

Текстове поле його функції та властивості

На попередніх уроках ми вчилися створенню об'єктів, надання їм властивостей та їх зміну. Тепер же ми почнемо працювати із даними та способами їх отримання.

Ми пам'ятаємо, що для введення даних в командному режимі використовується функція **input()**, а у модулі **tkinter** існують спеціальні об'єкти для введення даних, такі як **текстове поле**, **прапорець** та **перемикачі**.

Текстове поле

Текстове поле - це об'єкт, що використовується при побудові графічного інтерфейсу користувача.

Мета текстового поля - дозволити користувачеві ввести текстову інформацію, яка буде використовуватися програмою.

Модуль **tkinter** має можливість створити текстове поле у програмі на мові Python за допомогою функції **Entry()**:

назва_поля=Entry(назва_вікна, атрибут1...)

Атрибути (властивості) текстового поля

До текстового поля застосовуються наступні атрибути:

- **bg**="колір" - колір поля;
- **fg**="колір" - колір тексту поля;
- **font**="шрифт та розмір шрифту" - шрифт тексту (встановлює висоту поля);
- **width**=число – ширина рамки текстового поля, зазначається у кількості знаків;
- **bd**=число — ширина контуру поля, зазначається у пікселях.

Приклад 1

Маємо вікно **Window1**, розміри вікна 370x300, колір нехай буде чорний. Створимо в ньому текстове поле білого кольору з шириною у 30 знаків, а шрифтом **Times 13**. Розмістимо його в 50 пікселів від лівого краю та в 100 від верхнього краю. Ось як це буде виглядати у програмному коді:

```
File Edit Format Run Options Window Help
1 from tkinter import *
2 Window1=Tk()
3 Window1.geometry("370x300")
4 Window1["bg"]="black"
5 entry1=Entry(Window1, bg="white", width=30, font="Times 13") # створення текстового поля
6 entry1.place(x=50, y=100) # розміщення текстового поля
7 Window1.mainloop()
8
```

Запустимо програму та побачимо що з цього вийде:



Отримання даних та створення повноцінних програм

Саме по собі текстове поле не може використовуватися для отримання даних, поле – лише допоміжний об’єкт. Тобто, *entry1* – це лише назва об’єкта, а не назва змінної, яка містить дані введені у це поле.

Для отримання даних з поля використовується спеціальний метод

get()

Застосовується він так:

назва поля/прапорця/перемикача.get()

Тобто, *entry1* – це назва поля, а *entry1.get()* - це значення цього поля.

Приклад 2

Нехай ми маємо те ж вікно і те ж поле, що і в прикладі 1. Тепер створимо кнопку з текстом “Start” та розмістимо її в 170 пікселях від лівого краю та в 130 від верхнього. Присвоїмо їй подію *Button-1* (натиснення на ліву кнопку миші), при виконанні якої буде створюватися вікно повідомлення із даними, які ввів користувач. Заголовком вікна повідомлення зробимо “Дані користувача”. Ось, як це буде виглядати:

```
File Edit Format Run Options Window Help
1 from tkinter import *
2 from tkinter import messagebox
3 def start(event):
4     # виведення вікна повідомлення із даними користувача
5     messagebox.showinfo("Дані користувача", entry1.get())
6 Window1=Tk()
7 Window1.geometry("370x300")
8 Window1["bg"]="black"
9 # створення текстового поля
10 entry1=Entry(Window1, bg="white", width=30, font="Times 13")
11 entry1.place(x=50, y=100) # розміщення текстового поля
12 but=Button(Window1, text="Start") #кнопка
13 but.place(x=170, y=130)
14 but.bind("<Button-1>", start)
15 Window1.mainloop()
```

Як ви можете бачити замість повідомлення в функції `messagebox.showinfo()` було встановлено дані отримані з текстового поля (`entry1.get()`).

Зверніть увагу! Лапки в даному випадку ставити зовсім не потрібно, так як за замовчуванням дані текстового поля є рядком тексту. Якщо потрібно отримати числа (цілі або дійсні) — потрібно використати функції `int()` або `float()` відповідно.

Щоб отримати дійсні числа із текстового поля, використовують конструкцію:

`float(назва текстового поля.get())`

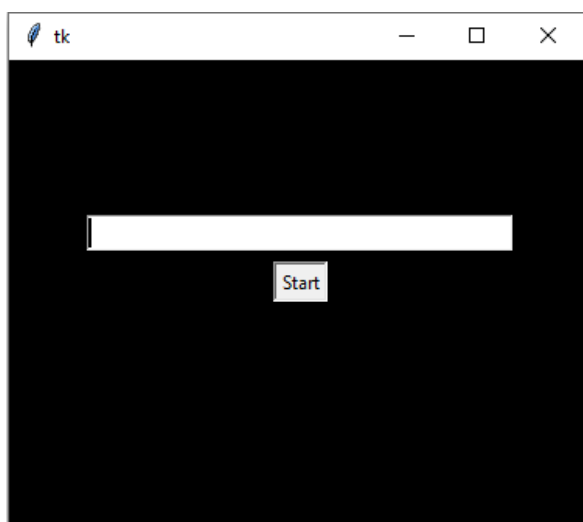
Щоб отримати цілі числа із текстового поля, використовують конструкцію:

`int(назва текстового поля.get())`

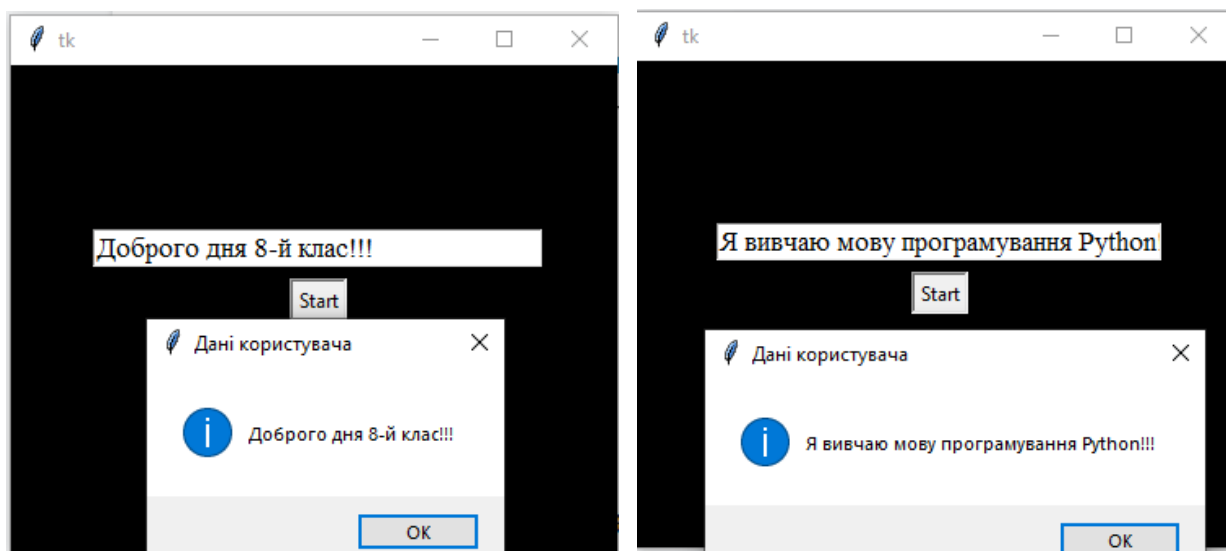
Щоб отримати текст (рядок) із текстового поля, використовують конструкцію:

`назва текстового поля.get()`

Маємо наступне вікно:



А тепер введем у текстове поле якийсь текст і натиснемо лівою кнопкою миші на кнопку “Start” У нас з’явиться вікно повідомлення із введеним текстом:



Приклад 3

Нехай ми матимемо те саме вікно з тими самими параметрами, що і в прикладі 1 та прикладі 2 (але без вікна повідомлення). Лише додамо до нього напис із початковим текстом “Введіть дані” який буде розміщено в 150-ти пікселях від лівого краю та в 50-ти від верхнього. В залежності від того, що введе користувач, воно буде відображатися замість напису “Введіть дані” (звичайно після натискання кнопки). Так це повинно виглядати у коді:

```
File Edit Format Run Options Window Help
1 from tkinter import *
2 from tkinter import messagebox
3 def start(event):
4     label["text"]=entry1.get() # зміна тексту напису на введений користувачем
5 Window1=Tk()
6 Window1.geometry("370x300")
7 Window1["bg"]="black"
8 # створення текстового поля
9 entry1=Entry(Window1, bg="white", width=30, font="Times 13")
10 entry1.place(x=50, y=100) # розміщення текстового поля
11 # створення напису
12 label=Label(Window1, text="Введіть дані", bg="black", fg="white", font="Times 14")
13 label.place(x=150, y=50) # розміщення напису
14 but=Button(Window1, text="Start") #кнопка
15 but.place(x=170, y=130)
16 but.bind("<Button-1>", start)
17 Window1.mainloop()
```

Як ви бачите, в даному випадку ми будемо вносити змінну тексту напису. **Зміна, або присвоєння одного атрибуту об'єкту застосовується так:**

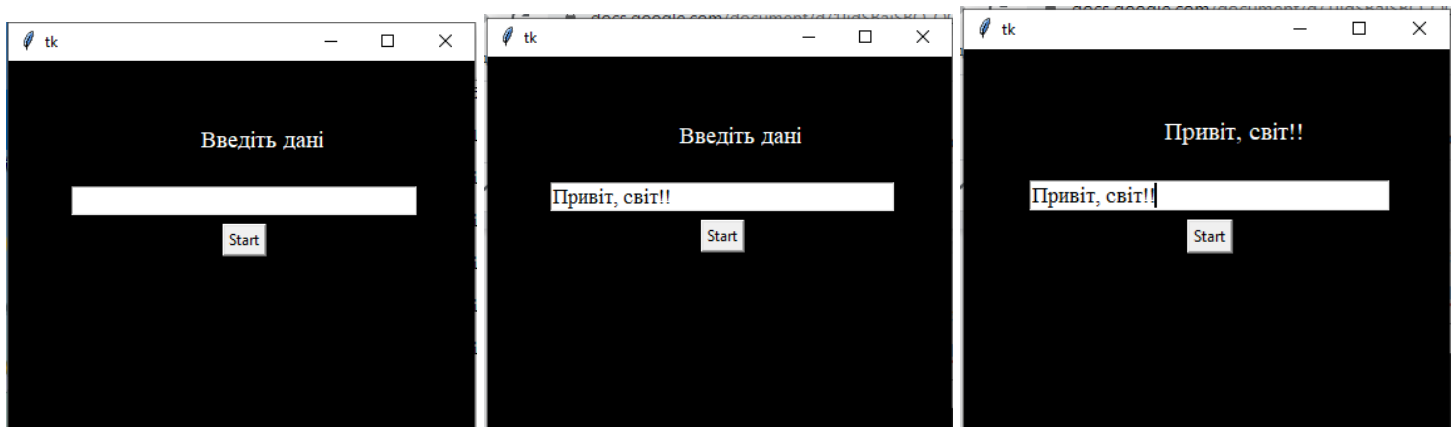
ім'я об'єкта ["назва атрибуту"]=значення

