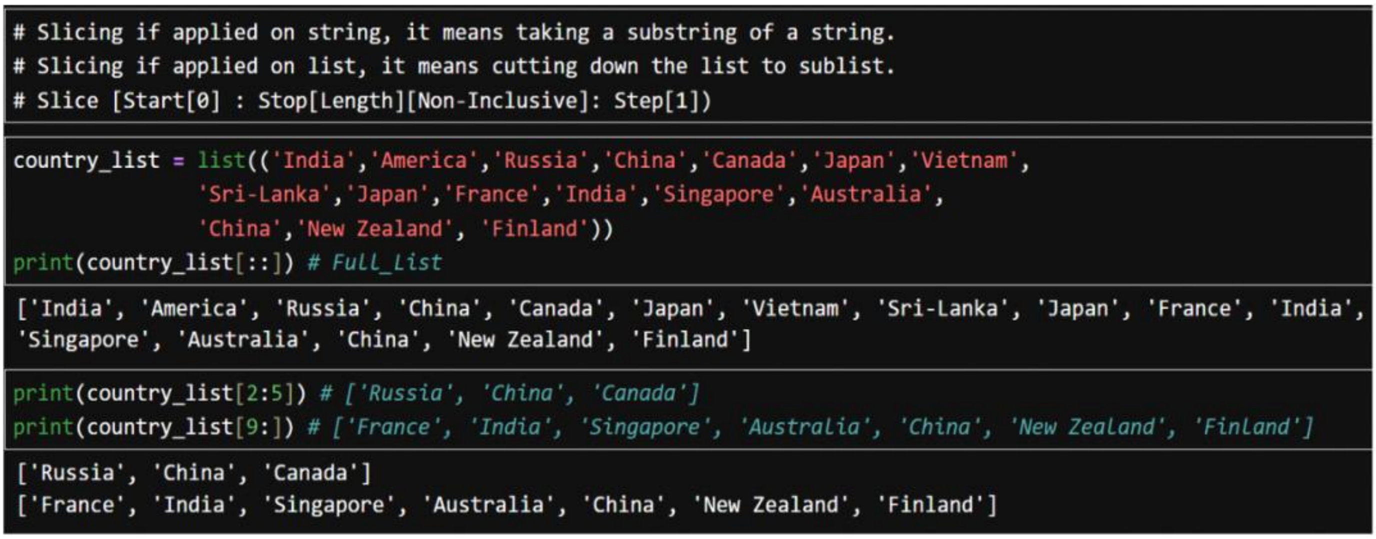
```
111
[11, 22, 33, 44, 55]
22
e
b
```



```
# list() Constructor
country_list = list(('India','America','Russia','China','Canada','Japan','Vietnam',
               'Sri-Lanka','Japan','France','India','Singapore','Australia',
               'China','New Zealand', 'Finland'))
print(country_list[0]) # 'India'
print(country_list[-5]) # 'Singapore'
print(country_list[5]) # 'Japan'
print(country_list[15]) # 'Finland'
print(country_list[-7]) # 'France'
print(country_list[9]) # 'France'
```

```
India
Singapore
Japan
Finland
France
France
```

```
print(country_list[17]) # IndexError: list index out of range
print(country_list[-17]) # IndexError: list index out of range
```



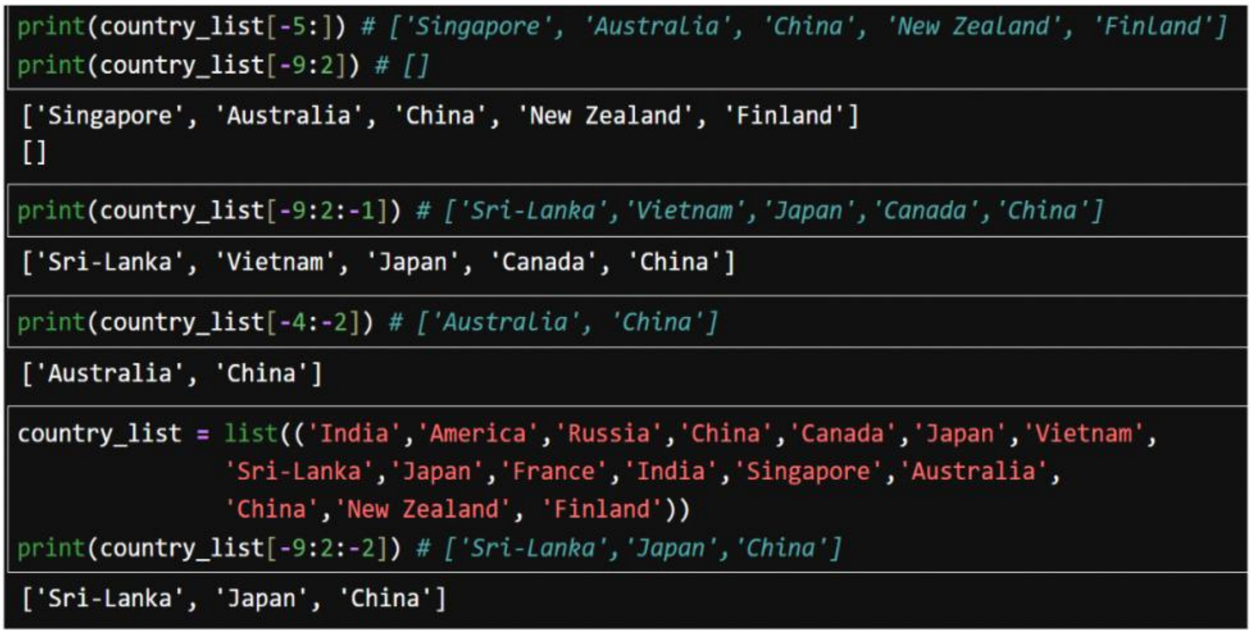
```
# Slicing if applied on string, it means taking a substring of a string.
# Slicing if applied on list, it means cutting down the list to sublist.
# Slice [Start[0] : Stop[Length][Non-Inclusive]: Step[1])
```

```
country_list = list(('India','America','Russia','China','Canada','Japan','Vietnam',
             'Sri-Lanka','Japan','France','India','Singapore','Australia',
             'China','New Zealand', 'Finland'))
print(country_list[::]) # Full_List
```

```
['India', 'America', 'Russia', 'China', 'Canada', 'Japan', 'Vietnam', 'Sri-Lanka', 'Japan', 'France', 'India',
'Singapore', 'Australia', 'China', 'New Zealand', 'Finland']
```

```
print(country_list[2:5]) # ['Russia', 'China', 'Canada']
print(country_list[9:]) # ['France', 'India', 'Singapore', 'Australia', 'China', 'New Zealand', 'Finland']
```

```
['Russia', 'China', 'Canada']
['France', 'India', 'Singapore', 'Australia', 'China', 'New Zealand', 'Finland']
```



```
print(country_list[-5:]) # ['Singapore', 'Australia', 'China', 'New Zealand', 'Finland']
print(country_list[-9:2]) # []
```

```
['Singapore', 'Australia', 'China', 'New Zealand', 'Finland']
[]
```

```
print(country_list[-9:2:-1]) # ['Sri-Lanka','Vietnam','Japan','Canada','China']
```

