



Урок 7

Модулі Python

Робота з модулями. Модуль "Turtle".

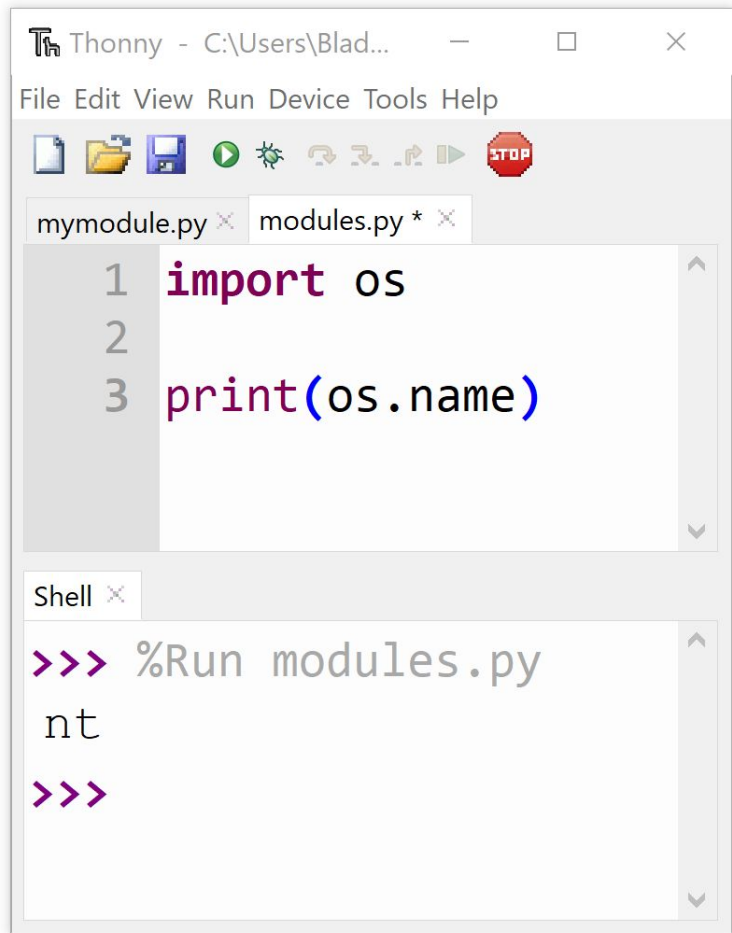


Модуль

це сукупність готових функцій та іншого корисного коду, які можна використовувати в своїй програмі.

Модуль підключається за допомогою команди **import** у верхній частині програми.

Після підключення модулю його ім'я використовується для доступу до функцій цього модулю.



The screenshot shows the Thonny Python IDE interface. The title bar reads "Thonny - C:\Users\Blad...". The menu bar includes "File", "Edit", "View", "Run", "Device", "Tools", and "Help". The toolbar contains icons for file operations, running, and debugging. Two tabs are open: "mymodule.py" and "modules.py *". The "modules.py" tab is active, displaying the following code:

```
1 import os
2
3 print(os.name)
```

