CN-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

2

'India','America','Russia','China', -9
'Canada','Japan','Vietnam','Sri-Lanka',
'Japan','France','India','Singapore', -5
'Australia','China','New Zealand','Finland'

12

```
start = 0 [default]
stop = len[default]
step = 1 [default]
```

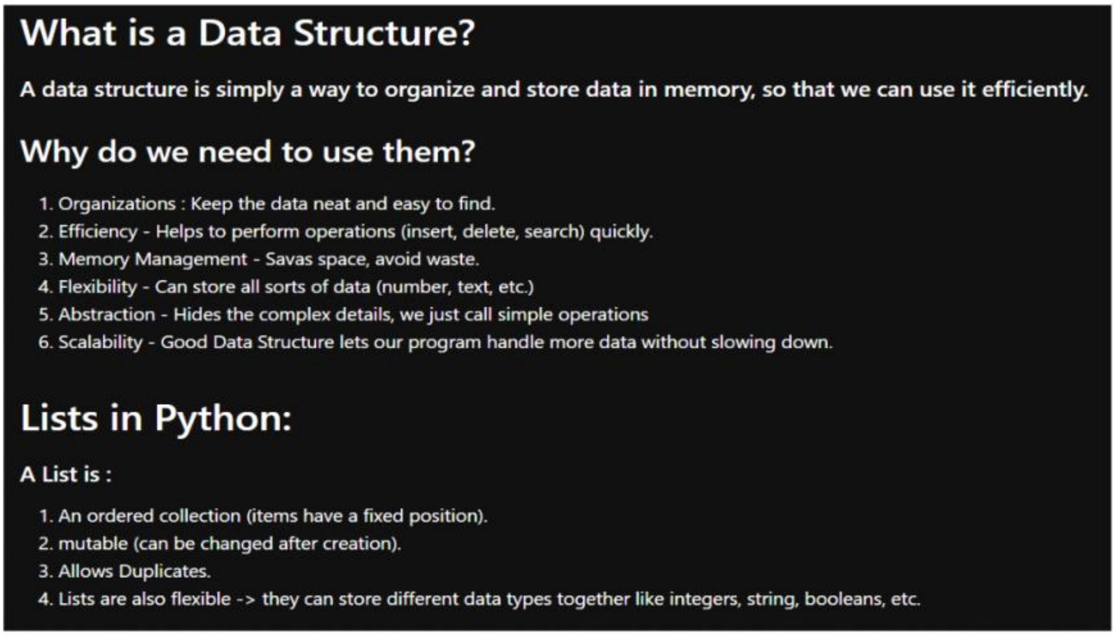
-9:12:1 ['Sri-Lanka','Japan','France','India','Singapore'

-9:12:-1 []

right to left

-9:2:-1 ['Sri Lanka' to 'China'] right to left

-9: -5: 2 ['Sri Lanka', 'France']



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.

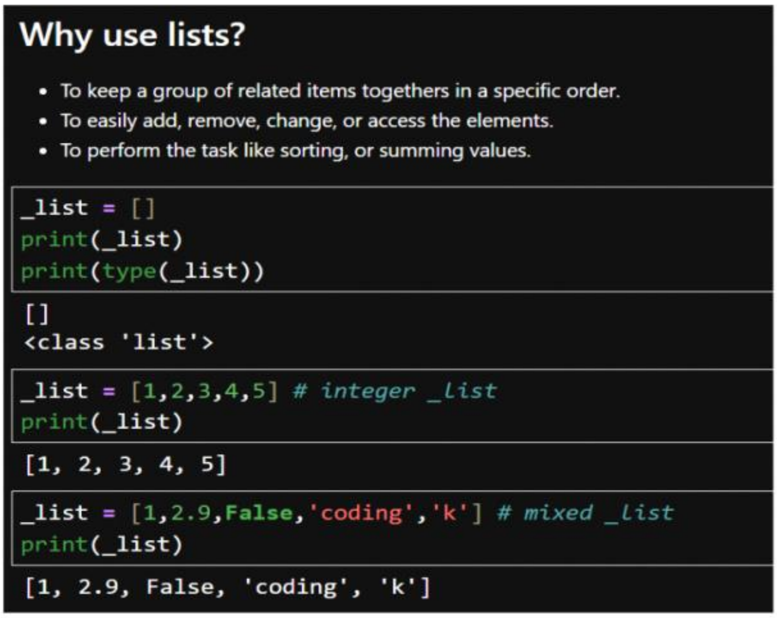
Why do we need to use them?

1. Organizations : Keep the data neat and easy to find.
2. Efficiency - Helps to perform operations (insert, delete, search) quickly.
3. Memory Management - Savas 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.

```python
_list = []
print(_list)
print(type(_list))
```

