Інформатика 9

Урок 50

за підручником

Морзе Н.В. та ін.

Виконання навчальних проєктів

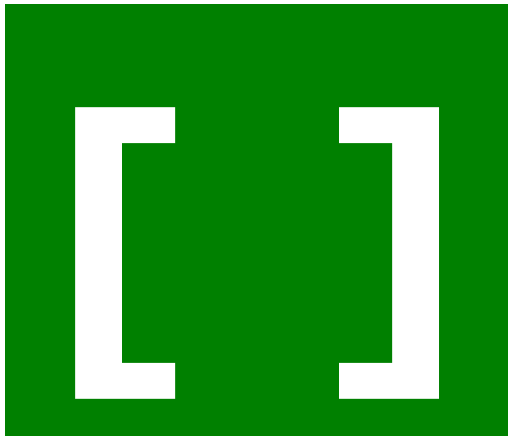
За навчальною програмою 2017 року

Розділ 3
§ 14

Повторюємо

Список (list) *— це упорядкований набір об'єктів різних типів (чисел, рядків, списків тощо), який можна змінювати.*

Значення списку *вкладаються у квадратні дужки [ ] та відокремлюються одне від одного за допомогою коми. Наприклад:*



```
a = [1, -2, 3.3, 'text']
```

У мові ***Python*** *списки використовуються для зберігання* ***масивів даних****.*

Розділ 3
§ 14

Створення списку

Створити порожній список можна двома способами:

за допомогою функції	list(): a = list()
використовуючи квадратні дужки	a = [ ]

ПРИКЛАД 1. *Створити об'єкт* ***animal*** *типу список і надати йому значення:*

```
animal = ["Кіт", "Собака", "Миша", "Хом'як"]
```

0 1 2 3

Список **animal** →

"Кіт"	"Собака"	"Миша"	"Хом'як"

