



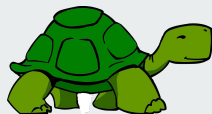
Урок 18

Малювання у Python #2

Робота з модулем "Turtle".



Модуль Turtle



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

Підключається за допомогою команд:

import turtle

ця команда імпортує модуль, але треба буде писати назву модулю перед методами

from turtle import *

ця команда дозволяє звертатися до методів без застосування імені модуля

A screenshot of the Thonny Python IDE. The window title is "Thonny - C:\Users\User\Desktop\PY...". The menu bar includes "File", "Edit", "View", "Run", "Device", "Tools", and "Help". The toolbar contains icons for opening files, saving, running, and stopping. The main editor window shows a file named "circleSpiral.py" with the following Python code:

```
1 from turtle import *
2
3 color("Dark Orchid")
4
5 for x in range(100):
6     circle(x)
7     left(91)
```

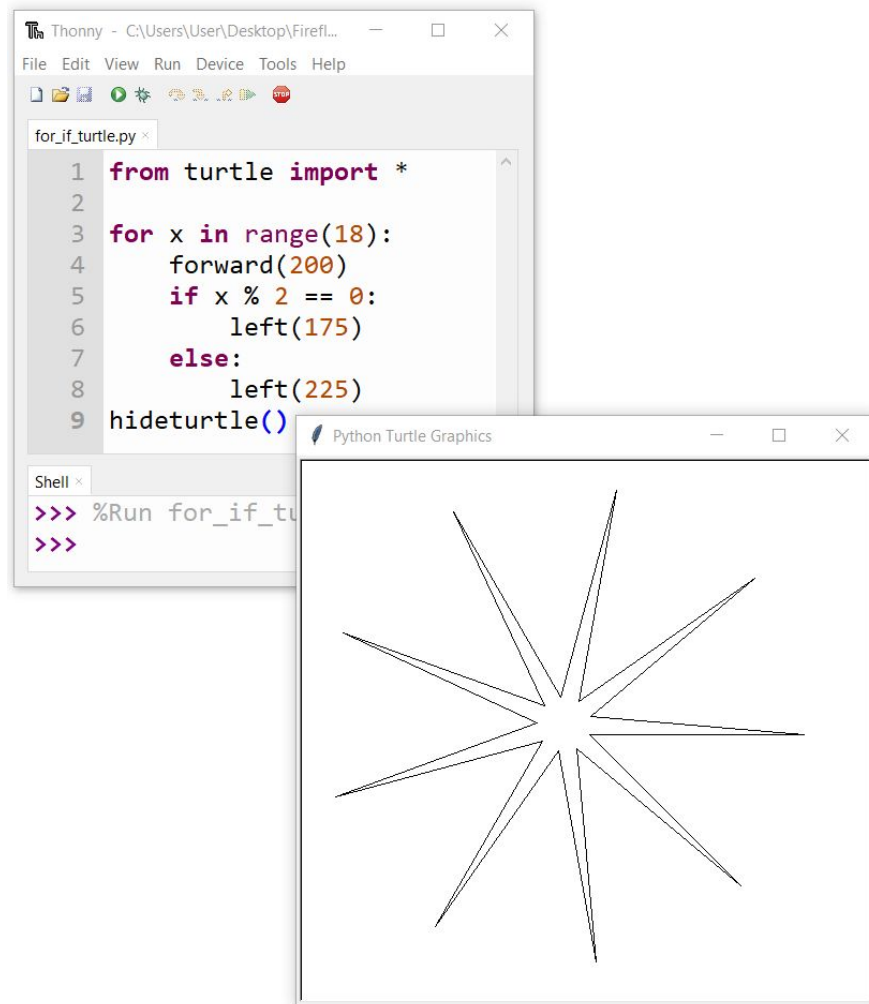
The bottom panel shows a "Shell" window with the prompt ">>>".

for-цикл та умови

Використання циклу **for** дозволяє виконувати схожі команди визначену кількість разів.

