Topics : Integer, Float, String, List, Dictionary, Escape character

Data Type in Python

1. Integer : Number with no Decimal. Example -5,-4,-3,0,5,7
2. Float : Numbers that have decimal part. Example 1.234,-0.23,12.01
3. String : It refer to Text. Example 'Sandipan', 'Paul'

In [1]:

```
print("Integer Example")
userAge = 20
print(userAge)
print(type(userAge))
```

```
Integer Example
20
<class 'int'>
```

In [2]:

```
print("Float Example")
userHeight = 1.82
print(userHeight)
print(type(userHeight))
```

```
Float Example
1.82
<class 'float'>
```

In [3]:

```
print("String Example")
userName = 'Sandipan'
print('userName')
print(type(userName))
```

```
String Example
userName
<class 'str'>
```

String Formatter

- Formatting string using the %
- Format : "string to be formatted" % (values or variables to be inserted into string, seperated by commas)
- %s is for string
- %d for integer
- %f is for float. %4.2f suggets 4 is total length and 2 refers decimal places

In [4]:

```
name = 'Sandipan'
age = 27.61222
message = 'My name is %s Paul, height is %d cm and age is %4.2f' % (name,168,age)
print(message)
```

```
My name is Sandipan Paul, height is 168 cm and age is 27.61
```

List

- It refers to collection of Data which are normally related
- Instead of storing these data as seperate variable we can store them in list
- Example - suppose we need to store Age of 5 users we will use List
- Format : listName = [values]
- Suppose we want to assign empty list then listName = []
- Index starts from ZERO
- We can access list from start userAge[0],userAge[1] or back userAge[-1]

Slicing

- Reassign to another list also possible, userAge2 = userAge or userAge3 = userAge[2:4]
- : is known as SLICING, item at the start index is included but item at the end is always excluded
- [2:4] index 2 to index 3

Stepper

- [1:5:2] will give sub list consisting of every second number from index 1 to index 5 i.e, index 1 and index 5

Modify Index

- Format : listName[index of the item to be modified] = newValue
- Example : userAge[1] = 5

Add to list

- Use append()
- userAge.append(99) , this will add value 99 to the end of the list

Remove from list

- del listName[index of the item]
- Example : del userAge[2]

In [5]:

```
# Declare List
myList = [1,2,3,4,5,"Hello"]
print(myList)
print(type(myList))
print("Third Item :",myList[2])

print("Last Item :",myList[-1])

print("Assign myList (from index 1 to 4) to myList2 and print myList2")
myList2 = myList[1:5]
print(myList2)

print("Modify the second item in myList to 20 and print the updated list")
myList[1] = 20
print(myList)

print("Append a new item to myList and print updated list")
myList.append("How are you")
print(myList)

print("Remove Sixth element from the list")
del myList[5]
print(myList)
```

```
[1, 2, 3, 4, 5, 'Hello']
<class 'list'>
Third Item : 3
Last Item : Hello
Assign myList (from index 1 to 4) to myList2 and print myList2
[2, 3, 4, 5]
Modify the second item in myList to 20 and print the updated list
[1, 20, 3, 4, 5, 'Hello']
Append a new item to myList and print updated list
[1, 20, 3, 4, 5, 'Hello', 'How are you']
Remove Sixth element from the list
[1, 20, 3, 4, 5, 'How are you']
```

Tuple

- Tuples are just like List but Tuple cannot modify values
- Initial values are the values that will stay for the rest of the program
- Format : tupleName = (initial values)
- Multiple input are seperated by comma

In [6]:

```
# Define Tuples
monthsOfYear = ('Jan','Feb','Mar','Apr')
print(monthsOfYear)
print(type(monthsOfYear))
print("First Element :",monthsOfYear[0])
print("Last Element :",monthsOfYear[-1])
```

```
('Jan', 'Feb', 'Mar', 'Apr')
<class 'tuple'>
First Element : Jan
Last Element : Apr
```

Dictionary

- Collection of related data pairs
- For instance, store userName and age of 5 users, store it in Dictionary
- Format : dictionaryName = {"dictionary key" : data} where dictionary key must be unique
- Multiple input are seperated by commas
- Another Format : dictionaryName = dict(dictionary key = data)

Modify Items in Dictionary

- dictionaryName[dictionary key to modify] = new data

In [7]:

```
# Define Dictionary
userNameAndAge = {"Sandipan" : 27, "Ashish" : 30, "Vaibhav" : 26}
print(userNameAndAge)
print(type(userNameAndAge))

# Another Method to define Dictionary
userNameAndAge = dict(Sandipan = 27, Ashish = 30, Vaibhav = 26)
print(userNameAndAge)
print(type(userNameAndAge))

print("\nPrint First Data :",userNameAndAge['Sandipan'])

print("\nModify Data")
userNameAndAge['Vaibhav'] = 30
print(userNameAndAge)

print("\nAdd new Data to Dictionary")
userNameAndAge['Sambit'] = 35
print(userNameAndAge)

print("\nRemove Data from Dictionary")
del userNameAndAge['Sambit']
print(userNameAndAge)
```

```
{'Sandipan': 27, 'Ashish': 30, 'Vaibhav': 26}
<class 'dict'>
{'Sandipan': 27, 'Ashish': 30, 'Vaibhav': 26}
<class 'dict'>

Print First Data : 27

Modify Data
{'Sandipan': 27, 'Ashish': 30, 'Vaibhav': 30}

Add new Data to Dictionary
{'Sandipan': 27, 'Ashish': 30, 'Vaibhav': 30, 'Sambit': 35}

Remove Data from Dictionary
{'Sandipan': 27, 'Ashish': 30, 'Vaibhav': 30}
```

In [8]:

```
# Triple Quotes : Display long message using print() and use ''' '''

print('''Sandipan is a good boy,
My name is Sandipan and
I am 28 year old''')
```

```
Sandipan is a good boy,
My name is Sandipan and
I am 28 year old
```

Escape Characters

- Some special character are "unprintable"
- Example Tab or a newline
- In this case we need to use \ (backslash)
- Following are the list
  - \' Single Quote
  - \ Backslash
  - \n New Line
  - \r Carriage Return
  - \t Tab
  - \b Backspace
  - \f Form Feed
  - \ooo Octal value
  - \xhh Hex value

In [9]:

```
print("Sandipan\nPaul")
print("I am 5'9 tall")
print('I am 5\'9 tall')
```

```
Sandipan
Paul
I am 5'9 tall
I am 5'9 tall
```