

Урок 19

Графіка з модулем

Tkinter #1

Знайомство з модулем "Tkinter".

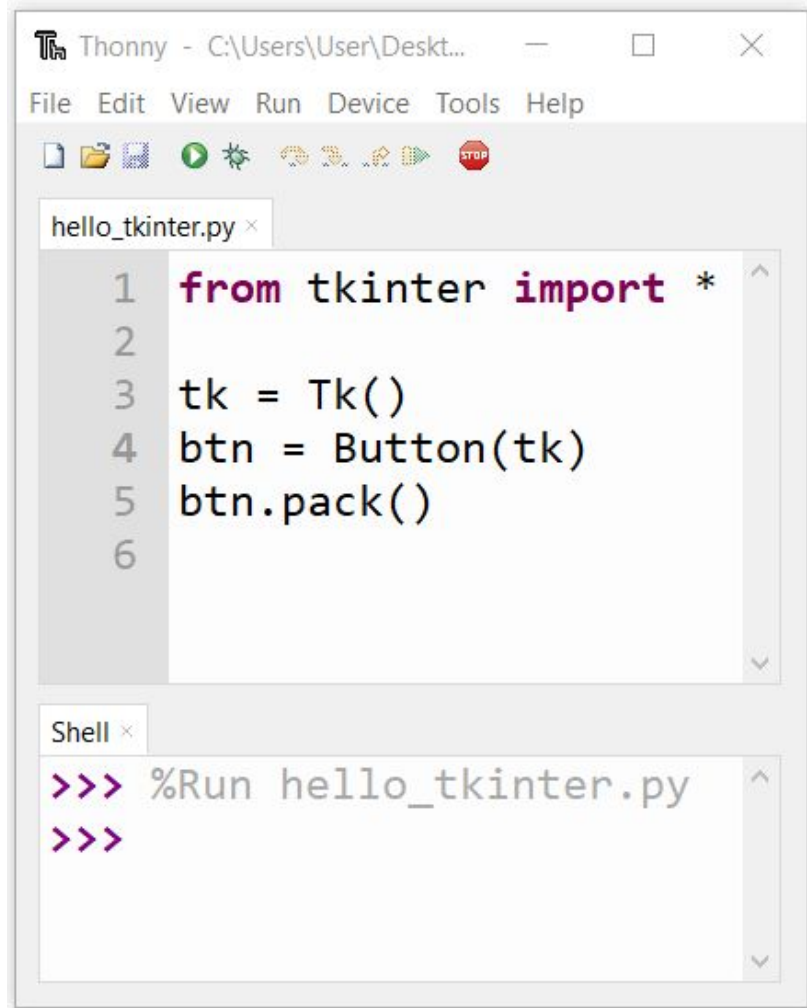


Модуль Tkinter

Це бібліотека для мови програмування Python, яка використовується для створення графіки та зображень.