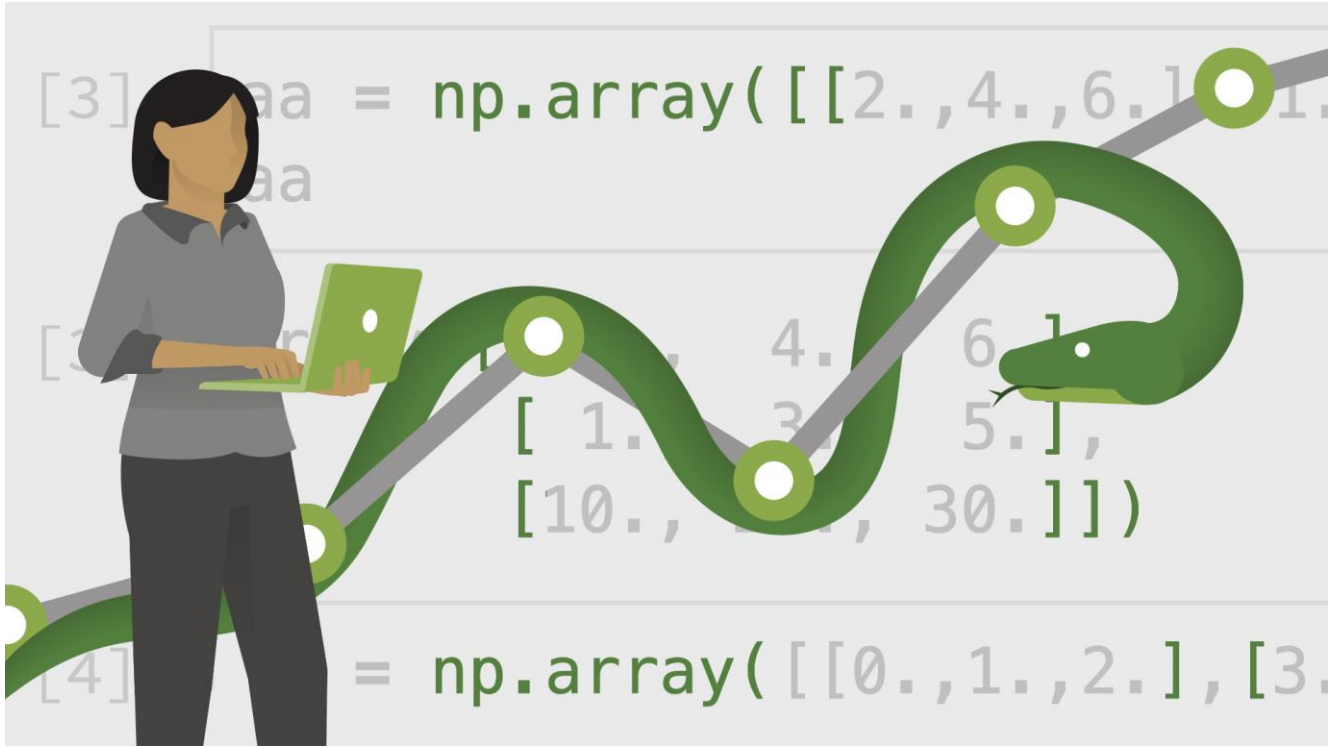
Розділ 3
§ 14

Методи списків

Виклик методів має загальний синтаксис:

Назва_списку. Назва_методу(<параметри>)

Зауважимо, що методи списків змінюють сам список, тому результат виконання не потрібно зберігати в іншу змінну.



Розділ 3
§ 14

Функції для роботи зі списками

Більшість функцій, на відміну від методів, не змінюють сам список, а повертають певне значення. Розглянемо на прикладі списку деякі корисні функції.

```
a = [1, 5, 7, 5, 31, -5]
```

`len(list)`	***Повертає довжину списку***

Приклад	***Результат***
`k = len(a)`	`k = 6`

Функції для роботи зі списками

Продовження…

```
a = [1, 5, 7, 5, 31, -5]
```

`max(list)`	***Повертає значення найбільшого елемента***

Приклад	***Результат***
`m = max(a)`	`m = 31`

Функції для роботи зі списками

Продовження…

```
a = [1, 5, 7, 5, 31, –5]
```

min(list)	*Повертає значення найменшого елемента*

Приклад	*Результат*
`m = min(a)`	`m = –5`

Розділ 3
§ 14

Функції для роботи зі списками

Продовження…

```
a = [1, 5, 7, 5, 31, –5]
```

sum(list)	*Повертає значення суми елементів*

Приклад	*Результат*
`s = sum(a)`	`s = 44`

Розділ 3
§ 14

Методи сортування масивів

Сортування елементів масиву — це впорядкування їх за деякою ознакою.

Щоб зрозуміти сутність алгоритмів сортування, розглянемо два найпростіші методи сортування масиву:

Сортування вибором максимального елемента

Сортування Обміном (метод бульбашки)

Розділ 3
§ 14

Об’єкти класу Listbox

Вибирати для виконання певних дій елементи списку із заданою ознакою зручніше, якщо ці елементи візуально представлені й доступні у графічному вікні. Таку можливість надають віджети класу Listbox.

Об’єкт класу Listbox дозволяє відображати список елементів, з якого користувач може обрати один або декілька пунктів.

```
class Attendee:
    'Common base class for all
    def __init__(self,
        self.name = name
        self.tickets =
    def displa
        print('Name : {},
```

Розділ 3
§ 14

Візуалізація елементів табличної величини

Два способи побудови діаграм:

із використанням графічних методів модуля tkinter

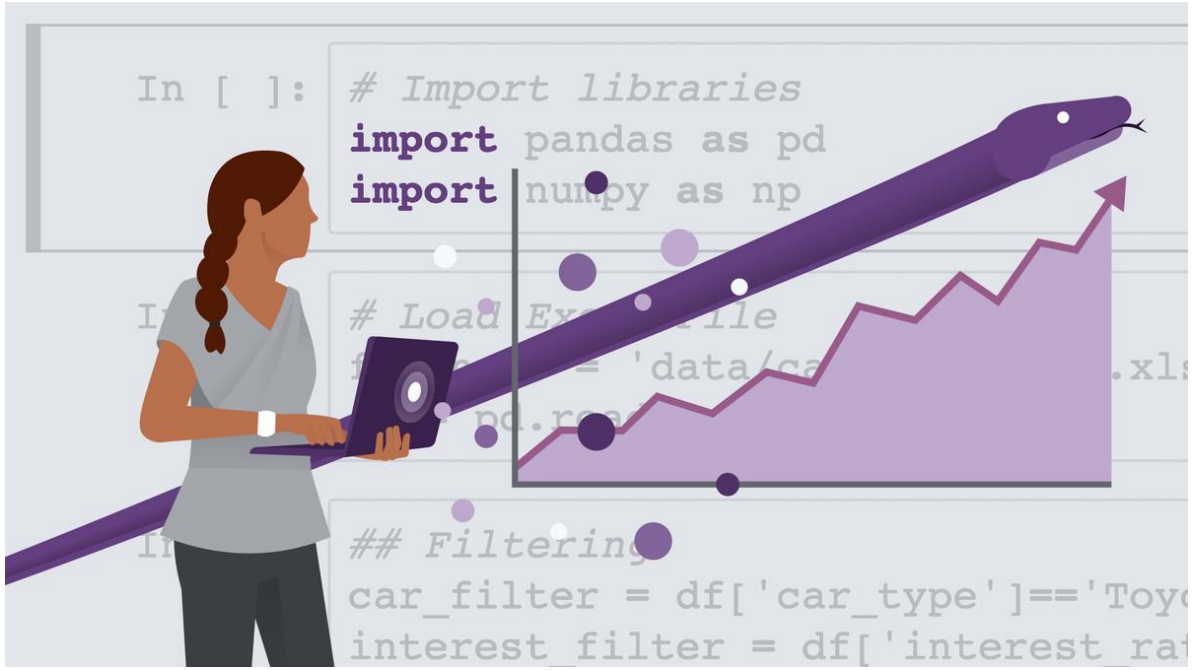
```
from tkinter import ttk

root = Tk()

button = ttk.Button(
button.pack()

def callback():
```

за допомогою методів бібліотеки matplotlib



Розділ 3
§ 14

Побудова діаграм за допомогою методів бібліотеки matplotlib

Matplotlib *— це кросплатформна бібліотека візуалізації даних і побудови діаграм для* ***Python****. Щоб установити бібліотеку для роботи в* ***IDLE PyCharm****, відкрийте вкладку* ***Terminal*** *і в командному рядку наберіть сполучення клавіш* ***Ctrl + Z****, щоб вийти з віртуального оточення; у наступному рядку наберіть команду* ***pip install matplotlib****:*

```
(venv) C:\Users\...\PycharmProjects\pythonProject4>^Z
(venv) C:\Users\...\PycharmProjects\pythonProject4>pip install matplotlib
```

Розділ 3
§ 14

Побудова діаграм за допомогою методів бібліотеки matplotlib

Після отримання повідомлення про успішне інсталювання matplotlib можна у Python-файлі

підключати модуль pyplot даної бібліотеки, який містить колекцію функцій для створення діаграм і налаштування їхнього вигляду. Імпортуємо пакет pyplot і коротко позначимо його як plt.