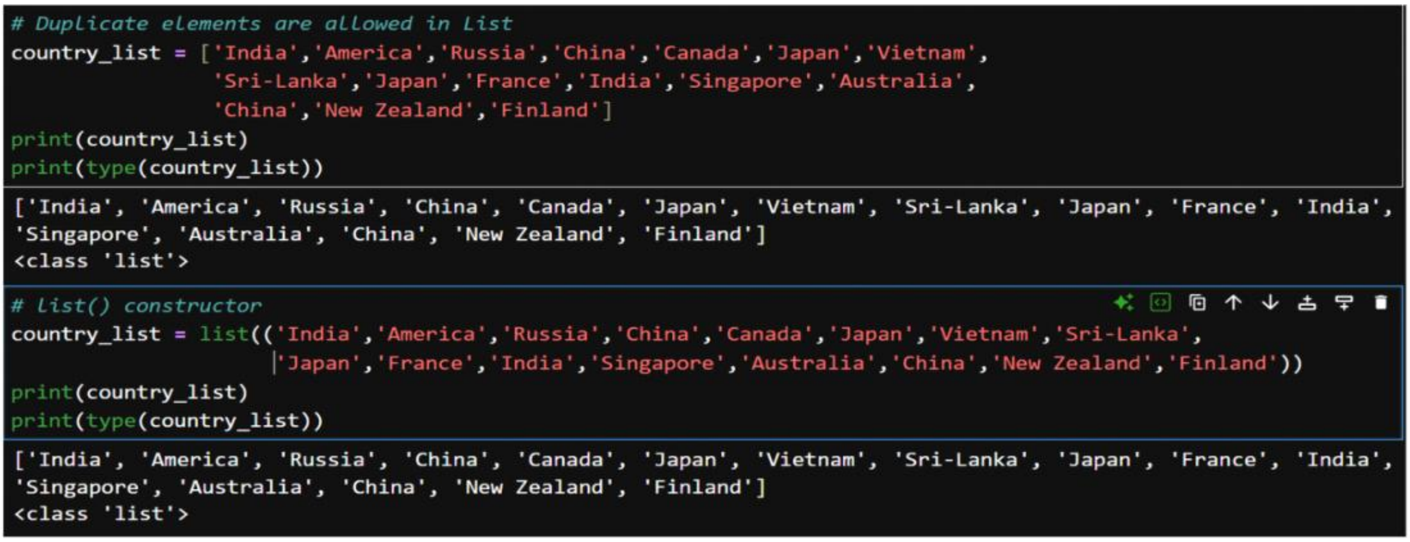
```
[]
<class 'list'>
```

```python
_list = [1,2,3,4,5] # integer _list
print(_list)
```

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

```python
_list = [1,2.9,False,'coding','k'] # mixed _list
print(_list)
```

```
[1, 2.9, False, 'coding', 'k']
```

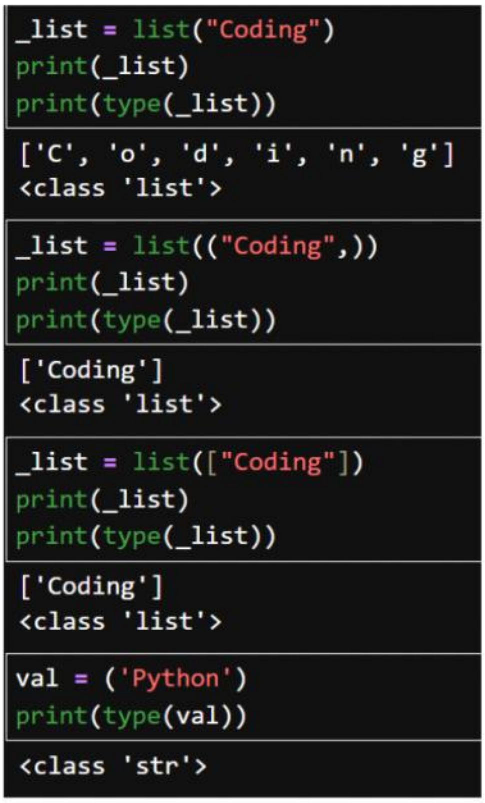


```python
# Duplicate elements are allowed in List
country_list = ['India','America','Russia','China','Canada','Japan','Vietnam',
                'Sri-Lanka','Japan','France','India','Singapore','Australia',
                'China','New Zealand','Finland']
print(country_list)
print(type(country_list))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

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

```python
# 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")
print(_list)
print(type(_list))
```

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

```
_list = list(("Coding",))
print(_list)
print(type(_list))
```

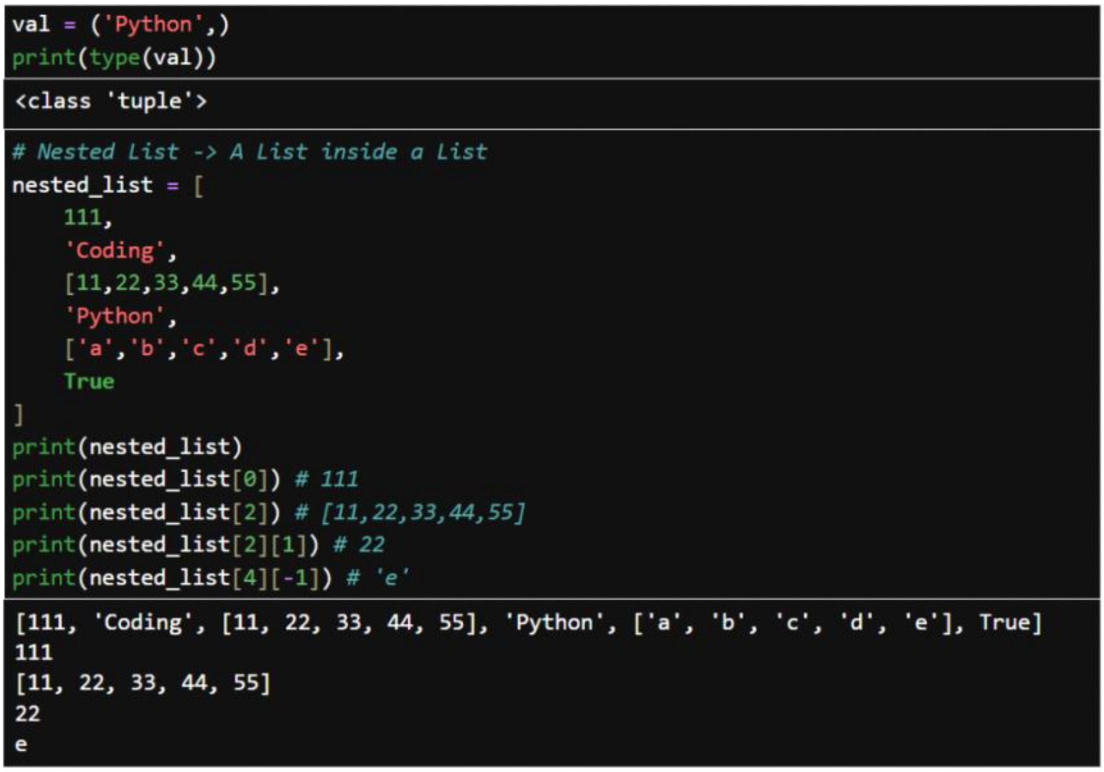
```
['Coding']
<class 'list'>
```

