PYTHON IS FOR EVERYONE

Tutorial 12:

PYTHON PROGRAMMING - MODULES AND LIBRARIES IN GOOGLE COLAB

Jeff Gentry

@www.linkedin.com/in/jefferycharlesgentry

Objectives

- Understand what modules and libraries are in Python.
- Learn how to import and use modules.
- Explore some commonly used standard libraries.
- Practice using modules through hands-on exercises.

What are Modules and Libraries?

- **Module: A module is a file containing Python code (functions, classes, variables) that can be reused in other Python programs. Modules help organize code and promote code reuse.**
- **Library: A library is a collection of modules. Libraries provide a set of functionalities that can be used to perform specific tasks.**

Importing Modules

You can import a module using the import statement. There are several ways to import modules:

Importing the entire module

Tutorial_12.ipynb

File Edit View Insert Runtime Tools Help

+ Code + Text

0s

```
import math
print(math.sqrt(16))  # Output: 4.0
```

```
4.0
```

Importing Modules

Importing specific functions or classes from a module

```
from math import sqrt, pi
print(sqrt(25))  # Output: 5.0
print(pi)        # Output: 3.141592653589793
```

```
5.0
3.141592653589793
```

Importing Modules

Importing a module with an alias

```
import numpy as np
print(np.array([1, 2, 3]))  # Output: [1 2 3]
```

```
[1 2 3]
```

Exploring Commonly Used Standard Libraries

Python comes with a rich set of standard libraries that provide various functionalities. Here are a few commonly used libraries:

“math”: Provides mathematical functions.

```
import math
print(math.factorial(5))  # Output: 120
```

```
120
```

Standard Libraries

“random”: Used for generating random numbers.

```
import random
print(random.randint(1, 10))  # Output: Random integer between 1 and 10
```

```
2
```

Standard Libraries

“datetime”: Used for manipulating dates and times.

```
from datetime import datetime
now = datetime.now()
print(now)  # Output: Current date and time
```

```
2024-12-26 19:54:47.958126
```

Standard Libraries

“os”: Provides functions for interacting with the operating system.

```python
import os
print(os.getcwd())  # Output: Current working directory
```

```
/content
```

Standard Libraries

“sys”: Provides access to system-specific parameters and functions.

```
import sys
print(sys.version)  # Output: Python version
```

```
3.10.12 (main, Nov  6 2024, 20:22:13) [GCC 11.4.0]
```

Creating Your Own Module

You can create your own module by saving a Python file with a ".py" extension. For example, create a file named "mymodule.py" with the following content:

```
# Define the contents of the file
file_contents = """def greet(name):
    return f"Hello, {name}!"

def add(a, b):
    return a + b
"""

# Create and write to the file
with open("mymodule.py", "w") as file:
    file.write(file_contents)
```

Creating Your Own Module

You can then import and use this module in another Python file:

```
import mymodule

print(mymodule.greet("Jeff"))  # Output: Hello, Jeff!
print(mymodule.add(5, 3))        # Output: 8
```

```
Hello, Jeff!
8
```

Practice Exercises

Using the “math” Module: Write a program that calculates the area of a circle using the “math” module.

```
import math

def area_of_circle(radius):
    return math.pi * (radius ** 2)

print(area_of_circle(5))  # Output: Area of the circle
```

```
78.53981633974483
```

Practice Exercises

Random Number Generator: Write a program that generates a random float between 0 and 1 using the “random” module.

```python
import random

random_float = random.random()
print(random_float)  # Output: Random float between 0 and 1
```

```
0.32584819337545934
```

Practice Exercises

Current Date and Time: Write a program that prints the current date and time using the "datetime" module.

```
from datetime import datetime

now = datetime.now()
print("Current date and time:", now)
```

```
Current date and time: 2024-12-26 20:08:49.640449
```

Practice Exercises

List Files in a Directory: Write a program that lists all files in the current directory using the os module.

```
import os

files = os.listdir('.')
print("Files in current directory:", files)
```

```
Files in current directory: ['.config', 'mymodule.py', 'mymodul.py', '__pycache__', 'sample_data']
```

Practice Exercises

System Information: Write a program that prints the Python version and the platform

```python import sys
```

```
import sys

# Print the Python version
print("Python version:", sys.version)

# Print the platform
print("Platform:", sys.platform)
```

```
Python version: 3.10.12 (main, Nov  6 2024, 20:22:13) [GCC 11.4.0]
Platform: linux
```

Conclusion

In this tutorial, you learned about modules and libraries in Python, how to import and use them, and explored some commonly used standard libraries. You also practiced creating your own module and using it in your programs. Understanding modules and libraries is essential for writing efficient and organized Python code.

Next Steps

In tutorial 13, we will cover the fundamental concepts of OOP (Object-Oriented Programming), including classes, objects, inheritance, encapsulation and polymorphism.

FOLLOW ME

for more tips you didn’t know you needed

Jeff Gentry

@www.linkedin.com/in/jefferycharlesgentry