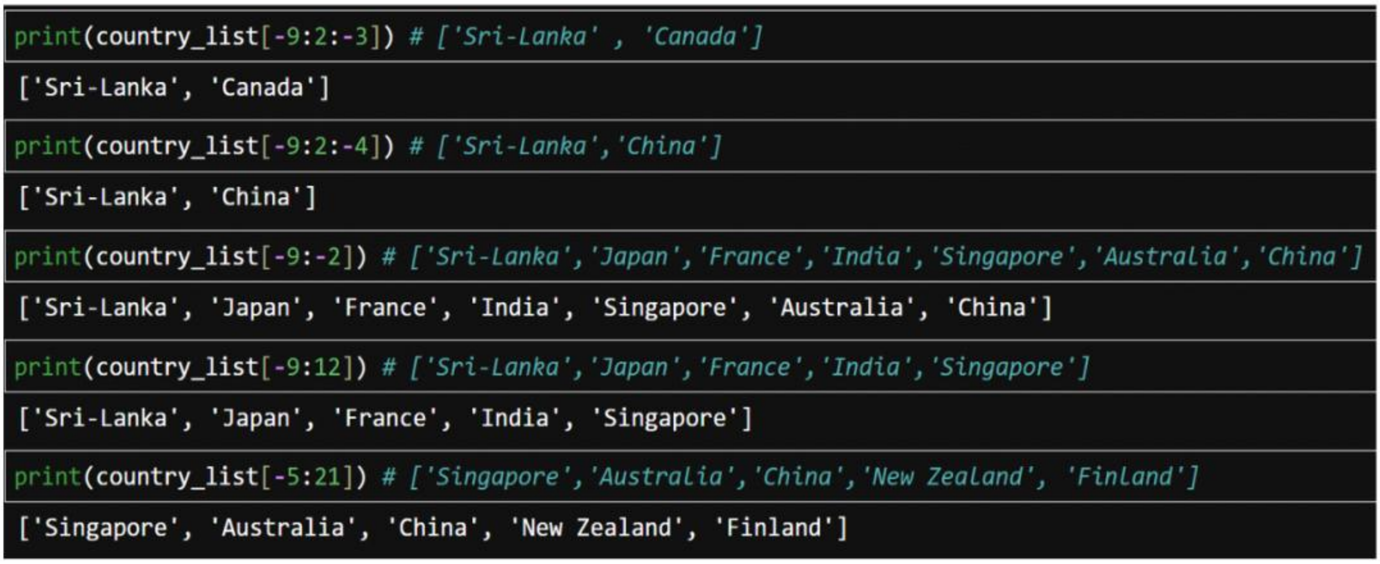
```
['Sri-Lanka', 'Vietnam', 'Japan', 'Canada', 'China']
```

```
print(country_list[-4:-2]) # ['Australia', 'China']
```

```
['Australia', 'China']
```

```
country_list = list(('India','America','Russia','China','Canada','Japan','Vietnam',
             'Sri-Lanka','Japan','France','India','Singapore','Australia',
             'China','New Zealand', 'Finland'))
print(country_list[-9:2:-2]) # ['Sri-Lanka','Japan','China']
```

```
['Sri-Lanka', 'Japan', 'China']
```



```
print(country_list[-9:2:-3]) # ['Sri-Lanka' , 'Canada']
```

```
['Sri-Lanka', 'Canada']
```

```
print(country_list[-9:2:-4]) # ['Sri-Lanka','China']
```

```
['Sri-Lanka', 'China']
```

```
print(country_list[-9:-2]) # ['Sri-Lanka','Japan','France','India','Singapore','Australia','China']
```

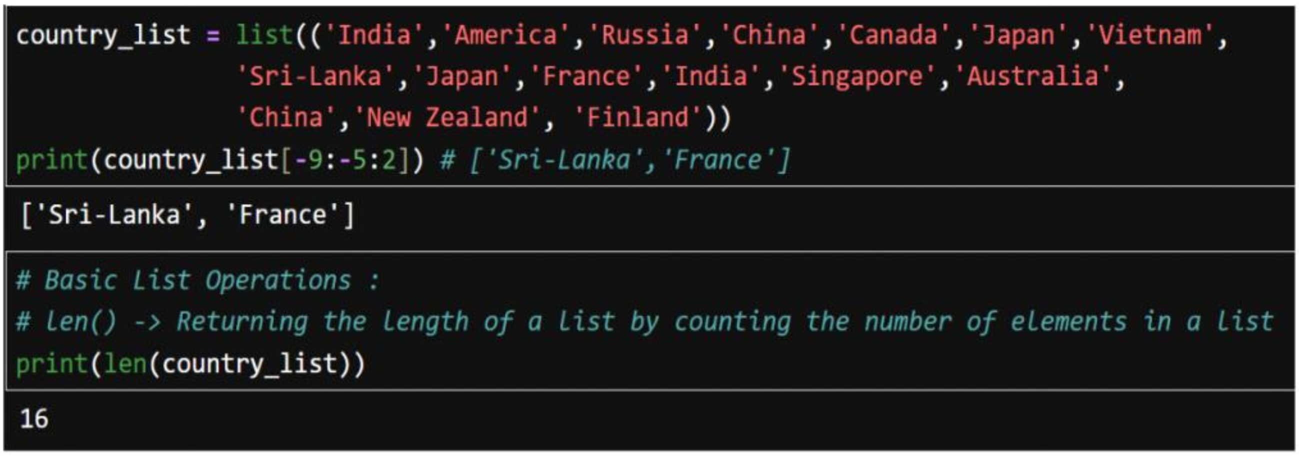
```
['Sri-Lanka', 'Japan', 'France', 'India', 'Singapore', 'Australia', 'China']
```

```
print(country_list[-9:12]) # ['Sri-Lanka','Japan','France','India','Singapore']
```

```
['Sri-Lanka', 'Japan', 'France', 'India', 'Singapore']
```

```
print(country_list[-5:21]) # ['Singapore','Australia','China','New Zealand', 'Finland']
```

```
['Singapore', 'Australia', 'China', 'New Zealand', 'Finland']
```

