

## Графічні алгоритми і виконавці. Черепашка

Покликання для вивчення графіки turtle і для роботи над завданням

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

посібник для початківців – вхід на сайт можливо з реєстрацією

<https://docs.python.org/uk/dev/library/turtle.html> повна документація про turtle від розробників

<https://www.datacamp.com/tutorial/turtle-graphics>

графіка Python Turtle: цікавий спосіб вивчити основи

[https://en.wikipedia.org/wiki/Turtle\\_graphics](https://en.wikipedia.org/wiki/Turtle_graphics)

графіка черепахи; в статті є багато покликань на інші публікації

<https://education.launchcode.org/lchs/appendices/turtles/turtle-properties.html>

властивості Turtle

<https://www.plus2net.com/python/tkinter-colors.php>

довідник кольорів стандарту X11

Самостійне практичне вивчення графіки. Приклади графіки є в пункті меню Turtle Demo:

Запуск Python -> вікно Python Shell -> меню Help -> пункт Turtle Demo

вікно turtle graphics examples прикладів для виконання і вивчення графіки

---

Завдання 1. Виконати вправи у вікні Python Shell інтерпретатора коду. Вправи скласти самостійно подібно до поданих на лекції:

```
>>> import turtle
>>> s = turtle.getscreen()
>>> t = turtle.Turtle()
>>> t.shape("turtle")
>>> t.speed(1)
>>> t.left(45)
>>> t.forward(120)
>>> t.right(90)
>>> t.backward(80)
>>> t.right(90)
>>> t.forward(150)
```

Скласти для демонстрації команди різного виду так, щоб показати базові можливості виконавця графічних алгоритмів. Можливо, записати не один великий приклад, а 3-4 менших, з врахуванням методики покрокового навчання учнів.

У відповідь надіслати копії тексту команд з вікна Python Shell чи іншого середовища.

Завдання 2. Розробити алгоритм і програму у вікні редактора коду для малювання складнішого малюнка шляхом виконання цілої програми. Малюнок придумати самому. Для оцінки цього завдання буде враховано складність малюнка і використання різноманітних команд. Використати лише стандартні команди черепашки.

У відповідь надішліть текст програми.

Завдання 3. Скласти визначення 1-2 функцій def, які би малювали складні *прямолінійні* фігури, і використати їх як незалежні функції для деякої програми малювання.

У відповідь надіслати текст програми малювання разом з визначеними функціями.

Завдання 4. Оскільки виконавець черепашки вміє малювати з усіх стандартних фігур лише "коло" і "крапка", то спробувати розшукати або скласти самому функцію малювання складнішої *криволінійної* фігури, наприклад, "еліпс", "серце", "сонце".

Надіслати текст функції разом з прикладом її застосування.

---

**Надіслати у відповідь в цілому:**

1) Файли всіх прикладів для завдань 1, 2, 3, 4 – окремими файлами в тому форматі, як визначено в завданні, або одним спільним файлом. Програмний код має бути коментований за правилами python.

У випадку текстового формату txt чи іншого обирайте кодування UNICODE. Якщо формат не визначено - обирайте довільний, в тому числі можна pdf.

2) Знятки екрана свого комп'ютера (отримані малюнки) за результатами виконання завдань з підписами.

3) Не забувайте нумерувати відповіді згідно номерів завдань і підпунктів.