Python Dictionaries

Dictionaries are ordered collection of data items. They store multiple items in a single variable. Dictionary items are key-value pairs that are separated by commas and enclosed within curly brackets {}.

Example:

```
info = {'name':'Karan', 'age':19, 'eligible':True}
print(info)
```

Output:

```
{'name': 'Karan', 'age': 19, 'eligible': True}
```

Accessing Dictionary items:

I. Accessing single values:

Values in a dictionary can be accessed using keys. We can access dictionary values by mentioning keys either in square brackets or by using get method.

Example:

```
info = {'name':'Karan', 'age':19, 'eligible':True}
print(info['name'])
print(info.get('eligible'))
```

Output:

```
Karan
True
```

II. Accessing multiple values:

We can print all the values in the dictionary using values() method.

Example:

```
info = {'name':'Karan', 'age':19, 'eligible':True}
print(info.values())
```

Output:

```
dict_values(['Karan', 19, True])
```

III. Accessing keys:

We can print all the keys in the dictionary using keys() method.

Example:

```
info = {'name':'Karan', 'age':19, 'eligible':True}
print(info.keys())
```

Output:

```
dict_keys(['name', 'age', 'eligible'])
```

IV. Accessing key-value pairs:

We can print all the key-value pairs in the dictionary using items() method.

Example:

```
info = {'name':'Karan', 'age':19, 'eligible':True}
print(info.items())
```

Output:

```
dict_items([('name', 'Karan'), ('age', 19), ('eligible', True)])
```

Dictionary Methods

Dictionary uses several built-in methods for manipulation.They are listed below

update()

The update() method updates the value of the key provided to it if the item already exists in the dictionary, else it creates a new key-value pair.

Example:

```
info = {'name':'Karan', 'age':19, 'eligible':True}
print(info)
info.update({'age':20})
info.update({'DOB':2001})
print(info)
```

Output:

```
{'name': 'Karan', 'age': 19, 'eligible': True}
{'name': 'Karan', 'age': 20, 'eligible': True, 'DOB': 2001}
```

Removing items from dictionary:

There are a few methods that we can use to remove items from dictionary.

clear():

The clear() method removes all the items from the list.

Example:

```
info = {'name':'Karan', 'age':19, 'eligible':True}
info.clear()
print(info)
```

Output:

```
{}
```

pop():

The pop() method removes the key-value pair whose key is passed as a parameter.

Example:

```
info = {'name':'Karan', 'age':19, 'eligible':True}
info.pop('eligible')
print(info)
```

Output:

```
{'name': 'Karan', 'age': 19}
```

popitem():

The popitem() method removes the last key-value pair from the dictionary.

Example:

```
info = {'name':'Karan', 'age':19, 'eligible':True, 'DOB':2003}
info.popitem()
print(info)
```

Output:

```
{'name': 'Karan', 'age': 19, 'eligible': True}
```

del:

we can also use the del keyword to remove a dictionary item.

Example:

```
info = {'name':'Karan', 'age':19, 'eligible':True, 'DOB':2003}
del info['age']
print(info)
```

Output:

```
{'name': 'Karan', 'eligible': True, 'DOB': 2003}
```

If key is not provided, then the del keyword will delete the dictionary entirely.

Example:

```
info = {'name':'Karan', 'age':19, 'eligible':True, 'DOB':2003}
del info
print(info)
```

Output:

```
NameError: name 'info' is not defined
```