Наприклад, для зміни тексту у написі:

`label["text"]=entry1.get()`

`label["text"]="приклад"`

Отже, після натискання кнопки, текст напису буде дорівнювати даним отриманим з текстового поля. Для прикладу запусимо програму і введемо “Привіт світ!” та натиснемо на кнопку “Start”:



Для створення більш складних задач, потрібно згадати програмування у

командному інтерфейсі, адже всі дії з даними та конструкції можна застосувати до даних у графічному інтерфейсі.

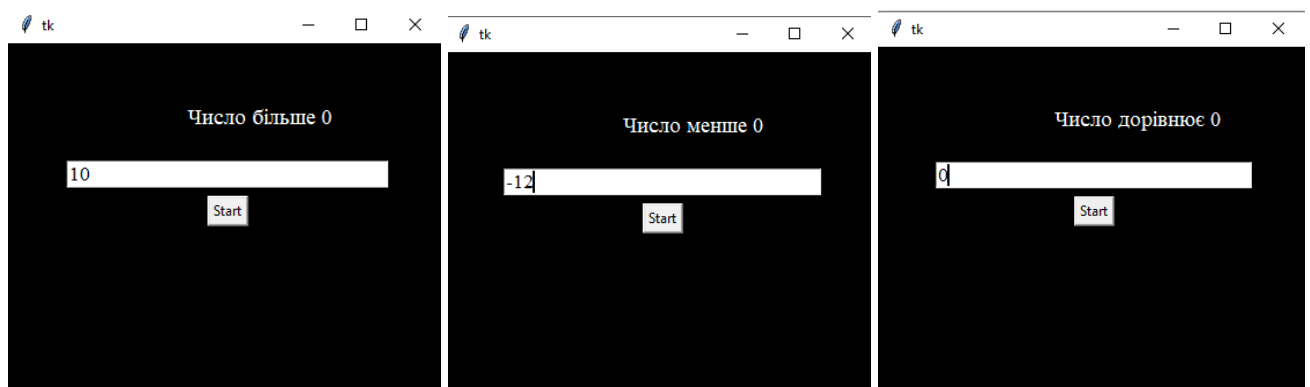
Давайте вдосконалимо нашу програму, щоб вона була цікавішою.

Приклад 4

Нехай користувач буде вводити числа в текстове поле і якщо це число більше 0 то буде виводитися напис “Число більше 0”, якщо менше то “Число менше 0” і відповідно “Число дорівнює 0”. Отже, маємо:

```
File Edit Format Run Options Window Help
1 from tkinter import *
2 def start(event):
3     # зміна тексту напису в залежності від значення, яке ввів користувач
4     if float(entry1.get())>0:
5         label["text"]="Число більше 0"
6     elif float(entry1.get())==0:
7         label["text"]="Число дорівнює 0"
8     else:
9         label["text"]="Число менше 0"
10 Window1=Tk()
11 Window1.geometry("370x300")
12 Window1["bg"]="black"
13 # створення текстового поля
14 entry1=Entry(Window1, bg="white", width=30, font="Times 13")
15 entry1.place(x=50, y=100) # розміщення текстового поля
16 # створення напису
17 label=Label(Window1, text="Введіть дані", bg="black", fg="white", font="Times 14")
18 label.place(x=150, y=50) # розміщення напису
19 but=Button(Window1, text="Start") #кнопка
20 but.place(x=170, y=130)
21 but.bind("<Button-1>", start)
22 Window1.mainloop()
```

Як ми бачимо, в залежності від того яким буде значення дійсного числа, яке введе користувач атрибут `text` напису `label` буде змінюватися по різному:



Приклад 5

Як і з іншими об'єктами вікна, з полем можна пов'язати процедури – обробники подій, використовуючи метод **bind()**.

Наприклад, на малюнку 4.13 наведено текст проєкту з процедурою – обробником події **Click** для поля. Ця процедура містить команди:

- установлення *жовтого* кольору фону поля;
- установлення *червоного* кольору для символів тексту в полі;
- установлення розміру *12* для символів тексту в полі;
- виведення тексту, уведеного в полі, у напис;
- присвоювання змінній *x* тексту з поля;
- виведення у вікно повідомлень тексту – значення змінної *x*, яке є текстом з поля, до якого в кінці додано знак оклику.

