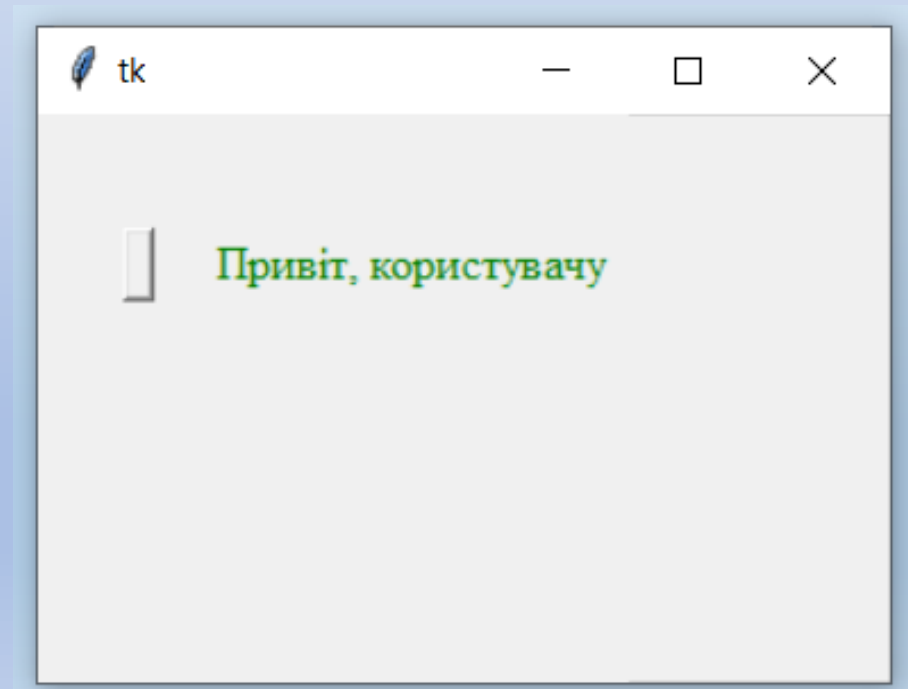




Елемент керування  
«*Напис*», «КНОПКА».

# Властивості напису

*Напис* — це компонент (елемент керування), який використовується для виведення текстових повідомлень.



# Алгоритм створення напису мовою Python

Для цього потрібно в тексті проєкту:

1. Увести команду створення напису.
2. Установити значення його властивостей.
3. Розмістити напис у вікні.

Команда створення напису має вигляд:

*<ім'я напису>=Label(text='<текст напису>')*

# Створення напису мовою Python

Встановлення напису у вікні проєкту робиться аналогічно, як із кнопкою. Кожному рядку потрібно дати ім'я та вказати функцію для його створення.

*Label* - (англ. етикетка, позначка) — команда створення напису.

Наприклад, для створення напису з іменем *label* і текстом *Я вчуся у 8 класі* потрібно виконати команду:

```
label = Label(text = 'Я вчуся у 8 класі')
```

## Атрибути:

**bg**=колір фону напису;

**text**=текст напису;

**fond**=шрифт тексту напису його розміри;

**fg**=колір символів;

**height**=висота напису;

**width**=ширина напису

та інші.

Наприклад, щоб створити напис зеленими літерами на жовтому фоні із текстом *Я навчаюся у 8 класі*, шрифт символів *Arial*, розмір символів 14, необхідно виконати команду:

