Basic Types of Data Structure

1. Tuple ()

2. List []

3. Dictionaries {}

4. Set {}

---

1. Tuples

- Ordered collection
- Enclosed in ()
- Store different types of elements
- Elements stored once can't be changed (immutable)

In [1]:
```
t1= (1, 2.4, 'python', True)
t1
```

Out[1]:
```
(1, 2.4, 'python', True)
```

In [2]:
```
type(t1)
```

Out[2]:
```
tuple
```

Indexing

In [3]:
```
t1[1]
```

Out[3]:
```
2.4
```

In [4]:
```
t1[3]
```

Out[4]:
```
True
```

In [5]:
```
# Last element is exclusive(not included)

t1[0:3]
```

Out[5]:
```
(1, 2.4, 'python')
```

Using different functions

In [6]:
```
# Find length of tuple
```

```
len(t1)
```

Out[6]: 4

In [7]:
```
t2 = (2, 8.9 ,' Baba Ammar', False)
t2
```

Out[7]: (2, 8.9, ' Baba Ammar', False)

In [8]:
```
# Adding 2 tuples

t1+t2
```

Out[8]: (1, 2.4, 'python', True, 2, 8.9, ' Baba Ammar', False)

In [9]:
```
# Multiplovction means repeatition of concatenation

t1*2
```

Out[9]: (1, 2.4, 'python', True, 1, 2.4, 'python', True)

In [10]:
```
# Multuply and addition

t1*2+t2
```

Out[10]: (1, 2.4, 'python', True, 1, 2.4, 'python', True, 2, 8.9, ' Baba Ammar', False)

In [11]:
```
t3= (20,30,40,50,60,70,80)
t3
```

Out[11]: (20, 30, 40, 50, 60, 70, 80)

In [12]:
```
# minimum and maximum

print (min(t3))
print (max(t3))
```

```
20
80
```

In [13]:
```
# multiply means repeatition
t3*2
```

Out[13]: (20, 30, 40, 50, 60, 70, 80, 20, 30, 40, 50, 60, 70, 80)

---

2. List

- Ordered collection
- Enclosed in []
- Store different types of elements
- Elements stored can be changed (mutable)

```
```

In [14]:
```
l1= [1, 2.4, 'python', True]
l1
```

Out[14]:
```
[1, 2.4, 'python', True]
```

In [15]:
```
type(l1)
```

Out[15]:
```
list
```

In [16]:
```
len(l1)
```

Out[16]:
```
4
```

In [17]:
```
l1[2]
```

Out[17]:
```
'python'
```

In [18]:
```
l2=[3,5, 'Ammar', 'Codanics', 478, 55.2, False]
l2
```

Out[18]:
```
[3, 5, 'Ammar', 'Codanics', 478, 55.2, False]
```

In [19]:
```
l1+l2
```

Out[19]:
```
[1, 2.4, 'python', True, 3, 5, 'Ammar', 'Codanics', 478, 55.2, False]
```

In [20]:
```
l1*2
```

Out[20]:
```
[1, 2.4, 'python', True, 1, 2.4, 'python', True]
```

Applying various funtions by -> listname. 'press-Tab'

In [21]:
```
l1.reverse()
l1
```

Out[21]:
```
[True, 'python', 2.4, 1]
```

In [22]:
```
l1.append('Codanics Youtube Channel')
l1
```

Out[22]:
```
[True, 'python', 2.4, 1, 'Codanics Youtube Channel']
```

In [23]:
```
l3=[20,45,67,34,23,2,68,678,34,78,11,45]

len(l3)
```

Out[23]:
```
12
```

In [24]:
```
l3.sort()

l3
```

Out[24]:
```
[2, 11, 20, 23, 34, 34, 45, 45, 67, 68, 78, 678]
```

In [25]:
```
l3*2
```

Out[25]:
```
[2,
 11,
 20,
 23,
 34,
 34,
 45,
 45,
 67,
 68,
 78,
 678,
 2,
 11,
 20,
 23,
 34,
 34,
 45,
 45,
 67,
 68,
 78,
 678]
```

In [26]:
```
l1+l3
```

Out[26]:
```
[True,
 'python',
 2.4,
 1,
 'Codanics Youtube Channel',
 2,
 11,
 20,
 23,
 34,
 34,
 45,
 45,
 67,
 68,
 78,
 678]
```

---

3. Dictionary

- Unordered collection
- Enclosed in {}
- It has KEY and VALUE
- Elements stored can be changed (mutable)

In [27]:
```
# Food and Prices

d1= {'Samosa':30 , 'Pakora': 100, 'Raita':20, 'Salad':50, 'Chicken Roll': 30}
```

```
d1
```

Out[27]:
```
{'Samosa': 30, 'Pakora': 100, 'Raita': 20, 'Salad': 50, 'Chicken Roll': 30}
```

In [28]:
```
type(d1)
```

Out[28]:
```
dict
```

In [29]:
```
# Extract Data

keys1 = d1.keys()
keys1
```

Out[29]:
```
dict_keys(['Samosa', 'Pakora', 'Raita', 'Salad', 'Chicken Roll'])
```

In [30]:
```
values1= d1.values()
values1
```

Out[30]:
```
dict_values([30, 100, 20, 50, 30])
```

In [31]:
```
# Updating / adding new value

d1['Tikki']=10

d1
```

Out[31]:
```
{'Samosa': 30,
 'Pakora': 100,
 'Raita': 20,
 'Salad': 50,
 'Chicken Roll': 30,
 'Tikki': 10}
```

In [32]:
```
# Updating / manipulating values

d1['Tikki']= 20
d1
```

Out[32]:
```
{'Samosa': 30,
 'Pakora': 100,
 'Raita': 20,
 'Salad': 50,
 'Chicken Roll': 30,
 'Tikki': 20}
```

In [33]:
```
d2={'Dates':100,'Choclates':50,'Swayyan':500}
d2
```

Out[33]:
```
{'Dates': 100, 'Choclates': 50, 'Swayyan': 500}
```

In [34]:
```
#Concatenate
d1.update(d2)
d1
```

Out[34]:
```
{'Samosa': 30,
 'Pakora': 100,
 'Raita': 20,
 'Salad': 50,
```

```
'Chicken Roll': 30,
'Tikki': 20,
'Dates': 100,
'Choclates': 50,
'Swayyan': 500}
```

---

4. Sets

- Unordered and Unindex collection
- Enclosed in {}
- No duplicates allowed
- Elements stored can be changed (mutable)

In [35]:
```
s1={2,44,6.7, 'Ammar','Codanics','Faisalabad', True}
s1
```

Out[35]:
```
{2, 44, 6.7, 'Ammar', 'Codanics', 'Faisalabad', True}
```

In [36]:
```
# Duplicates cant add
s1.add('Ammar')
s1
```

Out[36]:
```
{2, 44, 6.7, 'Ammar', 'Codanics', 'Faisalabad', True}
```

In [ ]:
```

```