

## Лабораторна робота 2.2. Бібліотека Matplotlib.

### Теорія.

*Matplotlib* - це бібліотека Python для візуалізації даних. Вона дозволяє створювати графіки, діаграми, діаграми розсіювання та інші типи візуалізації даних. <https://matplotlib.org/stable/>

Matplotlib можна встановити, використовуючи менеджер пакетів `pip`.

Для цього виконайте команду:

```
pip install
```

Підключення до проєкту.

```
import matplotlib.pyplot as plt
```

```
%matplotlib inline - #для того, щоб відкривати графік в поточному вікні
```

### Приклад 1.

```
# from matplotlib import pyplot as plt
%matplotlib inline
import numpy as np #
x= np.linspace(0, 5, 30)y=x**2 —створення нитру масиву.
```

```
# Підпис x і y
```

```
plt.xlabel("X")
plt.ylabel("Y")
# заголовок до графіку
plt.title('Parabola')
```

```
# колір та стиль лінії на графіку
```

```
a=plt.plot(x,y, color="green", linestyle="dashed")
#Декілька графіків
```

```
x2 = [1, 2, 3, 4]
```

```
y2 = [2, 4, 6, 8]
plt.plot(x2, y2, color='red', linestyle='dashed', linewidth=2)
```

```
# Експорт графіка
```

```
plt.savefig("parabola.png").
```

## **Приклад 2. Subplot.**

```
# створення даних
```

```
x= np.linspace(0,10,100) y1=np.sin(x)
```

```
y2=np.cos(x)
```

```
# створення полотна з двома підграфіками
```

```
plt.subplot(1,2,1)
```

```
plt.plot(x, y1, "b*")
```

```
plt.title("sin")
```

```
plt.subplot(1,2,2)
```

```
plt.plot(x, y2, "go")
```

```
plt.title("cos")
```

## **Приклад 3. Figure.**

```
# створення даних
```

```
x= np.linspace(0,10,100) y1=np.sin(x)
```

```
y2=np.cos(x)
```

```
fig, (ax1, ax2) = plt.subplots(1,2)
```

```
ax1.plot(x, y1,"b*")
```

```
ax1.set_title("sin")
```

```
ax2.plot(x, y1,"go")
```

```
ax2.set_title("cos")
```

## **Приклад 4. Стовпчаста діаграма**

```
import matplotlib.pyplot as plt
```

```
%matplotlib inline
```

```
fig = plt.figure()
```

*# Визначення положення та розміру об'єкта axes на figure*

```
ax= fig.add_axes([0,0,1,1])
marks=["90-100","75-89","60-74", "0-59"]
students=[23,45, 12,7]

ax.bar(marks,students, color=["red","green","tomato","blue"], label=["Excellent","Good",
"Aver","Poor"])

ax.legend()

ax.set(ylim=[0,46], title="University")
```

### **Приклад 5. Гістограма.**

```
import matplotlib.pyplot as plt import numpy as np
%matplotlib inline

# створення вибірки з нормального розподілу.
x=np.random.normal(size=10000)
x1=np.random.normal(size=5000)

# створення гістограми
plt.hist([x,x1], bins=100, density=True, alpha=0.5, color=["tomato","green"])
```

## **Завдання (100б).**

### **Завдання.1. Побудувати графіки функції (20б).**

$y=\log(x)$

З допомогою subplot: а)  $y=2\sin(x)$  та б)  $y =2\cos x+5$

При побудові використовувати: підписи осей, subplot, Figure, легенди.

### **Завдання.2(40б).**

а) Створіть функцію для оцінки значення  $\cos(x)$  на основі ряду Тейлора.

$$\cos(x) \approx \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n}}{(2n)!} \approx 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} - \frac{x^{10}}{10!} + \dots$$

При описі функції, для розрахунку факторіала використовуйте бібліотеку math.

б) Побудуйте графік для порівняння апроксимації ряду Тейлора з функцією `np.cos()`.

- Для масиву кутів використовуйте `np.arange(min,max, крок)`
- с) Для побудови використовуйте `fig, ax = plt.subplots()`.

### Завдання.3(40б).

Використовуючи відкрите API

(<https://www.visualcrossing.com/resources/documentation/weather-api/timeline-weather-api/>)

<https://www.visualcrossing.com/resources/documentation/weather-api/how-to-load-weather-data-into-a-jupyter-notebook/>

побудувати графіки Логгер значень [values].

1. Написати клас з методом `get_Api`, який буде надавати дані про погоду (response) (температуру, вологість, і т.д.) в форматі json за заданий період (дата з\_ по\_) і з заданим кроком (хв, або рік) і в указаній місцевості.
2. Вхідні дані для запиту (LOCATION, START, END, STEP), подавати при визові методу класу.
3. Реалізувати метод `get_values`, який приймає вхідні параметри (LOCATION, START, END, STEP) і повертає масиви даних з values.
4. Створити клас побудови графіка, подаєш масив даних і цей клас будує криві (наприклад, передасте масив даних температури і вологості і в результаті 2 криві). Через клас можна задати колір кожній кривій, підписати і відповідна легенда, Побудувати графіки.
5. В результаті, маєте отримати, схоже на рис. 1 (стилізація у кожного індивідуальна).

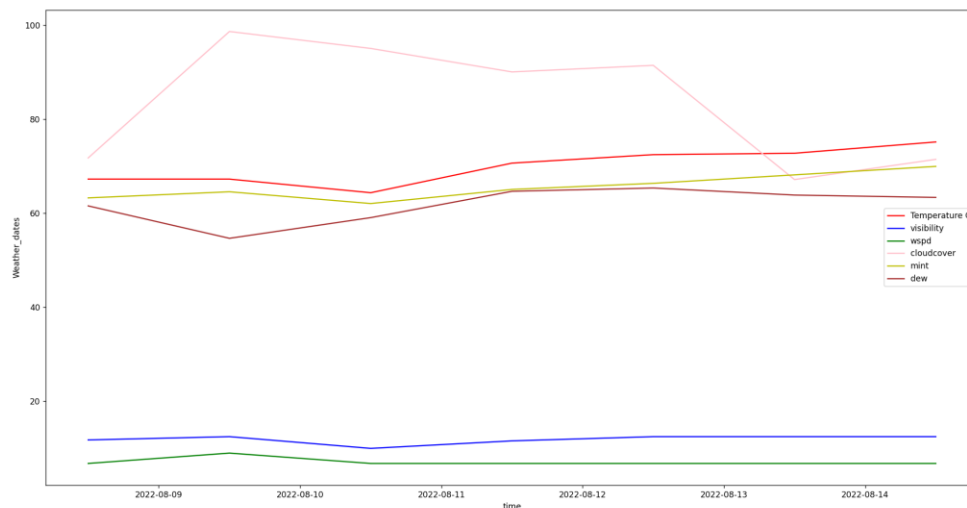


Рис. 1. Логгер значень [values].

**!!! Роботу здавати через гугл-форму: (є можливість додати посилання або файли)**

<https://docs.google.com/forms/d/e/1FAIpQLSdrxnys3Jagx0cAxRTG7LSaxzQHZh85uwAD7jELUm--3I75Lg/viewform?usp=sharing>

**Дата здачі завдання: 03.05.2023**