

Тема. Практикум з використання інформаційних технологій

Очікувані результати заняття

Після цього заняття потрібно вміти:

- створювати проекти на мові Python.

Поміркуйте

- Що таке підпрограма?
- Як організувати підпрограму мовою Python?

Перегляньте відео на повторення

[Робота в середовищі Repl.it](#)

Шпаргалка

Підпрограма – це логічно незалежна спеціальним чином оформлена частина програми для розв'язування певної задачі. До підпрограм можна багаторазово звертатися з інших частин програми. Таке звернення називають **викликом підпрограми**.

У Python усі підпрограми називають **функціями**. Кожна функція це «державка в державі», мініпрограма в програмі. В ній діють всі закони, що притаманні будь-якій програмі. Функція являє собою набір команд, як і сама програма, але вона розв'язує менш складне завдання.

Інколи функції групуються у **модулі**. Ми вже розглядали модуль підключення графічних можливостей **turtle**. В модулях зберігається опис його функції, які можна підключати і використовувати в будь-якій частині проекту або навіть в інших проектах. Існують вбудовані функції і **функції, створені користувачем**. **Вбудовані функції** є частиною мови і можуть викликатися за іменем без попереднього оголошення. Наприклад, **forward(30)** це ні що інше, як виклик вбудованої функції, яка прописана в стандартному модулі **turtle** і промальовує відрізок довжиною у 30 пікселів у напрямку руху.

Будь-яка функція має такий загальний вигляд:

```
def <назва> ([параметри]):  
    <блок команд>  
    [return <результат>]
```

Параметр (також називають аргументом) – це змінна, яка отримує конкретне значення під час звернення до функції. Параметри вказувати не обов'язково, але при цьому круглі дужки опускати не можна.

Інструкція **return** повертає результат виконання функції в основну програму.

Перегляньте відео

[Проект «Зоряне небо»](#)

Запитання до відео

- Які алгоритмічні структури розглянуто у відео?
- Які ще процеси можна реалізувати за допомогою такого коду?

Робота над проектом

1. Відкрийте середовище програмування, наприклад <https://replit.com/languages/python3>.

2. Напишіть код і запустіть на виконання, за потреби зробіть виправлення, керуючись вказівками у відео.

```
1 from turtle import*
2 from random import*
3
4 w = Screen()
5 w.bgcolor("dark blue")
6
7 def star(x,y):
8     t=Turtle()
9     t.hideturtle()
10    t.speed(0)
11    t.pensize(4)
12    t.color("yellow")
13
14    t.up()
15    t.goto(x,y)
16    t.down()
17
18    n=randint(5,7)
19    t.left(randint(1,100))
20    l=randint(5,10)
21
```

```
22 for i in range(n):
23     t.fd(1)
24     t.bk(1)
25     t.left(360/n)
26
27 del t
28
29 w.onclick(star)
30 w.mainloop()
```

3. Внесіть в код потрібні доповнення і корективи, щоб на «небі» відображались цілі сузір'я.
4. Надішліть файл з проєктом на HUMAN або на електронну пошту nataliartemiuk.55@gmail.com

Домашнє завдання

Повторити тему «Проекти з вікнами».

Джерело

[Магія двох бітів](#)