Після запуску проєкту на виконання в поле можна ввести довільний текст (на малюнку 4.14, а в поле введено текст *Працюємо з полем*) і вибрати поле. Результат виконання обробника події **Click** для поля наведено на малюнку 4.14, б.

```
from tkinter import*
from tkinter.messagebox import*

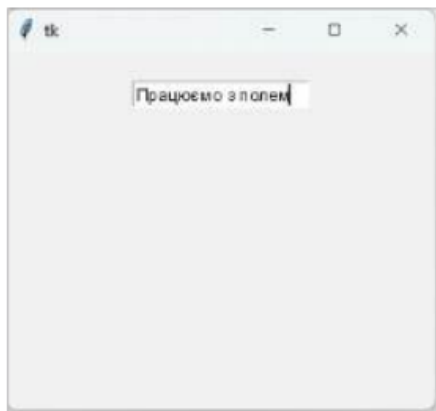
def click(event):
    entry['bg']='yellow'
    entry['fg']='red'
    entry['font']=12
    label['text'] = entry.get()
    x = entry.get()
    showinfo('Поле', x+'!')

root = Tk()
root.geometry('300x250')

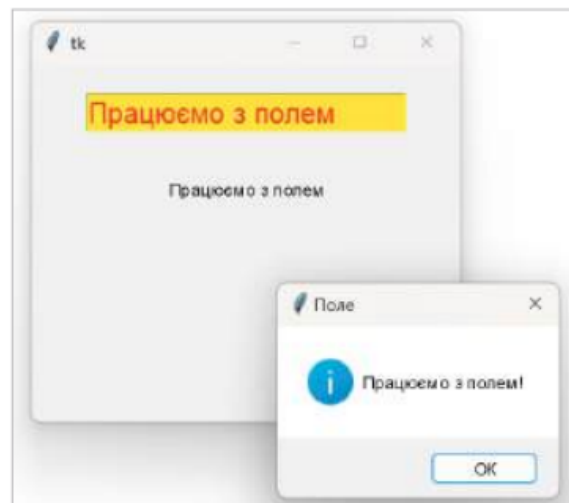
entry = Entry()
entry.pack(pady = 20)
entry.focus_set()
entry.bind('<1>', click)

label = Label()
label.pack(pady = 10)
```

Мал. 4.13. Текст проєкту з обробником події **Click** для поля



а)



б)

Мал. 4.14. Вікно виконання проєкту після введення тексту в поле і до настання події **Click** для поля (а) та після настання події (б)

Контрольні питання

1. Дайте визначення поняттю текстове поле.
2. Які атрибути можна застосувати для поля?
3. Який метод слугує для отримання даних з текстового поля?

Практичні завдання до уроку №60

Увага! При роботі за комп'ютером пам'ятайте про дотримання правил техніки безпеки та виконання санітарно-гігієнічних норм.

Виберіть і виконайте будь-яке **одне** завдання.

Виконувати можна як в онлайн середовищі для програмування Python так і в будь-якому встановленому на ваш пристрій або в зошиті.

Важливо! [Організація роботи по темі: "Програмування в Python"](#)

Завдання 1

Створіть новий файл Python та підключіть відповідний модуль. Створіть нове вікно Window1, розміри вікна 370x300, колір нехай буде чорний. Створіть в ньому текстове поле білого кольору з шириною у 30 знаків, а шрифтом *Times* 13. Розмістіть його в 50 пікселів від лівого краю та в 100 від верхнього краю.

```
File Edit Format Run Options Window Help
1 from tkinter import *
2 Window1=Tk()
3 Window1.geometry("370x300")
4 Window1["bg"]="black"
5 entry1=Entry(Window1, bg="white", width=30, font="Times 13") # створення текстового поля
6 entry1.place(x=50, y=100) # розміщення текстового поля
7 Window1.mainloop()
8
```

Завдання 2

Створіть новий файл Python та підключіть відповідний модуль. Створіть нове вікно Window1, розміри вікна 400x350, колір нехай буде зелений, заголовок вікна "Вікно №2". Створіть в ньому текстове поле рожевого кольору з шириною у 50 знаків, а шрифтом *Times* 14. Розмістіть його в 60 пікселів від лівого краю та в 150 від верхнього краю. (див. зд №1)

Завдання 3

Створіть новий файл Python та підключіть відповідний модуль.

- Створіть нове **вікно** Window1, розміри вікна 370x300, колір нехай буде чорний.
- Створіть в ньому **текстове поле** білого кольору з шириною у 30 знаків, а шрифтом *Times* 13. Розмістіть його в 50 пікселів від лівого краю та в 100 від

верхнього краю.

- Створіть **кнопку** з текстом “Start” та розмістіть її в 170 пікселях від лівого краю та в 130 від верхнього.
- Присвойте кнопці **подію** *Button-1* (натиснення на ліву кнопку миші), при виконанні якої буде створюватися вікно повідомлення із даними, які ввів користувач. Заголовком вікна повідомлення зробимо “Дані користувача”.

```
File Edit Format Run Options Window Help
1 from tkinter import *
2 from tkinter import messagebox
3 def start(event):
4     # виведення вікна повідомлення із даними користувача
5     messagebox.showinfo("Дані користувача", entry1.get())
6 Window1=Tk()
7 Window1.geometry("370x300")
8 Window1["bg"]="black"
9 # створення текстового поля
10 entry1=Entry(Window1, bg="white", width=30, font="Times 13")
11 entry1.place(x=50, y=100) # розміщення текстового поля
12 but=Button(Window1, text="Start") #кнопка
13 but.place(x=170, y=130)
14 but.bind("<Button-1>", start)
15 Window1.mainloop()
```

Завдання 4

Створіть новий файл Python та підключіть відповідний модуль.