```
_list = list(["Coding"])
print(_list)
print(type(_list))
```

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

```
val = ('Python')
print(type(val))
```

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

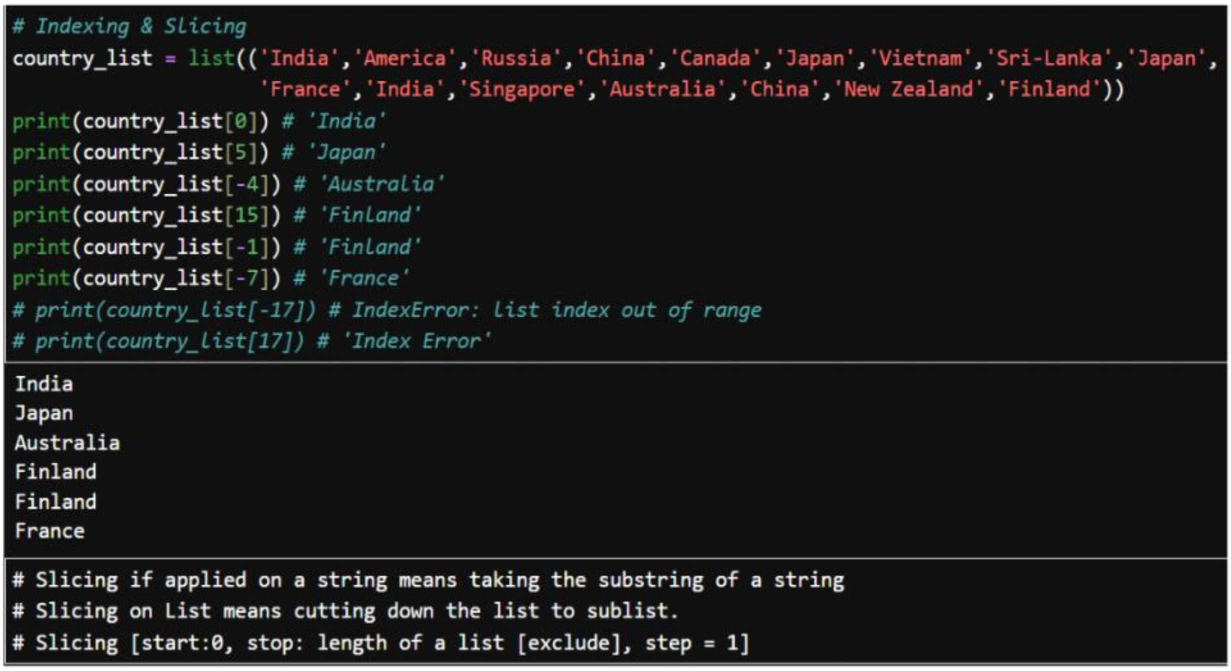


```
val = ('Python',)
print(type(val))
```

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

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

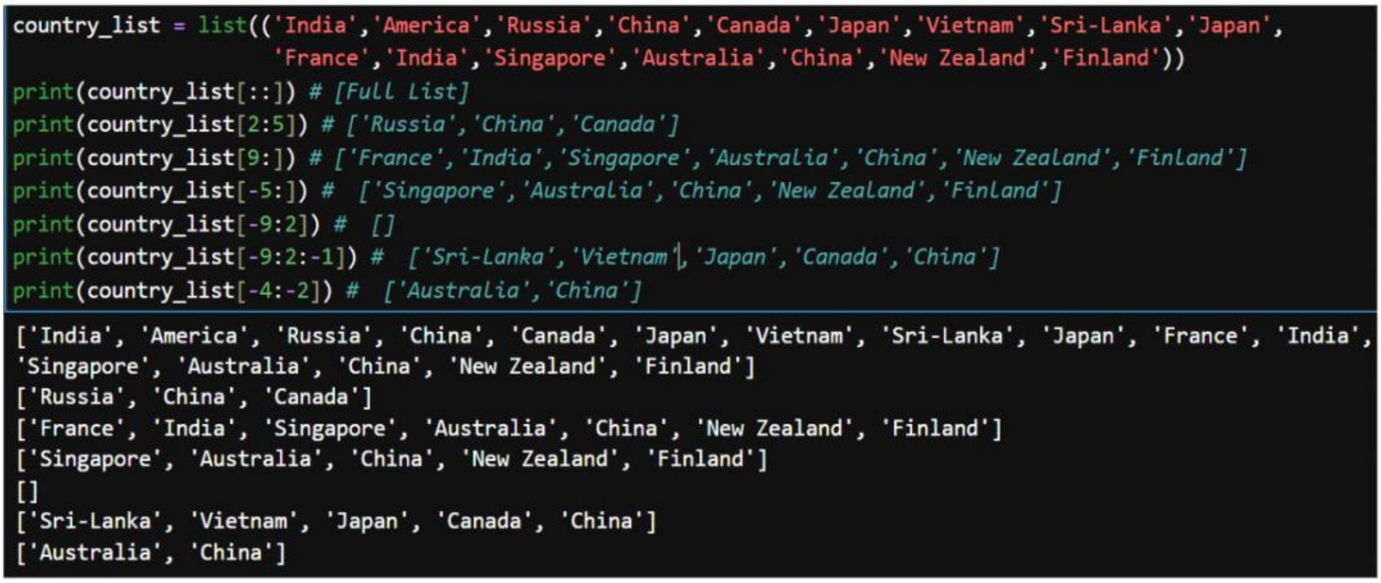
```
[111, 'Coding', [11, 22, 33, 44, 55], 'Python', ['a', 'b', 'c', 'd', 'e'], True]
111
[11, 22, 33, 44, 55]
22
e
```