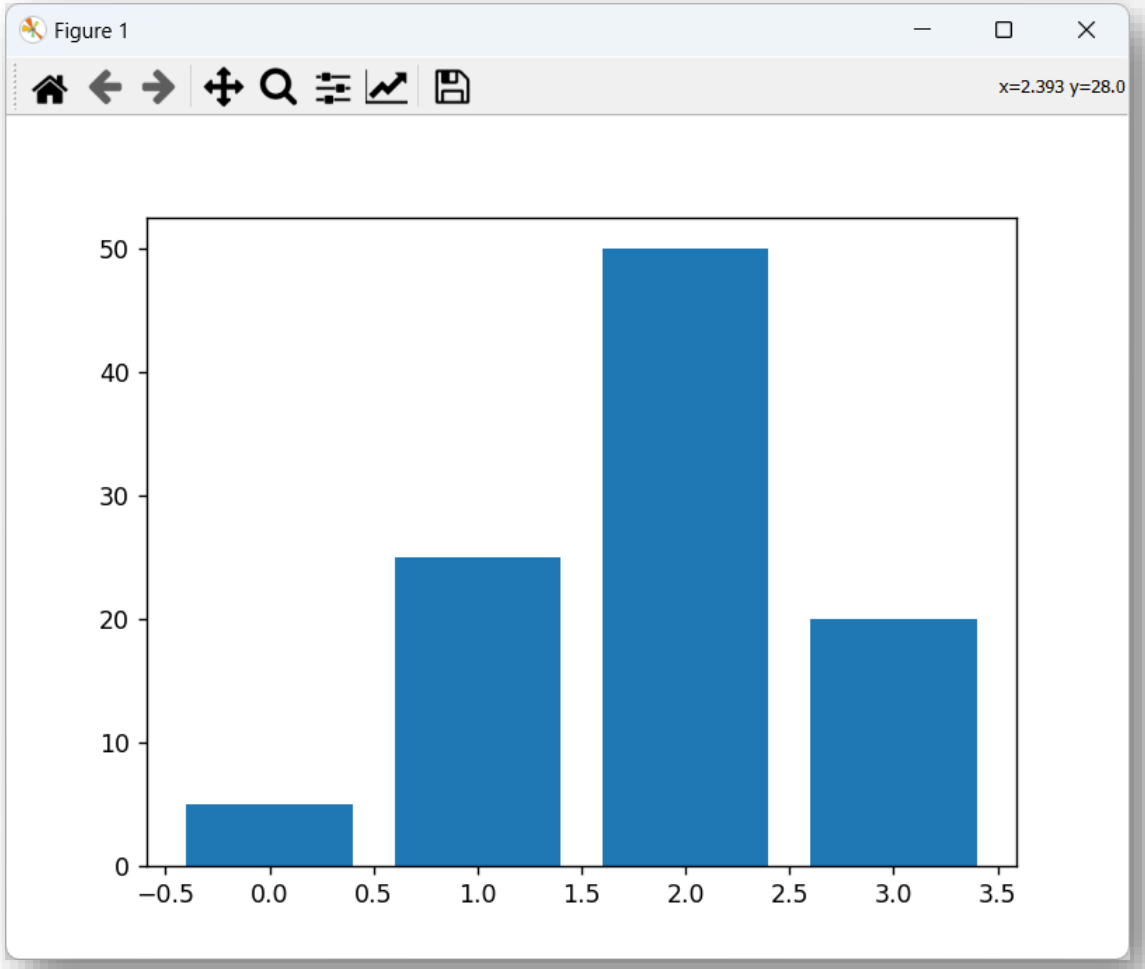

```
import matplotlib.pyplot as plt
```

Розділ 3
§ 14

Побудова діаграм за допомогою методів бібліотеки matplotlib

ПРИКЛАД 3. *Побудуємо стовпчикову діаграму для відображення елементів списку* ***data.***

```
import matplotlib.pyplot as plt
data = [5., 25., 50., 20.]
plt.bar(range(len(data)), data)
plt.show()
```



Розділ 3
§ 14

Побудова діаграм за допомогою методів бібліотеки matplotlib

***ПРИКЛАД 5**. Для побудови графіків призначена функція **plot()**. Побудуємо графік із прикладу 2, задавши червоний колір лінії (параметр 'r').*

