PYTHON

From Simple to Complex
With
Examples

AYESHA NOREEN

Bachelor's in Software Engineering,

Master's in Computer Science

from COMSATS University, Islamabad

NOTE!!!

In these notes Screenshots of practice examples and coding are added. The code files are also available in code folder that contain .ipynb files that are created on Jupyter notebook.

Chapter12
Dictionaries in Python

Dictionaries are mutable, ordered and not allow duplication. Dictionary contain data in key: value pair. e.g.

mixed={'name':'Ayesha', 'city' : 'Kamalia'}

There is no indexing in dictionary we can access its values through its keys.

```
user=dict(name='Ayesha',age=22)  #creating dictionary through another method
print(user)
print(user['name'])  #we can access values of dictionary trough key
#we can create dictionary inside dictionary
users={'user1':{'name1':'ayesha','age1':22,'city':'kamalia'},'user2':{'name2':'noreen','age2':22}}
print(users['user1'])
emp={}  #we can add data in empty dictionary as
emp['name']='ayesha'
print(emp)
```

```
{'name': 'Ayesha', 'age': 22}
Ayesha
{'name1': 'ayesha', 'age1': 22, 'city': 'kamalia'}
{'name': 'ayesha'}
```

- # In keyword in Dictionary

```
mixed={'name':'Ayesha','city':'Lahore'}
if 'name' in mixed:
    print("Present")
else:
    print("Not present")
#it prints present
if 'kamalia' in mixed:
    print("present")
else:
    print("not present")
#it prints not present bcz it chk keys only
if 'kamalia' in mixed.values():
    print("present")
else:
    print("not present")
#it prints present bcz values mehod is called
```

```
Present
not present
not present
```

- Keys() and values() method in Dictionary

```
mixed_dictionary_values=mixed.values()
print(f"values of dictinary is:{mixed_dictionary_values}")
mixed_dictionary_keys=mixed.keys()
print(f"keys of dictinary is:{mixed_dictionary_keys}")
```

```
values of dictinary is:dict_values(['Ayesha', 'Lahore'])
keys of dictinary is:dict_keys(['name', 'city'])
```

- Items() method in Dictionary

```
#items methods
mixed={'name':'Ayesha noreen','city':'kamalia'}
items_of_dictionary=mixed.items()
print(items_of_dictionary)
# it prints dict_items([('name', 'Ayesha noreen'), ('city', 'kamalia')])
```

```
dict_items([('name', 'Ayesha noreen'), ('city', 'kamalia')])
```

- # Looping in Dictionary

```
mixed={'name':'Ayesha noreen','city':'kamalia'}
for i in mixed:
    print(i)
    #gives only key values
for i in mixed.values():
    print(i)
    #gives only values
for i,j in mixed.items():
    print(f"keys are:{i} and values are:{j}")
    #it prints keys are:name and values are:Ayesha noreen
    #keys are:city and values are:kamalia
    #so items is best bcz give bith keys and values
```

```
name
city
Ayesha noreen
kamalia
keys are:name and values are:Ayesha noreen
keys are:city and values are:kamalia
```

- # Add or delete data from Dictionary

```
#add and delete data
mixed={'name':'Ayesha noreen','city':'lahore'}
mixed['songs']=['song1','song2']
print(mixed)
poped_item=mixed.pop('city')
print(poped_item)
print(mixed)
poped_item=mixed.popitem()
print(poped_item)
print(type(poped_item))
print(mixed)
#both pop method and popitem method are used to delete values of dictionary
#but pop method take exactly one parameter and popitem take no parameter and
#delete randomly from dictionary.popitem return type tuple and pop method type
#is according to data which is deleted.
```

```
{'name': 'Ayesha noreen', 'city': 'lahore', 'songs': ['song1', 'song2']}
lahore
{'name': 'Ayesha noreen', 'songs': ['song1', 'song2']}
('songs', ['song1', 'song2'])
<class 'tuple'>
{'name': 'Ayesha noreen'}
```

- # Update() method

```
#update method
user_info={'name':'Ayesha noreen','city':'kamalia'}
more_info={'state':'pakistan'}
user_info.update(more_info)
print(user_info)
user_info.update({})
#this does not have any change bcz update method is empty
print(user_info)
```

```
{'name': 'Ayesha noreen', 'city': 'kamalia', 'state': 'pakistan'}
{'name': 'Ayesha noreen', 'city': 'kamalia', 'state': 'pakistan'}
```