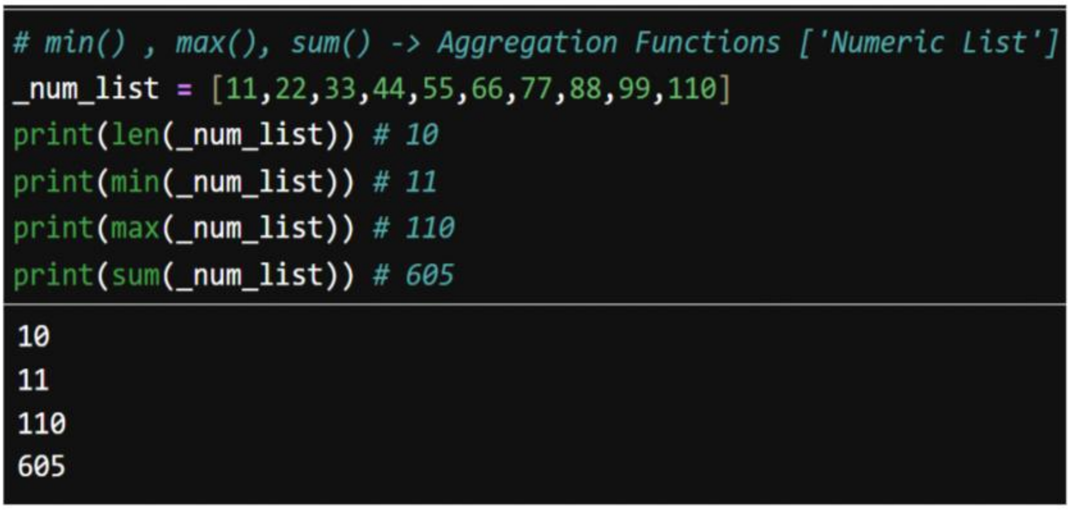


```
country_list = list(('India','America','Russia','China','Canada','Japan','Vietnam',
              'Sri-Lanka','Japan','France','India','Singapore','Australia',
              'China','New Zealand', 'Finland'))
print(country_list[-9:-5:2]) # ['Sri-Lanka','France']
```

```
['Sri-Lanka', 'France']
```

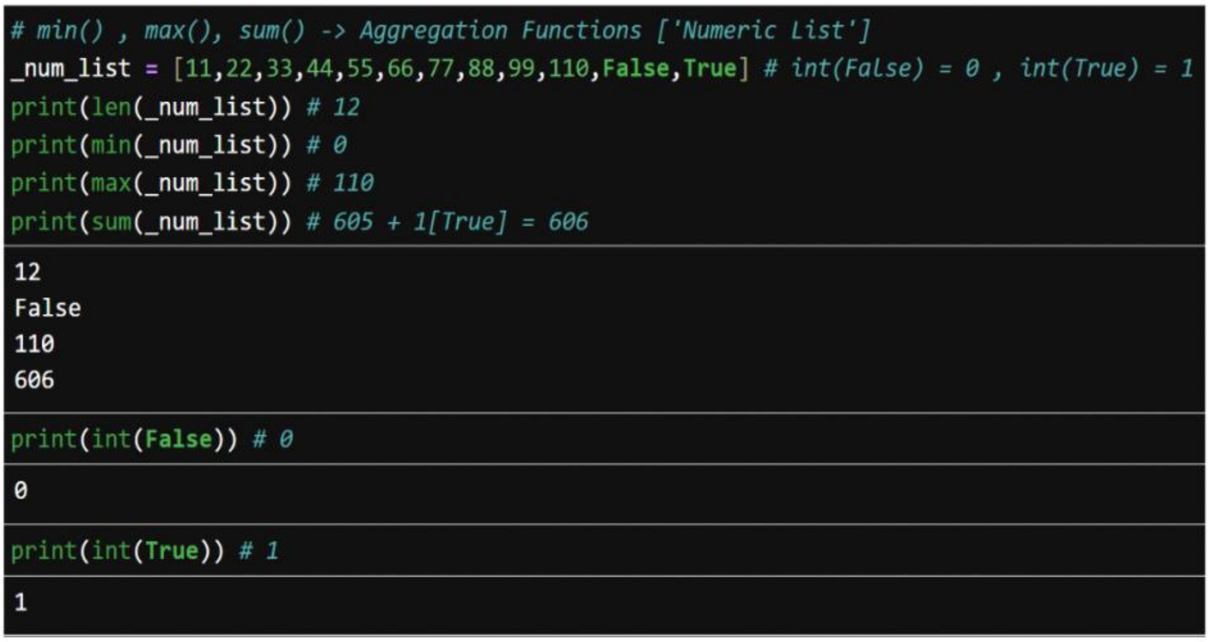
```
# Basic List Operations :
# len() -> Returning the length of a list by counting the number of elements in a list
print(len(country_list))
```

```
16
```



```
# min() , max(), sum() -> Aggregation Functions ['Numeric List']
_num_list = [11,22,33,44,55,66,77,88,99,110]
print(len(_num_list)) # 10
print(min(_num_list)) # 11
print(max(_num_list)) # 110
print(sum(_num_list)) # 605
```

```
10
11
110
605
```



```
# min() , max(), sum() -> Aggregation Functions ['Numeric List']
_num_list = [11,22,33,44,55,66,77,88,99,110,False,True] # int(False) = 0 , int(True) = 1
print(len(_num_list)) # 12
print(min(_num_list)) # 0
print(max(_num_list)) # 110
print(sum(_num_list)) # 605 + 1[True] = 606
```