```
# Indexing & Slicing
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]) # 'Japan'
print(country_list[-4]) # 'Australia'
print(country_list[15]) # 'Finland'
print(country_list[-1]) # 'Finland'
print(country_list[-7]) # 'France'
# print(country_list[-17]) # IndexError: list index out of range
# print(country_list[17]) # 'Index Error'
```

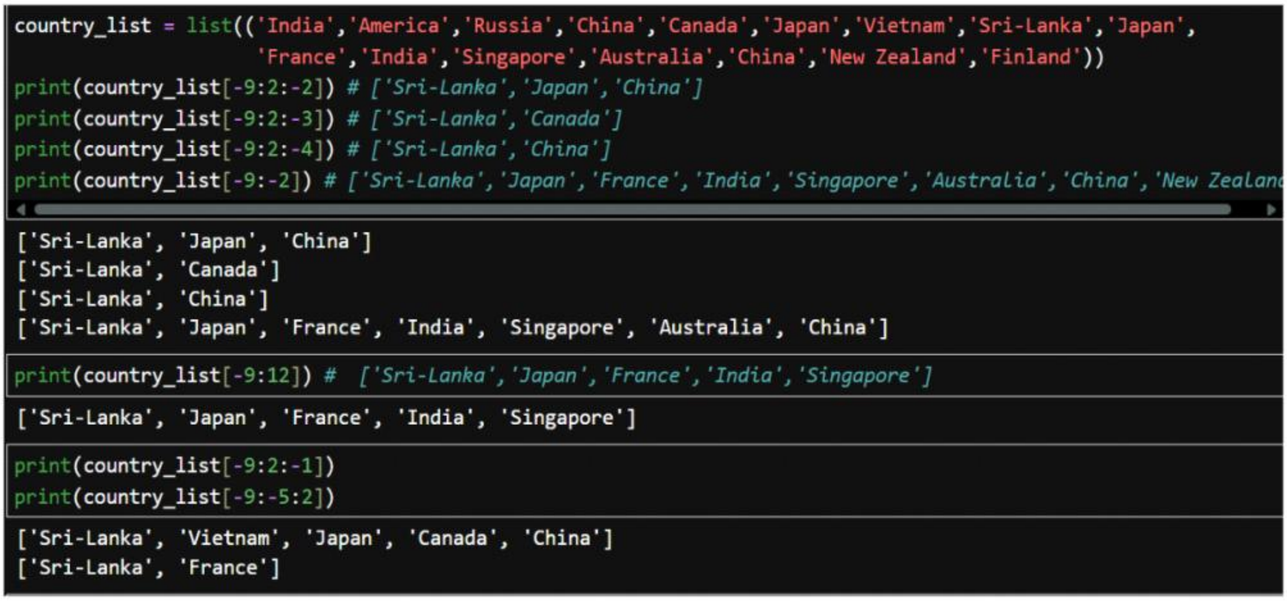
```
India
Japan
Australia
Finland
Finland
France
```

```
# Slicing if applied on a string means taking the substring of a string
# Slicing on List means cutting down the list to sublist.
# Slicing [start:0, stop: length of a list [exclude], 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]
print(country_list[2:5]) # ['Russia','China','Canada']
print(country_list[9:]) # ['France','India','Singapore','Australia','China','New Zealand','Finland']
print(country_list[-5:]) #  ['Singapore','Australia','China','New Zealand','Finland']
print(country_list[-9:2]) #  []
print(country_list[-9:2:-1]) #  ['Sri-Lanka','Vietnam','Japan','Canada','China']
print(country_list[-4:-2]) #  ['Australia','China']
```

```
['India', 'America', 'Russia', 'China', 'Canada', 'Japan', 'Vietnam', 'Sri-Lanka', 'Japan', 'France', 'India',
'Singapore', 'Australia', 'China', 'New Zealand', 'Finland']
['Russia', 'China', 'Canada']
['France', 'India', 'Singapore', 'Australia', 'China', 'New Zealand', 'Finland']
['Singapore', 'Australia', 'China', 'New Zealand', 'Finland']
[]
['Sri-Lanka', 'Vietnam', 'Japan', 'Canada', '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']
print(country_list[-9:2:-3]) # ['Sri-Lanka','Canada']
print(country_list[-9:2:-4]) # ['Sri-Lanka','China']
print(country_list[-9:-2]) # ['Sri-Lanka','Japan','France','India','Singapore','Australia','China','New Zealan
```

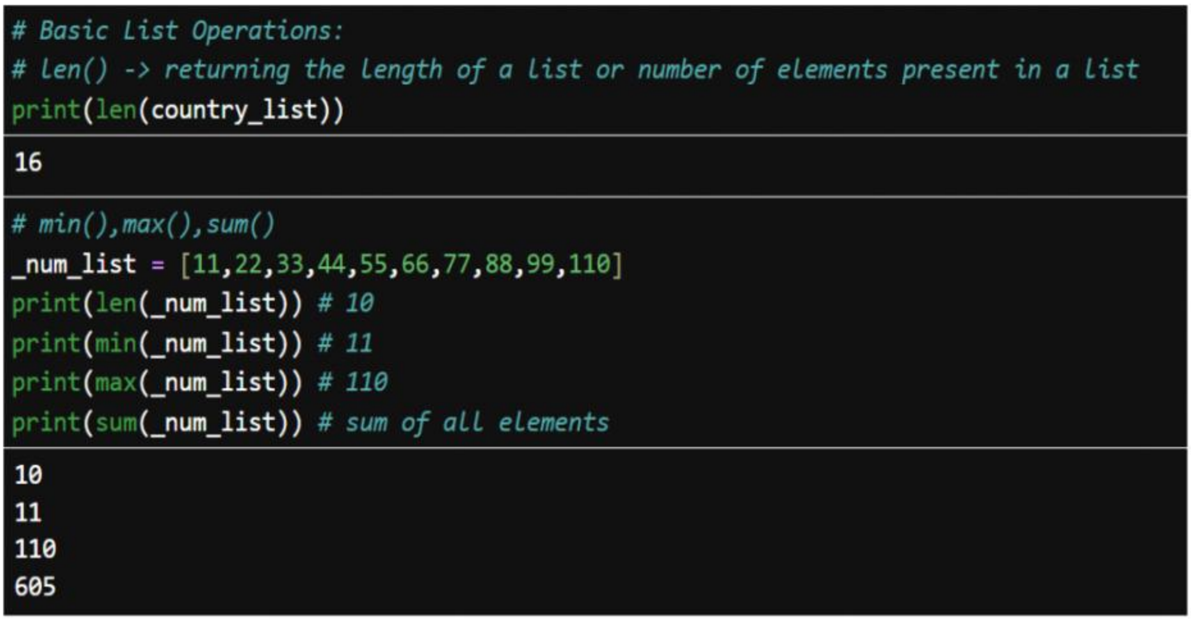
```
['Sri-Lanka', 'Japan', 'China']
['Sri-Lanka', 'Canada']
['Sri-Lanka', 'China']
['Sri-Lanka', 'Japan', 'France', 'India', 'Singapore', 'Australia', 'China']
```

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

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

