

Lecture10

October 10, 2024

```
[1]: import os

print(os.name)
print(os.getcwd())
os.mkdir("new_directory")
```

```
nt
c:\Users\ALIF\Documents\Python67\Lecture10
```

```
[3]: import datetime

now = datetime.datetime.now()
print(now)

date = datetime.date(2024, 1, 1)
print(date)
```

```
2024-09-16 13:16:57.343547
2024-01-01
```

```
[1]: import json

date = {"name": "Alice", "age": 25}
json_str = json.dumps(date)
print(json_str)

parsed_date = json.loads(json_str)
print(parsed_date)
```

```
{"name": "Alice", "age": 25}
{'name': 'Alice', 'age': 25}
```

```
[2]: import numpy as np

random_matrix = np.random.randint(1, 11, size=(3, 3))
print("Random 3x3 matrix:\n", random_matrix)

matrix_sum = np.sum(random_matrix)
print(f"\nSum of matrix elements: {matrix_sum}")
```

```

matrix_mean = np.mean(random_matrix)
print(f"\nMean of matrix elements: {matrix_mean:.2f}")

transport_matrix = np.transpose(random_matrix)
print("\nTransport matrix:\n", transport_matrix)

```

Random 3x3 matrix:

```

[[9 3 1]
 [9 5 3]
 [7 8 6]]

```

Sum of matrix elements: 51

Mean of matrix elements: 5.67

Transport matrix:

```

[[9 9 7]
 [3 5 8]
 [1 3 6]]

```

```

[3]: import pandas as pd

data = {'Name': ['Alice', 'Bob', 'Charlie'],
        'Age': [25, 30, 35],
        'City': ['New York', 'Los Angeles', 'Chicago']}

df = pd.DataFrame(data)
print("DataFrame:\n", df)

average_age = df['Age'].mean()
print("\nAverage Age:", average_age)

filtered_df = df[df['Age'] > 28]
print("\nFiltered DataFrame (Age > 20):\n", filtered_df)

df['Salary'] = [50000, 60000, 70000]
print("\nDataFrame with Salary column:\n", df)

```

DataFrame:

	Name	Age	City
0	Alice	25	New York
1	Bob	30	Los Angeles
2	Charlie	35	Chicago

Average Age: 30.0

Filtered DataFrame (Age > 20):

	Name	Age	City
1	Bob	30	Los Angeles
2	Charlie	35	Chicago

DataFrame with Salary column:

	Name	Age	City	Salary
0	Alice	25	New York	50000
1	Bob	30	Los Angeles	60000
2	Charlie	35	Chicago	70000

```
[9]: import matplotlib.pyplot as plt
import numpy as np
import matplotlib.animation as animation

fig, ax = plt.subplots()

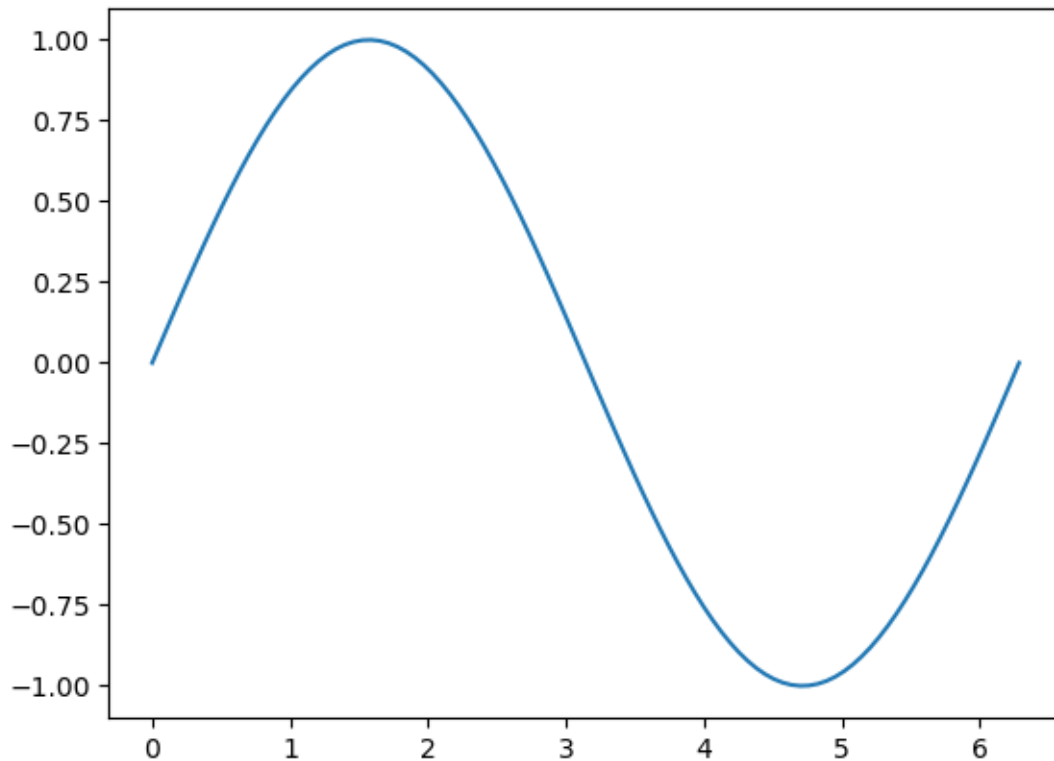
x = np.linspace(0, 2*np.pi, 100)
y = np.sin(x)

line, = ax.plot(x, y)

def update(frame):
    line.set_ydata(np.sin(x + frame/10.0))
    return line,

ani = animation.FuncAnimation(fig, update, frames=100, interval=50, blit=True)

plt.show()
```



```
[1]: import requests

response = requests.get("https://api.github.com/users/octocat")

if response.status_code == 200:
    user_data = response.json()

    print(f"Username: {user_data['login']}")
    print(f"Name: {user_data['name']}")
    print(f"Bio: {user_data['bio']}")
    print(f"Public Repos: {user_data['public_repos']}")
    print(f"Followers: {user_data['followers']}")
    print(f"Fowllowing: {user_data['following']}")
else:
    print("Failed to retrieve data.")
```

```
Username: octocat
Name: The Octocat
Bio: None
Public Repos: 8
Followers: 15294
Fowllowing: 9
```

```
[ ]: from flask import Flask

app = Flask(__name__)

@app.route('/')
def home():
    return "Hello, Flask!"

if __name__ == '__main__':
    app.run(debug=True)
```

```
[ ]: from flask import Flask, render_template_string

app = Flask(__name__)

html_template = """
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Greeting Page</title>
</head>
<body>
    <h1>Hello, {{name}}!</h1>
    <p>Welcome to your simple Flask web app.</p>
</body>
</html>
"""

@app.route('/')
def home():
    return render_template_string(html_template, name='Alice')

@app.route('/greeter/<name>')
def greeter(name):
    return render_template_string(html_template, name=name)

if __name__ == '__main__':
    app.run(debug=True)
```

```
[5]: import sqlite3

conn = sqlite3.connect('mydatabase.db')

cur = conn.cursor()
```

```

cur.execute('''
    CREATE TABLE IF NOT EXISTS users (
        id INTEGER PRIMARY KEY AUTOINCREMENT,
        name TEXT NOT NULL,
        age INTEGER NOT NULL,
        city TEXT NOT NULL
    )
''')

cur.execute('''INSERT INTO users (name, age, city) VALUES ('Alice', 25, 'New_
↳York')''')
cur.execute('''INSERT INTO users (name, age, city) VALUES ('Bob', 30, 'Los_
↳Angeles')''')
cur.execute('''INSERT INTO users (name, age, city) VALUES ('Charlie', 35, _
↳Chicago')''')

conn.commit()

cur.execute('''SELECT * FROM users WHERE age > 28''')
rows = cur.fetchall()

print("Users older than 28:")
for row in rows:
    print(f"ID: {row[0]}, Name: {row[1]}, Age: {row[2]}, City: {row[3]}")

conn.close()

```

Users older than 28:

ID: 2,	Name: Bob,	Age: 30,	City: Los Angeles
ID: 3,	Name: Charlie,	Age: 35,	City: Chicago