

# 1. Черепашача графіка

## Теорія:

У світі Python Черепашкою зветься уявний робот — пристрій, який переміщається по екрану й повертається в заданих напрямках, залишаючи за собою намальований слід заданого кольору й ширини.

Положення та напрямок руху Черепашки відображає невелика чорна стрілочка, яка повільно пересувається по екрану. Це дає змогу відстежити рух Черепашки та зрозуміти, яким чином кожен рядок коду впливає на траєкторію її руху.

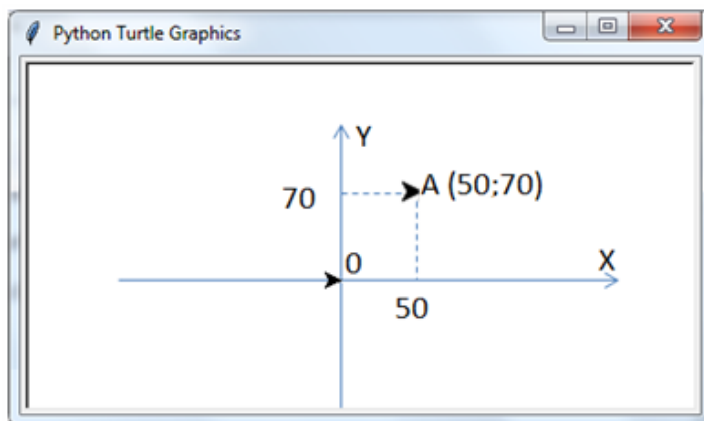
## Система координат

Як визначити положення Черепашки у графічному вікні? Для малювання нам доведеться користуватися орієнтирами не тільки по горизонталі, а й по вертикалі - визначити положення Черепашки на площині за допомогою координатних осей.

**Координатні осі — це дві координатні прямі, які перетинаються під прямим кутом. Центр вікна Python Turtle Graphics — точка перетину невидимих координатних осей — точка з координатами (0, 0).**

## Зверни увагу!

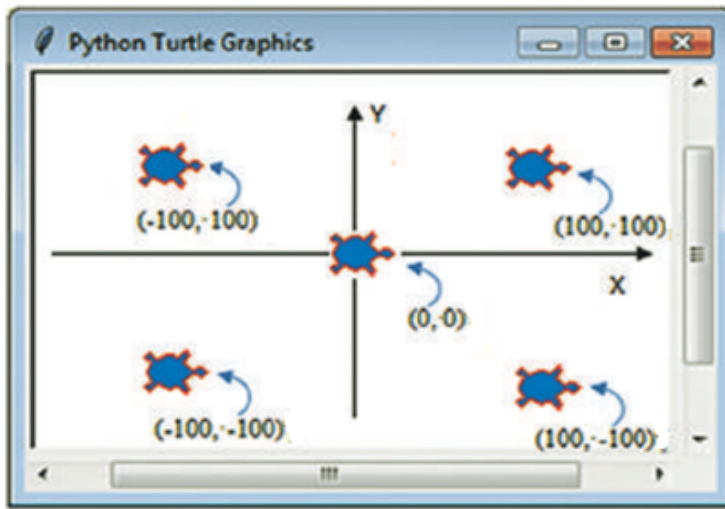
Вертикальна координата  $Y$  зростає знизу догори, а горизонтальна  $X$  — зліва направо.



Місцезнаходження Черепашки у вікні Python Turtle Graphics задається парою чисел  $(X, Y)$ . Початкове положення Черепашки - в точці  $O$ , яка має координати  $(0; 0)$ . Якщо Черепашку пересунути на 50 пікселів вправо і на 70 пікселів вгору, вона опиниться в точці  $A (50; 70)$ . Але як визначити координати Черепашки, якщо з точки  $O$  вона зрушить вліво чи донизу? На променях ліворуч і знизу від  $O$  розташовані від'ємні числа, тобто числа зі знаком мінус  $(-)$ .

## Приклад:

*Проаналізуй координати Черепашки на малюнку:*



## Команди Черепашки

Набір команд виконавця Черепашка міститься в модулі **turtle**.

**В Python модулем називається спосіб підключення корисного коду до іншої програми. Модулі зазвичай містять функції, до яких можна звертатися, щоб розширити перелік можливостей, що здатна виконувати програма.**

Тому для завантаження команд для роботи з Черепашкою потрібно програму почати з оператора:

```
from turtle import *
```

Після введення цієї команди ви можете давати Черепашці команди малювання.

Розглянемо команди малювання та їх призначення:

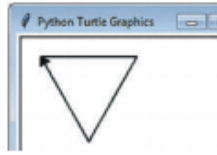
Команда	Призначення
forward (n)	Проповзти вперед n кроків (пікселів)
left (angle)	Повернути ліворуч на angle градусів
right (angle)	Повернути праворуч на angle градусів
circle (r)	Намалювати коло радіуса r, центр якого розташований ліворуч від Черепашки
circle (r,angle)	Намалювати дугу радіуса   r   із градусною мірою angle
goto (x,y)	Перемістити Черепашку в точку з координатами (x, y)
down ()	Опустити перо
up ()	Підняти перо
width (n)	Установити ширину сліду Черепашки в n пікселів
write (s)	Вивести текстовий рядок s у точці перебування Черепашки
clear ()	Очищення області малювання

### Приклад:

*Напишемо програму для малювання трикутника. Відкриємо вікно IDLE і створимо нове вікно програми. Напишемо код:*

```
from turtle import *  
forward (100)  
right (120)  
forward (100)  
right (120)  
forward (100)
```

Після збереження коду у файлі і виконання команди *Run / Run Module* побачимо малюнок:



Не забувай піднімати перо, якщо потрібно пересунути Черепашку, не залишаючи сліду.

### Приклад:

Намалюємо 2 кола:

```
circle(45) # малюємо коло з радіусом 45  
up() # піднімаємо перо  
goto(100, 50) # переходимо в потрібну точку  
down() # опускаємо перо  
circle(45) # малюємо друге коло
```