Підключається за допомогою команди:

```
from tkinter import *
```

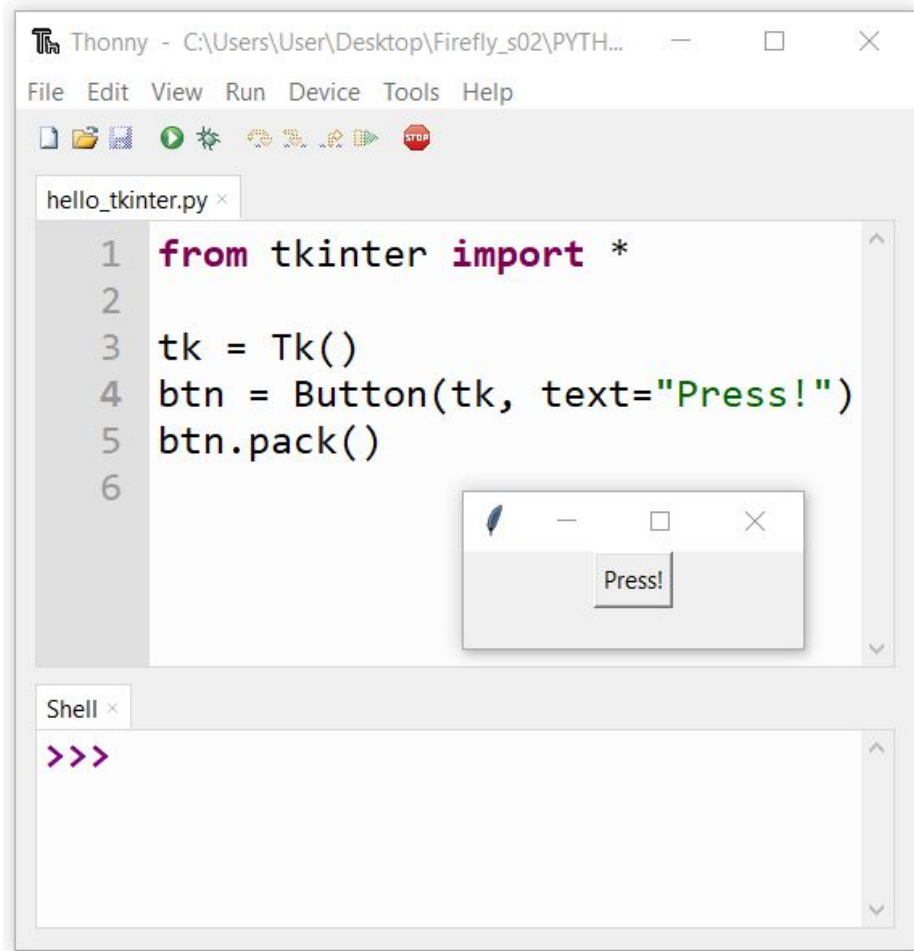


Створюємо кнопку

tk = Tk() - створення змінної з об'єктом класу Tk, який створює порожнє вікно. Це головний клас модулю Tkinter.

btn = Button (tk, text = 'Натисни')
- створення об'єкта класу **Button()** з аргументами **tk** та **text**.

btn.pack() - метод класу **Button()**, який відповідає за відображення кнопки на екрані.

