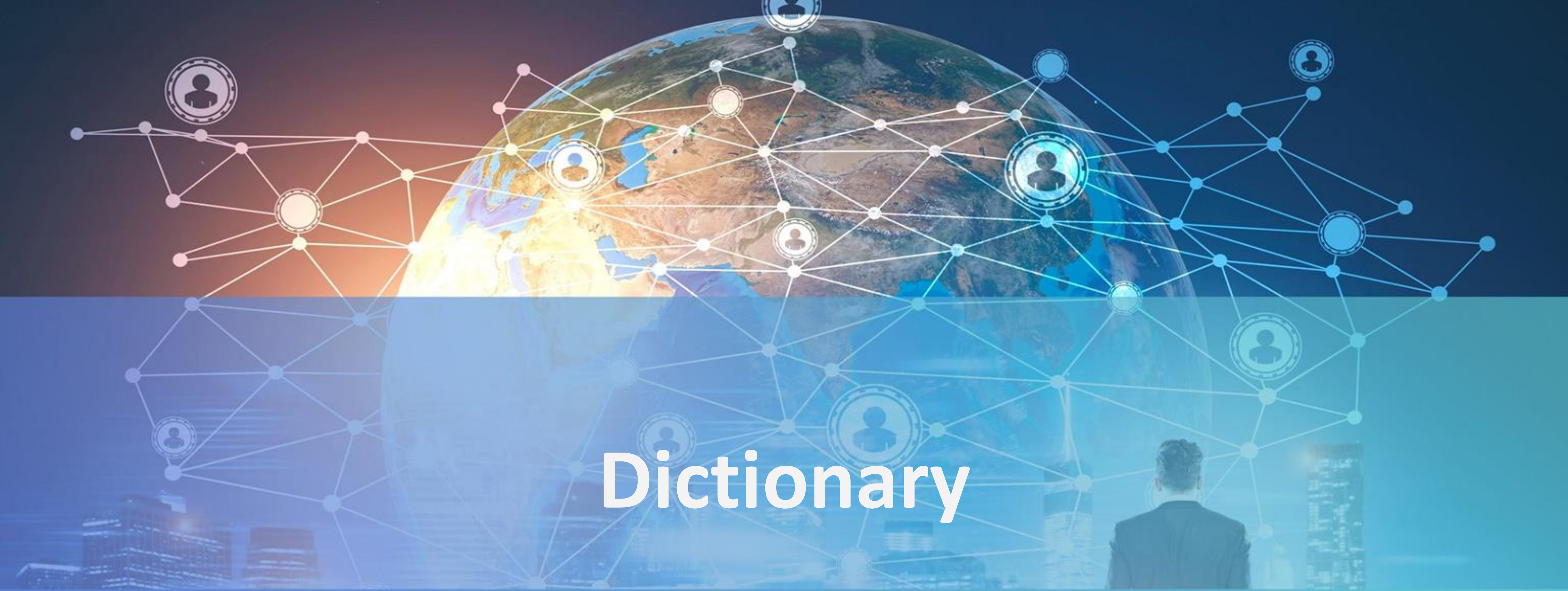


Dictionary

GITAA
Transforming careers

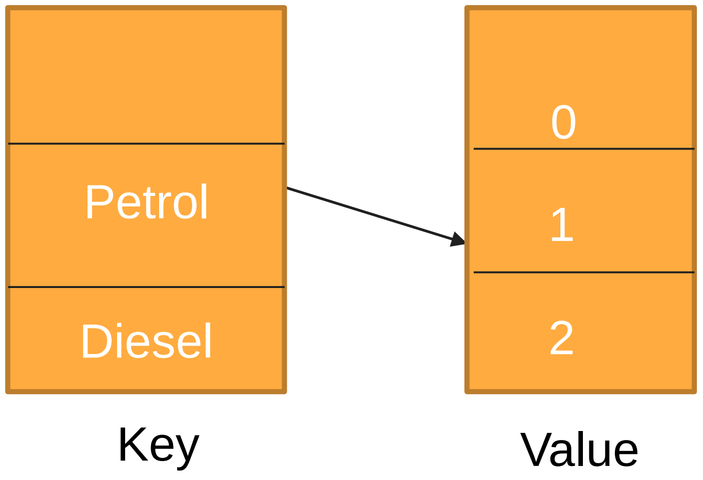
In this lecture

- Dictionary
- Creating a dictionary
- Accessing components
- Modifying components