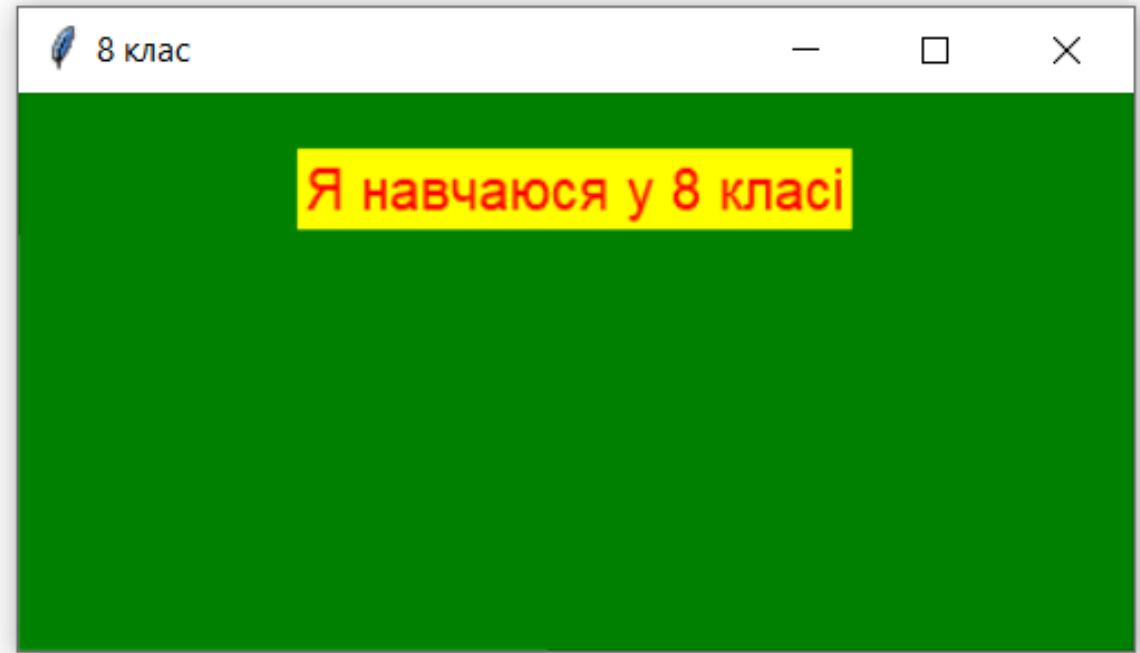
```
label = Label(text = 'Я вчуся у 8 класі', bg='yellow', fg='green', font='Arial 14')
```

# Створимо напис мовою Python


IDLE Shell 3.11.1

File Edit Shell Debug Options Window Help

```
Python 3.11.1 (tags/v3.11.1:a7a450f, Dec 6 2022, 19:58:39) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> from tkinter import*
>>> root=Tk()
>>> root.title('8 клас')
>>> ''
>>> root.geometry('400x200+200+100')
>>> ''
>>> root['bg']='green'
>>> label=Label(text = 'Я навчаюся у 8 класі')
>>> label.pack(pady = 20)
>>> label['font'] = 14
>>> label['fg'] = 'red'
>>> label['bg']='yellow'
>>>
```