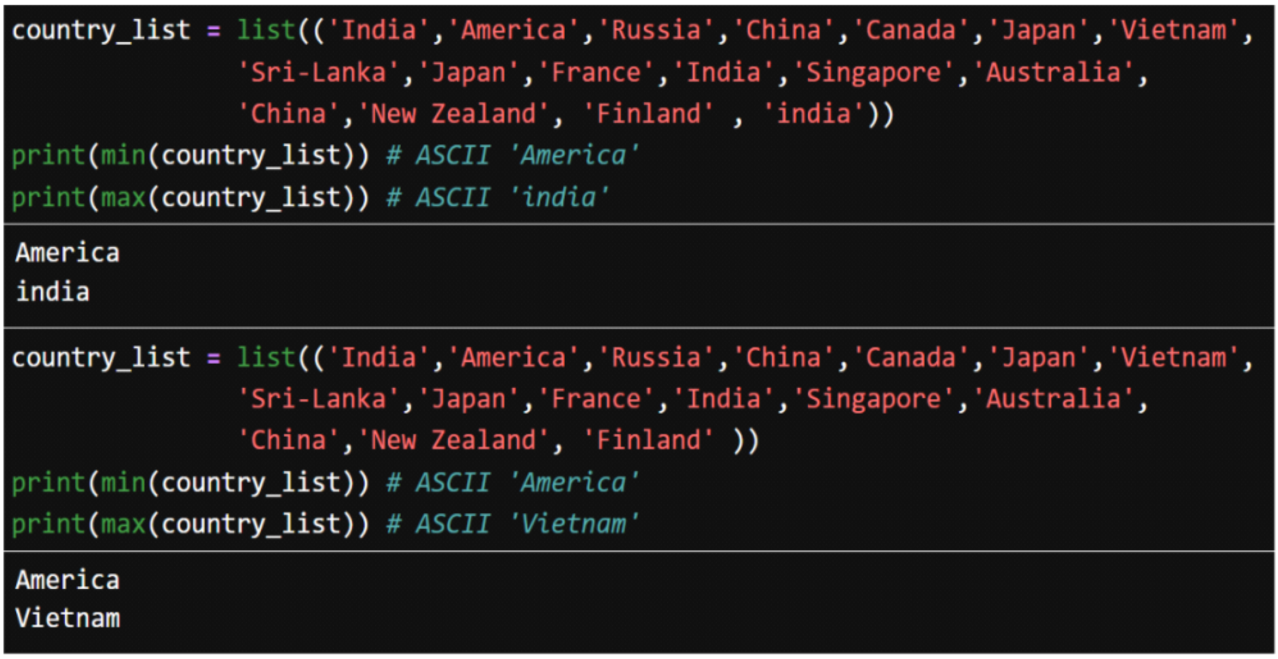
```
12
False
110
606
```

```
print(int(False)) # 0
```

```
0
```

```
print(int(True)) # 1
```

```
1
```

