

# Dictionaries

- A dictionary is a data structure.
- A dictionary is a collection which is
  - unordered,
  - changeable and
  - indexed.
- 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
- 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 variable `myDict`.
- `print(myDict)`
- `{'a': 'apple', 'b': 'boy', 3: 'third class', 'd': 400}`

- 2. Access Dictionary Elements **Dictionaries**
- Once a dictionary is created, you can access each value using the key to which it is assigned during creation.

```
>>> myDict['a']
```

```
'apple'
```

```
>>> myDict["b"]
```

```
'boy'
```

```
>>> myDict[3]
```

```
'third class'
```

```
>>> myDic["d"]
```

```
400
```

# Dictionaries

- To access all **key** : **values** the variable myDict can be used to access the dictionary elements.
- `>>> myDict`
- `{'A': 'Apple', 'C': 'Cat', 'B': 'Boy'}`
- Only dictionary keys can be used as indexes.
- This means that `myDict["A"]` would produce 'Apple' in output but `myDict["Apple"]` cannot produce 'A' in the output.

```
>>> myDict["Apple"]
```

```
Traceback (most recent call last):  File "<stdin>", line 1, in  
    <module>
```

```
KeyError: 'Apple'
```

# 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', 'C': 'Cat', 'B': 'Boy'}
```
- **Note:-** There could not be two keys with same name in a dictionary.
- ```
>>> mydict = {'a': 'application', 'a' : 'app'}
```
- ```
>>> mydict
```
- ```
{'a': 'app'}
```

# Dictionaries

## 4. Delete Dictionary Elements

- Individual elements can be deleted easily from a dictionary.
- We need to use **del** and the key of value to be deleted:-
- ```
>>> myDict
```
- ```
{'A': 'Application', 'C': 'Cat', 'B': 'Boy'}
```
- ```
>>> del myDict["A"]
```
- ```
>>> myDict {'C': 'Cat', 'B': 'Boy'}
```

# Dictionaries

- If you want to delete complete dictionary ie all the elements in the dictionary then it can be done using the `clear()` function. Here is an example :
- ```
>>> myDict
```
- ```
{'C': 'Cat', 'B': 'Boy'}
```
- ```
>>> myDict.clear()
```
- ```
>>> myDict
```
- ```
{}
```
- All the elements were deleted making the dictionary empty.

# Dictionaries

- Characteristics of Python Dictionaries
  1. Dictionaries are Unordered
  2. The dictionary elements (key-value pairs) are not in ordered form.
- `>>> myDict {'A': 'Apple', 'C': 'Cat', 'B': 'Boy'}`
- `{'C': 'Cat', 'B': 'Boy', 'A': 'Apple'}`
- You can observe that the order of elements while the dictionary was being created is different from the order in which they are actually stored and displayed.



# Dictionaries

- Even if you try to add other elements to python dictionary:
- `>>> myDict["D"] = "Dog"`
- `>>> myDict`
- `{'A': 'Apple', 'C': 'Cat', 'B': 'Boy', 'D': 'Dog'}`
- `>>> myDict["E"] = "Elephant"`
- `>>> myDict`
- `{'A': 'Apple', 'C': 'Cat', 'B': 'Boy', 'E': 'Elephant', 'D': 'Dog'}`
- You'll observe that it's not necessary that elements will be stored in the same order in which they were created.

# Dictionaries

## 2. Dictionary Keys are Case Sensitive

- Same key name but with different case are treated as different keys in python dictionaries.
- Here is an example :
- `>>> myDict["F"] = "Fan"`
- `>>> myDict["f"] = "freeze"`
- `>>> myDict`
- `{'A': 'Apple', 'C': 'Cat', 'B': 'Boy', 'E': 'Elephant', 'D': 'Dog', 'F': 'Fan', 'f': 'freeze'}`

- Required arguments
- Required arguments are the arguments passed to a function in correct positional order.
- Here, the number of arguments in the function call should match exactly with the function definition.
- # Function Arguments are required
- `def squarerect(sqrec):`
- `for i in sqrec:`
- `if (i==1):`
- `print("Square area = ",sqrec[i]*sqrec[i])`
- `if (i==2):`
- `print("Rectangle area = ",sqrec[1]*sqrec[2] )`
- `sr = {1:10,2:20}` # sr is a dictionary with 1 as key & 10 as value
- `squarerect(sr)`

# Dictionaries

- > Generate Electricity Bill with following scales:
  1. Between 0 to 100 @ 2/unit and service Charges 20
  2. Between 101 to 200 @ 3/unit and service Charges 30
  3. 201 to 300 @ 4/unit and service Charges 40
  4. above 300 @ 5/unit and service Charges 50
  5. Add sales tax 14%

For example: Bill for 253

Charge for first 100 = 200

Charge for second 100 = 300

Charge for Remaining 53 = 212

Total Electricity Bill is:  $712 + 99.68(14\% \text{ of } 712) + 40 = 851.68$

prompt the user " Do you want to generate another bill?"

if no stop

else prompt the user "Do you want to change the rate "

if yes increase each unit rate by .1 and generate the bill with revised rate

else generate the bill with old rate

# Dictionaries

