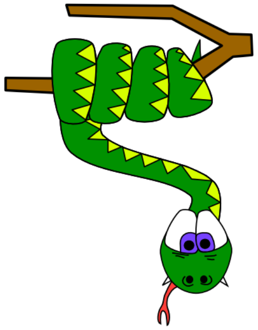

Dictionary

Session-VI

The *dictionary* data structure

- In Python, a *dictionary* is mapping between a set of indices (**keys**) and a set of **values**
  - The items in a dictionary are key-value pairs
- Keys can be any Python data type
  - Because keys are used for indexing, they should be immutable
- Values can be any Python data type
  - Values can be mutable or immutable

#In general, the order of items in a dictionary is unpredictable

#Dictionaries are indexed by keys, not integers

#Note that the in operator works differently for dictionaries than for other sequences

- For offset indexed sequences (strings, lists, tuples), x in y checks to see whether x is an item in the sequence
- For dictionaries, x in y checks to see whether x is a key in the dictionary

Creating a Dcitionary

```
#Creating Dictionary
d=dict()#Empty Dictionary
print(type(d))#<class 'dict'>
d={}
print(type(d))#<class 'dict'>
#Initialization-ex a key Value pair of Name & Age
d={"Ram":30, "Jai":40}
print(d)
```

Adding Elements

```
#Adding values to dictionary
d={}
d['James']=78
print(d)
d.update({'Harry':42})#if key present updates value else add key to dictionary
print(d)
d.setdefault('Syam',26)
print(d)
#using user input
d[input('Enter Key')]=eval(input('Enter Value'))
print(d)
d.setdefault(input('Enter key'),eval(input('Enter Value')))
print(d)
t={input('Enter key'):input('Enter Value'),input('Enter key'):input('Enter 
Value')}
print(t)
d.update({input('Enter key'):eval(input('Enter value'))})
print(d)
```

Accessing Values

```
#Accessing Dictionary Values
d={'Ram': 30, 'Jai': 40, 'Hari': 25, 'Shyam': 30, 'Jhon': 25}
x=d['Ram']
print(x)# 30
print(d['Ram'])# 30
print(d.get('Ram'))# 30
print(d.setdefault("Ram"))# 30
#KeyError--if key is not present python raises exception
print(d['James'])# KeyError: 'James'
```

Suppressing keyError

```
#Supressing KeyError
d={"Jhon":34, 1:23, "Harry":25, 22:'abc'}
#Accessing a key not present--suppressing keyError-- Method--1
print(d.get("Alice"))#returns None by default
print(d)#Alice will not be added as a key in d
##Customizing get
#Override default None by Some customized value
print(d.get("Alice", "Key not found in dict"))
#Method--2
print(d.setdefault("Alice"))
#returns None and key added to dictionary--value None
print(d.setdefault("Joe", 48))
#returns 48 and key added--value 48
print(d)
#{'Jhon': 34, 1: 23, 'Harry': 25, 22: 'abc', 'Alice': None, 'Joe': 48}
```

Keys, values, items

```
#keys,values and items
d={'Jhon': 34, 1: 23, 'Harry': 25, 22: 'abc', 'Alice': 32, 'Joe': 48}
l=d.keys()#returns object dict_keys
#dict_keys(['Jhon', 1, 'Harry', 22, 'Alice', 'Joe'])
print(l)
t=d.values()#returns dict_values
#dict_values([34, 23, 25, 'abc', 32, 48])
print(t)
s=d.items()#returns dict_items-- (key,value)
#dict_items([('Jhon', 34), (1, 23), ('Harry', 25), (22, 'abc'),
('Alice', 32), ('Joe', 48)])
print(s)
print(type(l))#<class 'dict_keys'>
print(type(t))#<class 'dict_values'>
print(type(s))#<class 'dict_items'>
l=list(l)#converting in list
print(l) #['Jhon', 1, 'Harry', 22, 'Alice', 'Joe']
```

Iterating Dictionary

```
#iterating dictionary
d={'Jhon': 34, 1: 23, 'Harry': 25, 22: 'abc', 'Alice': None,
'Joe': 48}
for i in d.keys():
    print(i,d[i])
for k,v in d.items():
    print(k,v)
```

Leftovers

- #Misc. Dictionary operations
- d={'Jhon': 34, 1: 23, 'Harry': 25, 22: 'abc', 'Alice': None, 'Joe': 48}
- s={'Jhon':32,'ken':19}
- d.update(s)
- print(d)#Existing key gets updated and new key gets added
- l=['ram',23.67,('shyam',67)]
- dt={}
- dt=dt.fromkeys(l)
- #creates dictioanry from a sequence of immutable-value is None
- print((dt))
- dt=dt.fromkeys(l,-1)#default value is -1
- print(dt)

Sorting

```
#sorting dictionary
d={'Jhon': 34, 'Ram': 23, 'Harry': 25, 'James': 97, 'Alice': 56,
'Joe': 48}
l=sorted(d)#returns sorted list of key
print(l)
l=sorted(d.keys())#returns list of key
print(l)
l=sorted(d.items())#returns list of sorted key value pair based on
key
print(l)
l=sorted(d.values())#returns list of values
print(l)
l=sorted(d.items(), key=lambda x:x[1])#sort dictionary on value
print(l)
l=sorted(d.items(), key=lambda x:x[0])#sort dictionary on keys
print(l)
```

Nested values(left)

```
#nested values & updation
d={1:['ab',34,45.67],23:['rt','yt','lo']}
d[1][1]=90
print(d)
d={'ab':{'a':2,'b':34},'bc':{'x':89,90:'n'}}
d['ab']['a']=4
print(d)
l=['ram',23.67,('shyam',67)]
dt={}
dt=dt.fromkeys(l)
#creates dictioanry from a sequence of immutable-value is None
print((dt))
dt=dt.fromkeys(l,-1)#default value is -1
print(dt)
```