

TUPLES AND DICTIONARIES

Accessing Items in a Dictionary

```
>>> dict3 = {'Mohan': 95, 'Ram': 89, 'Suhel': 92,  
             'Sangeeta': 85}  
>>> dict3['Ram']  
89
```

Dictionaries are Mutable

① Adding a new item

```
>>> dict1 = {'Mohan': 95, 'Ram': 99, 'Advait': 100}  
>>> dict1['Ashwini'] = 99  
>>> dict1
```

Modifying an Existing Item

```
>>> dict1 = {'Mohan': 95, 'Ram': 99, 'Advait': 100}  
>>> dict1['Advait'] = 100.5
```

Traversing a Dictionary

We can access each item of the dictionary or traverse a dictionary

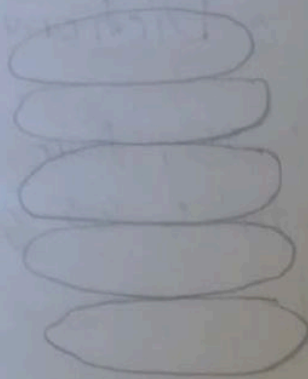
using

M-1

```
for >>> dict1 = {'Mohan': 95, 'Ram': 99,  
>>> for key in dict1:  
    print(key, ': ', dict1[key])
```

PDSA

Stack



LIFO

Last In First Out

Applications of Stack

While browsing the web we visit many webpages and when we click back button we go to the previous page while browsing history stored in as stack.

OPERATIONS ON STACK

PUSH and POP Operations

→ PUSH adds a new element at the TOP of the stack.
It is an insertion operation

→ POP operation is used to remove the top most element of the stack

The simple way to implement a stack in Python is using the data type list

- Let us create an empty stack named glassStack
glassStack = list()

```
def isEmpty(glassStack):  
    return True  
    else:  
        return False
```

A function named onPush to insert (PUSH)
anew element in a stack.

This function has to two parameters
→ the name of the stack in which
the element is to be inserted
(glassStack) and the elements that
needs to be inserted.

```
def onPush(glassStack, element):  
    glassStack.append(element)
```

The built-in-function method append() of
list to add an element to the stack
that always adds at the end of
the list.

→ A function named size to read the number
of elements in the glassStack.
We will use the len() function
of list in Python to find the no. of
elements in the glassStack.