# Створимо кнопку події з написом

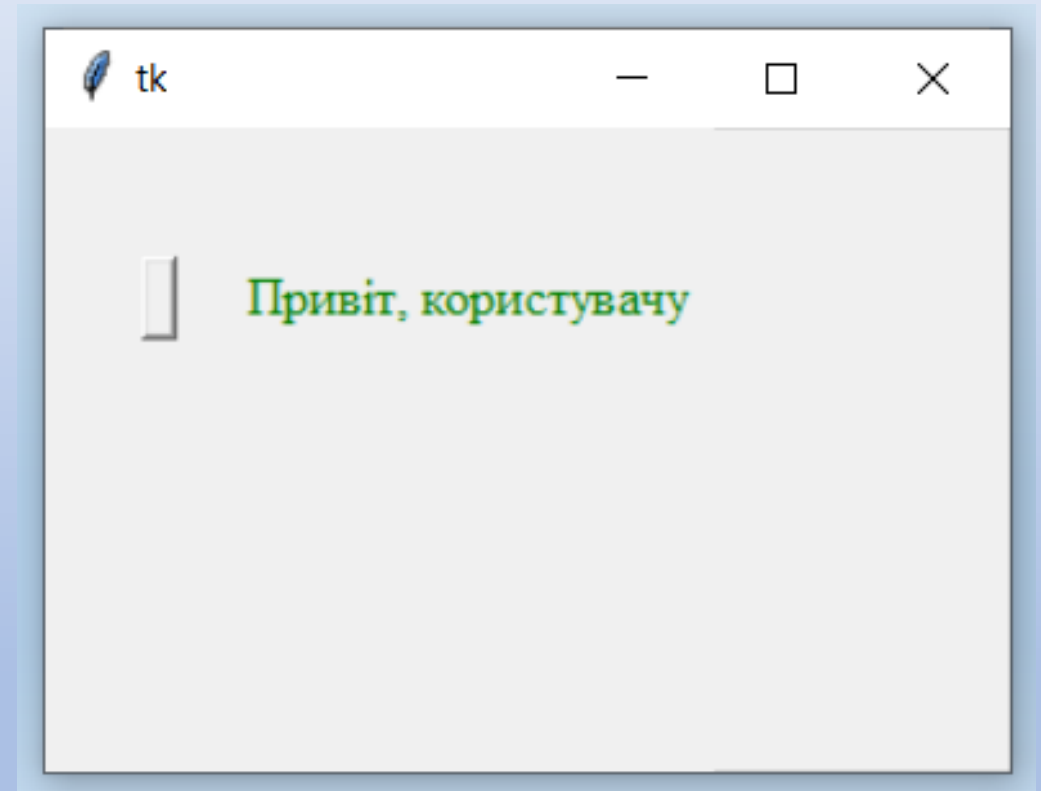
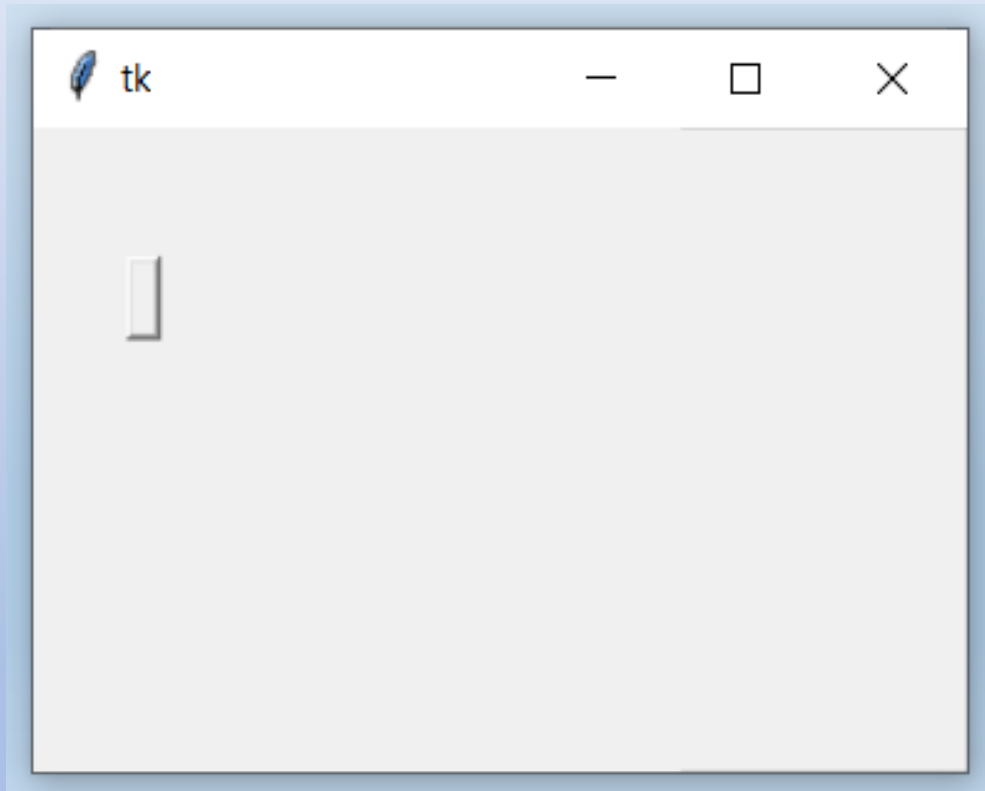
 IDLE Shell 3.11.1

File Edit Shell Debug Options Window Help

```
Python 3.11.1 (tags/v3.11.1:a7a450f, Dec 6 2022, 19:58:39) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> from tkinter import*
>>> def fun1(event):
...     labell=Label (Window, text="Привіт, користувачу", font="Times 12", fg="green")
...     #створення напису
...     labell.place(x=60, y=40)
...     #розміщення напису
...
...
>>> Window=Tk()
>>> Window.geometry("300x200")
...
>>> but=Button(Window)
>>> but.place(x=30, y=40)
>>> but.bind('<Button>', fun1)
```