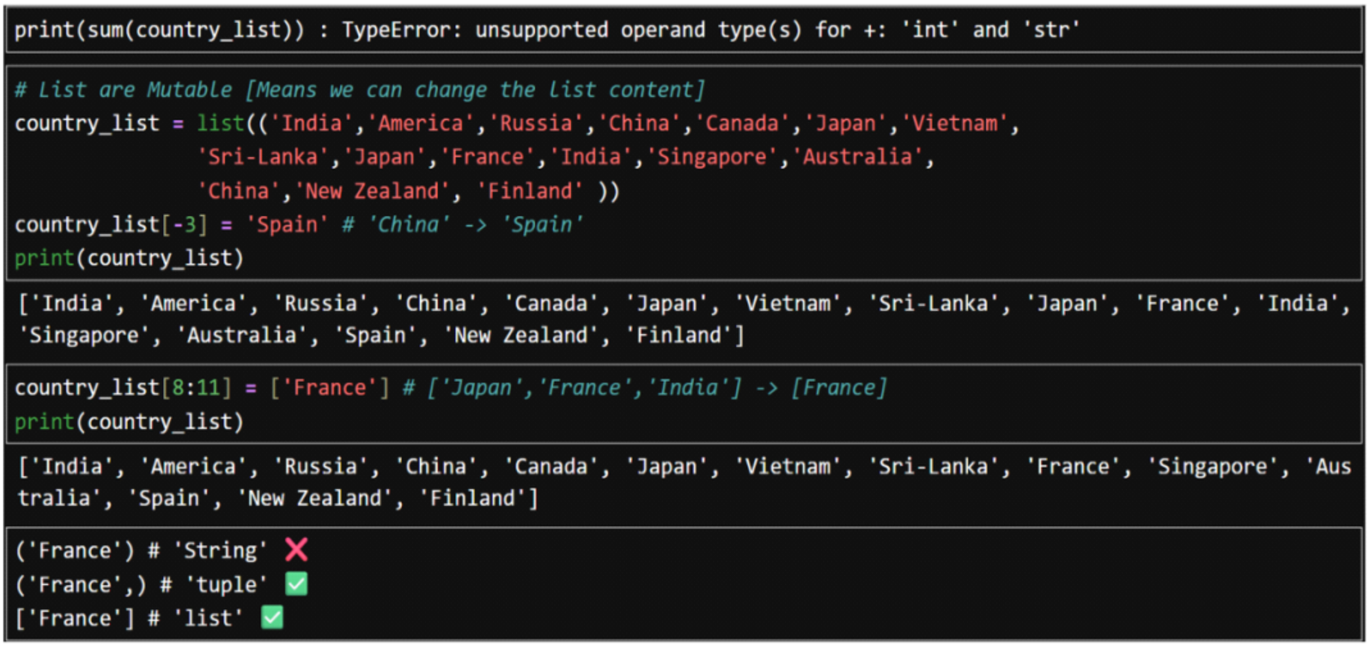


```
country_list = list(('India','America','Russia','China','Canada','Japan','Vietnam',
                'Sri-Lanka','Japan','France','India','Singapore','Australia',
                'China','New Zealand', 'Finland' , 'india'))
print(min(country_list)) # ASCII 'America'
print(max(country_list)) # ASCII 'india'
```

```
America
india
```

```
country_list = list(('India','America','Russia','China','Canada','Japan','Vietnam',
                'Sri-Lanka','Japan','France','India','Singapore','Australia',
                'China','New Zealand', 'Finland' ))
print(min(country_list)) # ASCII 'America'
print(max(country_list)) # ASCII 'Vietnam'
```

```
America
Vietnam
```



```
print(sum(country_list)) : TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

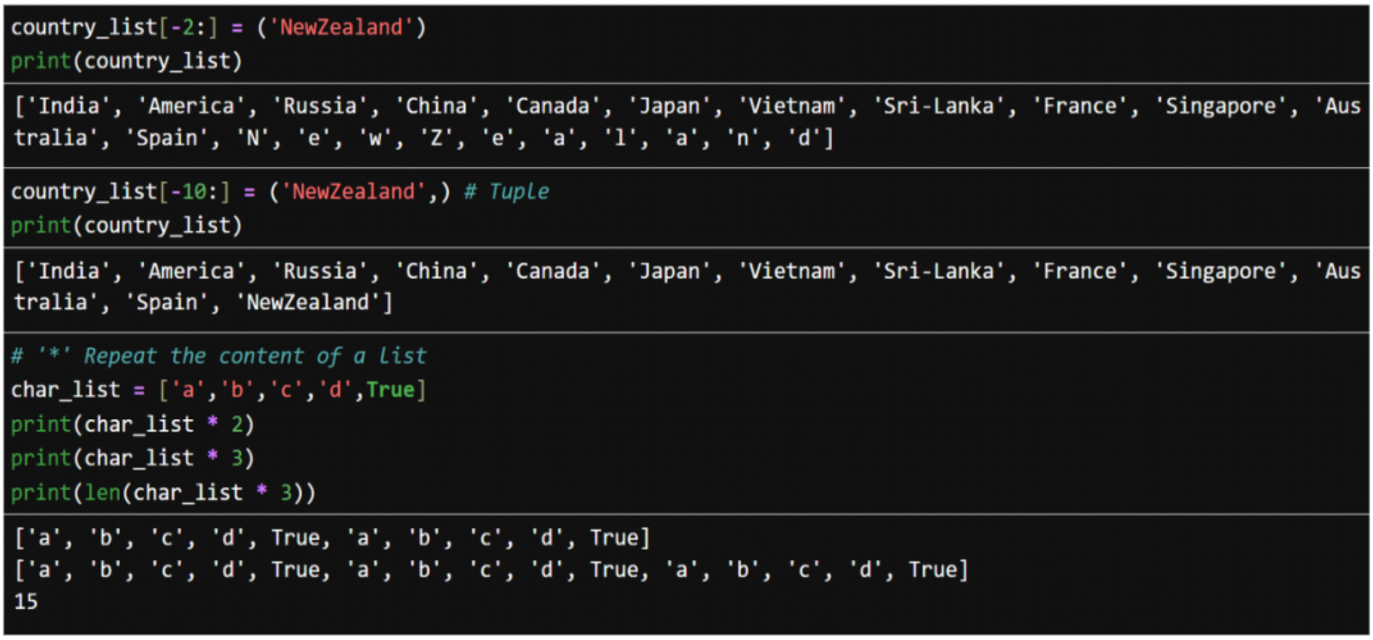
```
# List are Mutable [Means we can change the list content]
country_list = list(('India','America','Russia','China','Canada','Japan','Vietnam',
                'Sri-Lanka','Japan','France','India','Singapore','Australia',
                'China','New Zealand', 'Finland' ))
country_list[-3] = 'Spain' # 'China' -> 'Spain'
print(country_list)
```

```
['India', 'America', 'Russia', 'China', 'Canada', 'Japan', 'Vietnam', 'Sri-Lanka', 'Japan', 'France', 'India',
'Singapore', 'Australia', 'Spain', 'New Zealand', 'Finland']
```

```
country_list[8:11] = ['France'] # ['Japan','France','India'] -> [France]
print(country_list)
```

```
['India', 'America', 'Russia', 'China', 'Canada', 'Japan', 'Vietnam', 'Sri-Lanka', 'France', 'Singapore', 'Aus
tralia', 'Spain', 'New Zealand', 'Finland']
```

```
('France') # 'String' ❌
('France',) # 'tuple' ✅
['France'] # 'list' ✅
```



```
country_list[-2:] = ('NewZealand')
print(country_list)
```

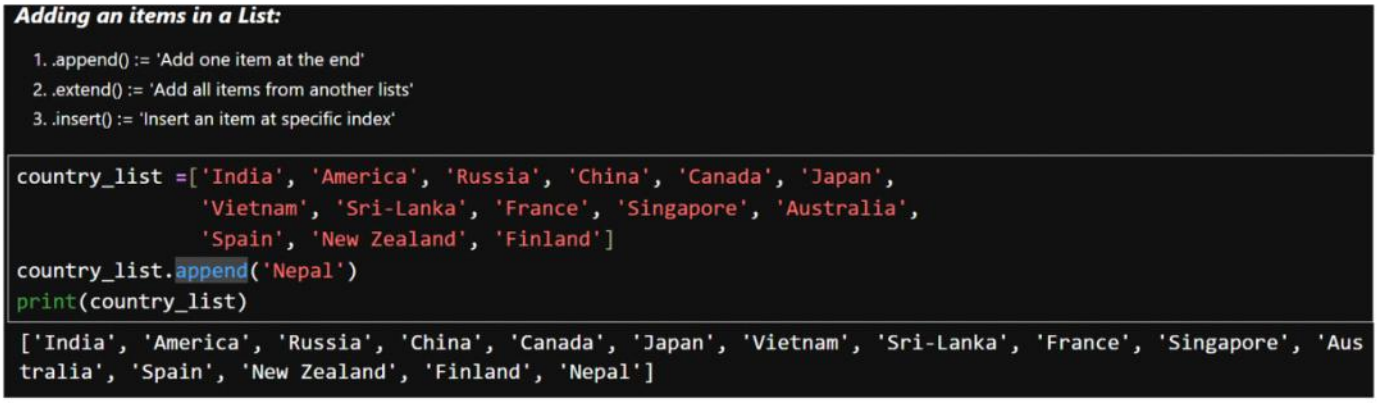
```
['India', 'America', 'Russia', 'China', 'Canada', 'Japan', 'Vietnam', 'Sri-Lanka', 'France', 'Singapore', 'Aus
tralia', 'Spain', 'N', 'e', 'w', 'Z', 'e', 'a', 'l', 'a', 'n', 'd']
```

