

## 1. List

- Sequential data type.
- List is mutable.

### 1.1 Creating a List

```
In [6]: courses = ['Python', 'C++', 'Scala', 'Java']
print(courses)

['Python', 'C++', 'Scala', 'Java']
```

### 1.2 Operators provided by List

- len(list) : Return the number of elements stored in list(length)
- list[idx] : Return the element stored in index of idx.
- list[-1] : Return the last element in list.

```
In [7]: print(len(courses))
print(courses[0])
print(courses[-1])

4
Python
Java
```

### 1.3 Methods provided by List

- list.append(item) : Push the item at the end of the list.
- list.insert(idx, item) : Push the item at the index of idx of the list.
- list.extend(sequence) : Push multiple items at the end of the list.
- list.remove(item) : Remove the item from the list.
- list.pop() : Remove the last value of the list and return its value. (Used for stack and queue.)
- list.reverse() : Sort the list in reversed order.
- list.sort() : Sort the element in list in ascending order.
- list.index(item) : Return the first index of item found in the list.
- item in list : Return True if item is in the list.

```
In [8]: courses.append('Go')
print(courses)
popped = courses.pop()
print(popped)

['Python', 'C++', 'Scala', 'Java', 'Go']
Go
```

```
In [9]: # Sorting without altering original list.
# courses.sort()
sorted_courses = sorted(courses)
print(sorted_courses)

['C++', 'Java', 'Python', 'Scala']
```

### 1.4 Built-in functions for List

- min(list) : Return the min value from the list.
- sum(list) : Return the sum of values from the list.

## 2. Tuples

- Tuples is immutable data type.
- Just for accessing and iterating.

```
In [11]: tuple1 = ('Python', 'C++', 'C#', 'Java', 'Scala')
tuple2 = tuple1

tuple1[0] = 'Art'
print(tuple1)
print(tuple2)

-----
TypeError                                Traceback (most recent call last)
<ipython-input-11-6b00e4a5318e> in <module>
      2 tuple2 = tuple1
      3
----> 4 tuple1[0] = 'Art'
      5 print(tuple1)
      6 print(tuple2)

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

## 3. Sets

- Data type without duplicates.
- Unordered

```
In [13]: courses = {'Python', 'C++', 'C#', 'Java', 'Scala'}

for _ in range(4):
    print(courses)

{'Scala', 'Python', 'C#', 'C++', 'Java'}
{'Scala', 'Python', 'C#', 'C++', 'Java'}
{'Scala', 'Python', 'C#', 'C++', 'Java'}
{'Scala', 'Python', 'C#', 'C++', 'Java'}
```

## 4. Creating a Empty Structures

```
In [18]: # Creating empty list
empty_list = []
empty_list = list()

# Creating empty tuple
empty_tuple = ()
empty_tuple = tuple()

# Creating empty sets
empty_set = set()

# Creating empty dictionary
empty_dictionary = {}
empty_dictionary = dict()
```