Графічні можливості модуля turtle

Розглянемо, що таке модуль.

Модуль - це самостійна програма, що розширює функціонал вже існуючих програм.

Функціонал - це перелік можливостей, що здатна виконувати програма.

Наприклад, ви їдете відпочивати за кордон і не володієте мовою, на якій розмовляють там. Ви можете купити розмовник, де розібрані вирази залежно від обставин. З розмовником розширюються ваші можливості у спілкуванні в конкретних ситуаціях, розширився функціонал спілкування.

Для того, щоб мати змогу створювати графічні об'єкти, до програми треба підключити модуль turtle (з англ. «черепашка»). Згадайте, черепаху з казки Олексія Миколайовича Толстого «Золотий ключик або Пригоди Буратіно». Її звали **Тортила**. Тепер зрозуміло, чому.

Підключення модуля turtle у Python відбувається наступним чином:

from turtle import *

Ця стрічка розширює графічний функціонал.

Тепер ми можемо створювати різні картинки. Після запуску такого роду програм відкривається окреме вікно (з назвою **tk**) з вказівником, що має вигляд трикутника. Будемо називати його «**черепашкою**».

Мал. 1

Черепашка дивиться строго направо і готова виконувати команди. Малює вона ніби пером. За настройками напочатку виконання перо в неї опущене і при пересуванні буде залишати слід. Яким же чином черепашку можна пересувати?

Команди переміщення черепашки

forward(60) - проповзти вперед 60 кроків (пікселів).

left(50) - повернутися наліво на 50 градусів.

right(90) - повернутися направо на 90 градусів.

circle(30) - намалювати коло радіуса 30, яке повністю знаходиться зліва від положення черепашки.

circle(-30) - намалювати коло радіуса 30, яке повністю знаходиться справа від положення черепашки.

Виявляється крім цих команд, черепашка ще орієнтується у вікні як у системі координат. Напочатку виконання програми вона знаходиться у точці (0, 0) і готова перейти у будь-яку точку з заданими координатами. Її система координат нічим не відрізняється від звичної для нас:

Мал. 2

Ця система координат невидима. Команда переміщення черепашки у координати (50, 90) має вигляд

goto(50, 90)

При цьому напрямок, в якому дивиться черепашка не змінюється, ця команда не повертає черепашку.

Команди малювання

down() - опустити перо. Після цієї команди черепашка почне залишати слід при будь-якому своєму пересуванні.

up() - підняти перо. Після цієї команди при пересуванні черепашка не буде залишати слід, поки ми знову не дамо команду **down()**

width(3) - встановити товщину сліду (пера) черепашки 3 пікселів.

color('red') - встановити червоний колір пера (сліду) черепашки. Колір має бути текстовим рядком з назвою кольору англійською мовою, наприклад, ‘red', ‘yellow', ‘green', ‘blue', ‘brown'. Без використання цієї команди черепашка малює чорним пером.

fill() - використовується для малювання зафарбованих областей. Починаючи малювати зафарбовану область, дайте команду fill(1), а закінчивши малювання області - fill(0).

Зверніть увагу! У версіях Python 2.6 та вище перед початком малювання області, що має заливатися, замість fill(1), треба писати команду **begin_fill()**, а закінчивши малювання області, замість fill(0), треба писати команду **end_fill()**

1 урок

https://www.programiz.com/python-programming/online-compiler/

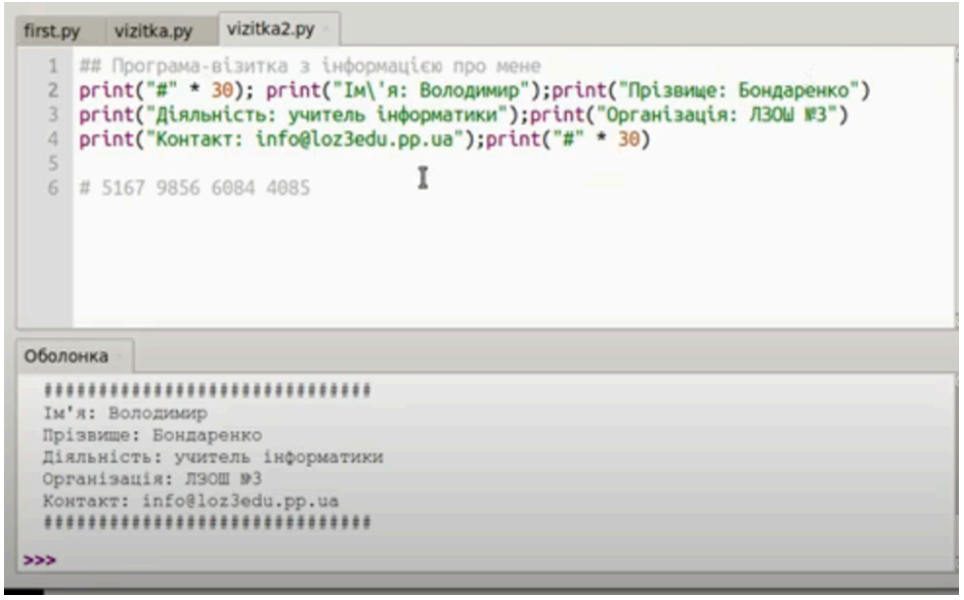
```
print(“Привіт!”)
print("Як тебе звати?")
name=input(‘Введи своє імя’)
print("Радий тебе вітати", name)
```