```
print(country_list[-9:2:-1])
print(country_list[-9:-5:2])
```

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

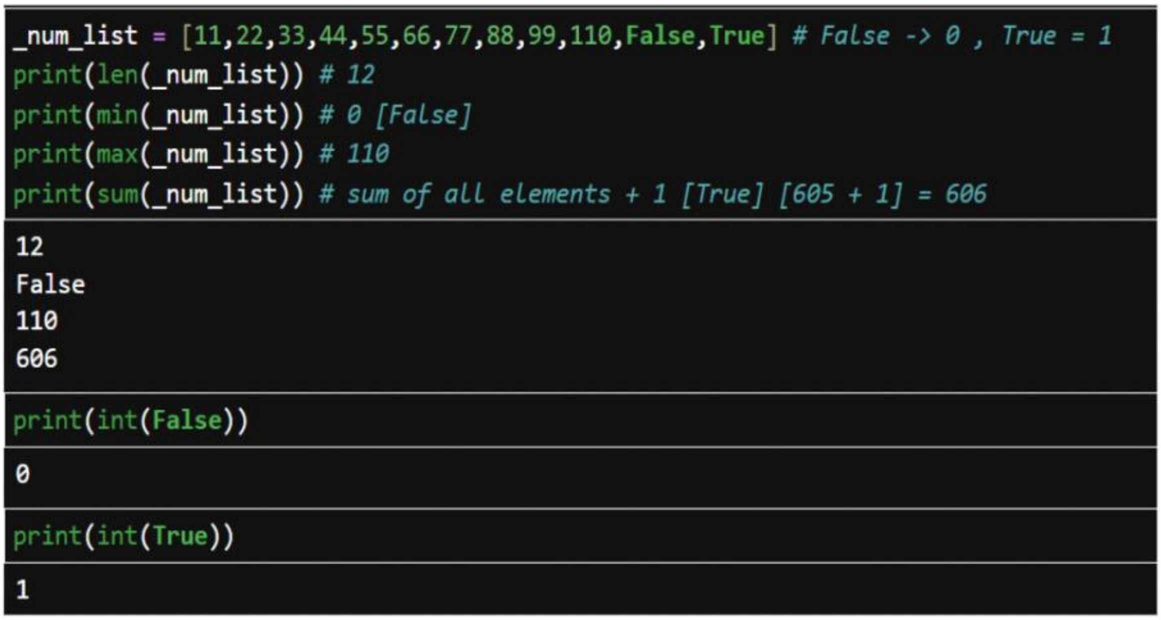


```
# Basic List Operations:
# len() -> returning the length of a list or number of elements present in a list
print(len(country_list))
```

```
16
```

```
# min(),max(),sum()
_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)) # sum of all elements
```

```
10
11
110
605
```



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

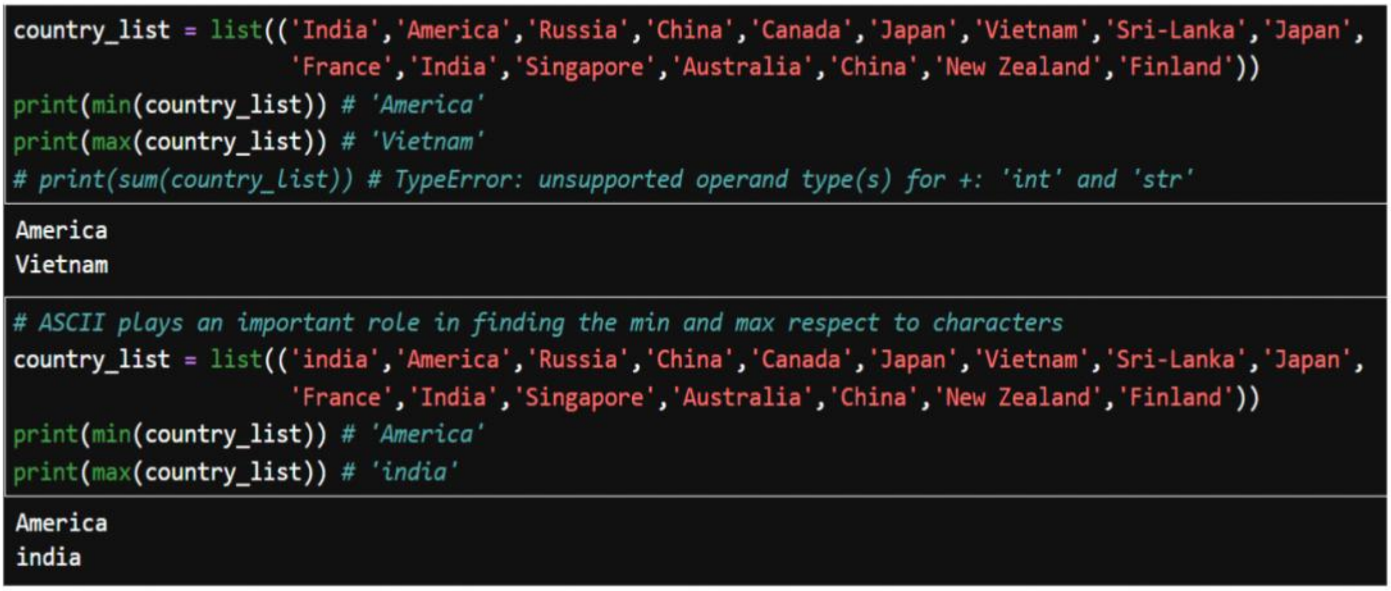
```
12
False
110
606
```