Dictionary

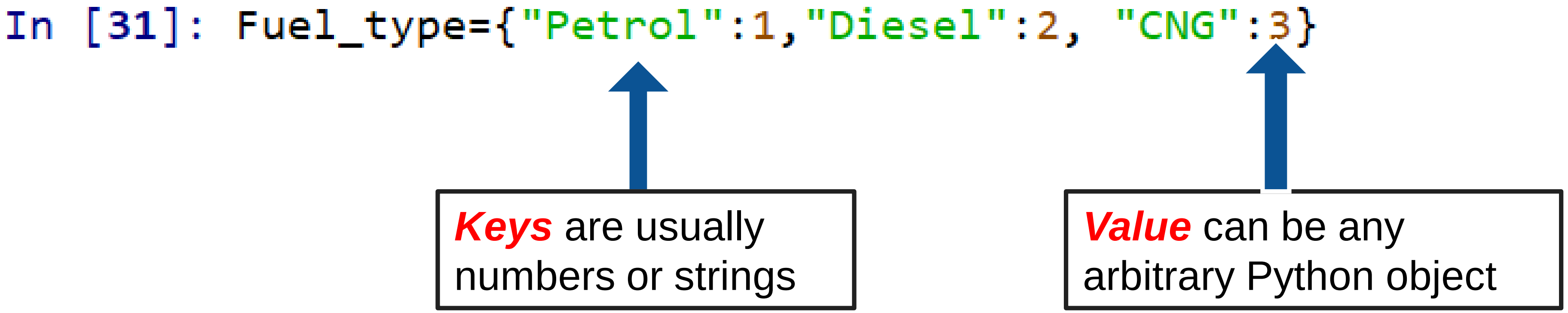
Dictionary

- Python dictionaries is an example of hash-table data structure
- Work like key-value pairs, where the keys are mapped to values
- Dictionaries are enclosed by curly braces { }



Creating a dictionary

- Create the dictionary with different fuel types category

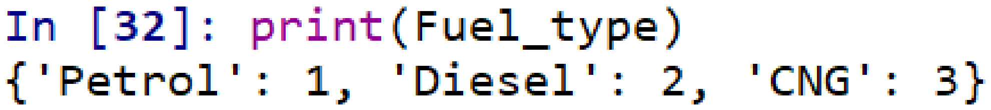


```
In [31]: Fuel_type={"Petrol":1,"Diesel":2, "CNG":3}
```

Keys are usually numbers or strings

Value can be any arbitrary Python object

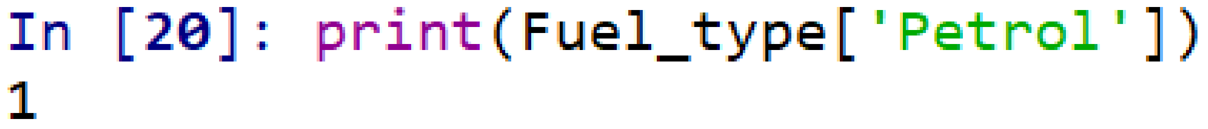
- To print the dictionary



```
In [32]: print(Fuel_type)
{'Petrol': 1, 'Diesel': 2, 'CNG': 3}
```

Accessing components of dictionary

- To know the value of the key ***Petrol*** from the ***Fuel_type***

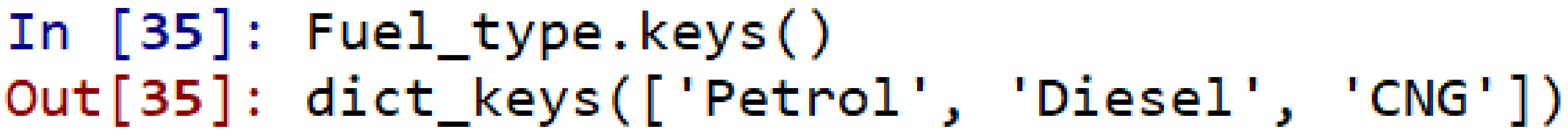


```
In [20]: print(Fuel_type['Petrol'])
1
```

