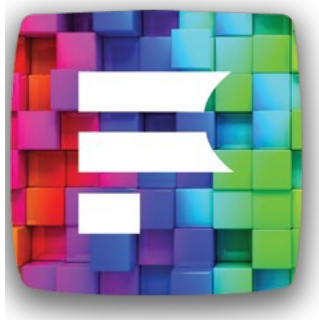

Future Connect Media

Python Part D

Part of Future Connect Media's IT Course

By Abhishek Sharma

Future Connect Media

Topics to be covered:

Set

Dictionary

Set

- Set: A Set is an iterable, changeable, and has no duplicate unordered collection data type. It is denoted by curly brackets {}.

```
>>> set={'audi','mercedes','bmw','jaguar','audi'}
>>> print(set)
{'mercedes', 'bmw', 'audi', 'jaguar'}
```

- Set() constructor:

```
>>> s=set(('audi','bmw','mercedes','jaguar'))
>>> print(s)
{'audi', 'bmw', 'mercedes', 'jaguar'}
```

- Adding to a set:

```
>>> s={'audi','bmw','mercedes','jaguar'}
>>> s.add('bentley')
>>> print(s)
{'audi', 'bmw', 'jaguar', 'mercedes', 'bentley'}
```

- Adding sets:

```
>>> s={'audi','bmw','mercedes','jaguar'}
>>> s1={'1','2','3','4'}
>>> s.update(s1)
>>> print(s)
{'audi', '4', '1', 'bmw', 'jaguar', 'mercedes', '2', '3'}
```

- Removing item from set:

```
>>> s={'audi','bmw','mercedes','jaguar'}
>>> s.remove('jaguar')
>>> print (s)
{'audi', 'bmw', 'mercedes'}
```

#Discard() can be also used to remove ‘jaguar’.

- Remove random item from the set by using pop():

```
>>> s={'audi','bmw','mercedes','jaguar'}
>>> s.pop()
'audi'
```

- Clearing the set:

```
>>> s={'audi','bmw','mercedes','jaguar'}
>>> s.clear()
>>> print(s)
set()
```

- Join two sets:

```
>>> s={'audi','bmw','mercedes','jaguar'}
>>> s1={'silver','white','black'}
>>> s.union(s1)
{'audi', 'jaguar', 'mercedes', 'white', 'black', 'bmw', 'silver'}
```

#del() is used to delete the set.

#update() can be used to update set s with set s1.

- Intersection_update(): Common items in both the sets.

```
>>> s={'audi','bmw','mercedes','jaguar'}
>>> s1={'black','white','bmw'}
>>> s.intersection_update(s1)
>>> print(s)
{'bmw'}
```

- Symmetric_difference():

```
>>> s={'audi','bmw','mercedes','jaguar'}
>>> s1={'black','white','bmw'}
>>> s.symmetric_difference(s1)
{'white', 'audi', 'jaguar', 'mercedes', 'black'}
```

Set method

Method	Description
add()	Adds an element to the set
clear()	Removes all the elements from the set
copy()	Returns a copy of the set
difference()	Returns a set containing the difference between two or more sets
difference_update()	Removes the items in this set that are also included in another, specified set
discard()	Remove the specified item
intersection()	Returns a set, that is the intersection of two other sets
intersection_update()	Removes the items in this set that are not present in other, specified set(s)
isdisjoint()	Returns whether two sets have a intersection or not
issubset()	Returns whether another set contains this set or not
issuperset()	Returns whether this set contains another set or not
pop()	Removes an element from the set
remove()	Removes the specified element
symmetric_difference()	Returns a set with the symmetric differences of two sets
symmetric_difference_update()	inserts the symmetric differences from this set and another
union()	Return a set containing the union of sets
update()	Update the set with the union of this set and others

- Dictionary: Unlike other data types, which can only retain a single value as an element, a dictionary in Python is a collection of keys and values that is used to store data values like a map.

```
>>> dict={'brand':'mercede','model':'c class','year':2014}
>>> print (dict)
{'brand': 'mercede', 'model': 'c class', 'year': 2014}
```

```
>>> dict={'brand':'mercede','model':'c class','year':2014}
>>> print (dict['model'])
c class
```

- Change item in the dictionary:

```
>>> dict={'brand':'mercede','model':'c class','year':2014}
>>> dict['model']= 'e calss'
>>> print(dict)
{'brand': 'mercede', 'model': 'e calss', 'year': 2014}
```

- Update():

```
>>> dict={'brand':'mercede','model':'c class','year':2014}
>>> dict.update({'model':'s63 amg'})
>>> print (dict)
{'brand': 'mercede', 'model': 's63 amg', 'year': 2014}
```

- Adding items is dictionary:

```
>>> dict={'brand':'mercede','model':'c class','year':2014}
>>> dict['colour']='black'
>>> print (dict)
{'brand': 'mercede', 'model': 'c class', 'year': 2014, 'colour': 'black'}
```

- Pop():

```
>>> dict={'brand':'mercede','model':'c class','year':2014}
>>> dict.pop('brand')
'mercede'
```

Future Connect Media

Dictionary method

Method	Description
clear()	Removes all the elements from the dictionary
copy()	Returns a copy of the dictionary
fromkeys()	Returns a dictionary with the specified keys and value
get()	Returns the value of the specified key
items()	Returns a list containing a tuple for each key value pair
keys()	Returns a list containing the dictionary's keys
pop()	Removes the element with the specified key
popitem()	Removes the last inserted key-value pair
setdefault()	Returns the value of the specified key. If the key does not exist: insert the key, with the specified value
update()	Updates the dictionary with the specified key-value pairs
values()	Returns a list of all the values in the dictionary

- Nested dictionary:

```
dict = {
    'mercedes':{
        'model':'c class',
        'colour':'black'    },
    'bmw':{
        'model':'5 series',
        'colour':'white'
    }
}
print(dict)
```

```
{'mercedes': {'model': 'c class', 'colour': 'black'}, 'bmw': {'model': '5 series', 'co
lour': 'white'}}
```

```
mercedes={
    'model':'c class',
    'colour':'black'
}
bmw={
    'model':'5 series',
    'colour':'white'
    }
dict={
    'mercedes':mercedes,
    'bmw':bmw
}
print(dict)
```

```
{'mercedes': {'model': 'c class', 'colour': 'black'}, 'bmw': {'model': '5 series', 'co
lour': 'white'}}
```

#Creating 2 different dictionaries and combining it into dict dictionary.