Lesson 04 Demo 01

Creating, Validating, and Parsing a YAML File Using Python

Objective: To create, validate, and parse a YAML file using Python for managing configurations or data serialization tasks, ensuring the file is read and validated correctly

Tools required: Linux terminal

Prerequisites: None

Steps to be followed:

1. Create a YAML file using Python
2. Run the Python file

Step 1: Create a YAML file using Python

1.1 Run the following command to create a Python file:

sudo nano yaml_demo.py

```
poojahksimplile@ip-172-31-36-118:~$ sudo nano yaml_demo.py
```

1.2 Add the following code to **yaml_demo.py** to create, validate, and parse a YAML file:

```
import yaml

# Step 1: Create a YAML filed
def create_yaml():
    data = {
        'name': 'Josh',
        'age': 36,
        'city': 'Springfield'
    }

    with open('data.yaml', 'w') as file:
        yaml.dump(data, file)

# Step 2: Validate the YAML file
def validate_yaml():
    try:
        with open('data.yaml', 'r') as file:
```

```
            data = yaml.safe_load(file)
        print("YAML file is valid.")
        return data
    except yaml.YAMLError as exc:
        print("Error in YAML file:", exc)
        return None

# Step 3: Parse the YAML file
def parse_yaml(data):
    if data:
        print("Name:", data['name'])
        print("Age:", data['age'])
        print("City:", data['city'])

if __name__ == "__main__":
    create_yaml()
    data = validate_yaml()
    parse_yaml(data)
```

```
  GNU nano 6.2                          yaml_demo.py *
import yaml

# Step 1: Create a YAML file
def create_yaml():
    data = {
        'name': 'Josh',
        'age': 36,
        'city': 'Springfield'
    }

    with open('data.yaml', 'w') as file:
        yaml.dump(data, file)

# Step 2: Validate the YAML file
def validate_yaml():
    try:
        with open('data.yaml', 'r') as file:
            data = yaml.safe_load(file)
        print("YAML file is valid.")
        return data
```

Step 2: Run the Python file

2.1 Use the following command to run the Python file:

sudo python yaml_demo.py

```
poojahksimplile@ip-172-31-36-118:~$ sudo python yaml_demo.py
YAML file is valid.
Name: Josh
Age: 36
City: Springfield
poojahksimplile@ip-172-31-36-118:~$
```

By following these steps, you have successfully created, validated, and parsed a YAML file using Python, ensuring both file reading and validation for configuration management or data serialization tasks.