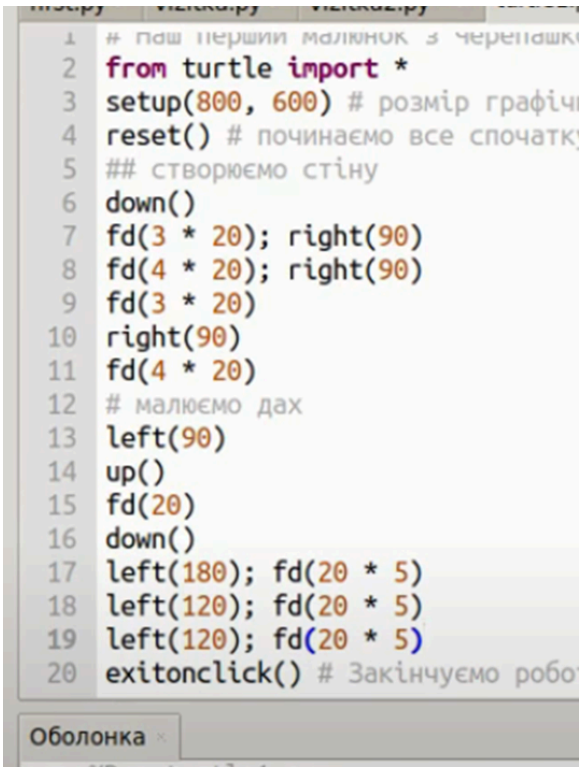
https://replit.com/

from tkinter import*

Window1=Tk()

Window1.mainloop()





https://replit.com/

```
from tkinter import*
Window1=Tk()
Window1.title('i')
Window1['bg']='green'
Window1.mainloop()
```

https://replit.com/

2 урок

```
print(“Hello world!”)
print("Привіт", " Мене звати Тетяна")
print("речення1", " речення2.")
```

input – введення

```
print("Привіт", input (" Яке ваше прізвище"),input (" Яке ваше імя"))
```

python Copy code

```python
# Запросити від користувача його ім'я, прізвище, школу та предмет
name = input("Введіть ваше ім'я: ")
surname = input("Введіть ваше прізвище: ")
school = input("Введіть назву вашої школи: ")
subject = input("Введіть предмет, з якого підписуєте зошит: ")

# Відкрити файл для запису
with open('зошит.txt', 'w') as file:
    # Записати дані в файл
    file.write("Зошит з Інформатики\n")
    file.write("Учень 8 класу\n")
    file.write("Середня школа №35\n")
    file.write("{} {}\n".format(surname, name))
    file.write("\n")
    file.write("Предмет: {}\n".format(subject))
    file.write("\n")
```

Regenerate response

3 урок

python Copy code

```python
# Запросити від користувача його ім'я, прізвище, школу та предмет
name = input("Введіть ваше ім'я: ")
surname = input("Введіть ваше прізвище: ")
school = input("Введіть вашу школу: ")
subject = input("Введіть предмет з якого підписуєте зошит: ")

# Вивести на екрані підписаний зошит
print("\n\n\n")
print("========================================")
print("           ПІДПИС НА ЗОШИТІ            ")
print("========================================")
print("\n")
print("              {} {}              ".format(name, surname))
print("\n")
print("              Школа: {}              ".format(school))
print("              Предмет: {}            ".format(subject))
print("\n")
print("========================================")
```

Regenerate response

№3.1

#Запросити від користувача його ім'я,прізвище, школу,клас та предмет

name=input("Введіть ваше ім\'я:'")

surname=input("Введіть ваше прізвище")

school=input("Введіть назву школи")

clas=input("В якому класі ви навчаєтесь")

subject=input("Введіть предмет з якого підписуєте зошит")

#Вивести на екрані підписаний зошит

print("\n\n\n")

print("================================")

print(" ЗОШИТ ")