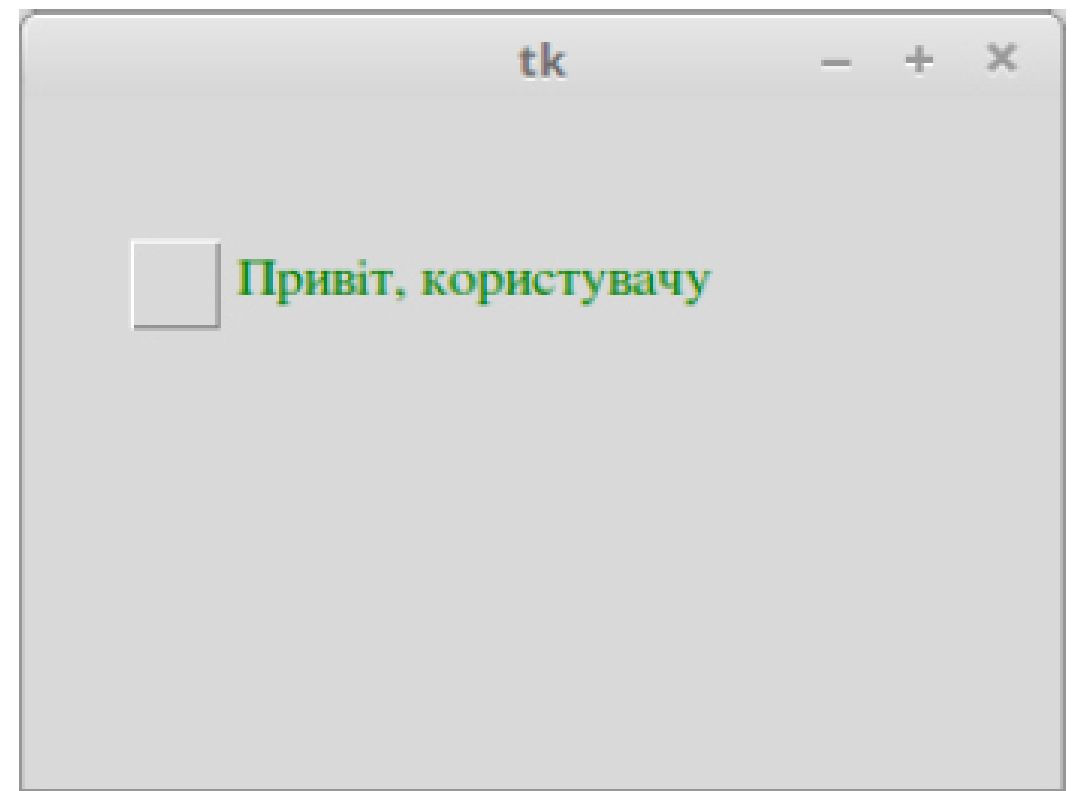
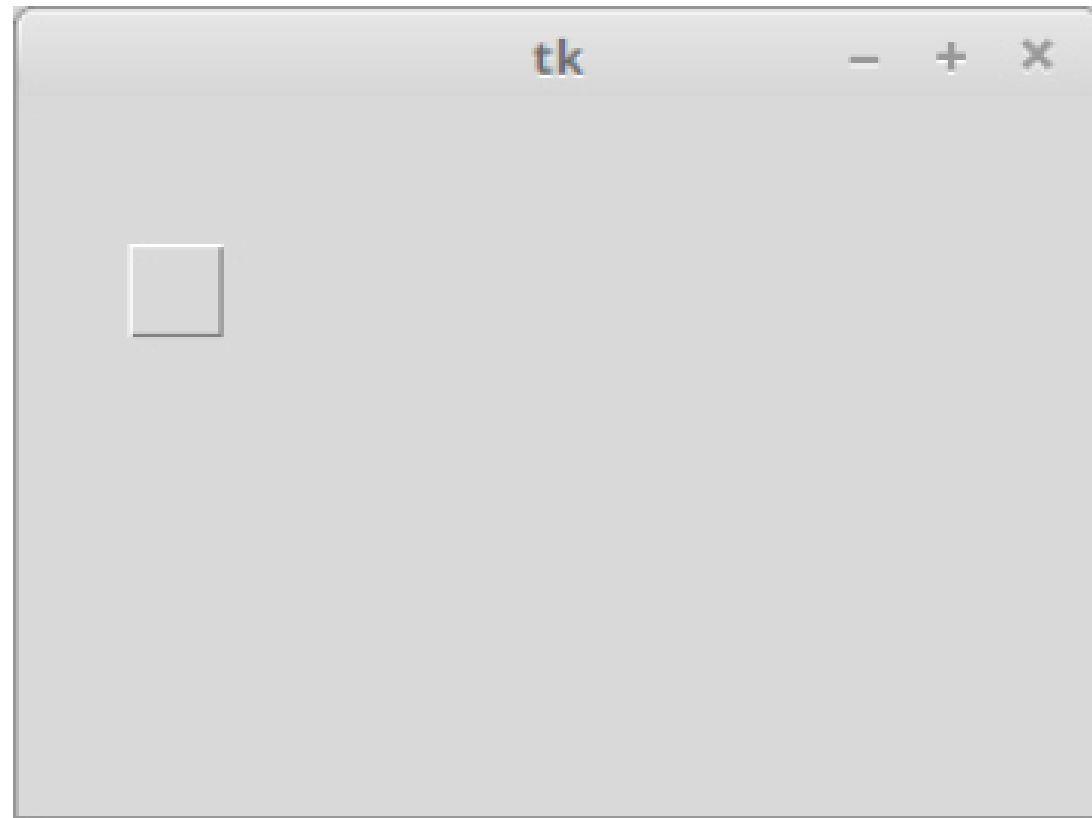
# Запустимо нашу команду натиснувши на кнопку