- We can also access remaining values in the above format

Accessing components of dictionary

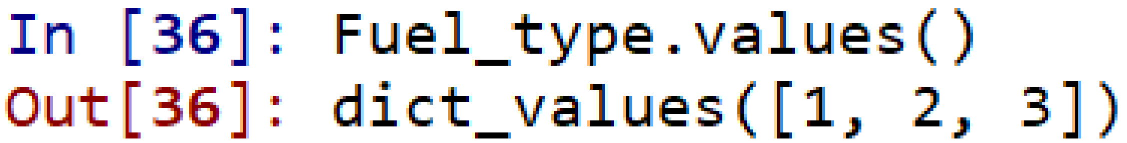
- To access the keys from dictionary ***Fuel_type***
- Syntax: **dictionary_name.keys( )**



```
In [35]: Fuel_type.keys()
Out[35]: dict_keys(['Petrol', 'Diesel', 'CNG'])
```

Accessing components of dictionary

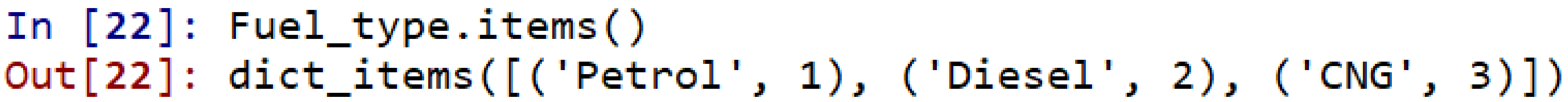
- To access the values from dictionary ***Fuel_type***
- Syntax: `dictionary_name.values( )`



```
In [36]: Fuel_type.values()
Out[36]: dict_values([1, 2, 3])
```

Accessing components of dictionary

- To access both keys and values simultaneously from dictionary ***Fuel_type***
- Syntax: `dictionary_name.items( )`

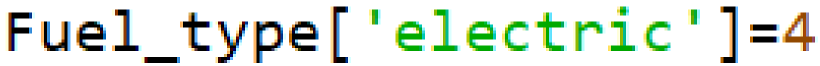


```
In [22]: Fuel_type.items()
Out[22]: dict_items([('Petrol', 1), ('Diesel', 2), ('CNG', 3)])
```

- Returns elements in a list format with (key , value) tuple pairs

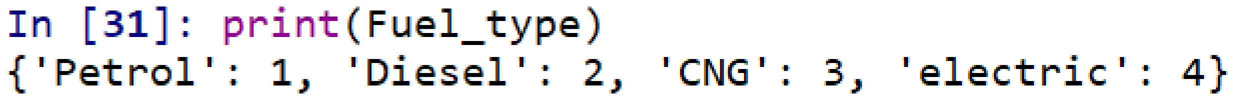
Modifying a dictionary

- Adding new key value pair to the existing dictionary ***Fuel_type*** using keys
- Syntax: `dictionary_name[key]=value`



```
Fuel_type['electric']=4
```

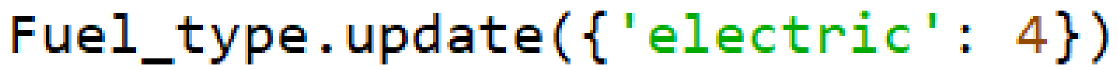
- Print the updated dictionary



```
In [31]: print(Fuel_type)
{'Petrol': 1, 'Diesel': 2, 'CNG': 3, 'electric': 4}
```

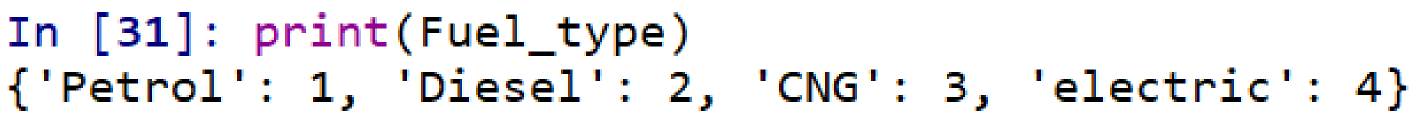
Modifying dictionary using update( )

- Adding a new key value pair to the existing dictionary ***Fuel_type*** using **update()** function
- Syntax: **dictionary_name.update({key:value})**



```
Fuel_type.update({'electric': 4})
```