```
print(int(False))
```

```
0
```

```
print(int(True))
```

```
1
```

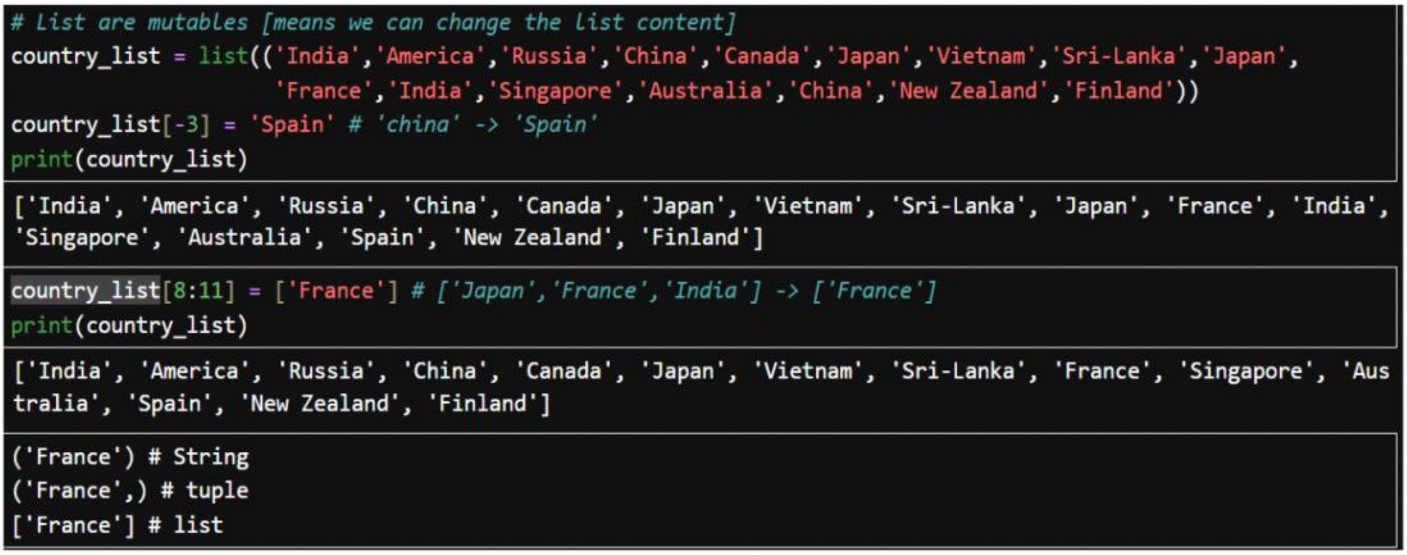


```
country_list = list(('India','America','Russia','China','Canada','Japan','Vietnam','Sri-Lanka','Japan',
                     'France','India','Singapore','Australia','China','New Zealand','Finland'))
print(min(country_list)) # 'America'
print(max(country_list)) # 'Vietnam'
# print(sum(country_list)) # TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

```
America
Vietnam
```

```
# ASCII plays an important role in finding the min and max respect to characters
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)) # 'America'
print(max(country_list)) # 'india'
```

```
America
india
```