```
country_list[-10:] = ('NewZealand',) # Tuple
print(country_list)
```

```
['India', 'America', 'Russia', 'China', 'Canada', 'Japan', 'Vietnam', 'Sri-Lanka', 'France', 'Singapore', 'Aus
tralia', 'Spain', 'NewZealand']
```

```
# '*' Repeat the content of a list
char_list = ['a','b','c','d',True]
print(char_list * 2)
print(char_list * 3)
print(len(char_list * 3))
```

```
['a', 'b', 'c', 'd', True, 'a', 'b', 'c', 'd', True]
['a', 'b', 'c', 'd', True, 'a', 'b', 'c', 'd', True, 'a', 'b', 'c', 'd', True]
15
```

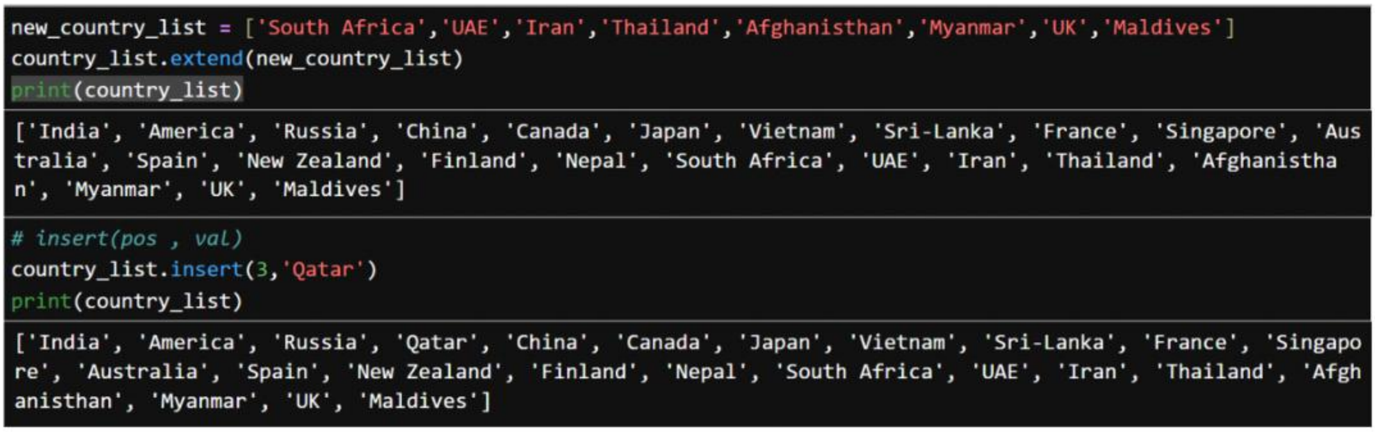


Adding an items in a List:

1. .append() := 'Add one item at the end'
2. .extend() := 'Add all items from another lists'
3. .insert() := 'Insert an item at specific index'

```
country_list =['India', 'America', 'Russia', 'China', 'Canada', 'Japan',
               'Vietnam', 'Sri-Lanka', 'France', 'Singapore', 'Australia',
               'Spain', 'New Zealand', 'Finland']
country_list.append('Nepal')
print(country_list)
```

```
['India', 'America', 'Russia', 'China', 'Canada', 'Japan', 'Vietnam', 'Sri-Lanka', 'France', 'Singapore', 'Aus
tralia', 'Spain', 'New Zealand', 'Finland', 'Nepal']
```

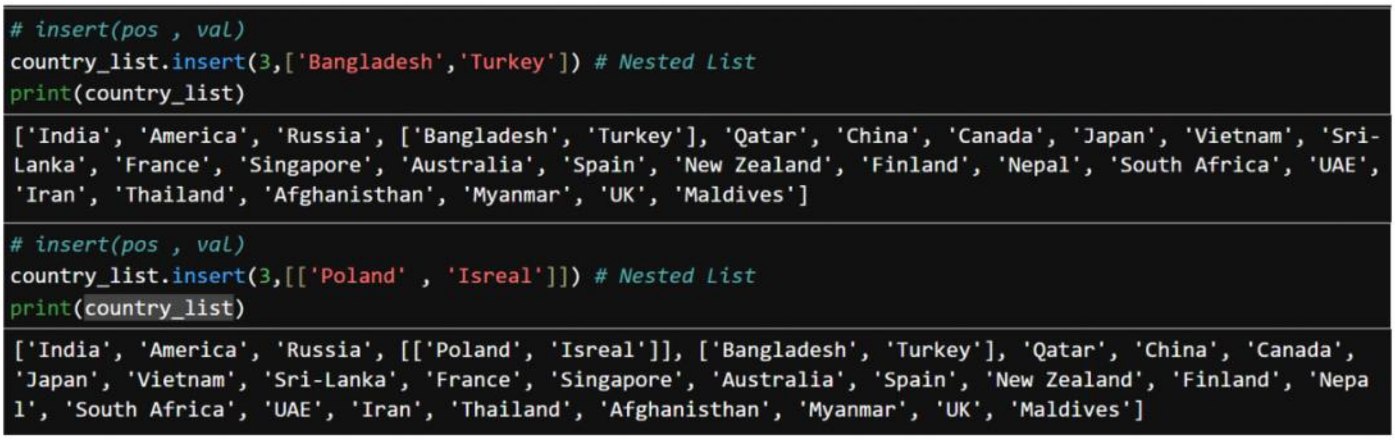


