Name: Akula Sharathchandra

Batch: Data Engineering

Date:31/01/2024

TOPICS: JSON

JSON:JSON stands for JavaScript Object Notation. It means that a script (executable) file which is made of text in a programming language, is used to store and transfer the data. Python supports JSON through a built-in package called JSON. To use this feature, we import the python JSON package into a Python script. The text in JSON is done through quoted-string which contains a value in key-value mapping within { }. It is similar to the dictionary in python.

JSON Syntax: Data is in name/value pairs and they are separated by commas. It uses curly brackets to hold the objects and square brackets to hold the arrays.

Function Used:

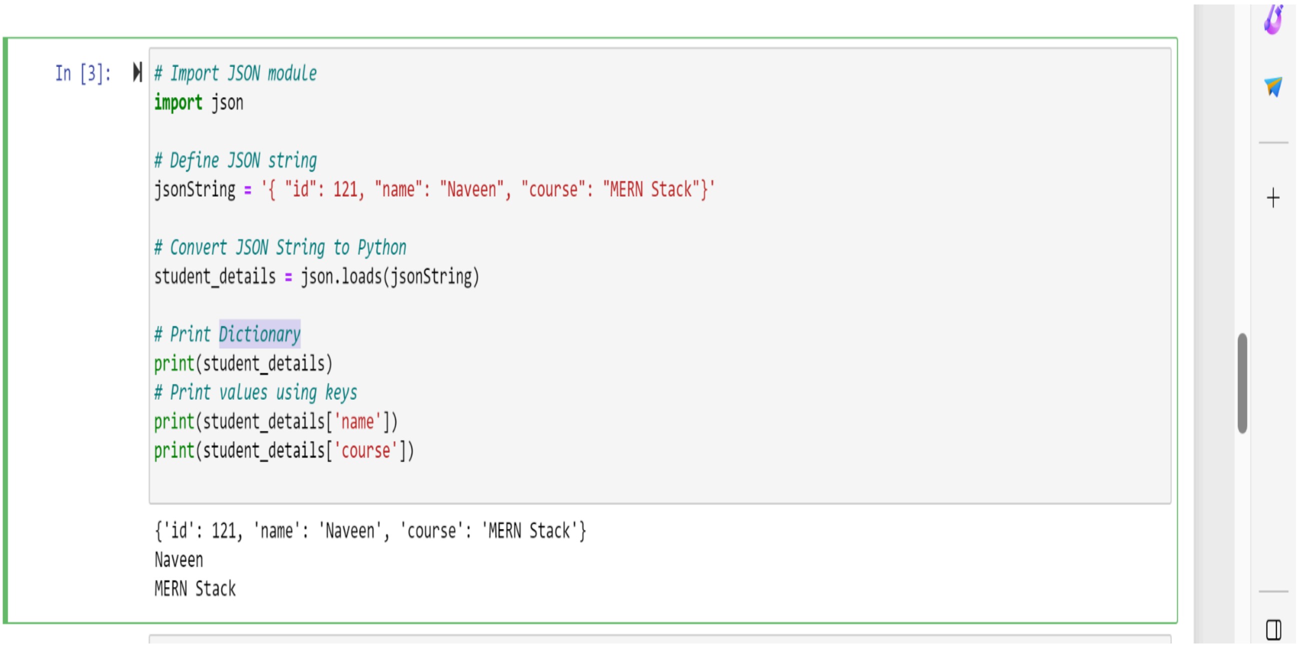
json.load():json.load() function is present in Python built-in 'JSON' module. This function is used to parse the JSON string.

json.loads():json.loads() function is present in Python built-in 'json' module. This function is used to parse the JSON string.

json.dump():The json.dump() function is used to serialize a Python object (such as a dictionary) into a JSON formatted string and write it to a file. It's particularly useful when you want to save Python data to a file in JSON format.

Example1:

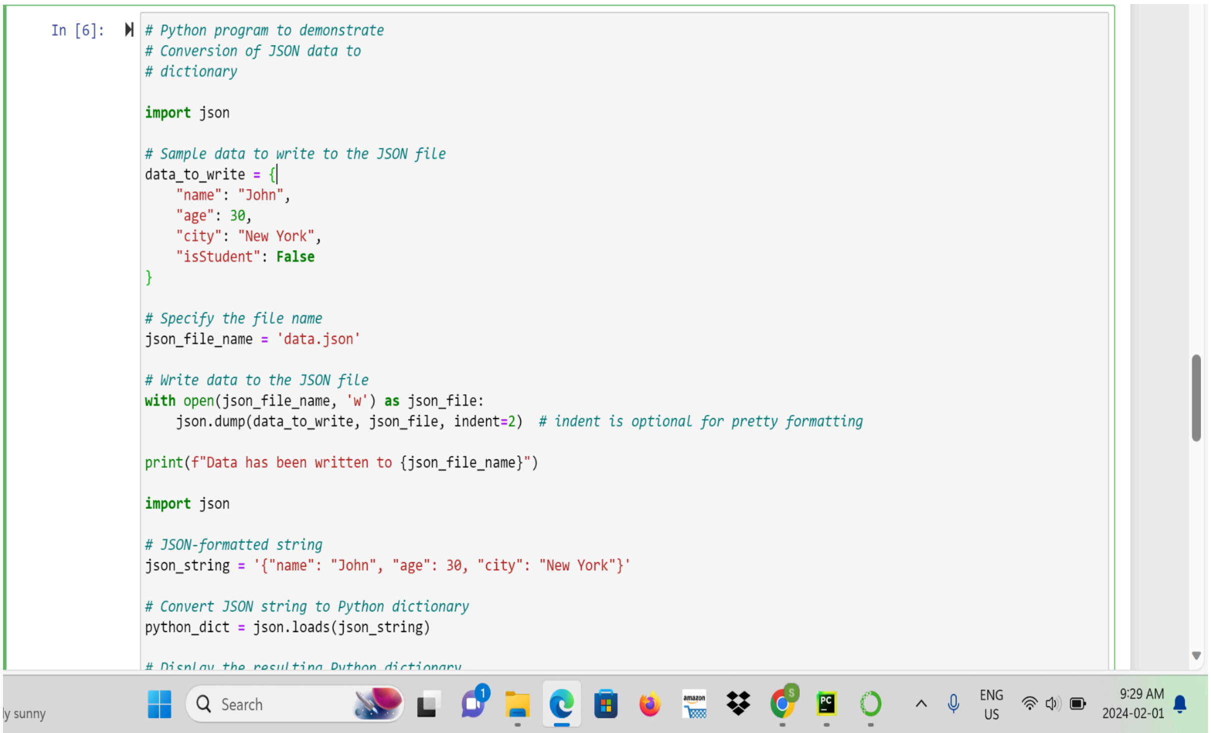
How to import , define , convert JSON string to python and print Dictionary.



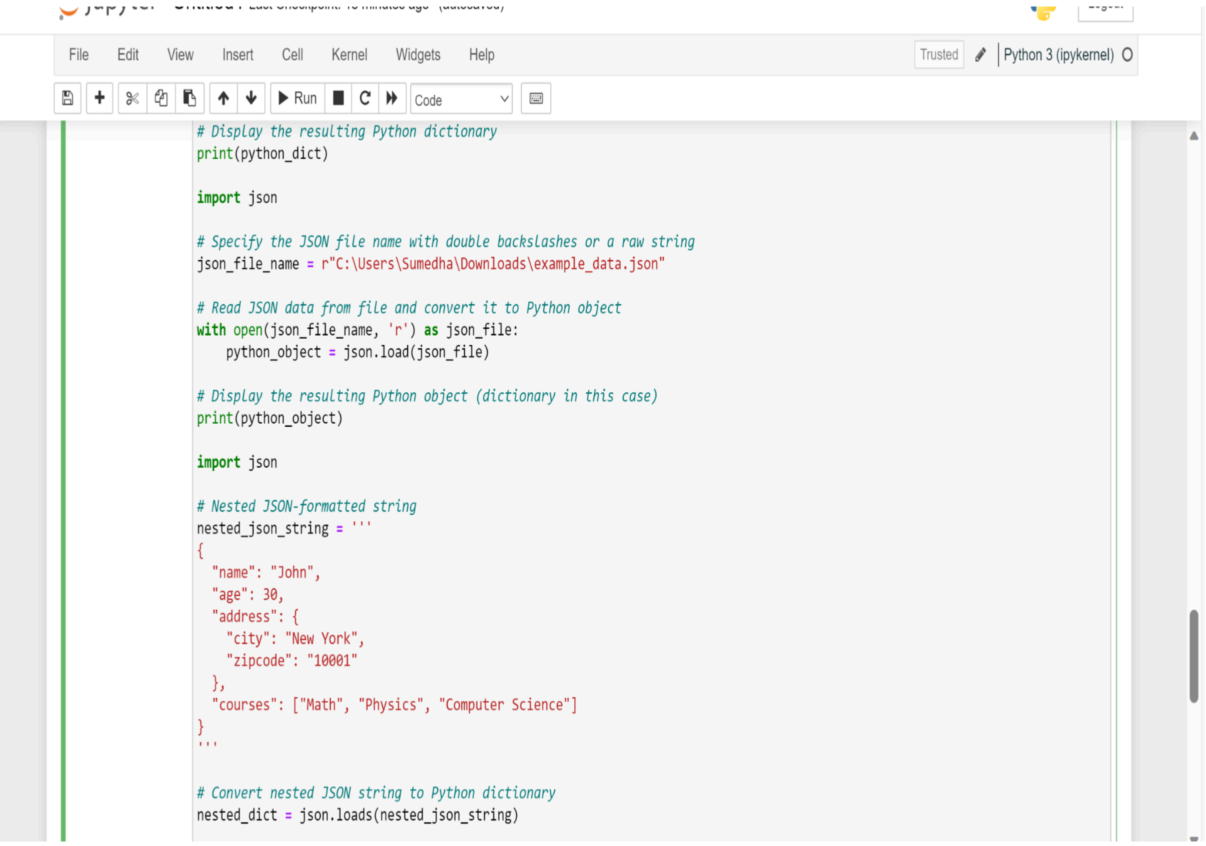
```
In [3]:
# Import JSON module
import json

# Define JSON string
jsonString = '{ "id": 121, "name": "Naveen", "course": "MERN Stack"}'

# Convert JSON String to Python
student_details = json.loads(jsonString)

# Print Dictionary
print(student_details)
# Print values using keys
print(student_details['name'])
print(student_details['course'])
```

```
{'id': 121, 'name': 'Naveen', 'course': 'MERN Stack'}
Naveen
MERN Stack
```