Створимо подію Button-1 та функцію fun1 та додамо відповідний КОД:

```
from tkinter import *
def fun1(event):
    labell=Label(Window, text="Привіт, користувачу",
        font="Times 12", fg="green")
    # створення напису
    labell.place(x=60, y=40)
    # розміщення напису
Window=Tk()
Window.geometry("300x200")
but=Button(Window)
but.place(x=30, y=40)
but.bind("<Button-1>", fun1)
# присвоєння події до кнопки
Window.mainloop()
```

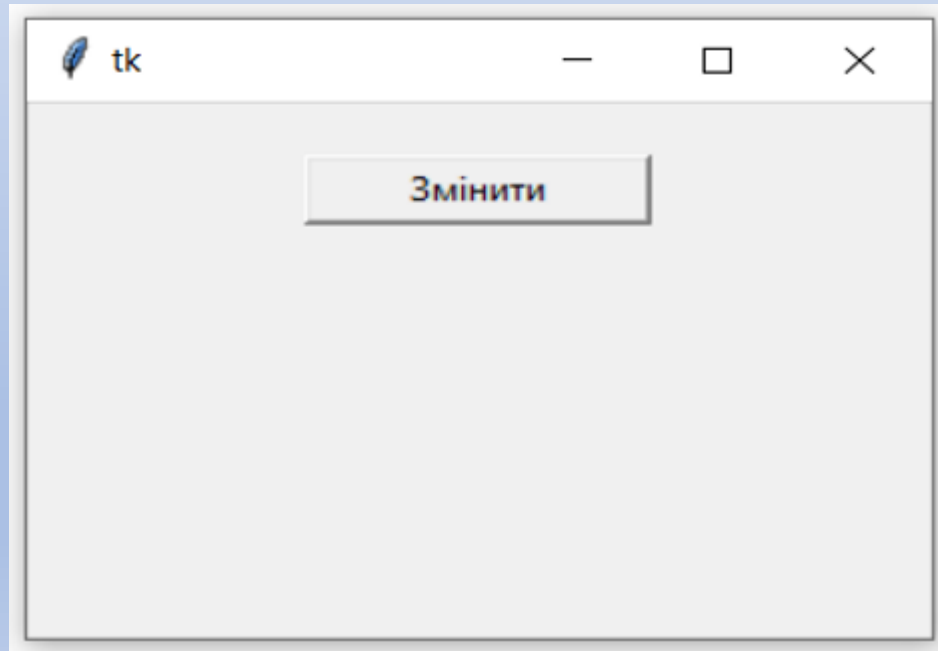
Запустимо нашу програму (мал. 23)



Мал. 23

# Властивості кнопки

- Одним з компонентів (елементів керування), який можна використати в проектах, є **кнопка**.



# Поняття елементів керування. Обробники подій.

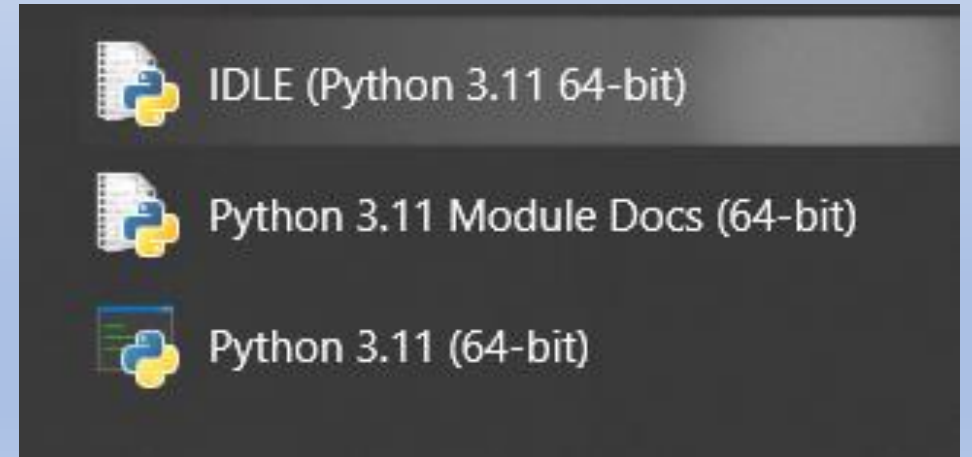
- **Елементи керування** — це графічні об'єкти, розташовані на вікні програми для показу або введення даних, виконання дій або полегшення роботи: текстові поля, списки, перемикачі, кнопки, прапорці тощо.
- **Обробник події** — це частина програми, що виконує певні дії у відповідь на подію (спричинену, наприклад, дією користувача).