```
• print("You want electricity Bill, ENTER 1 : ")
• n=1
• EBill={1:1.50,2:2.50,3:4.00,4:6.50,5:8}
• while(n):
•     units=int(input("Enter units consumed: "))
•     if units<=100:
•         first=EBill.get(1)
•         SCharge=25
•         DBill=units*first
•         STax=DBill*.14
•         print("Unit Charge is :", DBill)
•         print("Sur Charge for units ",units," is :", SCharge)
•         print("Service Charge for units ",units," is :", STax)
•         Bill=DBill+STax+SCharge
•     elif units>100 and units<=200:
•         second=EBill.get(2)
•         SCharge=35
•         CH1=100*EBill.get(1)
•         remaining=(units-100)*second
•         DBill=remaining+CH1
•         STax=DBill*.14
•         Bill=DBill+STax+SCharge
•         print("Charge for 100 units is :",CH1)
•         print("Charge for remaining ",units-100," units is :",remaining)
•         print("Sur Charge for units ",units," is :", SCharge)
•         print("Service Charge for units ",units," is :", STax)
•         print("Total Electricity Bill for units ",units," is :",Bill)
•     elif units>200 and units<=300:
•         third=EBill.get(3)
•         SCharge=50
•         CH1=100*EBill.get(1)
•         CH2=100*EBill.get(2)
•         remaining=(units-200)*third
•         DBill=remaining+CH1+CH2
•         STax=DBill*.14
•         Bill=DBill+STax+SCharge
•         print("Charge for first 100 units is :",CH1)
•         print("Charge for second 100 units is :",CH2)
•         print("Charge for remaining ",units-200," units is :",remaining)
•         print("Sur Charge for units ",units," is :", SCharge)
•         print("Service Charge for units ",units," is :", STax)
•         print("Total Electricity Bill for units ",units," is :",Bill)
•     elif units>300 and units<=400:
•         fourth=EBill.get(4)
•         SCharge=75
•         CH1=100*EBill.get(1)
•         CH2=100*EBill.get(2)
•         CH3=100*EBill.get(3)
•         remaining=(units-300)*fourth
•         DBill=remaining+CH1+CH2+CH3
•         STax=DBill*.14
```

# Dictionaries

```
print("You want electricity Bill, ENTER 1 : ")
n=1
EBill={1:1.50,2:2.50,3:4.00,4:6.50,5:8}
while(n):
    units=int(input("Enter units consumed: "))
    if units<=100:
        first=EBill.get(1)
        SCharge=25
        DBill=units*first
        STax=DBill*.14
        print("Unit Charge is :", DBill)
        print("Sur Charge for units ",units," is :", SCharge)
        print("Service Charge for units ",units, " is :", STax)
        Bill=DBill+STax+SCharge
    elif units>100 and units<=200:
        second=EBill.get(2)
        SCharge=35
        CH1=100*EBill.get(1)
        remaining=(units-100)*second
        DBill=remaining+CH1
        STax=DBill*.14
        Bill=DBill+STax+SCharge
        print("Charge for 100 units is :",CH1)
        print("Charge for remaining ",units-100," units is :",remaining)
        print("Sur Charge for units ",units," is :", SCharge)
        print("Service Charge for units ",units, " is :", STax)
        print("Total Electricity Bill for units ",units," is :",Bill)
```

```
elif units>200 and units<=300:
    third=EBill.get(3)
    SCharge=50
    CH1=100*EBill.get(1)
    CH2=100*EBill.get(2)
    remaining=(units-200)*third
    DBill=remaining+CH1+CH2
    STax=DBill*.14
    Bill=DBill+STax+SCharge
    print("Charge for first 100 units is :",CH1)
    print("Charge for second 100 units is :",CH2)
    print("Charge for remaining ",units-200," units is :",remaining)
    print("Sur Charge for units ",units," is :", SCharge)
    print("Service Charge for units ",units, " is :", STax)
    print("Total Electricity Bill for units ",units," is :",Bill)
elif units>300 and units<=400:
    fourth=EBill.get(4)
    SCharge=75
    CH1=100*EBill.get(1)
    CH2=100*EBill.get(2)
    CH3=100*EBill.get(3)
    remaining=(units-300)*fourth
    DBill=remaining+CH1+CH2+CH3
    STax=DBill*.14
    Bill=DBill+STax+SCharge
    print("Charge for first 100 units is :",CH1)
    print("Charge for second 100 units is :",CH2)
    print("Change for third 100 units is :",CH3)
    print("Charge for remaining ",units-300," units is :",remaining)
    print("Sur Charge for units ",units," is :", SCharge)
    print("Service Charge for units ",units, " is :", STax)
    print("Total Electricity Bill for units ",units," is :",Bill)
```

# Dictionaries

else:

```
fifth=EBill.get(5)
SCharge=100
CH1=100*EBill.get(1)
CH2=100*EBill.get(2)
CH3=100*EBill.get(3)
CH4=100*EBill.get(4)
remaining=(units-400)*fifth
DBill=remaining+CH1+CH2+CH3+CH4
STax=DBill*.14
Bill=DBill+STax+SCharge
print("Charge for first 100 units is :",CH1)
print("Charge for second 100 units is :",CH2)
print("Change for third 100 units is :",CH3)
print("Charge for fourth 100 units is :",CH4)
print("Charge for remaining ",units-400," units is
:",remaining)
print("Sur Charge for units ",units," is :",
SCharge)
print("Service Charge for units ",units," is :",
STax)
print("Total Electricity Bill for units ",units," is
:",Bill)
```

```
yn=input("You want generate another Bill: ")
if yn=='y':
    print("Generate next Print : ")
else:
    break
yesno=input("Do you want to modify charges: ")
if yesno=='y':
    print(EBill)
    for i in EBill:
        EBill[i]=EBill[i]+.10
    print("After Modification Rates are:")
    print(EBill)
```

# Dictionaries



# Dictionaries

# Dictionaries

# Dictionaries