- Створіть нове **вікно** Window1, розміри вікна 490x360, колір нехай буде синій, заголовок вікна “Вікно №4”
- Створіть в ньому **текстове поле** жовтого кольору з шириною у 50 знаків, а шрифтом *Times 14*. Розмістіть його в 40 пікселів від лівого краю та в 80 від верхнього краю.
- Створіть **кнопку** з текстом “Натисни” та розмістіть її в 80 пікселях від лівого краю та в 150 від верхнього.
- Присвойте кнопці **подію** *Button-1* (натиснення на ліву кнопку миші), при виконанні якої буде створюватися вікно повідомлення із даними, які ввів користувач. Заголовком вікна повідомлення зробимо “Ти ввів наступне повідомлення”. (див. зд №3)

Завдання 5

Створіть новий файл Python та підключіть відповідний модуль.

- Створіть нове **вікно** Window1, розміри вікна 370x300, колір нехай буде чорний.
- Створіть в ньому **текстове поле** білого кольору з шириною у 30 знаків, а шрифтом *Times 13*. Розмістіть його в 50 пікселів від лівого краю та в 100 від

верхнього краю.

- Створіть **кнопку** з текстом “Start” та розмістіть її в 170 пікселях від лівого краю та в 130 від верхнього.
- Додайте до нього **напис** із початковим текстом “Введіть дані” який буде розміщено в 150-ти пікселях від лівого краю та в 50-ти від верхнього.
- Присвойте кнопці **подію** *Button-1* (натиснення на ліву кнопку миші), при виконанні якої, текст, введений у текстове поле, буде відображатися замість напису “Введіть дані”.

File Edit Format Run Options Window Help

```
1 from tkinter import *
2 from tkinter import messagebox
3 def start(event):
4     label["text"]=entry1.get() # зміна тексту напису на введений користувачем
5 Window1=Tk()
6 Window1.geometry("370x300")
7 Window1["bg"]="black"
8 # створення текстового поля
9 entry1=Entry(Window1, bg="white", width=30, font="Times 13")
10 entry1.place(x=50, y=100) # розміщення текстового поля
11 # створення напису
12 label=Label(Window1, text="Введіть дані", bg="black", fg="white", font="Times 14")
13 label.place(x=150, y=50) # розміщення напису
14 but=Button(Window1, text="Start") #кнопка
15 but.place(x=170, y=130)
16 but.bind("<Button-1>", start)
17 Window1.mainloop()
```

Завдання 6

Створіть новий файл Python та підключіть відповідний модуль.

- Створіть нове **вікно** Window1, розміри вікна 600x500, колір нехай буде зелений, заголовок вікна “Вікно №6”
- Створіть в ньому **текстове поле** рожевого кольору з шириною у 40 знаків, а шрифтом *Arial* 12. Розмістіть його в 60 пікселів від лівого краю та в 200 від верхнього краю.
- Створіть **кнопку** з текстом “Start” та розмістіть її в 220 пікселях від лівого краю та в 300 від верхнього.
- Додайте до нього **напис** із початковим текстом “Тут будуть текстові дані” який буде розміщено в 100 пікселях від лівого краю та в 70-ти від верхнього.
- Присвойте кнопці **подію** *Button-1* (натиснення на ліву кнопку миші), при виконанні якої, текст, введений у текстове поле, буде відображатися замість напису “Тут будуть текстові дані”. (див. зд №5)

Завдання 7

Створіть новий файл Python та підключіть відповідний модуль.

- Створіть нове **вікно** Window1, розміри вікна 370x300, колір нехай буде чорний.

- Створіть в ньому **текстове поле** білого кольору з шириною у 30 знаків, а шрифтом *Times 13*. Розмістіть його в 50 пікселів від лівого краю та в 100 від верхнього краю.
- Створіть **кнопку** з текстом “Start” та розмістіть її в 170 пікселях від лівого краю та в 130 від верхнього.
- Додайте до нього **напис** із початковим текстом “Введіть дані” який буде розміщено в 150-ти пікселях від лівого краю та в 50-ти від верхнього.
- Нехай користувач буде вводити числа в текстове поле і якщо це число більше 0 то буде виводитися напис “Число більше 0”, якщо менше то “Число менше 0” і відповідно “Число дорівнює 0”.

```
File Edit Format Run Options Window Help
1 from tkinter import *
2 def start(event):
3     # зміна тексту напису в залежності від значення, яке ввів користувач
4     if float(entry1.get())>0:
5         label["text"]="Число більше 0"
6     elif float(entry1.get())==0:
7         label["text"]="Число дорівнює 0"
8     else:
9         label["text"]="Число менше 0"
10 Window1=Tk()
11 Window1.geometry("370x300")
12 Window1["bg"]="black"
13 # створення текстового поля
14 entry1=Entry(Window1, bg="white", width=30, font="Times 13")
15 entry1.place(x=50, y=100) # розміщення текстового поля
16 # створення напису
17 label=Label(Window1, text="Введіть дані", bg="black", fg="white", font="Times 14")
18 label.place(x=150, y=50) # розміщення напису
19 but=Button(Window1, text="Start") #кнопка
20 but.place(x=170, y=130)
21 but.bind("<Button-1>", start)
22 Window1.mainloop()
```

Завдання 8 (10 балів)

Створіть новий файл Python та підключіть відповідний модуль.