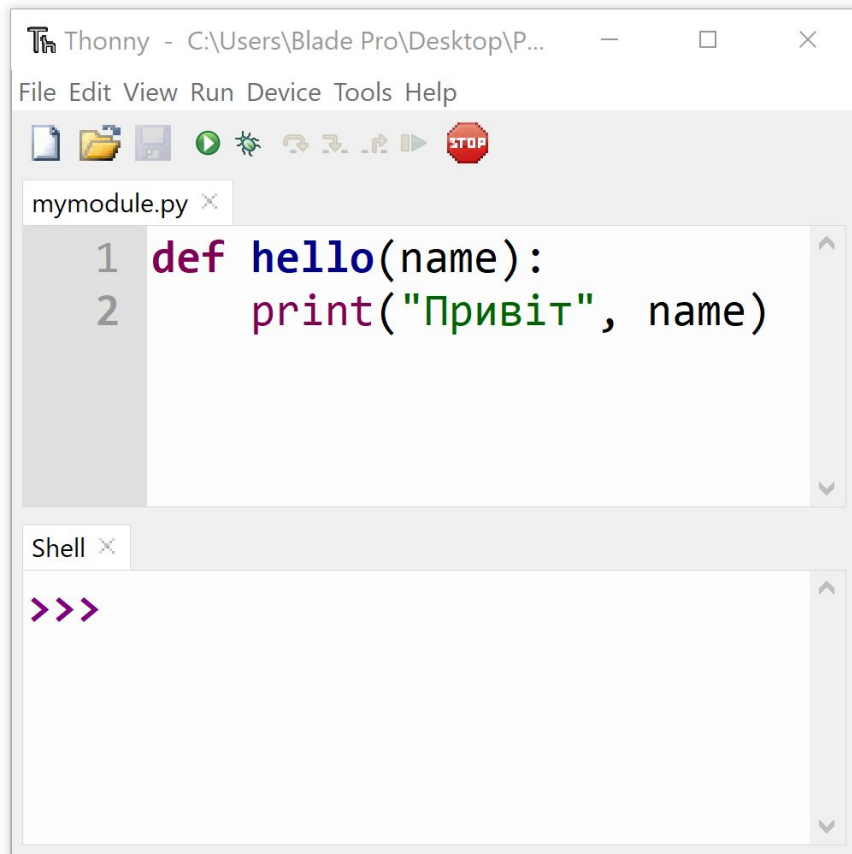
Below the code editor is a "Shell" window. It shows the command prompt with the command "%Run modules.py" and the output "nt".

```
>>> %Run modules.py
nt
>>>
```

Свій модуль

Цей модуль створює привітання "Привіт (ім'я)". Те, що буде відображатись на місці "ім'я" обирає користувач.

- 1) Створюємо новий файл
- 2) Пишемо в ньому код, що наведений справа
- 3) Зберігаємо файл з назвою **mymodule.py**



Thonny - C:\Users\Blade Pro\Desktop\P...

File Edit View Run Device Tools Help

mymodule.py

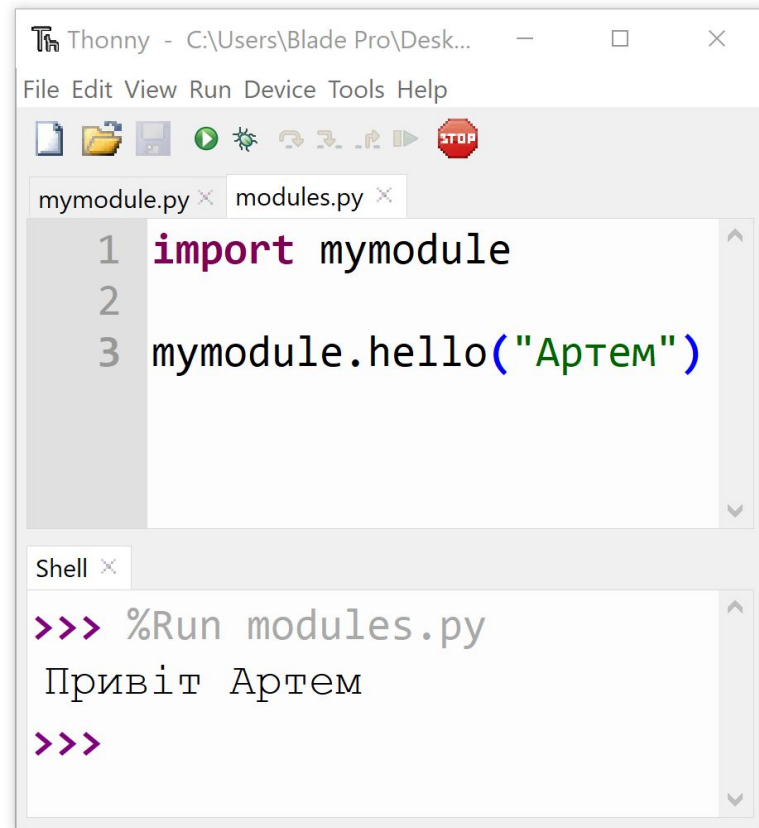
```
1 def hello(name):  
2     print("Привіт", name)
```

