

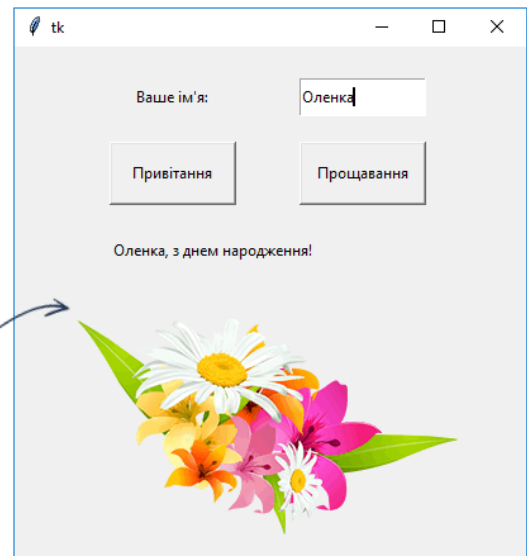
## Урок 24. Зображення

### Вивчення нового матеріалу

#### Слайд № 1

Створена на попередньому уроці програма виводила лише текст привітання певній людині. Сьогодні ми змінимо цю програму так, щоб подарувати дорогій нам людині справжню листівку зі словами привітання та квітами.

У разі натискання кнопки **Привітання** буде виведено текст привітання та вітальне зображення.



Спочатку змінимо напис.

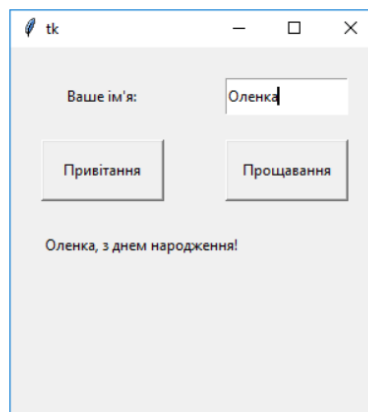
Далі

### Вправа

#### Вправа № 1



**Вправа 1.** Замініть команду, що виводила звичайне привітання, на привітання з днем народження у форматі **“Оленка, з днем народження!”** або **“Андрійко, з днем народження!”**.



Змініть код програми, створеної на попередньому уроці.  
Запустіть програму та переконайтеся у правильності її виконання.

Готово

Слайд № 2

### Додамо до привітання зображення.

Для виведення зображення у вікні необхідно створити **полотно** для малювання – об'єкт **Canvas**.

Для створення полотна у конструкторі необхідно вказати його ширину (**width**) та висоту (**height**), а потім виконати команду розташування у вікні:

```
canvas=Canvas(width=500,height=500)
canvas.pack()
```

Метод **pack** розташовує полотно під попереднім об'єктом. Він застосовується до об'єкта Canvas так само, як і до інших об'єктів.

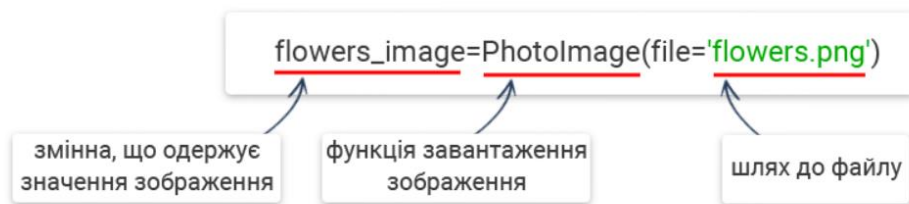


Замість чисел 500 можна вказати інші ширину та висоту.

Слайд № 3

### Розташуємо на полотні зображення квітів.

Для завантаження зображення до програми слід скористатися такою командою:



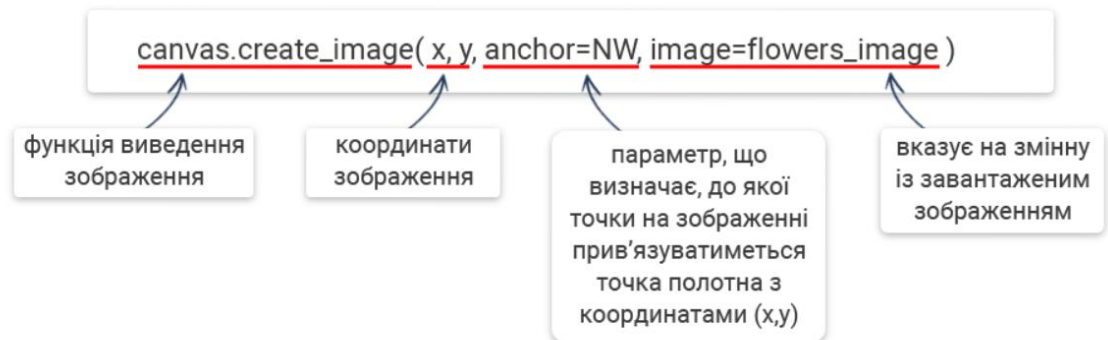
Якщо ім'я файлу зображення вказане без шляху, то файл має розміщуватися **у тій же папці, що і сама програма**.