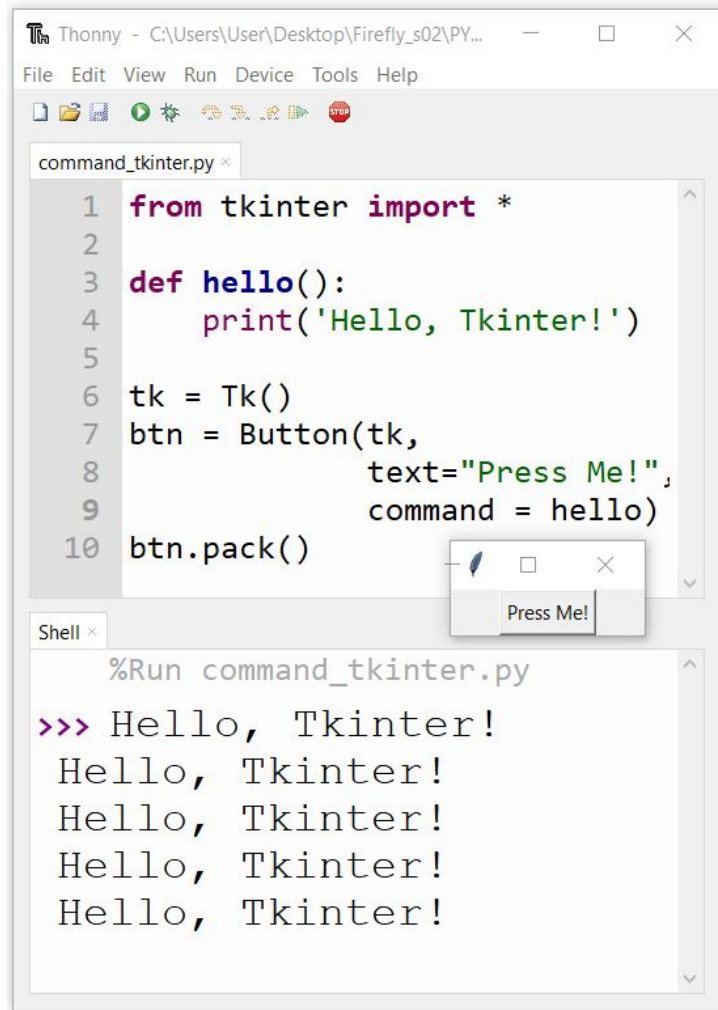


Взаємодія з кнопкою

Додавши ще один **іменований аргумент** до об'єкту **Button**, можна зробити так, щоб кнопка виконувала певну команду при натисненні.

```
btn = Button (tk, text = 'Натисни',  
command = hello)
```

command - аргумент, в якому ми можемо вказати назву функції, яка буде виконуватись.



The screenshot shows the Thonny IDE interface. The main editor window displays a Python script named `command_tkinter.py` with the following code:

```
1 from tkinter import *  
2  
3 def hello():  
4     print('Hello, Tkinter!')  
5  
6 tk = Tk()  
7 btn = Button(tk,  
8             text="Press Me!",  
9             command = hello)  
10 btn.pack()
```

Below the editor is a `Shell` window showing the output of the script after running it:

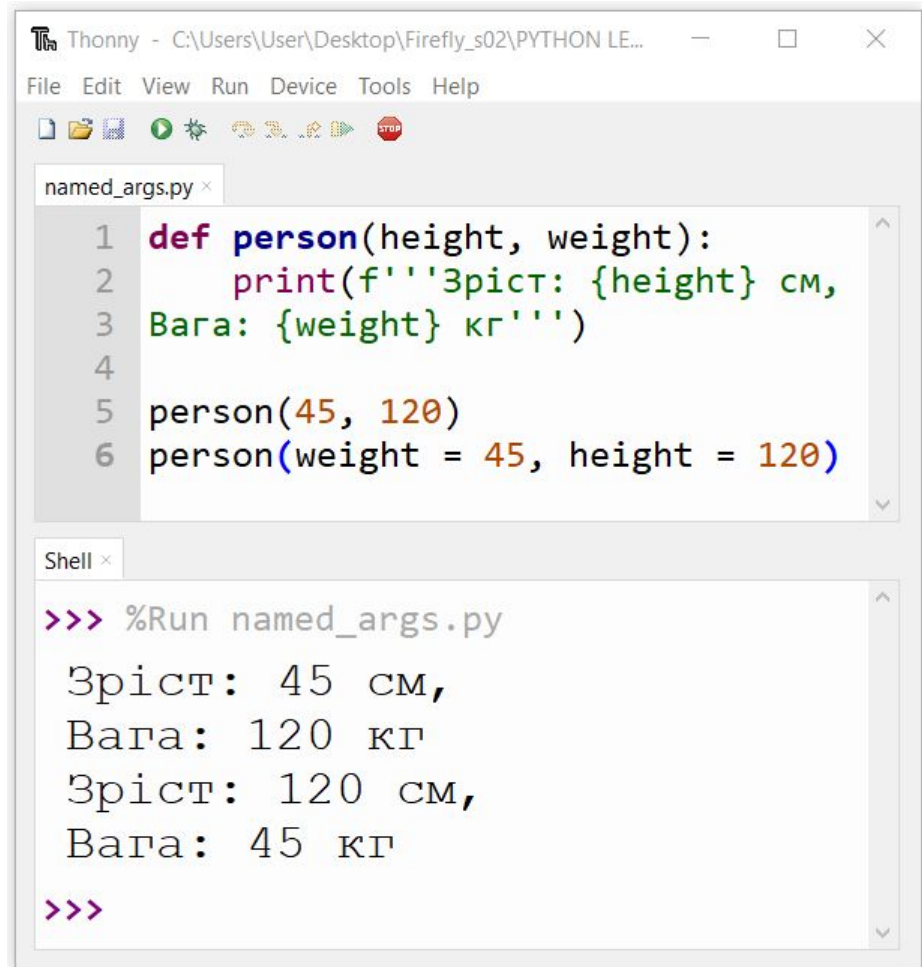
```
%Run command_tkinter.py  
>>> Hello, Tkinter!  
Hello, Tkinter!  
Hello, Tkinter!  
Hello, Tkinter!  
Hello, Tkinter!
```