# Поняття елементи керування

- Елементи керування можна умовно описати за формулою:
- **Елементи керування=властивості (стан)+методи (дії)+події (зворотні зв'язки)**

# Кнопка у вікні

- Щоб розмістити кнопку у вікні, потрібно:
- 1. Створити новий об'єкт типу **Button**, пов'язати його зі змінною, яка визначатиме ім'я об'єкта.
- 2. Установити значення властивостей кнопки або залишити їх за замовчуванням
- 3. Розмістити створені об'єкти у вікні.



Для створення кнопки та пов'язування її із змінною використовують команду вигляду:

**<ім'я\_кнопки>= Button()** - буде створено кнопку зі значеннями властивостей **висота, ширина, колір**, які встановлено за замовчуванням, без напису на ній.

**АБО**

**<ім'я\_кнопки>= Button(<набір властивостей та їх значень>)** – буде створено кнопку, значення властивостей якої задано в цій команді. У дужках через кому вказують пари **<властивість>=<значення>**.



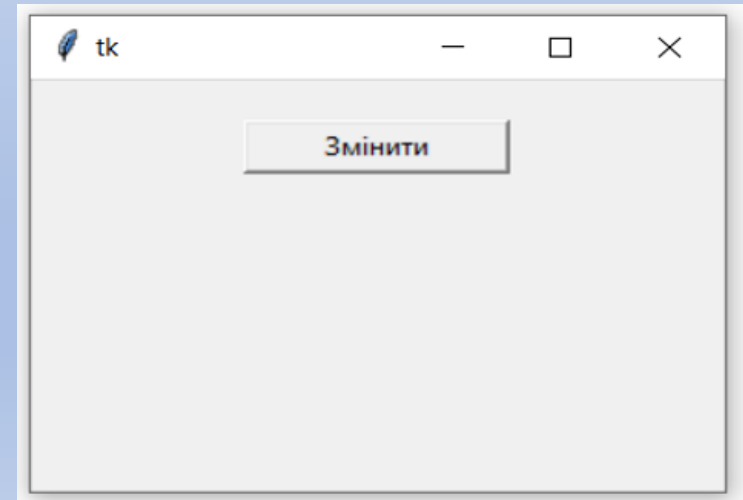
# Властивості об'єкту Кнопка

- Так само, як і напис, об'єкт кнопка може мати і свої властивості, які описують службовими словами:
- **text** – напис на кнопці;
- **width** – ширина кнопки (кількість символів, які можуть бути розміщені поруч по горизонталі);
- **height** – висота (довжина) кнопки (кількість символів, які можуть бути розміщені один під одним по вертикалі);
- **bg** – колір фону кнопки (значенням може бути назва кольору англійською);
- **fg** – колір символів тексту;
- **font** – шрифт і його розмір;
- **padx** – відступ від межі до об'єкта по горизонталі;
- **pady** – відступ від межі до об'єкта по вертикалі;
- **state** – стан кнопки (**normal** – звичайний, **disable** – недоступний) та інші.

# Наприклад,

кнопка з іменем **button** з текстом **Змінити** завширшки 15 символів може бути створена командою:

```
button= Button(text='Змінити', width=15
```



# Розміщення кнопки у вікні:

- **pack()** – кнопка автоматично розміщується під рядком заголовка вікна з однаковими відступами від лівої та правої меж вікна. Якщо у вікні розміщується кілька кнопок, то вони будуть відображатися одна під одною без відступів між ними. Якщо потрібно розмістити кнопку з деяким відступом від рядка заголовка або від попередньої кнопки, то в дужках можна вказати цей відступ у пікселях як значення властивості **pady**.