- Створіть нове **вікно** Window1, розміри вікна 570x600, колір нехай буде зелений, заголовок вікна “Вікно №8”
- Створіть в ньому **текстове поле** помаранчевого кольору з шириною у 35 знаків, а шрифтом *Times 14*. Розмістіть його в 70 пікселів від лівого краю та в 150 від верхнього краю.
- Створіть **кнопку** з текстом “Start” та розмістіть її в 200 пікселях від лівого краю та в 230 від верхнього.
- Додайте до нього **напис** із початковим текстом “Введіть дані” який буде розміщено в 120-ти пікселях від лівого краю та в 70-ти від верхнього.

- Нехай користувач буде вводити числа в текстове поле і якщо це число більше 50 то буде виводитися напис “Число більше 50”, якщо менше то “Число менше 50” і відповідно “Число дорівнює 50” (при натисненні на кнопку). (див. зд №7)

Завдання 9 (11 балів)

Створіть новий файл Python та підключіть відповідний модуль.

- Створіть нове **вікно** світло червоного кольору із розмірами 600 на 700 пікселів із заборонаю змінювати розміри вікна, із заголовком “Вікно №9”
- Створити **текстове поле** (всі атрибути встановити за власним бажанням),
- **кнопку** поруч із текстовим полем (текст “Ок”, колір білий, шрифт “Calibri 13”),
- та **напис** із початковим текстом «Ви нічого не ввели» (розмістіть його в довільному місці вікна).
- Створіть подію: після натискання на кнопку текст напису повинен змінитися на введений користувачем. (див. зд №5)

Завдання 10 (11 балів)

- Створіть **вікно** жовтогарячого кольору з розмірами 600x600 та з заголовком “10-те вікно”.
- Створіть **текстове поле** будь-якого кольору. Ширина поля 30 знаків та шрифт “Calibri 12”.
- Створіть **напис**, колір тексту темно-синій, шрифт аналогічний полю.
- При натисканні правою клавішею **по області вікна**, текст напису буде змінюватися на введений користувачем та виводитися на екран **вікно повідомлення** з заголовком “Підтвердження події” та з змістом “Дія виконана!”.

Завдання 12 (12 балів)

- Створіть **вікно** з розмірами 875x578 з заголовком «Це є вікно №12:)».
- Створіть **текстове поле** з шириною у 35 знаків, та розмістити його на 100 пікселів від лівого краю та на 100 пікселів від верхнього краю.
- Створіть кнопку “Розрахунки”, та розмістіть її під текстовим полем.
- В поле користувач буде вводити ціле число. При натисканні правою клавішею миші на кнопку “Розрахунки” на екран буде виводитися вікно повідомлення із добутком введеного числа на нього самого.

Заголовок вікна повідомлення, оформлення кнопки та колір фону головного вікна встановити самостійно.

Додаток

Кольори для Python (tkinter)