```
new_country_list = ['South Africa','UAE','Iran','Thailand','Afghanisthan','Myanmar','UK','Maldives']
country_list.extend(new_country_list)
print(country_list)
```

```
['India', 'America', 'Russia', 'China', 'Canada', 'Japan', 'Vietnam', 'Sri-Lanka', 'France', 'Singapore', 'Aus
tralia', 'Spain', 'New Zealand', 'Finland', 'Nepal', 'South Africa', 'UAE', 'Iran', 'Thailand', 'Afghanistha
n', 'Myanmar', 'UK', 'Maldives']
```

```
# insert(pos , val)
country_list.insert(3,'Qatar')
print(country_list)
```

```
['India', 'America', 'Russia', 'Qatar', 'China', 'Canada', 'Japan', 'Vietnam', 'Sri-Lanka', 'France', 'Singapo
re', 'Australia', 'Spain', 'New Zealand', 'Finland', 'Nepal', 'South Africa', 'UAE', 'Iran', 'Thailand', 'Afgh
anisthan', 'Myanmar', 'UK', 'Maldives']
```

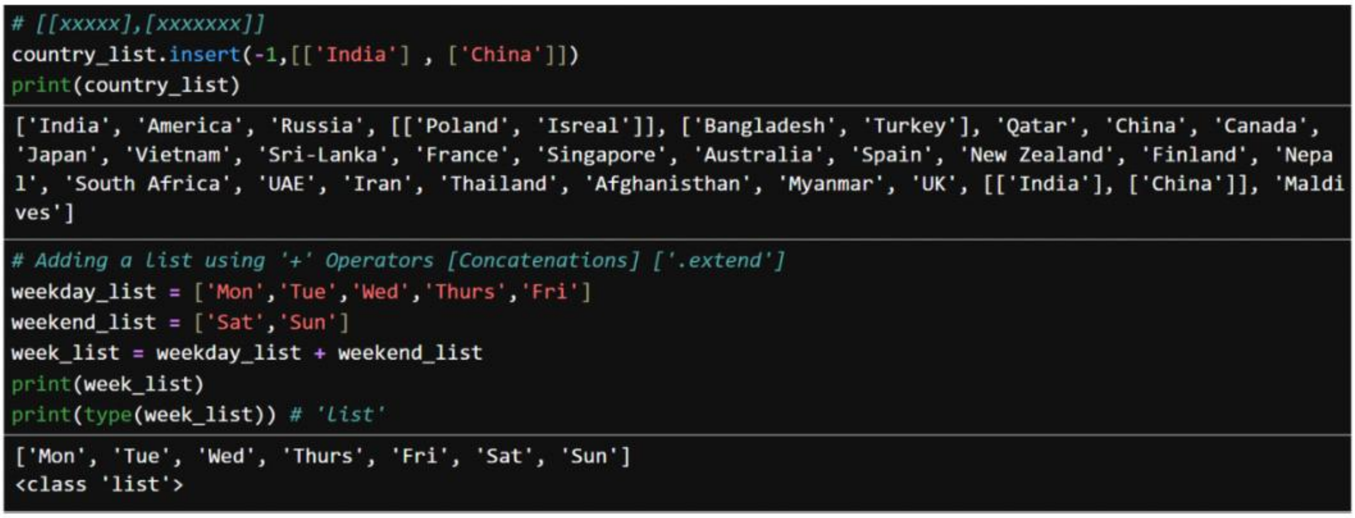


```
# insert(pos , val)
country_list.insert(3,['Bangladesh','Turkey']) # Nested List
print(country_list)
```

```
['India', 'America', 'Russia', ['Bangladesh', 'Turkey'], 'Qatar', 'China', 'Canada', 'Japan', 'Vietnam', 'Sri-
Lanka', 'France', 'Singapore', 'Australia', 'Spain', 'New Zealand', 'Finland', 'Nepal', 'South Africa', 'UAE',
'Iran', 'Thailand', 'Afghanisthan', 'Myanmar', 'UK', 'Maldives']
```

```
# insert(pos , val)
country_list.insert(3,[['Poland' , 'Isreal']]) # Nested List
print(country_list)
```

```
['India', 'America', 'Russia', [['Poland', 'Isreal']], ['Bangladesh', 'Turkey'], 'Qatar', 'China', 'Canada',
'Japan', 'Vietnam', 'Sri-Lanka', 'France', 'Singapore', 'Australia', 'Spain', 'New Zealand', 'Finland', 'Nepa
l', 'South Africa', 'UAE', 'Iran', 'Thailand', 'Afghanisthan', 'Myanmar', 'UK', 'Maldives']
```

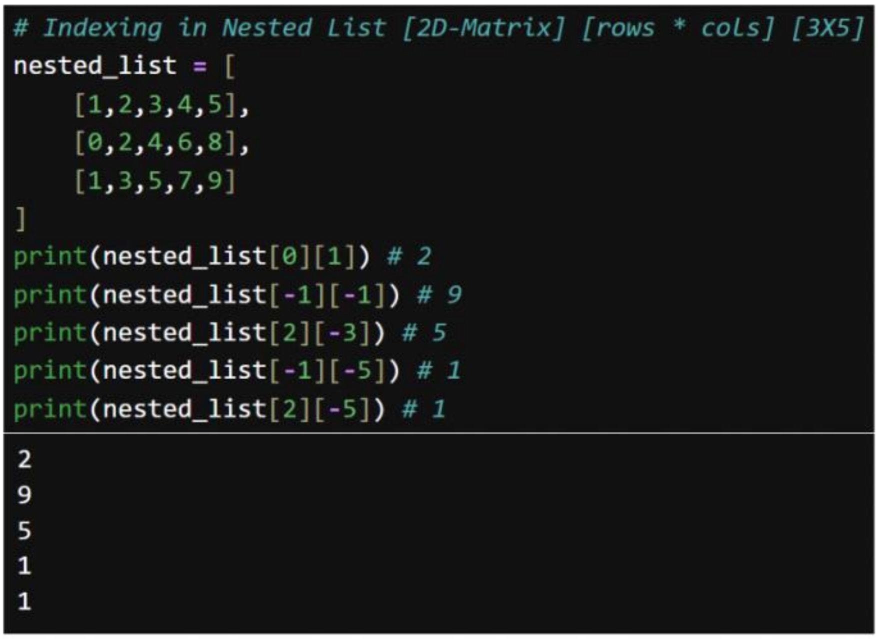