Shell

```
>>>
```

Використання свого модулю

- 1) Створюємо новий файл у тій же директорії, що і модуль
- 2) Підключаємо модуль за допомогою команди **import**
- 3) Використовуємо функцію **hello** з підключеного модулю
- 4) Зберігаємо файл з назвою **modules.py**

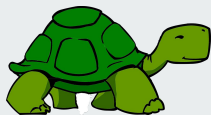


The screenshot shows the Thonny Python IDE interface. The title bar reads "Thonny - C:\Users\Blade Pro\Desktop...". The menu bar includes "File", "Edit", "View", "Run", "Device", "Tools", and "Help". The toolbar contains icons for file operations and execution. Two tabs are open: "mymodule.py" and "modules.py". The "modules.py" tab is active, displaying the following code:

```
1 import mymodule
2
3 mymodule.hello("Артем")
```

Below the code editor is a "Shell" window. It shows the command prompt with the command `>>> %Run modules.py` and the output `Привіт Артем`. The prompt `>>>` is shown again on the next line.

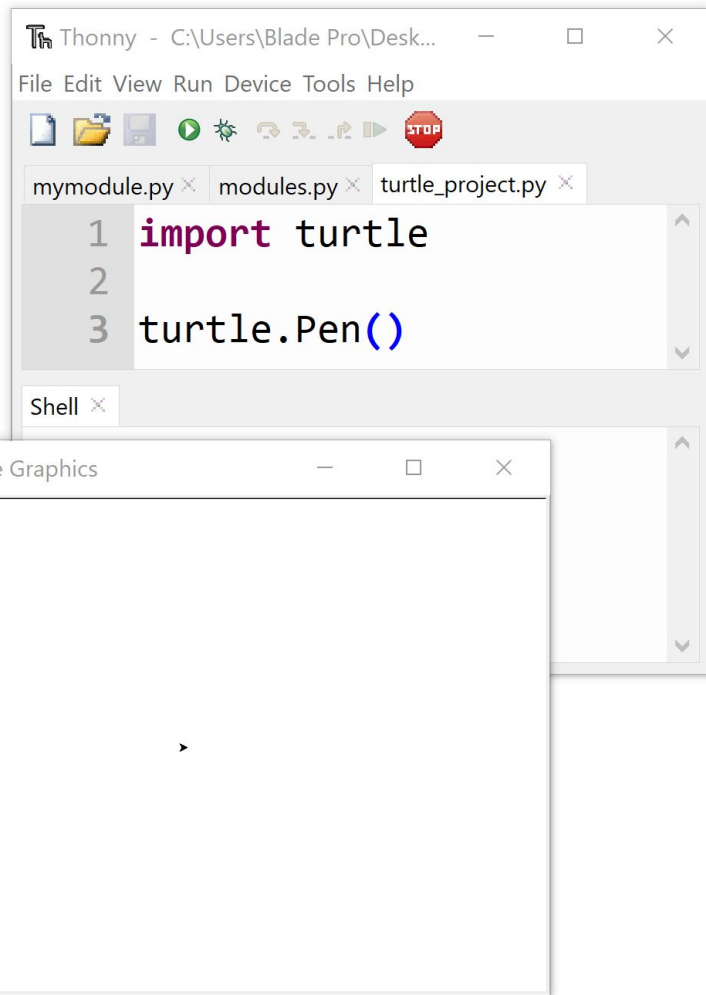
Модуль turtle



надає можливість створювати прості малюнки за допомогою ліній, крапок та кривих. Щоб почати ним користуватись, треба вивести наступну команду:

import turtle

Команда **turtle.Pen()** дозволяє створити полотно на якому можна малювати.