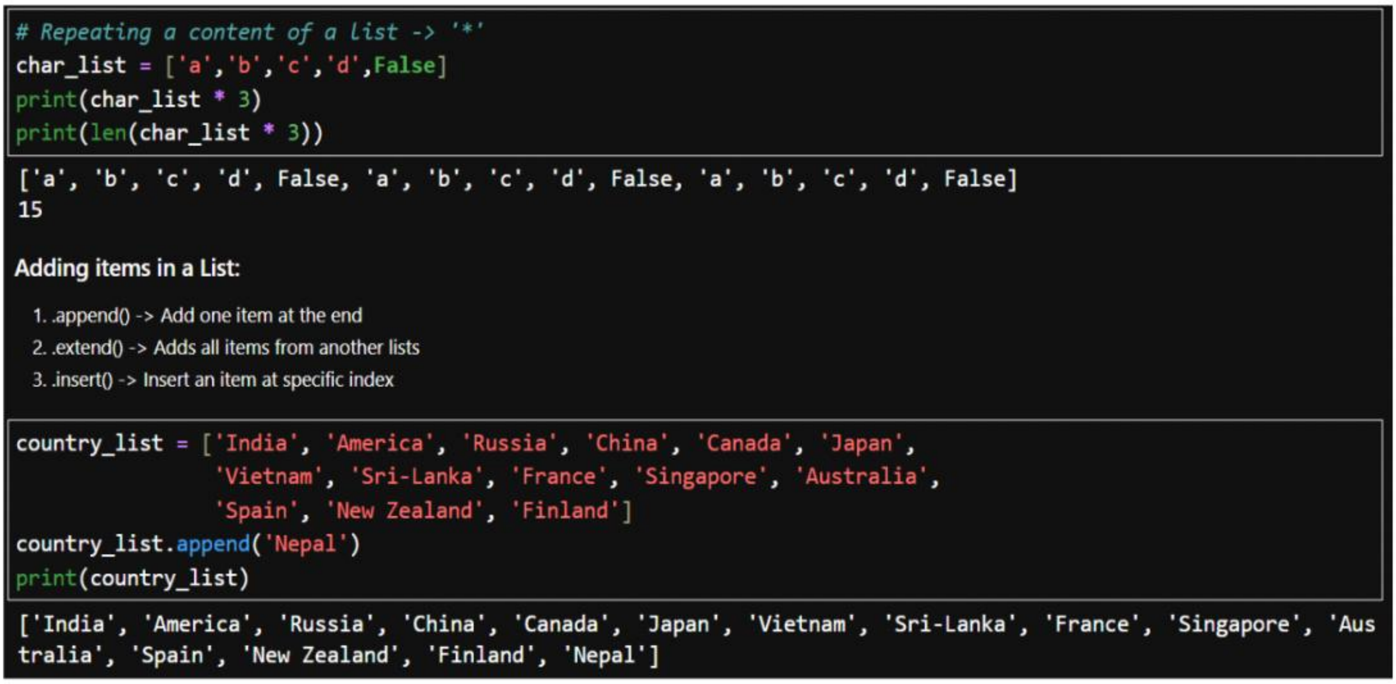
```
# List are mutables [means we can change the list content]
country_list = list(('India','America','Russia','China','Canada','Japan','Vietnam','Sri-Lanka','Japan',
                     'France','India','Singapore','Australia','China','New Zealand','Finland'))
country_list[-3] = 'Spain' # 'china' ->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



```
# Repeating a content of a list -> '*'
char_list = ['a','b','c','d',False]
print(char_list * 3)
print(len(char_list * 3))
```

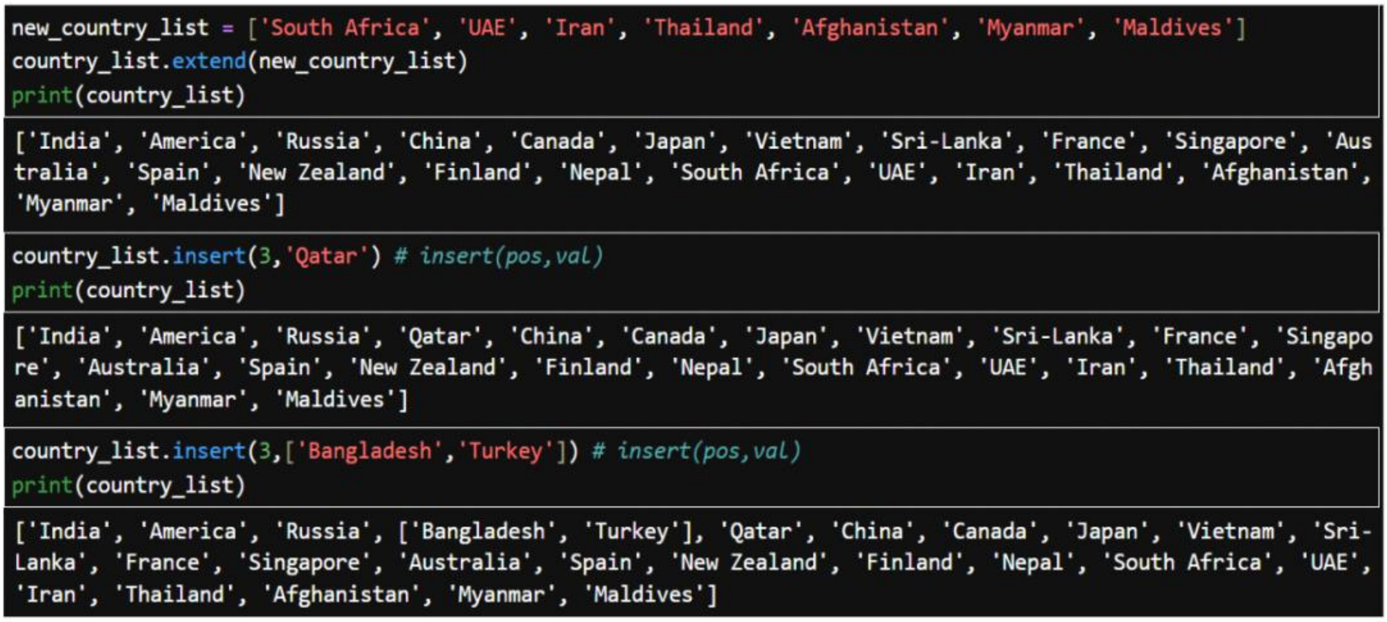
```
['a', 'b', 'c', 'd', False, 'a', 'b', 'c', 'd', False, 'a', 'b', 'c', 'd', False]
15
```

Adding items in a List:

1. .append() -> Add one item at the end
2. .extend() -> Adds 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', 'Afghanistan', 'Myanmar', 'Maldives']
country_list.extend(new_country_list)
print(country_list)
```