Example 2: Python program to demonstrate Conversion of JSON data to dictionary

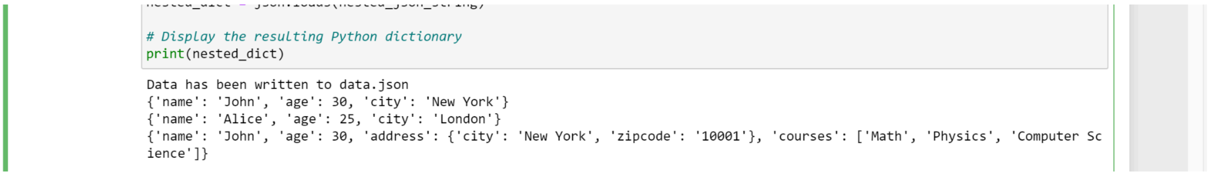


```python
# Python program to demonstrate
# Conversion of JSON data to
# dictionary

import json

# Sample data to write to the JSON file
data_to_write = {
    "name": "John",
    "age": 30,
    "city": "New York",
    "isStudent": False
}

# Specify the file name
json_file_name = 'data.json'

# Write data to the JSON file
with open(json_file_name, 'w') as json_file:
    json.dump(data_to_write, json_file, indent=2)  # indent is optional for pretty formatting

print(f"Data has been written to {json_file_name}")

import json

# JSON-formatted string
json_string = '{"name": "John", "age": 30, "city": "New York"}'

# Convert JSON string to Python dictionary
python_dict = json.loads(json_string)

# Display the resulting Python dictionary
```



```python
# Display the resulting Python dictionary
print(python_dict)

import json

# Specify the JSON file name with double backslashes or a raw string
json_file_name = r"C:\Users\Sumedha\Downloads\example_data.json"

# Read JSON data from file and convert it to Python object
with open(json_file_name, 'r') as json_file:
    python_object = json.load(json_file)

# Display the resulting Python object (dictionary in this case)
print(python_object)

import json

# Nested JSON-formatted string
nested_json_string = '''
{
  "name": "John",
  "age": 30,
  "address": {
    "city": "New York",
    "zipcode": "10001"
  },
  "courses": ["Math", "Physics", "Computer Science"]
}
'''

# Convert nested JSON string to Python dictionary
nested_dict = json.loads(nested_json_string)
```

OUTPUT:



```
nested_dict = json.loads(nested_json_string)

# Display the resulting Python dictionary
print(nested_dict)
```

```
Data has been written to data.json
{'name': 'John', 'age': 30, 'city': 'New York'}
{'name': 'Alice', 'age': 25, 'city': 'London'}
{'name': 'John', 'age': 30, 'address': {'city': 'New York', 'zipcode': '10001'}, 'courses': ['Math', 'Physics', 'Computer Sc
ience']}
```

SUMMARY:

i)Writing JSON Data to a File:

- The code starts by creating a Python dictionary `data_to_write` representing some sample data.
- It then writes this data to a JSON file named "data.json" using json.dump()
- The indent=2 parameter is used to add indentation for better readability in the JSON file.

ii)Converting JSON String to Python Dictionary:

- The code provides a JSON-formatted string json_string.
- It uses json.loads()to convert this JSON string into a Python dictionary.

iii)Reading JSON Data from a File:

- The code specifies the file path to a JSON file named "example_data.json" using a raw string
- It reads the JSON data from the file using json.load() and converts it into a Python object .

iv)Nested JSON to Python Dictionary:

- The code provides a nested JSON-formatted string nested_json_string
- It uses json.loads() to convert this nested JSON string into a Python dictionary.