Named colour chart												
snow	deep sky blue	gold	seashell3	SlateBlue2	LightBlue3	SpringGreen2	DarkGoldenrod1	brown4	pink3	purple1	gray26	
ghost white	sky blue	light goldenrod	seashell4	SlateBlue3	LightBlue4	SpringGreen3	DarkGoldenrod2	salmon1	pink4	purple2	gray27	
white smoke	light sky blue	goldenrod	AntiqueWhite1	SlateBlue4	LightCyan2	SpringGreen4	DarkGoldenrod3	salmon2	LightPink1	purple3	gray28	
gainsboro	steel blue	dark goldenrod	AntiqueWhite2	RoyalBlue1	LightCyan3	green2	DarkGoldenrod4	salmon3	LightPink2	purple4	gray29	
floral white	light steel blue	rosy brown	AntiqueWhite3	RoyalBlue2	LightCyan4	green3	RosyBrown1	salmon4	LightPink3	MediumPurple1	gray30	
old lace	light blue	indian red	AntiqueWhite4	RoyalBlue3	PaleTurquoise1	green4	RosyBrown2	LightSalmon2	LightPink4	MediumPurple2	gray31	
linen	powder blue	saddle brown	bisque2	RoyalBlue4	PaleTurquoise2	chartreuse2	RosyBrown3	LightSalmon3	PaleVioletRed1	MediumPurple3	gray32	
antique white	pale turquoise	sandy brown	bisque3	blue2	PaleTurquoise3	chartreuse3	RosyBrown4	LightSalmon4	PaleVioletRed2	MediumPurple4	gray33	
papaya whip	dark turquoise	dark salmon	bisque4	blue4	PaleTurquoise4	chartreuse4	IndianRed1	orange2	PaleVioletRed3	thistle1	gray34	
blanched almond	medium turquoise	salmon	PeachPuff2	DodgerBlue2	CadetBlue1	OliveDrab1	IndianRed2	orange3	PaleVioletRed4	thistle2	gray35	
bisque	turquoise	light salmon	PeachPuff3	DodgerBlue3	CadetBlue2	OliveDrab2	IndianRed3	orange4	maroon1	thistle3	gray36	
peach puff	cyan	orange	PeachPuff4	DodgerBlue4	CadetBlue3	OliveDrab4	IndianRed4	DarkOrange1	maroon2	thistle4	gray37	
navajo white	light cyan	dark orange	NavajoWhite2	SteelBlue1	CadetBlue4	DarkOliveGreen1	sienna1	DarkOrange2	maroon3		gray38	
lemon chiffon	cadet blue	coral	NavajoWhite3	SteelBlue2	turquoise1	DarkOliveGreen2	sienna2	DarkOrange3	maroon4		gray39	
mint cream	medium aquamarine	light coral	NavajoWhite4	SteelBlue3	turquoise2	DarkOliveGreen3	sienna3	DarkOrange4	VioletRed1		gray40	
azure	aquamarine	tomato	LemonChiffon2	SteelBlue4	turquoise3	DarkOliveGreen4	sienna4	coral1	VioletRed2		gray41	gray42
alice blue	dark green	orange red	LemonChiffon3	DeepSkyBlue2	turquoise4	khaki1	burlywood1	coral2	VioletRed3		gray43	gray44
lavender	dark olive green	red	LemonChiffon4	DeepSkyBlue3	cyan2	khaki2	burlywood2	coral3	VioletRed4		gray45	gray46
lavender blush	dark sea green	hot pink	cornsilk2	DeepSkyBlue4	cyan3	khaki3	burlywood3	coral4	magenta2		gray47	gray48
misty rose	sea green	deep pink	cornsilk3	SkyBlue1	cyan4	khaki4	burlywood4	tomato2	magenta3		gray49	gray50
dark slate gray	medium sea green	pink	cornsilk4	SkyBlue2	DarkSlateGray1	LightGoldenrod1	wheat1	tomato3	magenta4		gray51	gray52
dim gray	light sea green	light pink	ivory2	SkyBlue3	DarkSlateGray2	LightGoldenrod2	wheat2	tomato4	orchid1		gray53	gray54
slate gray	pale green	pale violet red	ivory3	SkyBlue4	DarkSlateGray3	LightGoldenrod3	wheat3	OrangeRed2	orchid2		gray55	gray56
light slate gray	spring green	maroon	ivory4	LightSkyBlue1	DarkSlateGray4	LightGoldenrod4	wheat4	OrangeRed3	orchid3		gray57	gray58
gray	lawn green	medium violet red	honeydew2	LightSkyBlue2	aquamarine2	LightYellow2	tan1	OrangeRed4	orchid4		gray59	gray60
light grey	medium spring green	violet red	honeydew3	LightSkyBlue3	aquamarine4	LightYellow3	tan2	red2	plum1		gray61	gray62
midnight blue	green yellow	medium orchid	honeydew4	LightSkyBlue4	DarkSeaGreen1	LightYellow4	tan4	red3	plum2		gray63	gray64
navy	lime green	dark orchid	LavenderBlush2	SlateGray1	DarkSeaGreen2	yellow2	chocolate1	red4	plum3		gray65	gray66
cornflower blue	yellow green	dark violet	LavenderBlush3	SlateGray2	DarkSeaGreen3	yellow3	chocolate2	DeepPink2	plum4		gray67	gray68
dark slate blue	forest green	blue violet	LavenderBlush4	SlateGray3	DarkSeaGreen4	yellow4	chocolate3	DeepPink3	MediumOrchid1		gray69	gray70
slate blue	olive drab	purple	MistyRose2	SlateGray4	SeaGreen1	gold2	firebrick1	DeepPink4	MediumOrchid2		gray71	gray72
medium slate blue	dark khaki	medium purple	MistyRose3	LightSteelBlue1	SeaGreen2	gold3	firebrick2	HotPink1	MediumOrchid3		gray73	gray74
light slate blue	khaki	thistle	MistyRose4	LightSteelBlue2	SeaGreen3	gold4	firebrick3	HotPink2	MediumOrchid4		gray75	gray76
medium blue	pale goldenrod	snow2	azure2	LightSteelBlue3	PaleGreen1	goldenrod1	firebrick4	HotPink3	DarkOrchid1		gray77	gray78
royal blue	light goldenrod yellow	snow3	azure3	LightSteelBlue4	PaleGreen2	goldenrod2	brown1	HotPink4	DarkOrchid2		gray79	gray80
blue	light yellow	snow4	azure4	LightBlue1	PaleGreen3	goldenrod3	brown2	pink1	DarkOrchid3		gray81	gray82
dodger blue	yellow	seashell2	SlateBlue1	LightBlue2	PaleGreen4	goldenrod4	brown3	pink2	DarkOrchid4		gray83	gray84

Шрифти в Python (tkinter)

