

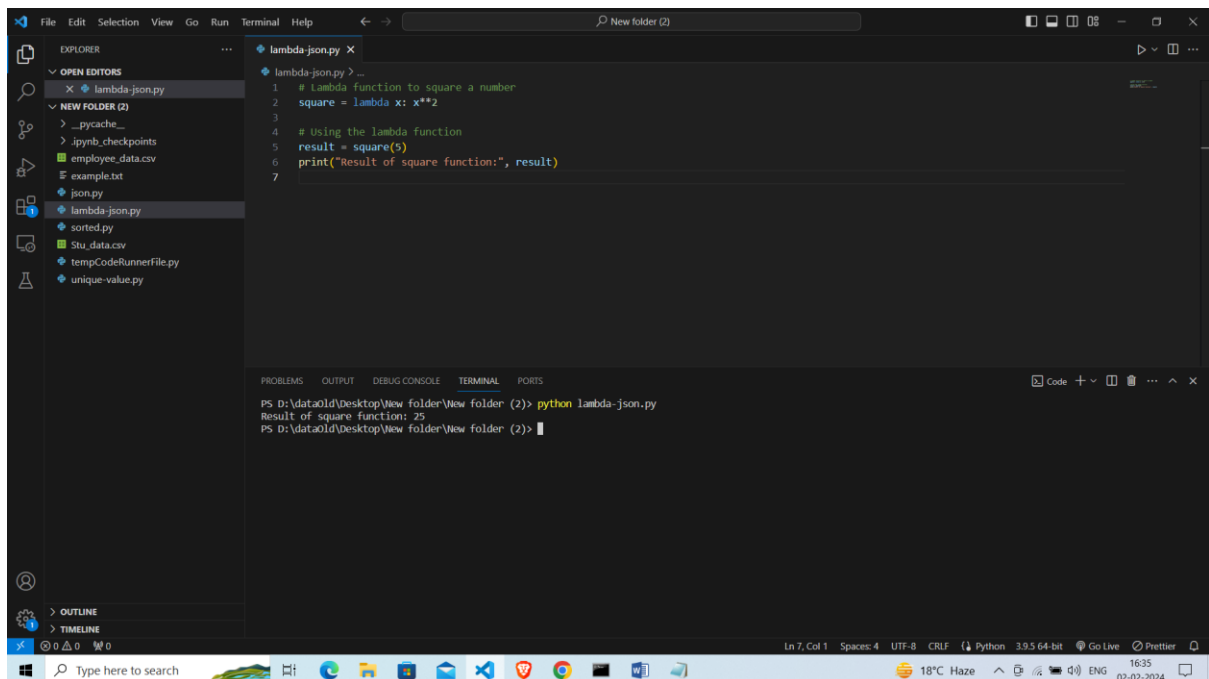
# Python Coding Challenge

## Q.2 Execute with one example Lambda Functions in Python&Read JSON Strings to Python dicts or lists

Ans.

### Example of Lambda Function in Python:

A lambda function is a small, anonymous function defined using the lambda keyword. It's often used for short, simple operations. Here's a simple example:

A screenshot of the Visual Studio Code (VS Code) editor interface. The Explorer sidebar on the left shows a project structure with files like 'lambda-json.py', 'sorted.py', 'Stu\_data.csv', 'tempCodeRunnerFile.py', and 'unique-value.py'. The main editor window is open to 'lambda-json.py', which contains the following Python code:

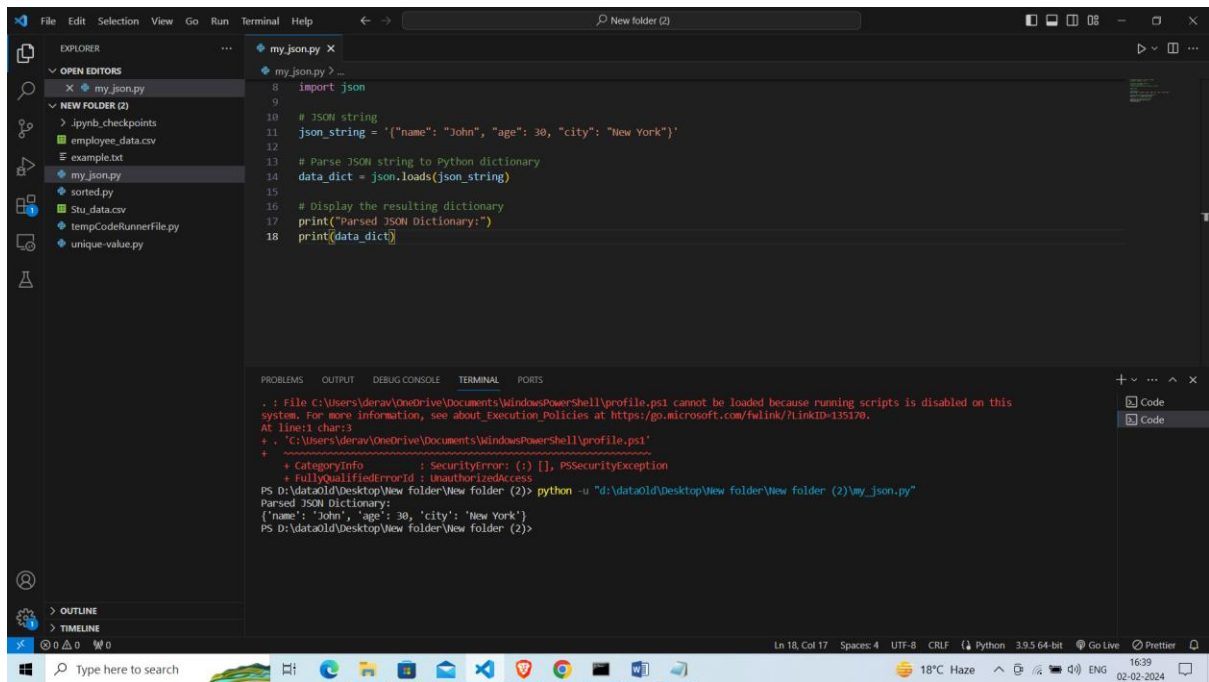
```
1 # Lambda function to square a number
2 square = lambda x: x**2
3
4 # Using the lambda function
5 result = square(5)
6 print("Result of square function:", result)
7
```

At the bottom, the Terminal panel shows the command 'python lambda-json.py' being executed, with the output 'Result of square function: 25'. The status bar at the very bottom indicates the current line and column (Ln 7, Col 1), file encoding (UTF-8), and other settings.

In this example, we define a lambda function square that takes an argument x and returns its square. The lambda function is then used to calculate the square of 5, and the result is printed.

### Read JSON Strings to Python Dicts or Lists:

Python provides the json module to work with JSON data. You can use the json.loads() function to parse JSON strings into Python dictionaries or lists.



The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left lists files in a project named 'New folder (2)', including 'my\_json.py'. The main editor window displays the content of 'my\_json.py', which is a Python script that imports the 'json' module, defines a JSON string, and uses 'json.loads()' to parse it into a dictionary. The script also prints the resulting dictionary. Below the editor, the TERMINAL panel shows the command 'python -u "d:\dataOld\Desktop\New folder\New folder (2)\my\_json.py"' being executed. The output shows a 'SecurityError' and 'UnauthorizedAccess' message, followed by the printed dictionary: {'name': 'John', 'age': 30, 'city': 'New York'}.

```
File Edit Selection View Go Run Terminal Help
my_json.py X
my_json.py > ...
0 import json
1
2 # JSON string
3 json_string = '{"name": "John", "age": 30, "city": "New York"}'
4
5 # Parse JSON string to Python dictionary
6 data_dict = json.loads(json_string)
7
8 # Display the resulting dictionary
9 print("Parsed JSON Dictionary:")
10 print(data_dict)
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
. : File C:\Users\derav\OneDrive\Documents\WindowsPowerShell\profile.ps1 cannot be loaded because running scripts is disabled on this
system. For more information, see about_Execution_Policies at https://go.microsoft.com/fwlink/?LinkID=135170.
At line:1 char:3
+ . 'C:\Users\derav\OneDrive\Documents\WindowsPowerShell\profile.ps1'
+ ~~~~~
+ CategoryInfo          : SecurityError: (:) [], PSecurityException
+ FullyQualifiedErrorId : UnauthorizedAccess
PS D:\dataOld\Desktop\New folder\New folder (2)> python -u "d:\dataOld\Desktop\New folder\New folder (2)\my_json.py"
Parsed JSON Dictionary:
{'name': 'John', 'age': 30, 'city': 'New York'}
PS D:\dataOld\Desktop\New folder\New folder (2)>
```

Ln 18, Col 17 Spaces: 4 UTF-8 CRLF Python 3.9.5 64-bit Go Live Prettier

In this example, the `json_string` is a JSON-formatted string representing an object with `name`, `age`, and `city`. The `json.loads()` function is used to convert this string into a Python dictionary (`data_dict`). You can also parse JSON arrays into Python lists using the same function.