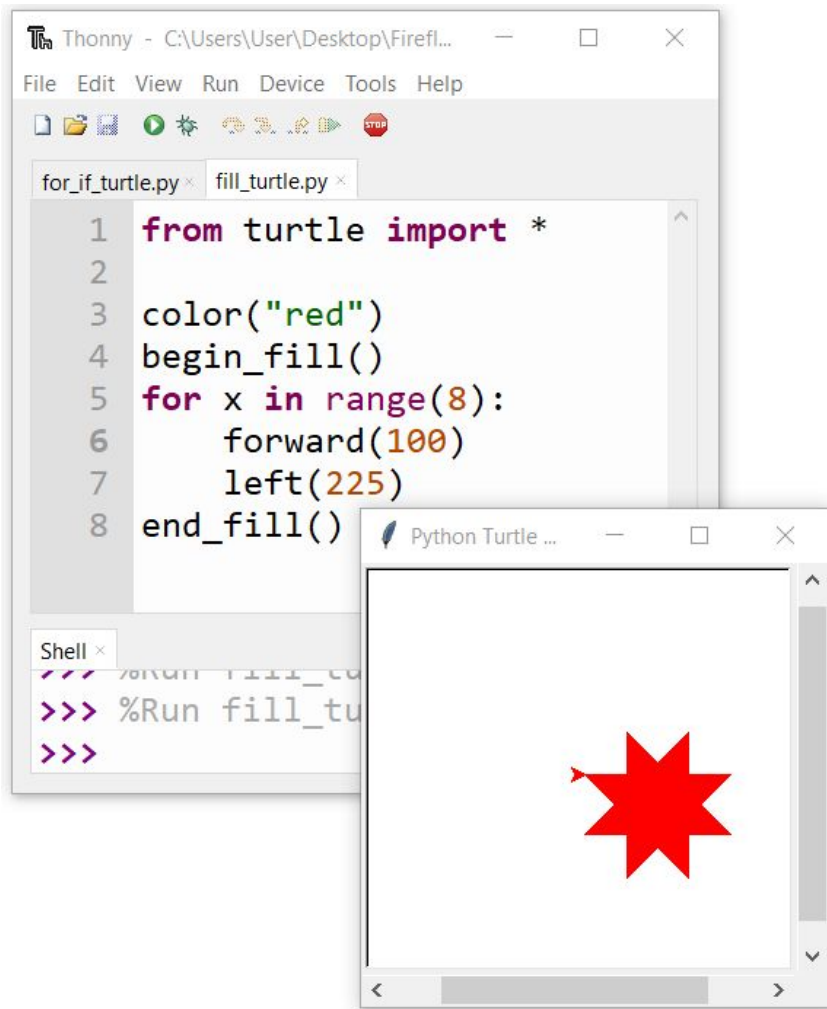
Умови дозволяють вносити зміни у поведінку черепашки під час кожного проходження циклу.

Оператор **%** (modulo) дозволяє отримати решту від ділення. Код **if x % 2 == 0** перевіряє, яке число міститься у змінній **x**, парне чи не парне.



Заливка кольором

Щоб зафарбувати фігуру обраним кольором, використовують метод **begin_fill()** перед початком малювання та метод **end_fill()** у кінці малювання цієї фігури.

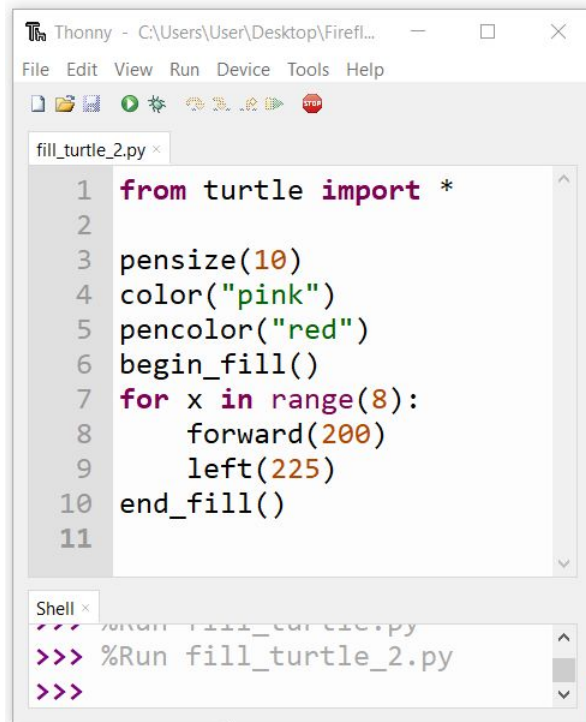


Колір пера

Щоб вибрати колір пера "черепашки", використовують метод:

pencolor("колір")

Також, можна використовувати метод **color("колір", "колір")** з двома аргументами. Перший буде відповідати за колір пера, а другий за заливку.