'System',	'Bahnschrift SemiCondensed',	'Gabriola',
'Terminal',	'Bahnschrift SemiBold SemiCondensed',	'Gadugi',
'Fixedsys',	'Bahnschrift Light Condensed',	'Georgia',
'Modern',	'Bahnschrift SemiLight Condensed',	'Impact',
'Roman',	'Bahnschrift Condensed',	'Ink Free',
'Script',	'Bahnschrift SemiBold Condensed',	'Javanese Text',
'Courier',	'Calibri',	'Leelawadee UI',
'MS Serif',	'Calibri Light',	'Leelawadee UI Semilight',
'MS Sans Serif',	'Cambria',	'Lucida Console',
'Small Fonts',	'Cambria Math',	'Lucida Sans Unicode',
'Marlett',	'Candara',	'Malgun Gothic',
'Arial',	'Candara Light',	'@Malgun Gothic',
'Arabic Transparent',	'Comic Sans MS',	'Malgun Gothic Semilight',
'Arial Baltic',	'Consolas',	'@Malgun Gothic Semilight',
'Arial CE',	'Constantia',	'Microsoft Himalaya',
'Arial CYR',	'Corbel',	'Microsoft JhengHei',
'Arial Greek',	'Corbel Light',	'@Microsoft JhengHei',
'Arial TUR',	'Courier New',	'Microsoft JhengHei UI',
'Arial Black',	'Courier New Baltic',	'@Microsoft JhengHei UI',
'Bahnschrift Light',	'Courier New CE',	'Microsoft JhengHei Light',
'Bahnschrift SemiLight',	'Courier New CYR',	'@Microsoft JhengHei Light',
'Bahnschrift',	'Courier New Greek',	'Microsoft JhengHei UI Light',
'Bahnschrift SemiBold',	'Courier New TUR',	'@Microsoft JhengHei UI Light',
'Bahnschrift Light SemiCondensed',	'Ebrima',	'Microsoft New Tai Lue',
'Bahnschrift SemiLight SemiConde',	'Franklin Gothic Medium',	'Microsoft PhagsPa',

'Microsoft Sans Serif',
'Microsoft Tai Le',
'Microsoft YaHei',
'@Microsoft YaHei',
'Microsoft YaHei UI',
'@Microsoft YaHei UI',
'Microsoft YaHei Light',
'@Microsoft YaHei Light',
'Microsoft YaHei UI Light',
'@Microsoft YaHei UI Light',
'Microsoft Yi Baiti',
'MingLiU-ExtB',
'@MingLiU-ExtB',
'PMingLiU-ExtB',
'@PMingLiU-ExtB',
'MingLiU_HKSCS-ExtB',
'@MingLiU_HKSCS-ExtB',
'Mongolian Baiti',
'MS Gothic',
'@MS Gothic',
'MS UI Gothic',
'@MS UI Gothic',
'MS PGothic',
'@MS PGothic',
'MV Boli',
'Myanmar Text',
'Nirmala UI',
'Nirmala UI Semilight',
'Palatino Linotype',
'Segoe MDL2 Assets',
'Segoe Print',
'Segoe Script',
'Segoe UI',
'Segoe UI Black',
'Segoe UI Emoji',
'Segoe UI Historic',
'Segoe UI Light',
'Segoe UI Semibold',
'Segoe UI Semilight',
'Segoe UI Symbol',
'SimSun',
'@SimSun',
'NSimSun',

'@NSimSun',
'SimSun-ExtB',
'@SimSun-ExtB',
'Sitka Small',
'Sitka Text',
'Sitka Subheading',
'Sitka Heading',
'Sitka Display',
'Sitka Banner',
'Sylfaen',
'Symbol',
'Tahoma',
'Times New Roman',
'Times New Roman Baltic',
'Times New Roman CE',
'Times New Roman CYR',
'Times New Roman Greek',
'Times New Roman TUR',
'Trebuchet MS',
'Verdana',
'Webdings',
'Wingdings',
'Yu Gothic',
'@Yu Gothic',
'Yu Gothic UI',
'@Yu Gothic UI',
'Yu Gothic UI Semibold',
'@Yu Gothic UI Semibold',
'Yu Gothic Light',
'@Yu Gothic Light',
'Yu Gothic UI Light',
'@Yu Gothic UI Light',
'Yu Gothic Medium',
'@Yu Gothic Medium',
'Yu Gothic UI Semilight',
'@Yu Gothic UI Semilight',
'HoloLens MDL2 Assets',
'BIZ UDGothic',
'@BIZ UDGothic',
'BIZ UDPGothic',
'@BIZ UDPGothic',
'BIZ UDMincho Medium',
'@BIZ UDMincho Medium',

'BIZ UDPMincho Medium',
'@BIZ UDPMincho Medium',
'Meiryo',
'@Meiryo',
'Meiryo UI',
'@Meiryo UI',
'MS Mincho',
'@MS Mincho',
'MS PMincho',
'@MS PMincho',
'UD Digi Kyokasho N-B',
'@UD Digi Kyokasho N-B',
'UD Digi Kyokasho NP-B',
'@UD Digi Kyokasho NP-B',
'UD Digi Kyokasho NK-B',
'@UD Digi Kyokasho NK-B',
'UD Digi Kyokasho N-R',
'@UD Digi Kyokasho N-R',
'UD Digi Kyokasho NP-R',
'@UD Digi Kyokasho NP-R',
'UD Digi Kyokasho NK-R',
'@UD Digi Kyokasho NK-R',
'Yu Mincho',
'@Yu Mincho',
'Yu Mincho Demibold',
'@Yu Mincho Demibold',
'Yu Mincho Light',
'@Yu Mincho Light',
'DengXian',
'@DengXian',
'DengXian Light',
'@DengXian Light',
'FangSong',
'@FangSong',
'KaiTi',
'@KaiTi',
'SimHei',
'@SimHei',
'Ubuntu',
'Raleway',
'Ubuntu Condensed',
'Ubuntu Light'