Практична робота

- 1) Створіть новий файл **(New)**
- 2) Імпортуйте модуль **turtle**
- 3) Створіть порожнє полотно за допомогою команди **turtle.Pen()**
- 4) Збережіть файл **(Save)** під назвою **picture.py**
- 5) Виконайте запуск програми **(Run)**.

Thonny - C:\Users\Blade Pro\Desktop

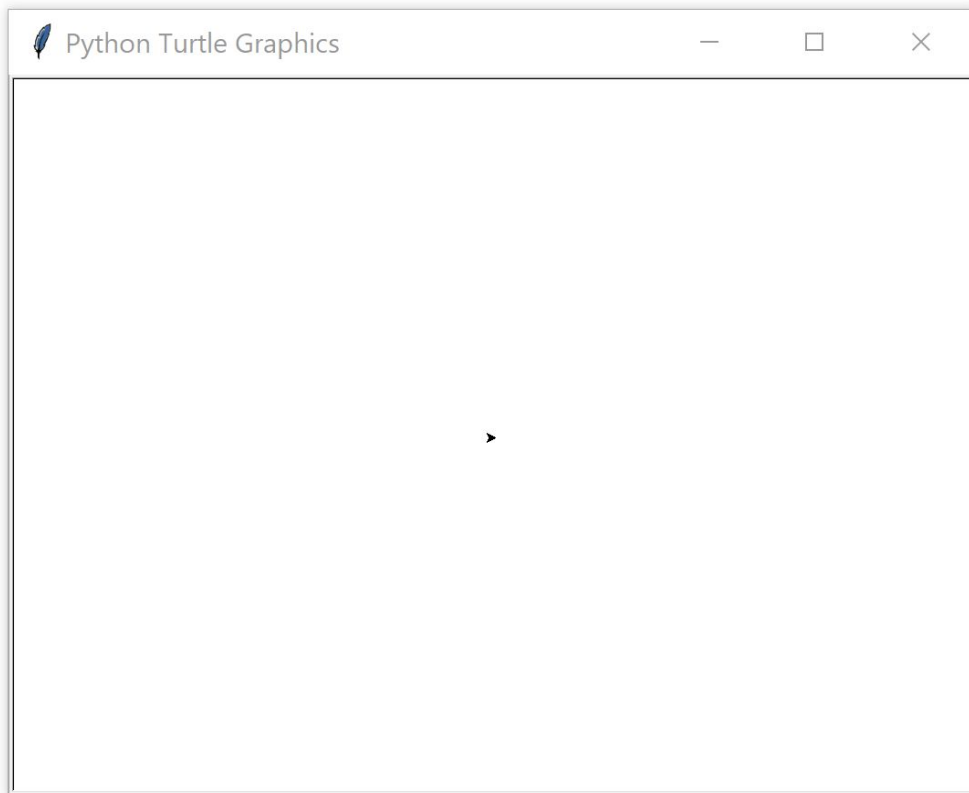
File Edit View Run Device Tools Help

mymodule.py x modules.py x picture.py x

```
1 import turtle
2
3 turtle.Pen()
```