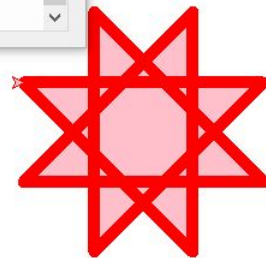


The screenshot shows the Thonny IDE interface. The main editor window displays a Python script named 'fill_turtle_2.py' with the following code:

```
1 from turtle import *
2
3 pensize(10)
4 color("pink")
5 pencolor("red")
6 begin_fill()
7 for x in range(8):
8     forward(200)
9     left(225)
10 end_fill()
11
```

Below the editor is a 'Shell' window showing the command to run the script:

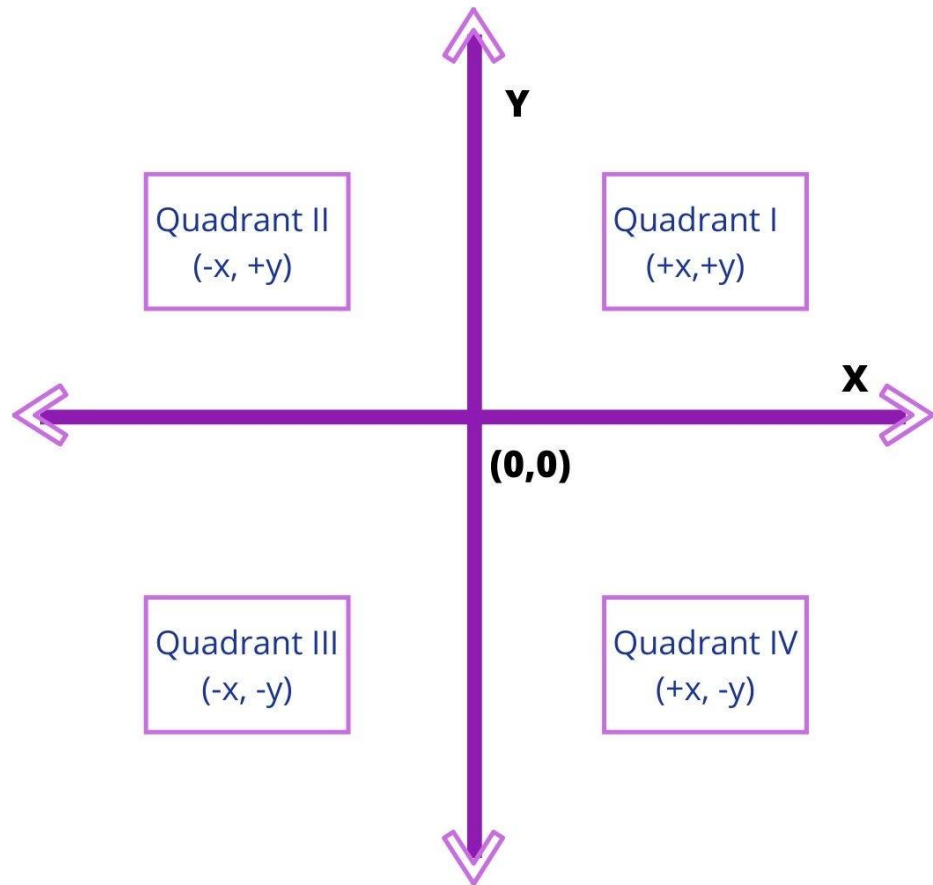
```
>>> %Run fill_turtle_2.py
>>>
```



Координати

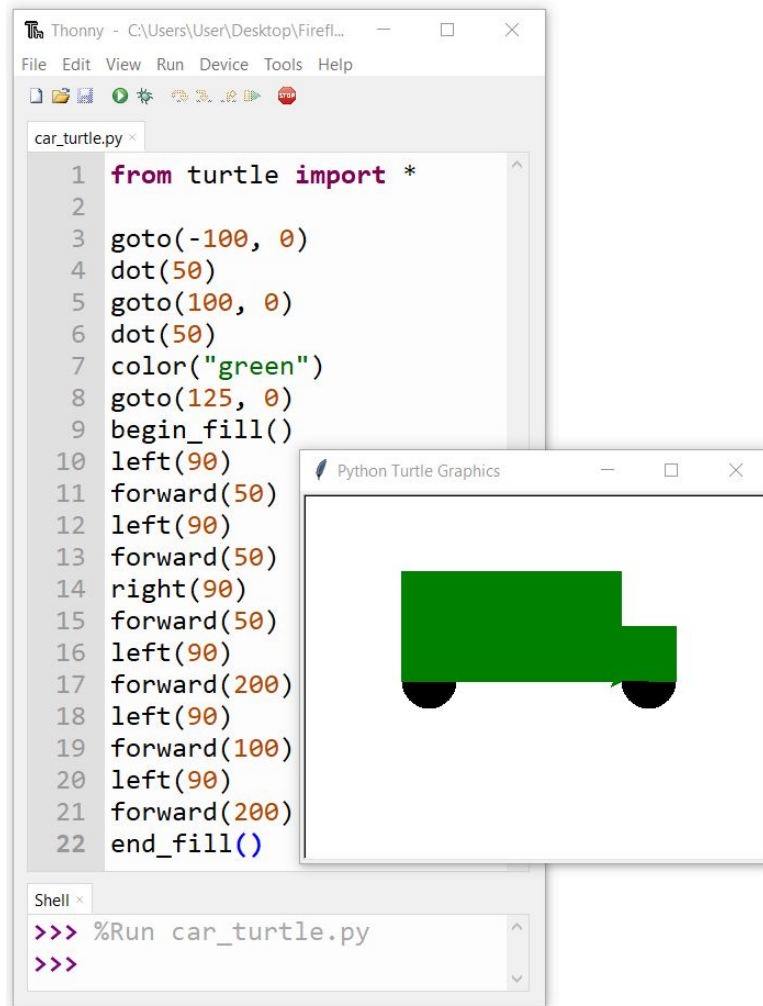
Щоб перемістити "черепашку" у потрібну точку на полотні, використовують метод:

`goto(50, 100)`



Складні фігури

Використовуючи координати, можна створювати зображення, які утворені з кількох фігур.



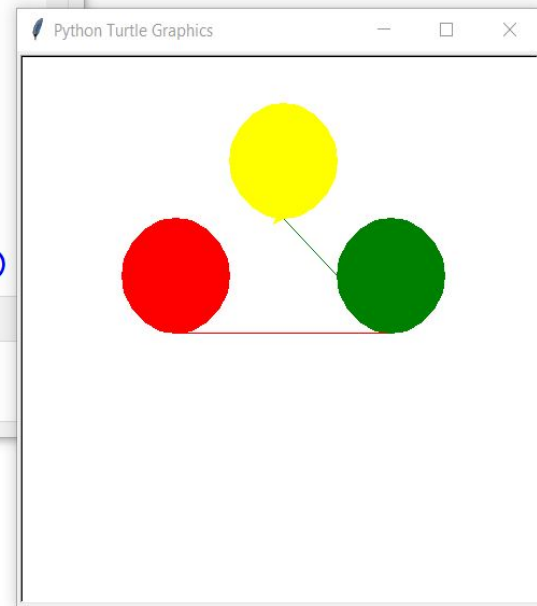
Функції і turtle

Використання власних функцій,
надає можливість створювати
складні та цікаві малюнки

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

def_turtle.py x
1 from turtle import *
2
3 def my_circle(clr):
4     color(str(clr))
5     begin_fill()
6     circle(50)
7     end_fill()
8
9 goto(-100, 0)
10 my_circle("red")
11 goto(100, 0)
12 my_circle("green")
13 goto(0, 100)
14 my_circle("yellow")

Shell x
>>> %Run def_turtle.py
>>>
```

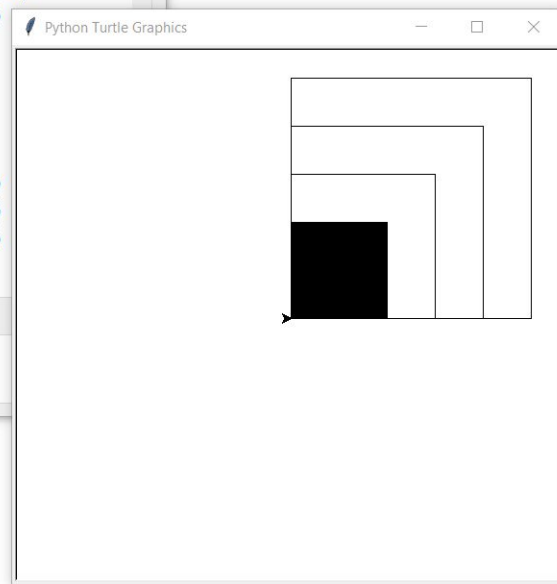


Функції і turtle

Використання власних функцій,
надає можливість створювати
складні та цікаві малюнки

```
Thonny - C:\Users\User\Desktop\Firefly_s02\PYT...
File Edit View Run Device Tools Help
def_turtle_2.py <
1  from turtle import *
2
3  def my_square(side, fill):
4      if fill == True:
5          begin_fill()
6      for x in range(1,5):
7          forward(side)
8          left(90)
9      if fill == True:
10         end_fill()
11
12 my_square(100, True)
13 my_square(150, False)
14 my_square(200, False)
15 my_square(250, False)
16

Shell <
>>>
```





Практична робота

- 1) Створіть новий файл **(New)**
- 2) Імпортуйте модуль **turtle**
- 3) Створіть візерунок з використанням набутих на цьому уроці навичок
- 4) Збережіть файл **(Save)** під назвою **my_cool_picture.py**
- 5) Виконайте запуск програми **(Run)**.



Підсумки

Дізнались додаткові можливості **модулю turtle** для створення малюнків

<https://realpython.com/beginners-guide-python-turtle/>