A small Tkinter window titled `Press Me!` is also visible, containing a button with the text `Press Me!`.

Іменовані аргументи

дозволяють передати значення визначеному параметру функції, незалежно від його порядку передачі.

Також, іменовані аргументи спрощують використання функцій з великою кількістю параметрів і дозволяють звертатись тільки до деяких з них.



```
Thonny - C:\Users\User\Desktop\Firefly_s02\PYTHON LE...
File Edit View Run Device Tools Help

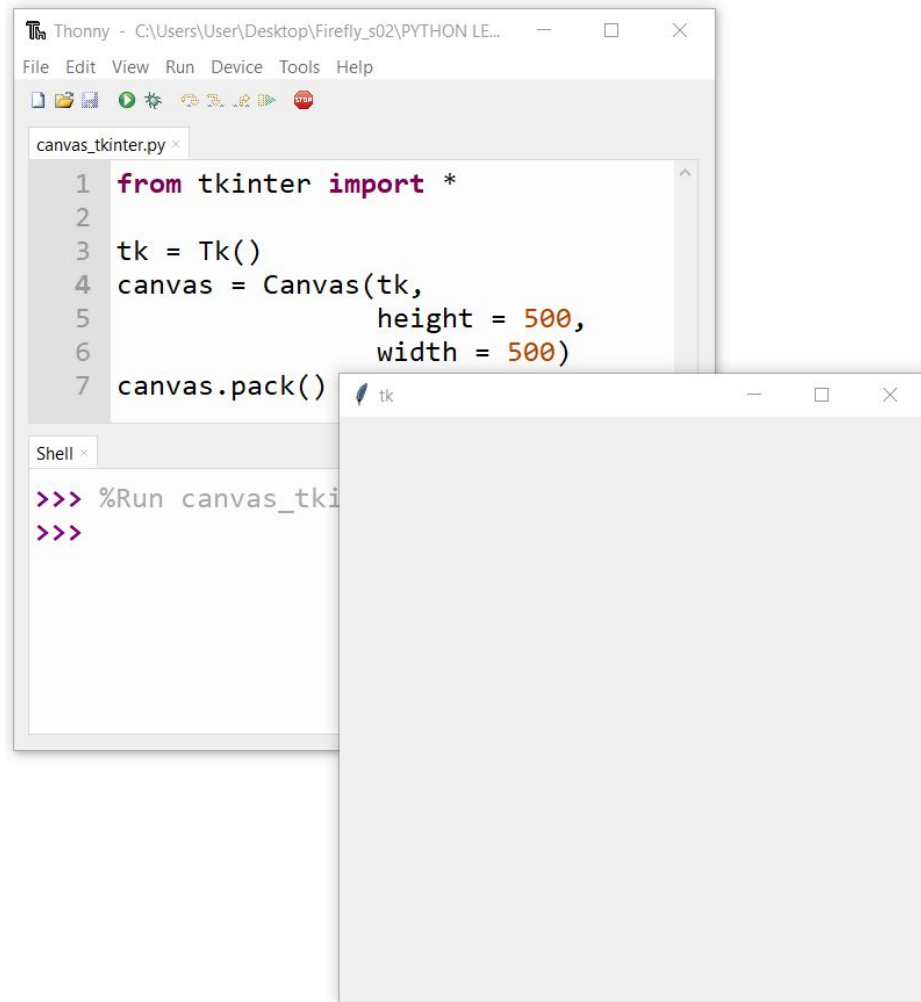
named_args.py x
1 def person(height, weight):
2     print(f'''Зріст: {height} см,
3     Вага: {weight} кг''')
4
5 person(45, 120)
6 person(weight = 45, height = 120)

Shell x
>>> %Run named_args.py
Зріст: 45 см,
Вага: 120 кг
Зріст: 120 см,
Вага: 45 кг
>>>
```