Наприклад, після виконання команди **button.pack(pady=20)** кнопку буде розміщено з відступом 20 пікселів від верхньої межі вікна або від попередньої кнопки.

## Розміщення кнопки у вікні:

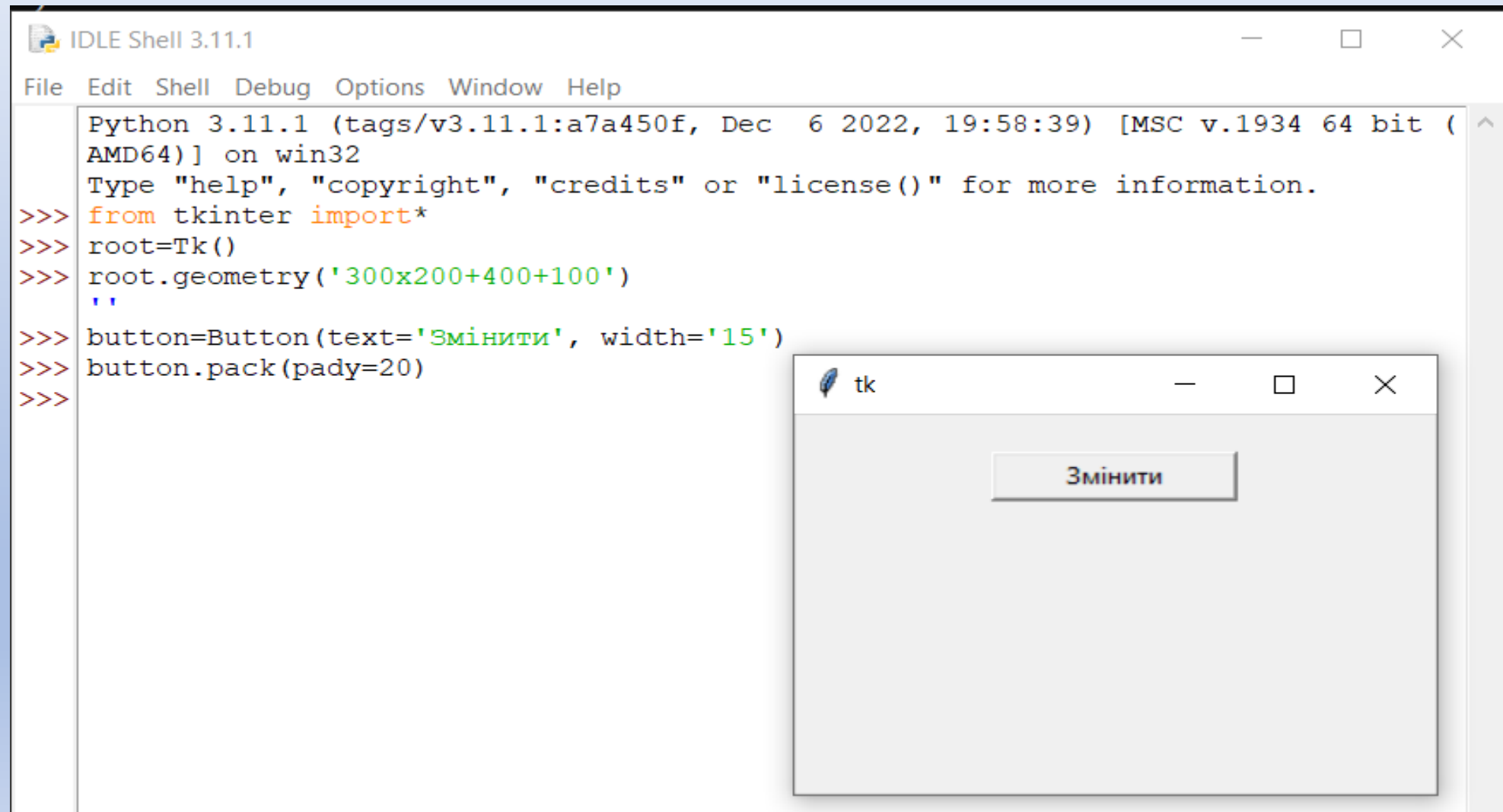
- **place(x=<відступ від лівої межі вікна>,y=<відступ від верхньої межі вікна>** - верхня ліва вершина кнопки розміщується в точці вікна із заданими координатами x та y.

Наприклад, після виконання команди **button.place(x=200, y=100)**, верхня ліва вершина кнопки буде розміщена в точці вікна з координатами (200, 100)

та інші.