Shell x

```
>>> %Run picture.py
>>>
```



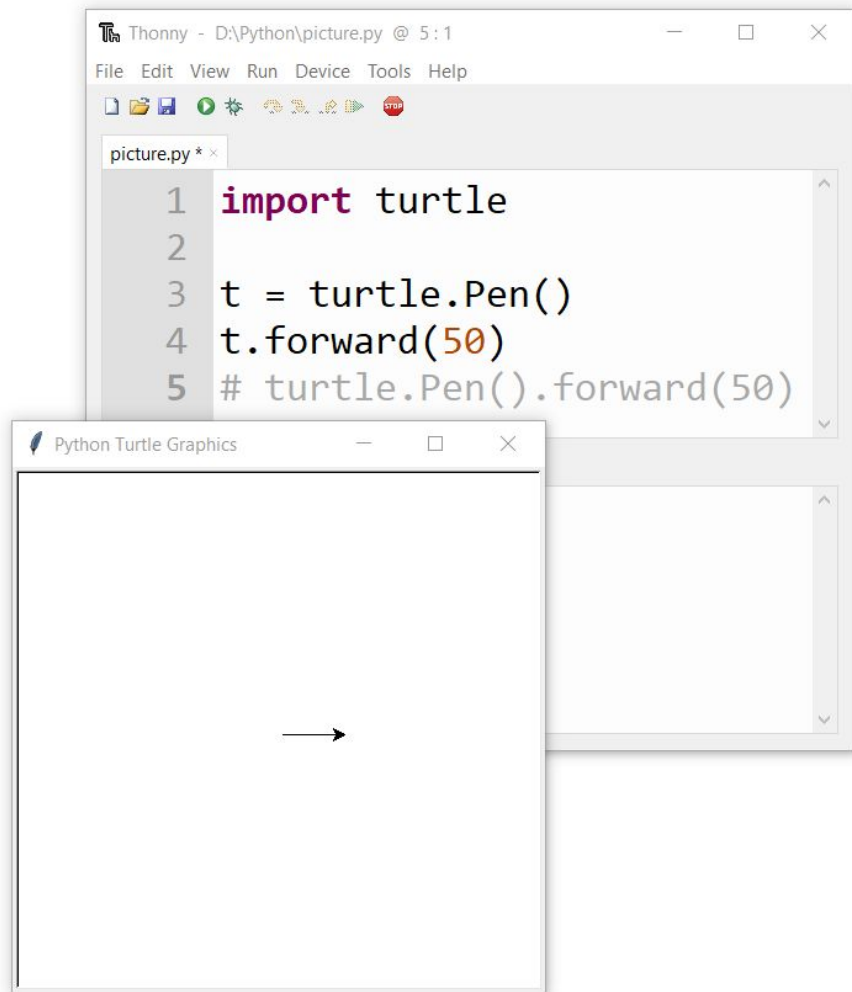
Якщо все зроблено правильно - ви побачите такий результат

Команди керування “черепашкою”

`turtle.Pen().forward(значення)` -
переміщення черепашки на кількість
пікселів, що зазначена у значенні **`forward`**

Щоб скоротити код, можна зберегти
об'єкт **`turtle.Pen()`** у змінній **`t`**.

`t.forward(50)` - переміщення черепашки
на 50 пікселів в той бік, куди “дивиться”
черепашка.



Команди керування “черепашкою”