```
# [[xxxxx],[xxxxxxx]]
country_list.insert(-1,[['India'] , ['China']])
print(country_list)
```

```
['India', 'America', 'Russia', [['Poland', 'Isreal']], ['Bangladesh', 'Turkey'], 'Qatar', 'China', 'Canada',
'Japan', 'Vietnam', 'Sri-Lanka', 'France', 'Singapore', 'Australia', 'Spain', 'New Zealand', 'Finland', 'Nepa
l', 'South Africa', 'UAE', 'Iran', 'Thailand', 'Afghanisthan', 'Myanmar', 'UK', [['India'], ['China']], 'Maldi
ves']
```

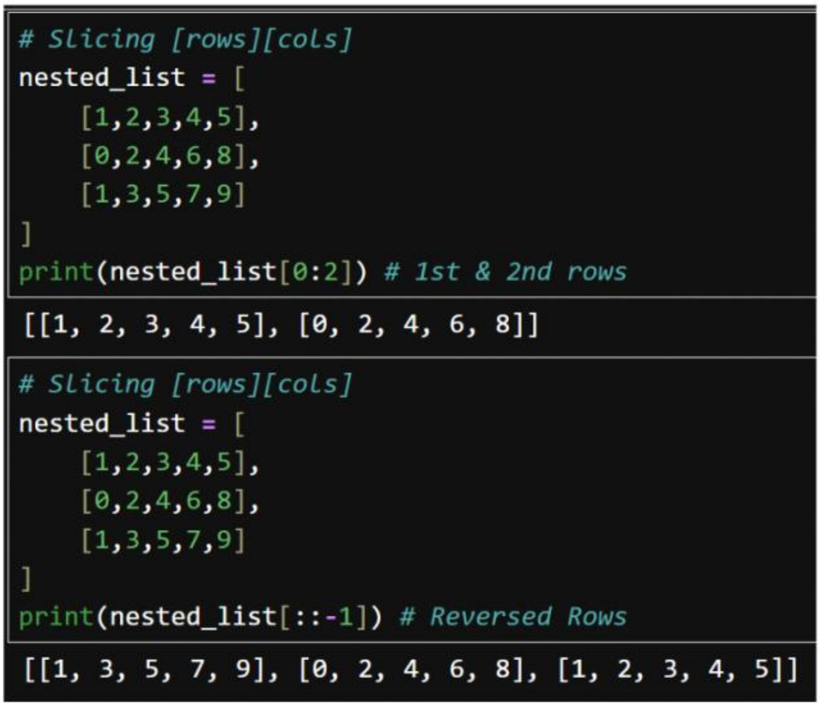
```
# Adding a list using '+' Operators [Concatenations] ['.extend']
weekday_list = ['Mon','Tue','Wed','Thurs','Fri']
weekend_list = ['Sat','Sun']
week_list = weekday_list + weekend_list
print(week_list)
print(type(week_list)) # 'list'
```

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



```
# Indexing in Nested List [2D-Matrix] [rows * cols] [3X5]
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][-1]) # 9
print(nested_list[2][-3]) # 5
print(nested_list[-1][-5]) # 1
print(nested_list[2][-5]) # 1
```

```
2
9
5
1
1
```

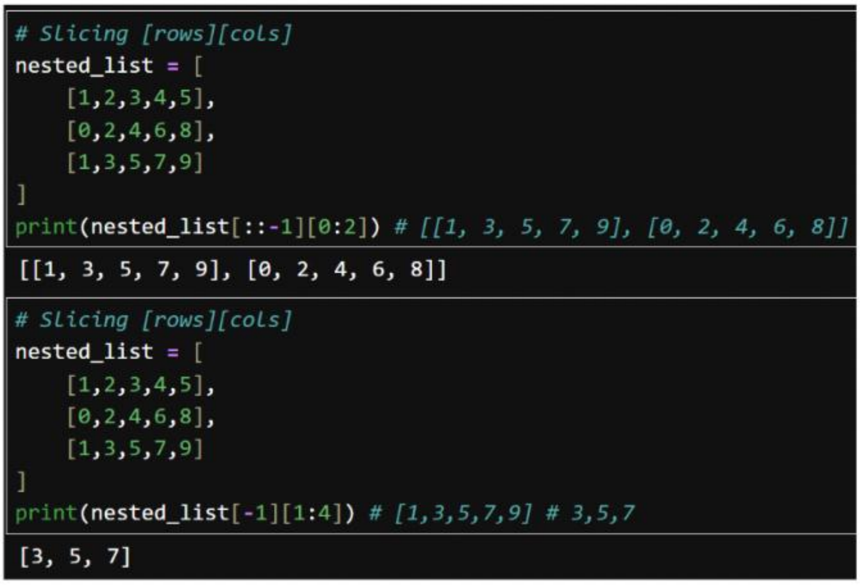


```
# Slicing [rows][cols]
nested_list = [
    [1,2,3,4,5],
    [0,2,4,6,8],
    [1,3,5,7,9]
]
print(nested_list[0:2]) # 1st & 2nd rows
```

```
[[1, 2, 3, 4, 5], [0, 2, 4, 6, 8]]
```

```
# Slicing [rows][cols]
nested_list = [
    [1,2,3,4,5],
    [0,2,4,6,8],
    [1,3,5,7,9]
]
print(nested_list[::-1]) # Reversed Rows
```

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



```
# Slicing [rows][cols]
nested_list = [
    [1,2,3,4,5],
    [0,2,4,6,8],
    [1,3,5,7,9]
]
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]]
```

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

```
[3, 5, 7]
```