

# Mini Project

Two classes

# Python - List

- **List.** Is a collection of items of same or different data types.
- A list stores elements one after another with comma separator.
- List is index based with index starting with zero.
- Elements are enclosed in square brackets [ ]
- Elements in the list can repeat
- It does not provide fast lookups.
- Finding an element is often slow.
- A search is required.

# Python - List

- List is collection of elements of different types:
- `>>> listMixed = [1,1.5,'A','KMIT Hyderabad']`
- `>>> print(listMixed)`
- `[1, 1.5, 'A', "KMIT Hyderabad"]`

# Exercise

```
items = [ "book", "computer", 'book',1,1.5]  
for element in items:  
    print(type(element))
```

Output

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

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

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

```
<class 'int'>
```

```
<class 'float'>
```

# Python - List

- **Append:** This method is called upon the list instance (which must not equal None). It receives the value we are adding.
- **Append**
- `list = []`
- `list.append(1)`
- `list.append(2)`
- `list.append(6)`
- `list.append(3)`
- `print(list)`
- **Output**
- `[1, 2, 6, 3]`

# Dictionaries

- A dictionary is a data structure.
- A dictionary is a collection which is
  - unordered,
  - indexed.
  - Changeable or updated and
- In Python dictionaries are written with curly brackets { },
- and every element is a
  - key and value pair.
    - 1: “First”
    - “B”:200

# Dictionaries

- Basics of python dictionaries
- In dictionary the values are accessed using key rather than index
- 1. Create a dictionary:
- `myDict = {'a':"apple",'b':'boy',3:'third class','d':400}`
- A dictionary is created.
- This dictionary contains three elements.
- Each element constitutes of **a key (A)** **value (Apple)** pair.
- This dictionary can be accessed using the dictionary identifier `myDict`.
- `print(myDict)`
- `{'a': 'apple', 'b': 'boy', 3: 'third class', 'd': 400}`

# Dictionaries

## 3. Update Dictionary Elements

- Just the way dictionary values are accessed using keys, the values can also be modified using the dictionary keys:
- ```
>>> myDict['a'] = "Application"
```
- ```
>>> myDict
```
- ```
{'a': 'Application', 'b': 'boy', 3: 'third class', 'd': 400}
```
- **Note:-** In a dictionary two keys cannot be same.
- ```
>>> mydict = {'a': 'Application', 'a' : 'App'}
```
- ```
>>> mydict
```
- ```
{'a': 'app'}
```



# Dictionary and for iterator

'''

**for iterator and dictionary**

**Using for iterator print all the key and value elements**

**d = {'x': 1, 'y': 2, 'z': 3}**

'''

**d = {'x': 1, 'y': 2, 'z': 3}**

**for key in d:**

**print (key, 'corresponds to', d[key])**

x corresponds to 1  
y corresponds to 2  
z corresponds to 3

# Mini Project

**Create Two Classes Student & StudentMarks.**

**Write a menu driven program where**

**1. Adding New student & his marks**

**3. Exit**

**Give Choice:**

**When user selects 1 ask the user**

**Give Roll No:**

**Give Student name:**

**Call addStudent() of this class and save the rollNo and sName pair in a dictionary stuDetails. From Student class call dispStudent() and display**

**100 : Ramesh**

**101 Call giveMarks() from the second class StudentMarks by passing rollNo as parameter by giving a prompt**

**Physics marks: 50**

**Maths marks: 60**

**Chemistry marks: 70**

**Followed by this call dispMarks() method of the same class and display**

**100 70 80 90**

# Mini Project

**1. Adding New student & his marks**

**3. Exit**

**Give choice 1**

**Give Roll No: 1000**

**Give Student name: Ramesh**

**1000 : Ramesh**

**Physics marks: 70**

**Maths marks: 80**

**Chemistry marks: 90**

**1000 70 80 90**

**1. Adding New student & his marks**

**3. Exit**

**Give choice 1**

**Give Roll No: 1001**

**Give Student name: Sita**

**1000 : Ramesh**

**1001 : Sita**

**Physics marks: 75**

**Maths marks: 85**

**Chemistry marks: 95**

**1000 70 80 90**

**1001 75 85 95**

**1. Adding New student & his marks**

**3. Exit**

**Give choice 3**

- class Student:
- stuDetails = {}
- def \_\_init\_\_(self):
- self.rollName = {}
- def addStudent(self, id, val):
- Student.stuDetails[id] = val
- self.dispStudent()
- def dispStudent(self):
- for key in Student.stuDetails:
- print("%d : %5s" %(key, Student.stuDetails[key]))
-

- class StudentMarks:
- stuMarks = []
- def giveMarks(self,rno):
- phy = int(input("Physics marks: "))
- mat = int(input("Maths marks: "))
- che = int(input("Chemistry marks: "))
- StudentMarks.stuMarks.append(rno)
- StudentMarks.stuMarks.append(phy)
- StudentMarks.stuMarks.append(mat)
- StudentMarks.stuMarks.append(che)
- self.dispMarks()
- def dispMarks(self):
- count = len(StudentMarks.stuMarks)
- for i in range(0,count):
- if(i%4 == 0):
- print()
- print(StudentMarks.stuMarks[i], end=' ')
-

- def menu():
- stu = Student()
- stuMks = StudentMarks()
- while(1):
- print()
- print("1. Adding New student & his marks")
- #print("2. Assigning marks of 3 subject")
- print("3. Exit")
- choice = int(input("Give choice "))
- if(choice == 1):
- rno = int(input("Give Roll No: "))
- nam = input("Give Student name: ")
- stu.addStudent(rno,nam)
- stuMks.giveMarks(rno)
- if(choice == 2):
- phy = int(input("Physics marks: "))
- mat = int(input("Maths marks: "))
- che = int(input("Chemistry marks: "))
- stu.addStudent(rno,nam)
- if(choice == 3):
- break
- 
- menu()