t.forward(пікс.) - переміщення вперед

t.backward(пікс.) - переміщення назад

t.up() - підняти черепашку над полотном

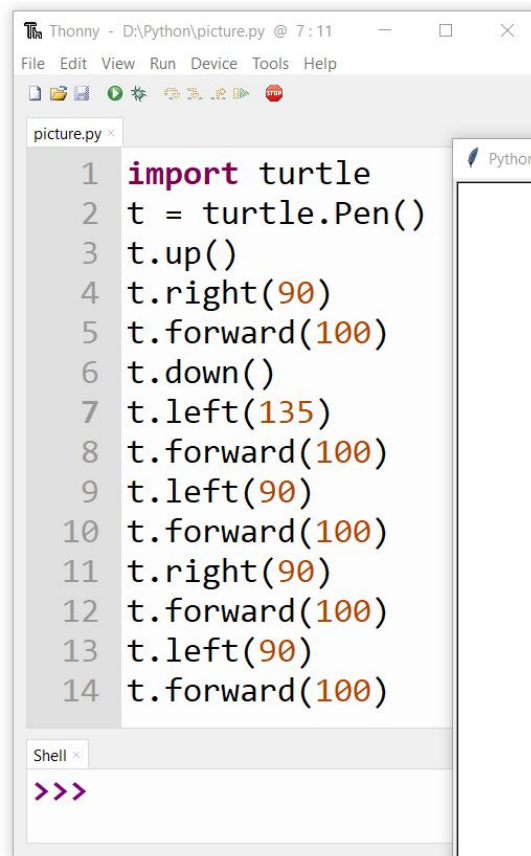
t.down() - опустити черепашку

t.left(град.) - поворот вліво

t.right(град.) - поворот вправо

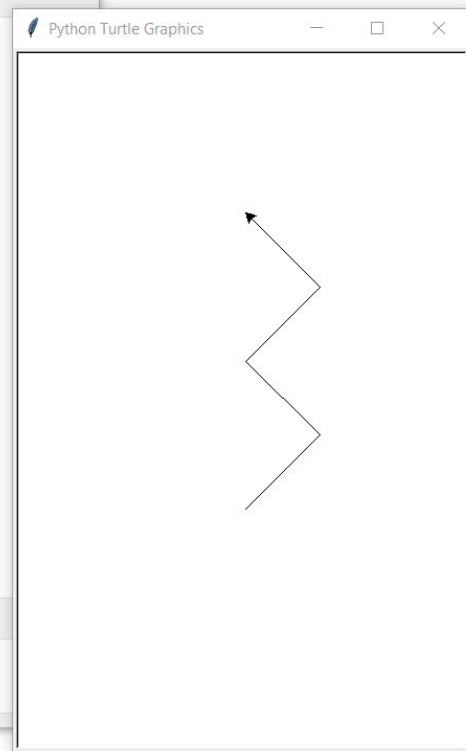
t.reset() - перезавантаження полотна

t.clear() - очистити полотно



```
1 import turtle
2 t = turtle.Pen()
3 t.up()
4 t.right(90)
5 t.forward(100)
6 t.down()
7 t.left(135)
8 t.forward(100)
9 t.left(90)
10 t.forward(100)
11 t.right(90)
12 t.forward(100)
13 t.left(90)
14 t.forward(100)
```