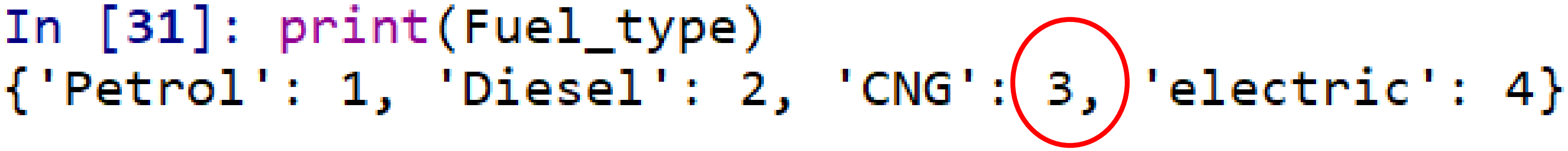
- Print the updated dictionary



```
In [31]: print(Fuel_type)
{'Petrol': 1, 'Diesel': 2, 'CNG': 3, 'electric': 4}
```

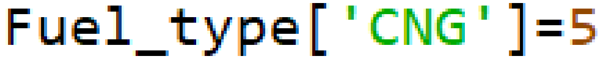
Modifying dictionary

- Modify the value of an existing key
- Assign value to be changed to corresponding key of the dictionary



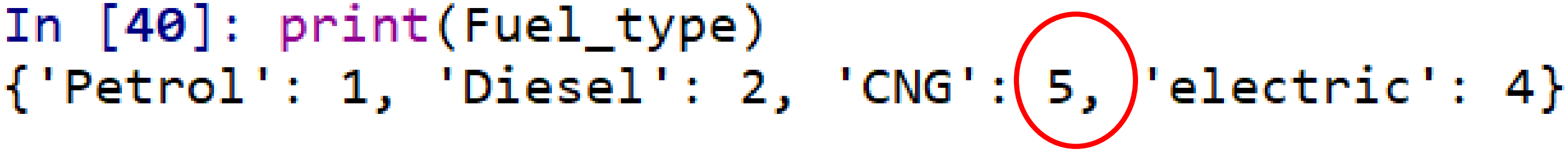
```
In [31]: print(Fuel_type)
{'Petrol': 1, 'Diesel': 2, 'CNG': 3, 'electric': 4}
```

- Here the value of 3 should be updated to 5



```
Fuel_type['CNG']=5
```

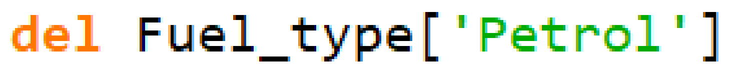
- Print the updated dictionary



```
In [40]: print(Fuel_type)
{'Petrol': 1, 'Diesel': 2, 'CNG': 5, 'electric': 4}
```

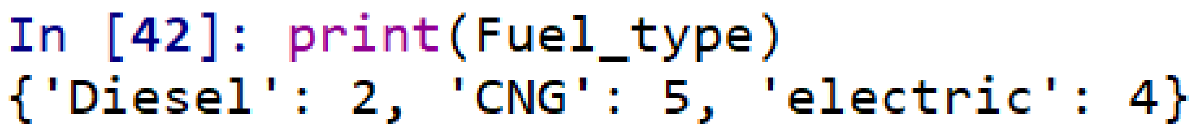
Modifying dictionary using del( )

- del- removes the key value pairs
- Syntax: del dictionary_name[key]
- Dropping the key ***Petrol*** from ***Fuel_type***



```
del Fuel_type['Petrol']
```

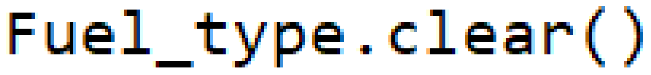
- Print the updated dictionary



```
In [42]: print(Fuel_type)
{'Diesel': 2, 'CNG': 5, 'electric': 4}
```

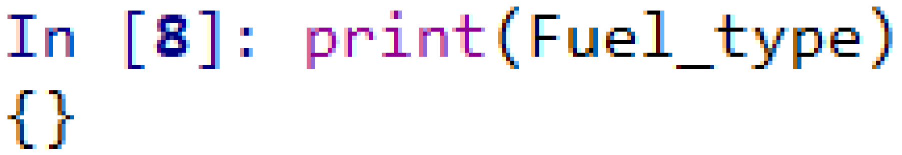
Modifying dictionary using clear ( )

- `clear()` -   removes all the key value pairs
- Syntax: `dictionary_name.clear()`



```
Fuel_type.clear()
```

- Print the updated dictionary



```
In [8]: print(Fuel_type)
{}
```

Summary

- Create dictionary
- Accessing components
- Modifying dictionary

THANK YOU