Створення холста

Якщо потрібно щось намалювати на екрані, використовують об'єкт **Canvas()**, який створює холст (полотно) з заданими висотою та довжиною..

```
canvas = Canvas(tk, width = 500,  
height = 500)
```

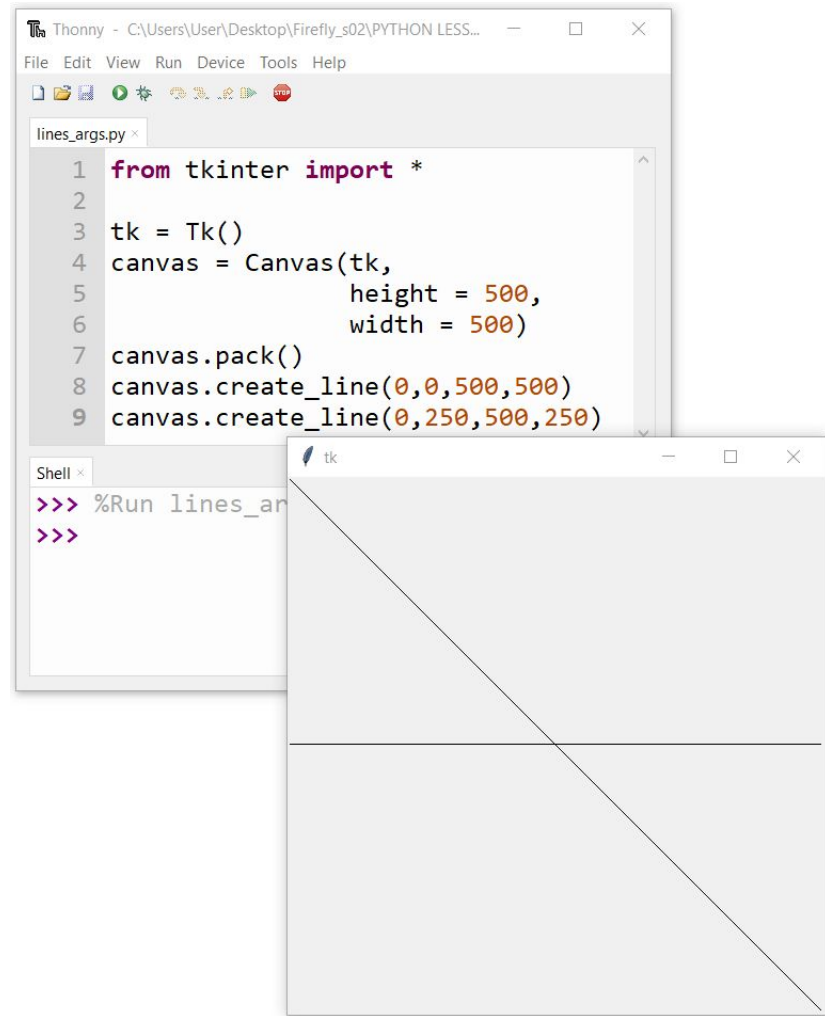


Створення ліній

Щоб створити лінію, використовують метод **create_line(координати)**.

