

Python Basics Assignment 5

1.What does an empty dictionary's code look like?

Ans: An empty dictionary is often represented by two empty curly brackets
`d = { }` or `d = dict()`

```
# Python3 code to demonstrate use of  
# {} symbol to initialize dictionary  
emptyDict = { }
```

```
# print dictionary  
print(emptyDict)
```

```
# print length of dictionary  
print("Length:", len(emptyDict))
```

```
# print type  
print(type(emptyDict))
```

```
{ }  
Length: 0  
<class 'dict'>
```

2.what is the value of dictionary value with key 'foo' and the value 42 ?

Ans: `{'foo':42}`

3.What is the most significant distinction between a dictionary and a list?

Ans: Dictionaries are represented by `{ }` where as listed are represented by `[]`. The Items stored in a dictionary are Unordered , while the items in a list are ordered.

Lists are just like the arrays, declared in other languages. Lists need not be homogeneous always which makes it a most powerful tool in Python. A single list may contain DataTypes like Integers, Strings, as well as Objects. Lists are mutable, and hence, they can be altered even after their creation.

Dictionary in Python on the other hand is an unordered collection of data values, used to store data values like a map, which unlike other Data Types that hold only single value as an element, Dictionary holds key:value pair. Key-value is provided in the dictionary to make it more optimized. Each key-value pair in a Dictionary is separated by a colon :, whereas each key is separated by a 'comma'.

```
#Dictionary  
Dict = { 1: 'Sharva', 2: 'Arra', 3: 'Mihir' }  
print("Dictionary with the use of Integer Keys: ")  
print(Dict)
```

```
#List
```

```
List = ["Sharva", "Nikki", "ABC"]
print("List containing multiple values: ")
print(List[0])
print(List[2])
```

Dictionary with the use of Integer Keys:

```
{1: 'Sharva', 2: 'Arra', 3: 'Mihir'}
```

List containing multiple values:

Sharva

ABC

4.What happens if you try to access spam ['foo'] if spam is {'bar':100} ?

Ans: we will get a keyError KeyError: 'foo'

```
spam = {'bar': 100}
spam ['foo']
```

```
-----
KeyError                                Traceback (most recent call last)
Input In [3], in <cell line: 2>()
      1 spam = {'bar': 100}
----> 2 spam ['foo']
```

KeyError: 'foo'

5.if a dictionary is stored in spam,what is the difference between the expressions 'cat' in spam and 'cat' in spam.keys() ?

Ans: There is no difference . The operator checks whether a value exists as a key in the dictionary or not.

6.if a dictionary is stored in spam,what is the difference between the expressions 'cat' in spam and 'cat' in spam.values() ?

Ans:'cat' in spam checks whether there is a 'cat' key in the dictionary, while 'cat' in spam.values() checks whether there is a value 'cat' for one of the keys in spam.

7.what is a shortcut for the following code ?

```
if 'color' not in spam:    spam['color'] ='black'
```

Ans: spam.setdefault('color','black')

```
if 'color' not in spam: spam['color'] ='black'
```

```
spam.setdefault('color','black')
```

'black'

8.How do you 'pretty print' dictionary values using which modules and function ?

Ans: we can pretty print a dictionary using three functions

1. by using pprint() function of pprint module
 - **Note:** pprint() function doesnot prettify nested dictionaries

2. by using dumps() method of json module
3. by using dumps() method of yaml module

```
ndict = [
    {'Name': 'Sharva', 'Age': '54', 'Residence': {'Country': 'Bangladesh', 'City': 'Dhaka'}},
    {'Name': 'Krish', 'Age': '23', 'Residence': {'Country': 'Spain', 'City': 'Madrid'}},
    {'Name': 'Ara', 'Age': '26', 'Residence': {'Country': 'UK', 'City': 'England'}},
    {'Name': 'Papa', 'Age': '64', 'Residence': {'Country': 'Japan', 'City': 'Osaka'}}
]
```

```
print('Printing using print() function\n',ndict)
print('-'*70)
import pprint
print('Printing using pprint() funciton')
pprint.pprint(ndict)
print('-'*70)
import json
dump = json.dumps(ndict, indent=4)
print('Printing using dumps() method\n', dump)
print('-'*70)
import yaml
dump = yaml.dump(ndict)
print('Printing using dump() method\n', dump)
```

Printing using print() function

```
[{'Name': 'Sharva', 'Age': '54', 'Residence': {'Country': 'Bangladesh', 'City': 'Dhaka'}}, {'Name': 'Krish',
'Age': '23', 'Residence': {'Country': 'Spain', 'City': 'Madrid'}}, {'Name': 'Ara', 'Age': '26', 'Residence':
{'Country': 'UK', 'City': 'England'}}, {'Name': 'Papa', 'Age': '64', 'Residence': {'Country': 'Japan', 'City':
'Osaka'}}]
```

Printing using pprint() funciton

```
[{'Age': '54',
  'Name': 'Sharva',
  'Residence': {'City': 'Dhaka', 'Country': 'Bangladesh'}},
 {'Age': '23',
  'Name': 'Krish',
  'Residence': {'City': 'Madrid', 'Country': 'Spain'}},
 {'Age': '26',
  'Name': 'Ara',
  'Residence': {'City': 'England', 'Country': 'UK'}},
 {'Age': '64',
  'Name': 'Papa',
  'Residence': {'City': 'Osaka', 'Country': 'Japan'}}]
```

Printing using dumps() method

```
[
  {
    "Name": "Sharva",
    "Age": "54",
    "Residence": {
      "Country": "Bangladesh",
      "City": "Dhaka"
```

```
    }  
  },  
  {  
    "Name": "Krish",  
    "Age": "23",  
    "Residence": {  
      "Country": "Spain",  
      "City": "Madrid"  
    }  
  },  
  {  
    "Name": "Ara",  
    "Age": "26",  
    "Residence": {  
      "Country": "UK",  
      "City": "England"  
    }  
  },  
  {  
    "Name": "Papa",  
    "Age": "64",  
    "Residence": {  
      "Country": "Japan",  
      "City": "Osaka"  
    }  
  }  
]  
]
```

Printing using dump() method

```
- Age: '54'  
  Name: Sharva  
  Residence:  
    City: Dhaka  
    Country: Bangladesh  
- Age: '23'  
  Name: Krish  
  Residence:  
    City: Madrid  
    Country: Spain  
- Age: '26'  
  Name: Ara  
  Residence:  
    City: England  
    Country: UK  
- Age: '64'  
  Name: Papa  
  Residence:  
    City: Osaka  
    Country: Japan
```