- From keys() method

```
d=dict.fromkeys(['name','age','city'],'123')
print(d)
d=dict.fromkeys(('name','age','city'),'123')
print(d)
#we can also use tuple in place of list it gives same output
#it prints {'name': '123', 'age': '123', 'city': '123'} 123 is assign to aach as its value
d=dict.fromkeys('state','unknown')
print(d)
#if we pass a string than it assign values to each element of string as
#{'s': 'unknown', 't': 'unknown', 'a': 'unknown', 'e': 'unknown'}
d=dict.fromkeys(range(1,11),'unknown')
print(d)
#we can also use range function it assign default value to each number in range it prints as
#{1: 'unknown', 2: 'unknown', 3: 'unknown', 4: 'unknown', 5: 'unknown', 6: 'unknown',
# 7: 'unknown', 8: 'unknown', 9: 'unknown', 10: 'unknown'}
d=dict.fromkeys(['name','age'],['unknown','unknown'])
print(d)
```

```
{'name': '123', 'age': '123', 'city': '123'}
{'name': '123', 'age': '123', 'city': '123'}
{'s': 'unknown', 't': 'unknown', 'a': 'unknown', 'e': 'unknown'}
{1: 'unknown', 2: 'unknown', 3: 'unknown', 4: 'unknown', 5: 'unknown', 6: 'unknown', 7: 'unknown', 8: 'unknown', 9: 'unknown', 10: 'unknow
n'}
{'name': ['unknown', 'unknown'], 'age': ['unknown', 'unknown']}
```

- # Get() method

```
#get method
#Get method is used to access data of dictionary.
user_info={'name':'ayesha','age':22}
print(user_info['name']) #gives ayesha
#print(user_info['names']) #gives error bcz names is not present
print(user_info.get('name')) #gives ayesha
print(user_info.get('names')) #gives none not error bcz names is none mean not present
```

```
ayesha
ayesha
None
```

- # Clear() method

```
#clear method
#clear method is used for clear of dictionary.
user_info={'name':'ayesha','age':22}
user_info.clear()
print(user_info) #prints {}
```

```
{}
```

- # Copy() method

```
#copy method
#copy method is used for coping one dictionary into another.
user_info={'name':'ayesha','age':22}
copied=user_info.copy()
print(copied)
#prints {'name':'ayesha','age':22}
print(copied is user_info)    #return false mean not same
print(copied==user_info)      #return true mean same it chks values
```

```
{'name': 'ayesha', 'age': 22}
False
True
```

- **TODO Task**

Take a number and print a dictionary from 1 to number as keys and values are cube of keys

```
#take a number and print a dictionary from 1 to numbera as keys and values are cube of keys
n=int(input("Enter any number:"))
def cube(num):
    d={}
    for i in range(1,num+1):
        d[i]=i*i*i
    return d
print(f"dictionary with cube of numbers is={cube(n)}")
#output
#dictionary with cube of numbers is={1: 1, 2: 8, 3: 27, 4: 64, 5: 125}
```

```
Enter any number:6
dictionary with cube of numbers is={1: 1, 2: 8, 3: 27, 4: 64, 5: 125, 6: 216}
```

- **TODO Task**

Define a counter function that take a string and counts how many times each character of string is occur

```python
#define a counter function that take a string and counts
#how many times each character of string is occur
def counter(s):
    count={}
    for i in s:
        count[i]=s.count(i)
    return count
print(counter("ayesha"))
#output
#it prints {'a': 2, 'y': 1, 'e': 1, 's': 1, 'h': 1}
#it we donot have to chk if any character occur twice bcz counter automatically override it
```

```
{'a': 2, 'y': 1, 'e': 1, 's': 1, 'h': 1}
```

- **TODO Task**

Take input of dictionary from user and print it

```python
#take input from user and print dictionary in this format
mixed={}
name=input("Enter your name:")
age=input("Enter your age:")
city=input("Enter your city:")
movies=input("Enter your fav movies seprated by comma:").split(",")
song=input("Enter your fav song seprated by comma:").split(",")
mixed['name']=name
mixed['age']=age
mixed['city']=city
mixed['movies']=movies
mixed['song']=song
for i,j in mixed.items():
    print(f"{i}:{j}")
```

```
Enter your name:ayesha
Enter your age:26
Enter your city:lhr
Enter your fav movies seprated by comma:coco,was,saa
Enter your fav song seprated by comma:awaz,zuban,haha
name:ayesha
age:26
city:lhr
movies:['coco', 'was', 'saa']
song:['awaz', 'zuban', 'haha']
```