Перша пара координат **(0, 0,** відповідає за початкову точку, звідки буде починатись лінія.

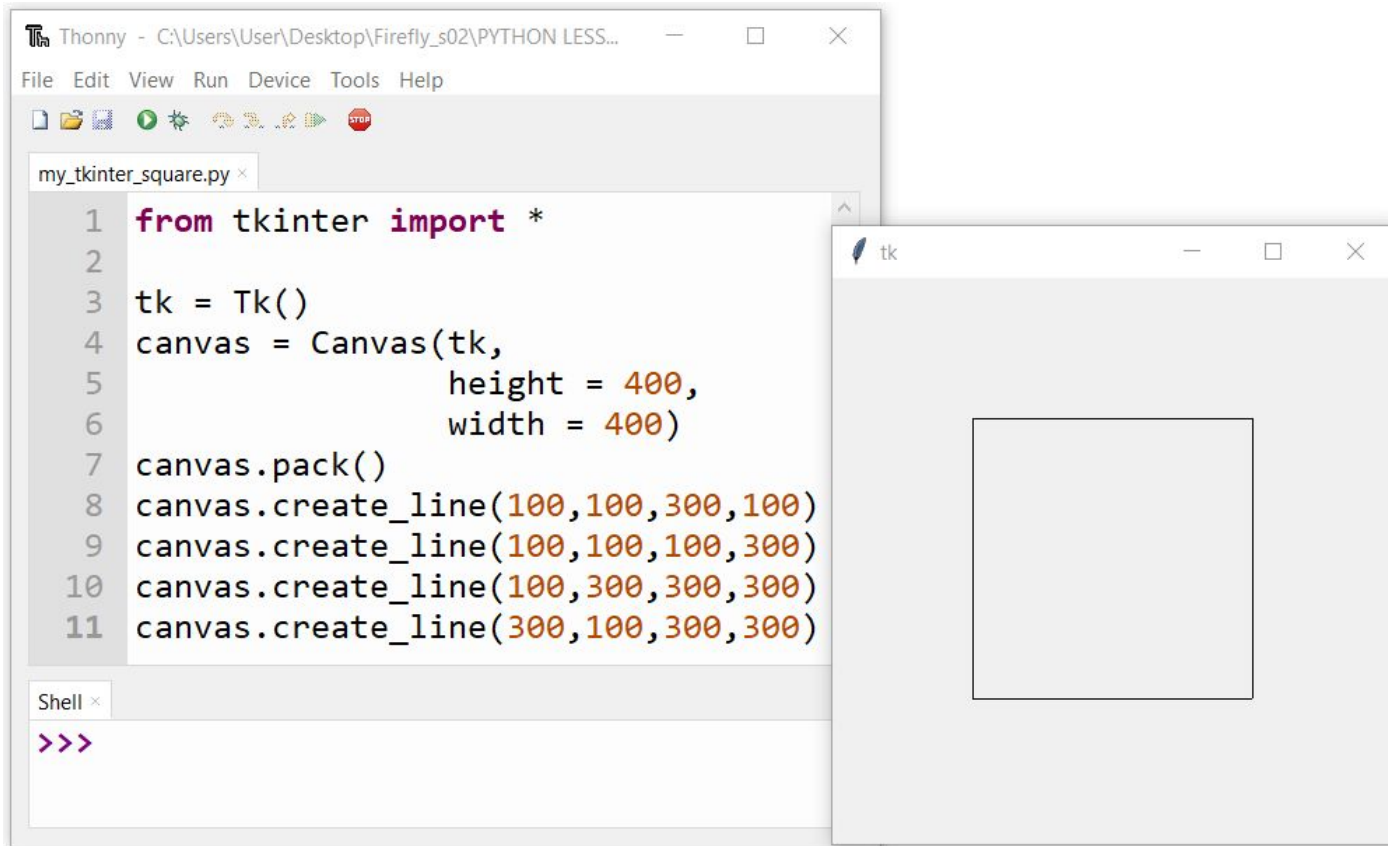
Друга пара координат **500, 500)** відповідає за точку, де лінія буде закінчуватись.





