Dictionaries

Creation of New Dictionary

➢ To create a new dictionary in Python, you can use curly braces {} and specify key-value pairs within them. For example:

➢
```python
my_dict = {'key1': 'value1', 'key2': 'value2'}
```

Accessing Items in the Dictionary\

➢ You can access values in a dictionary by referencing the key associated with the value. For example:

➢
```python
value = my_dict['key1']
```

Change Values in the Dictionary

➢ To change the value of a specific key in a dictionary, you can simply assign a new value to that key. For example:

➢
```python
my_dict['key1'] = 'new_value'
```

Loop Through a Dictionary Values

➢ You can loop through the values of a dictionary using a for loop. For example:

➢
```python
for value in my_dict.values():
    print(value)
```

Check if Key Exists in the Dictionary

➢ You can check if a key exists in a dictionary using the in keyword. For example:

➢
```python
if 'key1' in my_dict:
    print('Key exists!')
```

Checking for Dictionary Length

➢ You can find the length of a dictionary using the len() function. For example:

➢
```python
length = len(my_dict)
```

Adding Items in the Dictionary

➢ To add a new key-value pair to a dictionary, you can simply assign a value to a new key. For example:

➢
```python
my_dict['new_key'] = 'new_value'
```

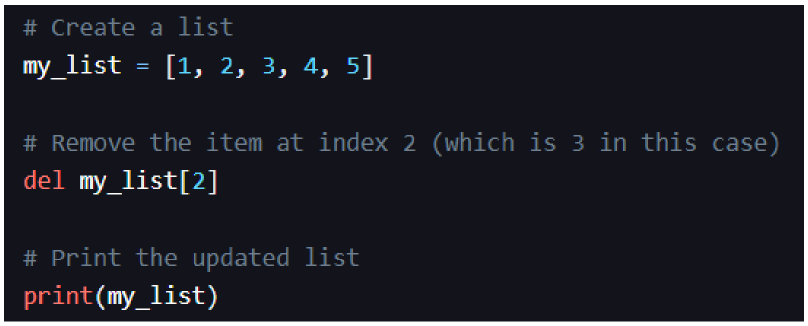
Removing Items in the Dictionary

➢ You can remove items from a dictionary using various methods like pop(), popitem(), or clear(). For example:

➢
```python
my_dict.pop('key1')
```

Remove an Item Using del Statement

➢ In Python, you can remove an item from a list using the del statement. The del statement removes the item at a specific index from the list. Here is an example of how you can use the del statement to remove an item from a list:

➢ 

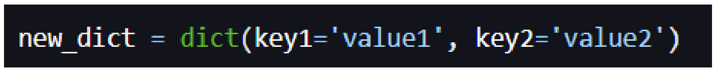
```python
# Create a list
my_list = [1, 2, 3, 4, 5]

# Remove the item at index 2 (which is 3 in this case)
del my_list[2]

# Print the updated list
print(my_list)
```

The dict() Constructor

➢ You can create a dictionary using the dict() constructor by passing key-value pairs as arguments. For example:

➢ 

```python
new_dict = dict(key1='value1', key2='value2')
```

Dictionary Methods

➢ Python dictionaries have built-in methods like keys(), values(), and items() for accessing keys, values, and key-value pairs respectively.

Jupyter notebook

Adding Folder

➢ You can add a folder in Jupyter Notebook by creating a new directory within the Jupyter environment.

Adding Text file

➢ To add a text file in Jupyter Notebook, you can use the Jupyter interface to upload or create a new text file

CSV file for data analysis and visualization

➢ CSV files are commonly used for data analysis and visualization in Jupyter Notebook. You can import CSV files using libraries like pandas and perform data analysis.

Import libraries

➢ In Jupyter Notebook, you can import libraries like pandas, numpy, matplotlib, and seaborn for data analysis and visualization.

Finding data

➢ You can find data for analysis from various sources like online datasets, APIs, or by creating your own datasets.

Importing data

➢ To import data into Jupyter Notebook, you can use libraries like pandas to read data from CSV files, Excel files, databases, or web sources.

Data attributes

➢ Data attributes in Jupyter Notebook refer to the characteristics and properties of the dataset being analyzed, such as columns, rows, data types, and values.