Побудова діаграм за допомогою методів бібліотеки matplotlib

Програмний код до прикладу 5

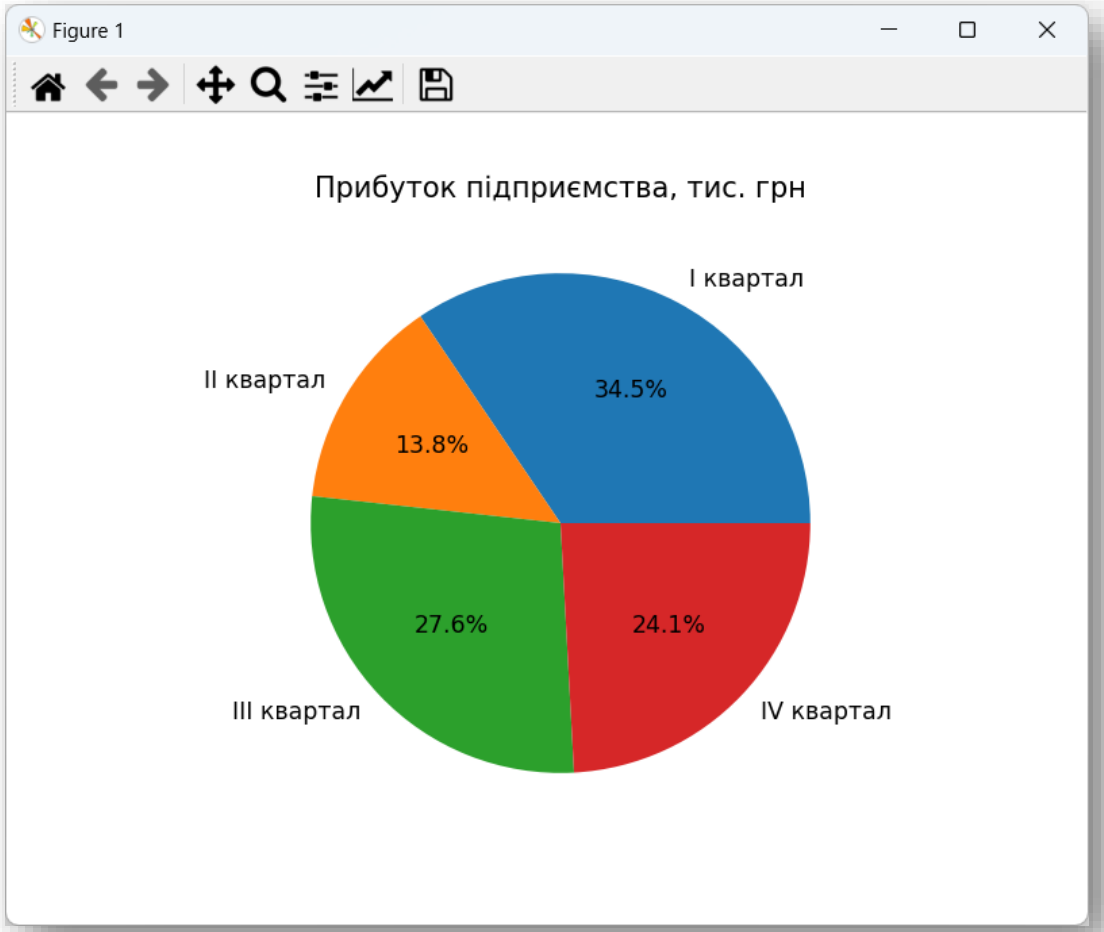
```
import matplotlib.pyplot as plt
plt.title('Середньодобова температура за тиждень')
a = [14, 12, 15, 12, 9, 10, 7]
day = ['Пн', 'Вт', 'Ср', 'Чт', 'Пт', 'Сб', 'Нд’]
plt.xlabel('Дні тижня', color = 'gray')
plt.ylabel('Середньодобова температура', color = 'gray')
plt.grid(True)
plt.plot(day, a, 'r')
plt.show()
```

Розділ 3
§ 14

Побудова діаграм за допомогою методів бібліотеки matplotlib

ПРИКЛАД 6. Побудувати кругову діаграму прибутку підприємства за 4 квартали року:

	I квартал	II квартал	III квартал	IV квартал
Прибуток підприємства, тис. грн	100	40	80	70



Побудова діаграм за допомогою методів бібліотеки matplotlib

Програмний код до прикладу 6

```
import matplotlib.pyplot as plt
plt.title('Прибуток підприємства, тис. грн')
a = [100, 40, 80, 70]
kv = ['I квартал', 'II квартал', 'III квартал', 'IV квартал']
plt.pie(a, labels = kv, autopct = "%.1f%%")
plt.show()
```

Атрибут autopct дозволяє відображати частку, використовуючи форматування рядка Python.

Розділ 3
§ 14

Працюємо за комп’ютером

Виконати завдання з файлу:

Практичне завдання_урок 50

Інформатика 9

Урок 50

за підручником

Морзе Н.В. та ін.

Дякую за увагу!

За навчальною програмою 2017 року