Практична робота

- 1) Створіть новий файл **(New)**
- 2) Імпортуйте модуль **tkinter**
- 3) Створіть полотно розміром **400** на **400** пікселів і намалюйте у його центрі квадрат розміром **200** на **200** пікселів, використовуючи лінії
- 4) Збережіть файл **(Save)** під назвою **my_tkinter_square.py**
- 5) Виконайте запуск програми **(Run)**.

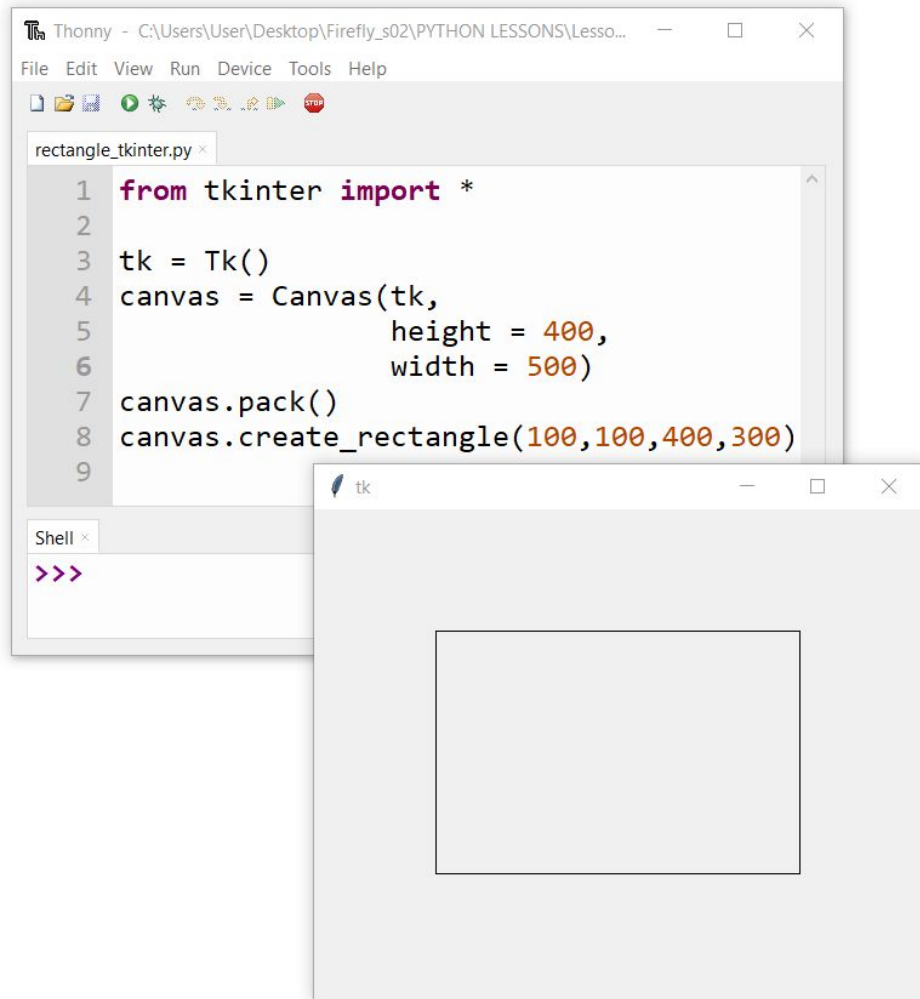


Якщо все зроблено правильно - ви побачите такий результат

Створення прямокутників

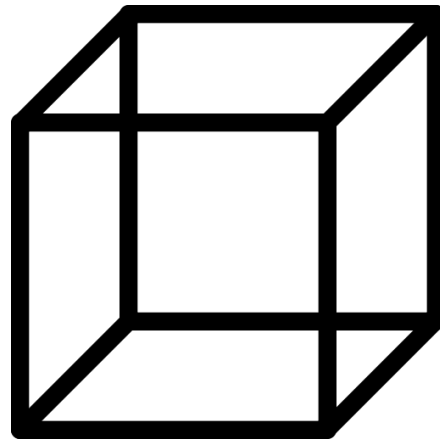
Щоб створити прямокутник,
використовують метод
create_rectangle(координати).