# Кнопка у вікні

- На слайді наведено текст проєкту зі створення та розміщення кнопки у вікні та результат його виконання.

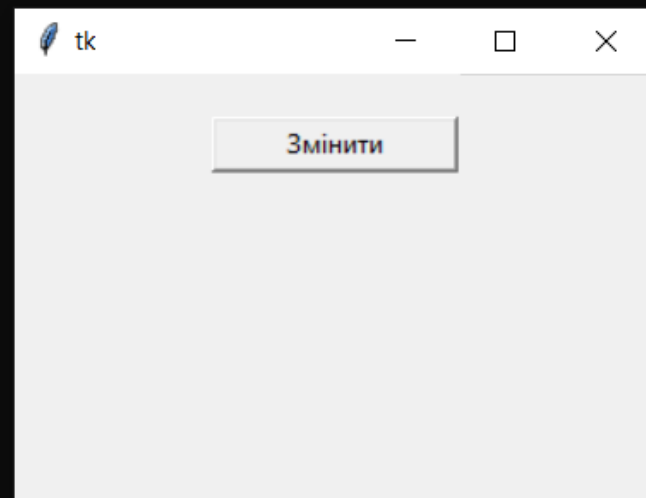


The image shows a screenshot of the IDLE Shell 3.11.1 window. The window has a menu bar with 'File', 'Edit', 'Shell', 'Debug', 'Options', 'Window', and 'Help'. The shell area displays the following Python code and its output:

```
Python 3.11.1 (tags/v3.11.1:a7a450f, Dec 6 2022, 19:58:39) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> from tkinter import *
>>> root=Tk()
>>> root.geometry('300x200+400+100')
>>> ''
>>> button=Button(text='Змінити', width='15')
>>> button.pack(pady=20)
>>>
```

Below the shell window, a separate Tkinter window titled 'tk' is displayed. It contains a single button with the text 'Змінити' (Change) centered on the screen.

```
Python 3.11.1 (tags/v3.11.1:a7a450f, Dec 6 2022, 19:58:39) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> from tkinter import *
>>> root=Tk()
>>> root.geometry('300x200+400+100')
>>> ''
>>> button=Button(text='Змінити', width='15')
>>> button.pack(pady=20)
>>>
```



# Обробники подій Click для кнопки в середовищі Python

Пов'язати з кнопкою обробник будь – якої події можна так само, як з вікном, використовуючи метод **bind()**. Наприклад, на слайді наведено приклад обробника події **Click** для кнопки, яка містить команду встановлення жовтого кольору вікна.

```
from tkinter import *
def click(event):
    root['bg'] = 'yellow'
root = Tk()
root.geometry('600x400+350+200')
button = Button(text = 'Змінити', width = 15)
button.pack(pady = 30)
button.bind('<1>', click)
```

# Обробники подій Click для кнопки

На слайді наведено приклад обробника подій **Click** для кнопки – процедури з іменем **Click ()**, яка містить команди:

- Установлення для кнопки червоного кольору фону;
- Установлення шрифту Arial розміром 14 пікселів;
- Виведення на кнопці вказаного тексту;
- Установлення для кнопки стану недоступна.



```
from tkinter import *
def click():
    button['bg'] = 'red'
    button['font'] = 'Arial 14'
    button['text'] = 'Змінюються значення властивостей кнопки'
    button['state'] = 'disable'
root = Tk()
root.geometry('600x400+350+200')
button = Button(text = 'Вибери мене', command = click)
button.pack(pady = 30)
```

## **Контрольні питання**

1. Що таке напис?
2. Як створити напис?
3. Які властивості має напис?
4. Як перенести текст на наступний рядок у написі?

## **Практичні завдання**

1. Створіть новий файл Python та вікно із заголовком “Вікно №1” та з розмірами 500x500. Додайте до вікна напис із текстом “Це вікно було створено в середовищі IDLE”. Колір напису повинен бути синім, а колір тексту білий, шрифт тексту “Arial 14”. Розмістити його довільно.
2. Створіть новий файл Python та вікно із заголовком “Вікно №2” та з розмірами 500x500. Додайте до вікна напис із текстом “Функція виконана”, який буде з’являтися після виконання події KeyPress відносно вікна. Колір напису повинен бути зеленим, а колір тексту білий, шрифт тексту “Calibri 14”. Розмістити його довільно.

# Домашнє завдання

*Проаналізувати конспект  
Виконати практичне завдання.  
Слайд 26.*