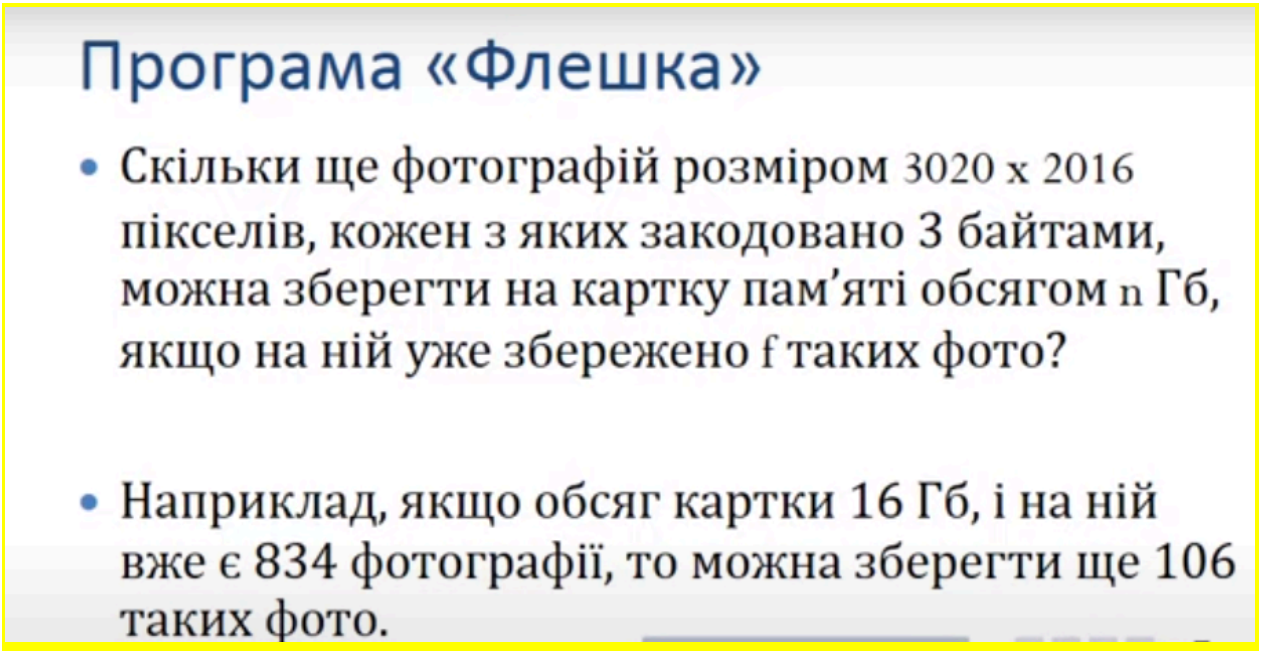
```
print("          з",subject)
print("          учня",clas,"класу")
print("          ",school)
print("          ",name,  surname)
print("="*30)
print("\n")
```

lesson 4

```
#отримуємо від користувача значення довжини та ширини клумби за допомогоб функції "input()"
a=float(input("Введіть довжину клумби:"))
b=float(input("Ведіть ширину клумби:"))
#знаходимо площу клумби
area=a*b
#знаходимо довжину огорожі
fence_length=2*(a+b)
#знаходимо кількість сажанців
```

```
plants_needed=area*4
#результат виводимо за допомогою функції print
print("Довжина огорожі:",fence_length)
print("Кількість саджанців, які потрібно закупити:",plants_needed)
```

lesson 5




```
width = 3020  # ширина фотографії
height = 2016  # висота фотографії
bytes_per_pixel = 3  # кількість байтів на піксель
photos_saved = int(input("Введіть кількість вже збережених фотографій: "))
memory_card_size = int(input("Введіть обсяг картки пам'яті в Гб: ")) * (1024 ** 3)  # перетворення Гб в байти
photo_size = width * height * bytes_per_pixel  # розмір однієї фотографії в байтах
remaining_space = memory_card_size - (photos_saved * photo_size)  # залишок місця на картці пам'яті
num_photos = remaining_space // photo_size  # кількість фотографій, які можна зберегти
print("На картку пам'яті можна зберегти ще", num_photos, "фотографій.")
```

lesson 6

колір фону напису ставав *жовтим*, колір символів – *червоним*, розмір символів – *14*, у напис виводився заданий текст. На малюнку 6.35 наведено текст відповідного проєкту та результат виконання процедури – обробника події **Click** для напису.

```
from tkinter import *
def label_click(event):
    label['text'] = 'Я навчаюсь у 8 класі'
    label['font'] = 14
    label['fg'] = 'red'
    label['bg'] = 'yellow'
root=Tk()
root.geometry('300x200')
label = Label(text = 'Це напис')
label.bind('<1>', label_click)
label.pack(pady = 20)
```

Мал. 6.35. Обробник події **Click** для напису та результат його виконання

from tkinter import*

def label click(event):

def label click (event):