Перша пара координат **(100, 100,**
відповідає за положення лівого
верхнього куту прямокутника.
Друга пара координат **400, 300)**
відповідає за положення нижнього
правого куту прямокутника.



Практична робота

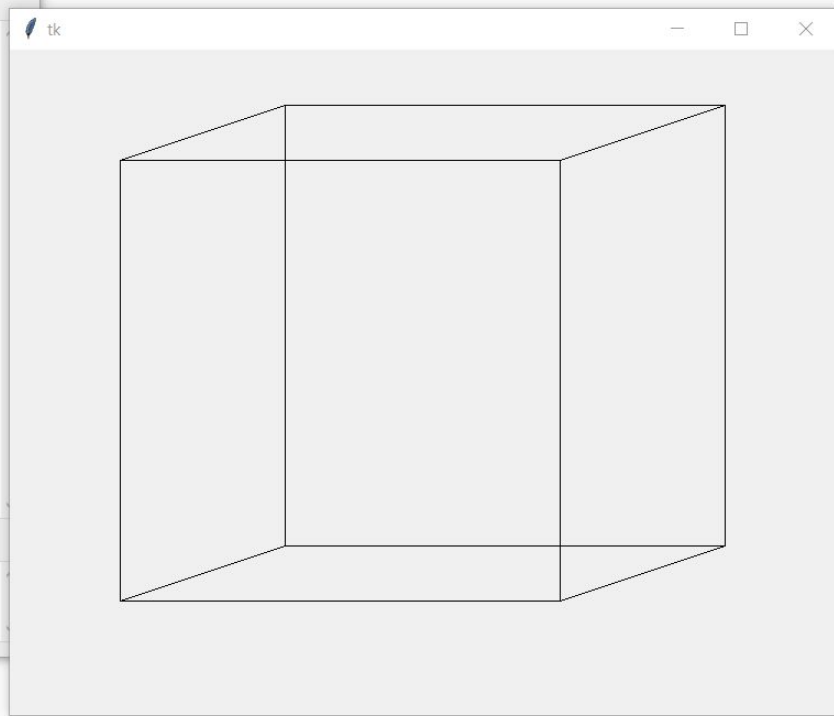
- 1) Створіть новий файл **(New)**
- 2) Імпоруйте модуль **tkinter**
- 3) За допомогою ліній та прямокутників створіть зображення кубу
- 4) Збережіть файл **(Save)** під назвою **my_tkinter_cube.py**
- 5) Виконайте запуск програми **(Run)**.



```
Thonny - C:\Users\User\Desktop\Firefly_s02\PYTHON LESSONS\Lesso...
File Edit View Run Device Tools Help

rectangle_tkinter.py x
1 from tkinter import *
2
3 tk = Tk()
4 canvas = Canvas(tk,
5                 height = 600,
6                 width = 750)
7 canvas.pack()
8 canvas.create_rectangle(100,100,500,500)
9 canvas.create_rectangle(250,50,650,450)
10
11 canvas.create_line(100,100,250,50)
12 canvas.create_line(650,450,500,500)
13 canvas.create_line(100,500,250,450)
14 canvas.create_line(500,100,650,50)

Shell x
>>>
```



Якщо все зроблено правильно - ви побачите такий результат



Підсумки

Познайомились з **модулем tkinter** для створення зображень