Shell >>>

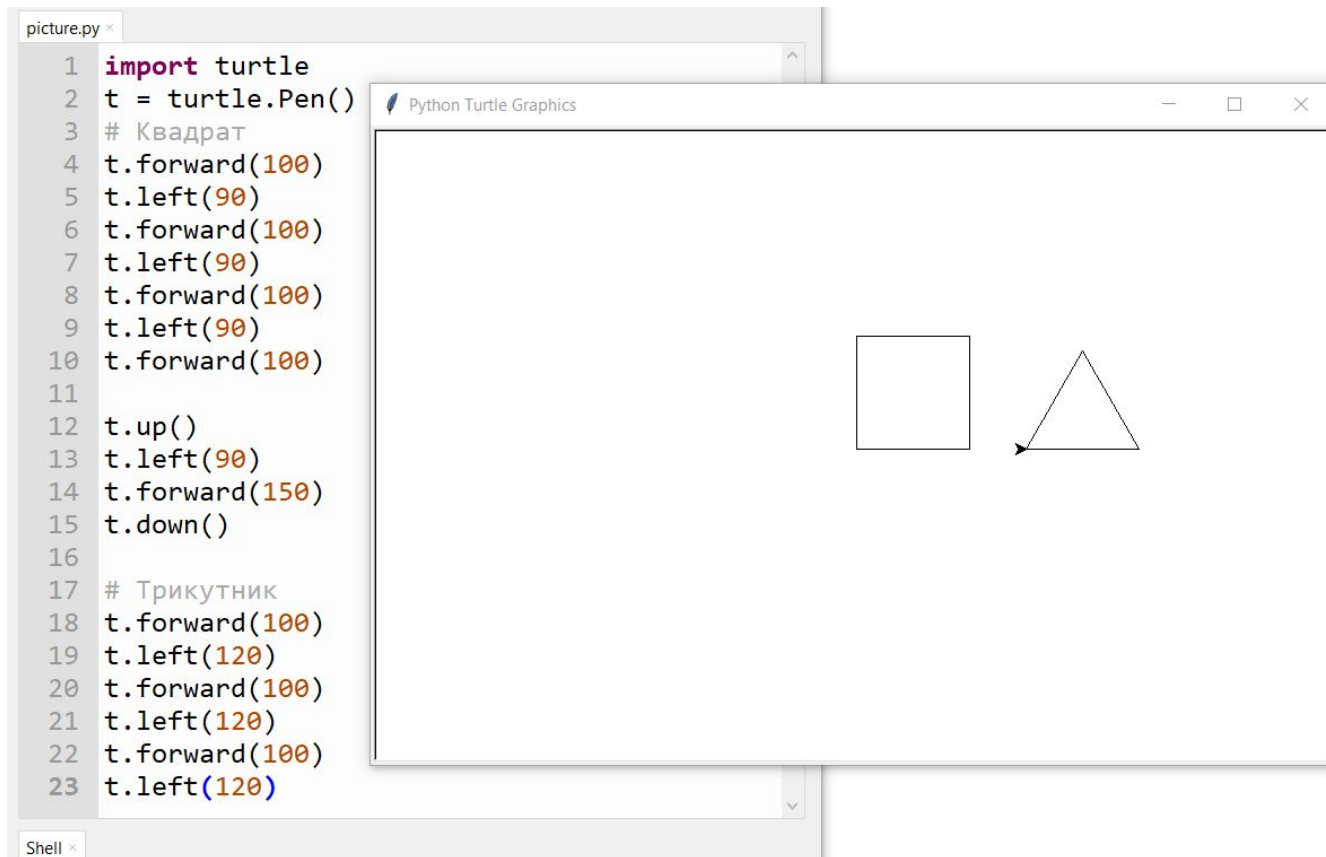




Практична робота

Спробуйте намалювати **квадрат** та **трикутник** на одному полотні.





Якщо все зроблено правильно - ви побачите такий результат

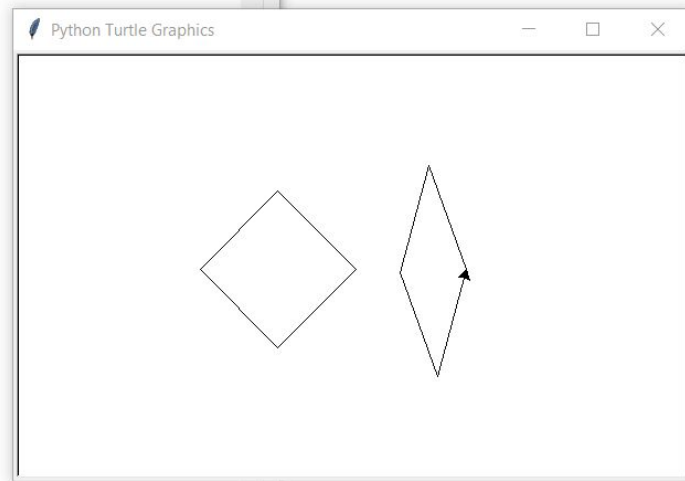


Практична робота

Спробуйте намалювати **ромб з рівними кутами** та **ромб з різними кутами** на одному полотні



```
picture.py x picture_2.py x
1 import turtle
2 t = turtle.Pen()
3 # Ромб 1
4 t.left(135)
5 t.forward(100)
6 t.left(90)
7 t.forward(100)
8 t.left(90)
9 t.forward(100)
10 t.left(90)
11 t.forward(100)
12
13 t.up()
14 t.right(45)
15 t.forward(100)
16 t.down()
17 # Ромб 2
18 t.left(110)
19 t.forward(100)
20 t.left(145)
21 t.forward(100)
22 t.left(35)
23 t.forward(100)
24 t.left(145)
25 t.forward(100)
```



Якщо все зроблено правильно - ви побачите такий результат



Підсумки

Дізнались, що таке **модулі** та для чого вони використовуються

Навчилися створювати **власні модулі**

Дізнались, як використовувати **модуль turtle** для малювання