Засобами tkinter, крім інших, можна завантажити файли графічних форматів **.gif** та **.png**. Їх перевага у тому, що вони підтримують прозорий фон зображення.

#### Слайд № 4

Але для виведення зображення на полотні команди `flowers_image=PhotoImage(file='flowers.png')` недостатньо. Необхідно вказати, у якому місці полотна воно відображатиметься. Для цього використовують команду:



Значення `anchor=NW` означає, що координати `x,y` відповідають лівому верхньому куту зображення (North West — північний захід).



Запам'ятайте синтаксис команди виведення зображення на полотні.

#### Слайд № 5

Звичайно атрибут `anchor` може набувати й інших значень. На рисунку представлено його можливі значення та відповідні точки прив'язки.

NW	N	NE
W	CENTER	E
SW	S	SE

Зауважимо, що якщо аргумент `anchor` опущений, то це означає прив'язку до центра зображення.



Запам'ятайте призначення та способи використання атрибута `anchor`.

#### Слайд № 6

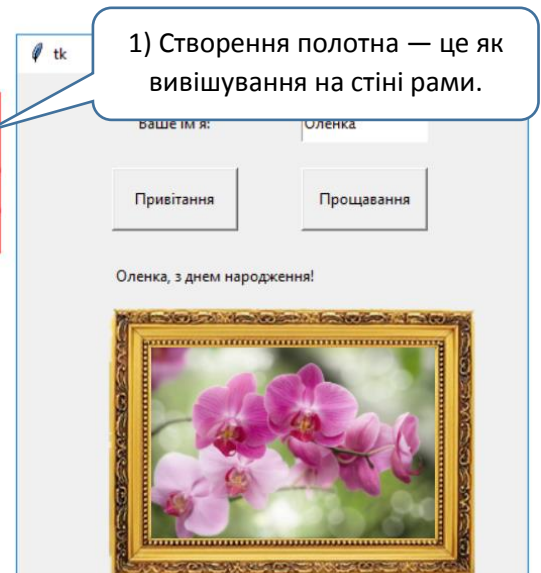
Отже, зображення у вікні виводить такий код:

```
canvas=Canvas(tk,width=400,height=350)
canvas.pack()
flower_image=PhotoImage(file="flowers.png")
canvas.create_image(200,175,image=flower_image)
```

2) Готуємо саму картину.

3) Розміщуємо картину в рамі.

Це дуже схоже на те, якби ми вивішували на стіні картину!



## Вправа № 2

**Вправа 2. Внесіть у програму такі зміни:**

- 1) розширте вікно програми по вертикалі до 600 пікселів;
- 2) створіть графічне полотно розміром 400x350 і розмістіть його під вітальним написом;
- 3) додайте коди виведення зображення з файлу **flowers.png** у разі натискання кнопки **Привітання**.

У разі використання методу **pack** об'єкт розташовується після попереднього об'єкта. Для точного розташування у вікні скористайтеся методом **place**.

```
from tkinter import *
tk.geometry("400x600")
canvas=Canvas(tk,width=400,height=350)
canvas.pack()
flower_image=PhotoImage(file="flowers.png")
def btn1_click():
    lbl2=Label(text=ent.get()+" , з днем народження!")
    lbl2.place(x=75,y=150)
    canvas.create_image(200,175,image=flower_image)
...
```



Для відображення полотна скористайтеся методом **pack** або **place**.  
Файл **flowers.png** завантажте з онлайнного підручника.

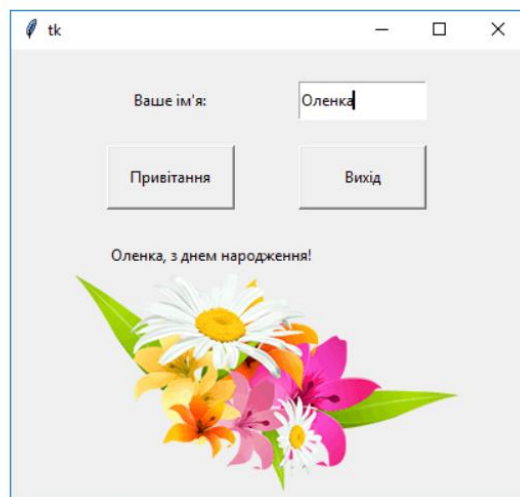
## Вивчення нового матеріалу

## Слайд № 7

Замінімо кнопку **Прощай** кнопкою **Вихід**. У разі натискання на неї буде закриватися вікно програми.