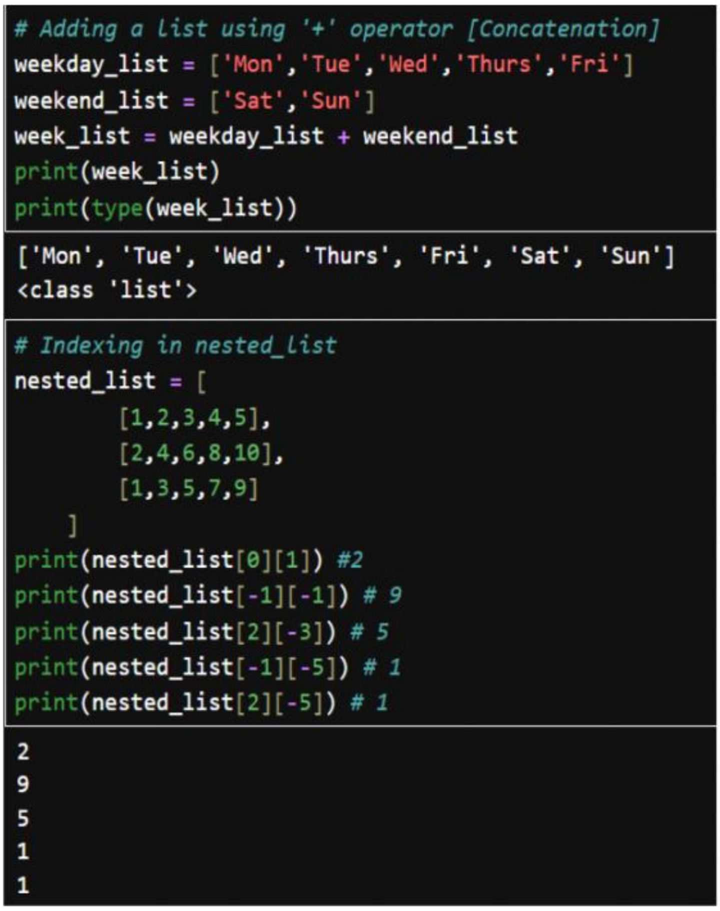
```
['India', 'America', 'Russia', 'China', 'Canada', 'Japan', 'Vietnam', 'Sri-Lanka', 'France', 'Singapore', 'Aus
tralia', 'Spain', 'New Zealand', 'Finland', 'Nepal', 'South Africa', 'UAE', 'Iran', 'Thailand', 'Afghanistan',
'Myanmar', 'Maldives']
```

```
country_list.insert(3,'Qatar') # insert(pos,val)
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
anistan', 'Myanmar', 'Maldives']
```

```
country_list.insert(3,['Bangladesh','Turkey']) # insert(pos,val)
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', 'Afghanistan', 'Myanmar', 'Maldives']
```

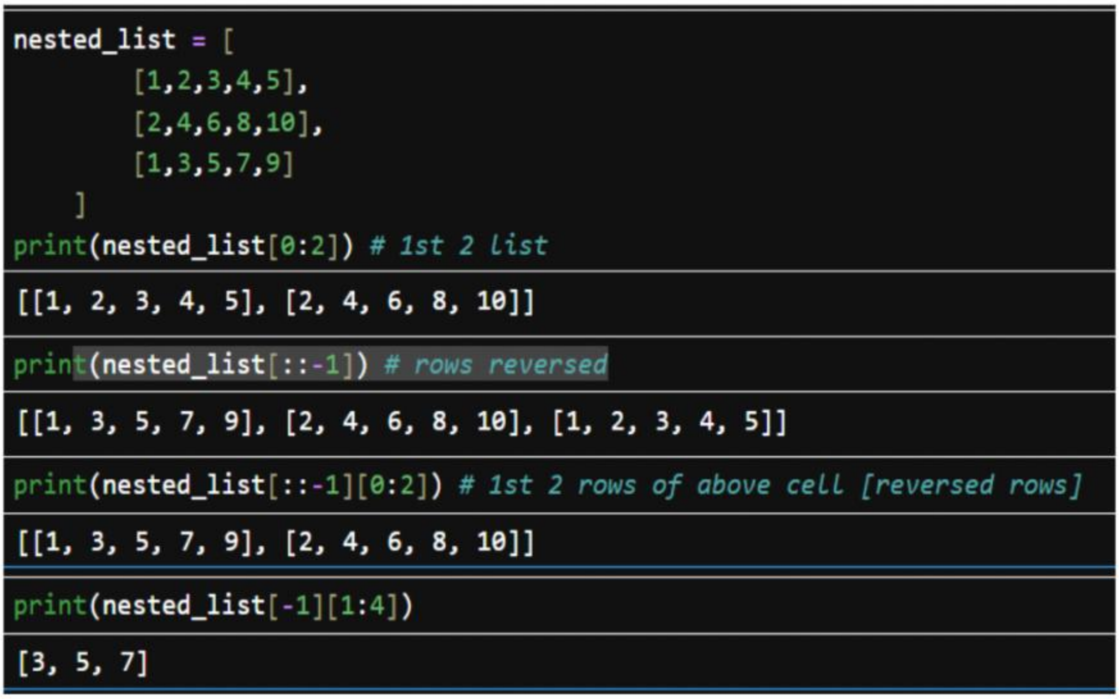


```
# Adding a list using '+' operator [Concatenation]
weekday_list = ['Mon','Tue','Wed','Thurs','Fri']
weekend_list = ['Sat','Sun']
week_list = weekday_list + weekend_list
print(week_list)
print(type(week_list))
```

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

```
# Indexing in nested_list
nested_list = [
        [1,2,3,4,5],
        [2,4,6,8,10],
        [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
```



```
nested_list = [
        [1,2,3,4,5],
        [2,4,6,8,10],
        [1,3,5,7,9]
    ]
print(nested_list[0:2]) # 1st 2 list
```

```
[[1, 2, 3, 4, 5], [2, 4, 6, 8, 10]]
```

```
print(nested_list[::-1]) # rows reversed
```

```
[[1, 3, 5, 7, 9], [2, 4, 6, 8, 10], [1, 2, 3, 4, 5]]
```

```
print(nested_list[::-1][0:2]) # 1st 2 rows of above cell [reversed rows]
```

```
[[1, 3, 5, 7, 9], [2, 4, 6, 8, 10]]
```

```
print(nested_list[-1][1:4])
```

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