Для завершення програми використовують команду **tk.destroy**.

`command=tk.destroy`



Щоб прив'язати команду **tk.destroy** до події натискання кнопки, надайте відповідне значення параметру **command** в конструкторі кнопки: **command = tk.destroy**.

## Слайд № 8

Вкажіть дії, які необхідно виконати для заміни кнопки **Прощавання** на кнопку **Вихід**.

```
...
def btn1_click():
    lbl2=Label(text=ent.get()+" , з днем народження!")
    lbl2.place(x=75,y=150)
    canvas.create_image(200,260,image=flower_image)
def btn2_click():
    print("Прощавай, ",ent.get(),"!",sep=" ")
btn1=Button(text="Привітання",command=btn1_click)
btn1.place(x=75,y=75,width=100,height=50)
btn2=Button(text="Прощавання",command=btn2_click)
btn2.place(x=225,y=75,width=100,height=50)
...
```

- ☐ замінити властивість **text** для **btn1**
- ☒ замінити властивість **text** для **btn2**
- ☒ для **btn2** замінити значення **command** на **tk.destroy**
- ☒ видалити функцію **btn2\_click**

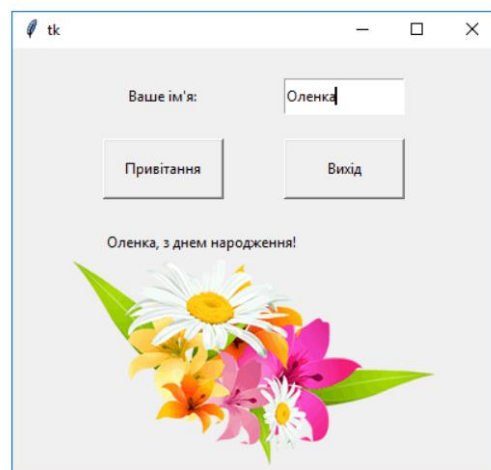


Оберіть усі правильні відповіді.

## Слайд № 9

Заповніть пропуски у коді створення кнопки **Вихід**.

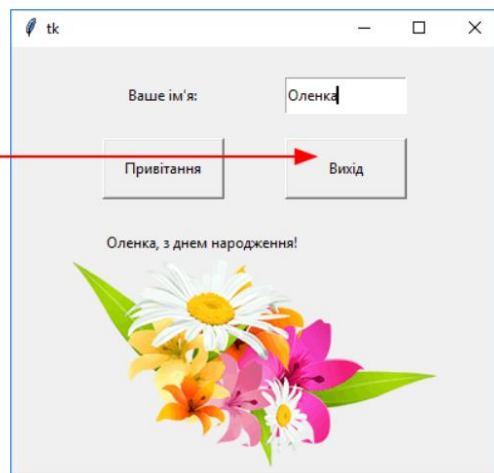
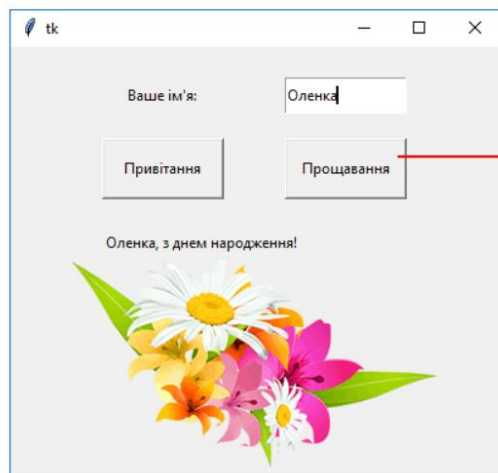
```
btn2=Button(text="Вихід",command=tk.destroy)
btn2.place(x=225,y=75,width=100,height=50)
```



Заповніть пропуски. Зауважте, що кнопка має виконувати команду **tk.destroy**.



**Вправа 3. Замініть кнопку Прощавання на кнопку Вихід.**  
Запрограмуйте вихід із програми по натисканні цієї кнопки.



Змініть програму, запустіть її та переконайтеся у правильності виконання.  
Зауважте, що вихід із програми виконує команда `tk.destroy`.