

ВІЗУАЛІЗАЦІЯ ЕЛЕМЕНТІВ ТАБЛИЧНОЇ ВЕЛИЧИНИ в Python. Елемент керування «багаторядкове текстове поле».

У різних сферах діяльності існує багато завдань, у яких вихідні дані й результати повинні бути подані у графічній формі. Уявіть собі, що ви готуетесь до виступу з доповіддю. Ваша доповідь виглядатиме більш наочною, якщо сухі цифри подати у вигляді кольорової діаграми. Діаграми будуються на основі числових даних, що містяться в таблицях, і допомагають аналізувати та порівнювати дані.

Розглянемо два способи побудови діаграм: із використанням

графічних методів модуля `tkinter` і за допомогою методів бібліотеки `matplotlib`.

Побудова діаграм із використанням графічних методів модуля `tkinter`

Діаграма — це графічне зображення, у якому числові дані подано геометричними фігурами. Діаграми можна створювати, програмуючи креслення графічних примітивів (лінія, прямокутник, сектор) графічними методами полотна `Canvas`, із якими ви знайомі з курсу 7 класу.

ПРИКЛАД 1. Гістограма (стовпчикова діаграма) використовується для порівняння елементів табличної величини. Побудуємо гістограму показника успішності учнів 9-А класу з чотирьох предметів списку `subject = ['Інф. ', 'Алг. ', 'Теом. ', 'Фіз. ']`, значення якого занесені до списку `n`:

`n=[9.8, 8.5, 9.1, 7.4]` за кольорами списку: `colors = ['red', 'orange', 'yellow', 'green']`

Вигляд вікна форми програми:



Областю побудови діаграми є полотно об'єкта `Canvas` розмірами 300 x 220 пікселів. Координатні осі намальовані за допомогою методу `canvas.create_line()`. Для малювання стрілок до переліку параметрів методу додано параметри `arrow` та `arrowshape`. Атрибуту `arrow` надано значення `LAST`, що задає розміщення стрілки в кінці лінії. Значення атрибуту `arrowshape` задає розміри стрілки.

Ряди даних побудовані методом `create_rectangle()`.

До діаграми слід додати підписи осей і рядів даних. Розмістити текст на полотні можна за допомогою методу `create_text(x, y, text)`. За замовчуванням у точці

(x, y) розташовується центр напису. Щоб розмістити за вказаними координатами ліву границю тексту, атрибуту anchor надається значення W (від англ. west — захід).

Код програми:

```
1 from tkinter import*
2 root = Tk()
3 root.title("Успішність учнів 9А класу")
4 canvas = Canvas(root, width = 300, height = 220)
5 canvas.pack()
6 n=[9.8, 8.5, 9.1, 7.4]
7 # Побудова координатних осей
8 canvas.create_line(10, 200, 10, 10, arrow = LAST, arrowshape = "10 20 10")
9 canvas.create_line(10, 200, 280, 200, arrow = LAST, arrowshape = "10 20 10")
10 colors = ['red', 'orange', 'yellow', 'green']
11 subject = ['Інф. ', 'Алг. ', 'Геом. ', 'Фіз. ']
12 x = 30
13 for i in range( 4):
14     y =220-(n[i])*20
15     canvas.create_rectangle(x, y, x+30, 200, fill = colors[i])
16     canvas.create_text(x, 210, text = subject[i], anchor = W)
17     x = x+50
18
```

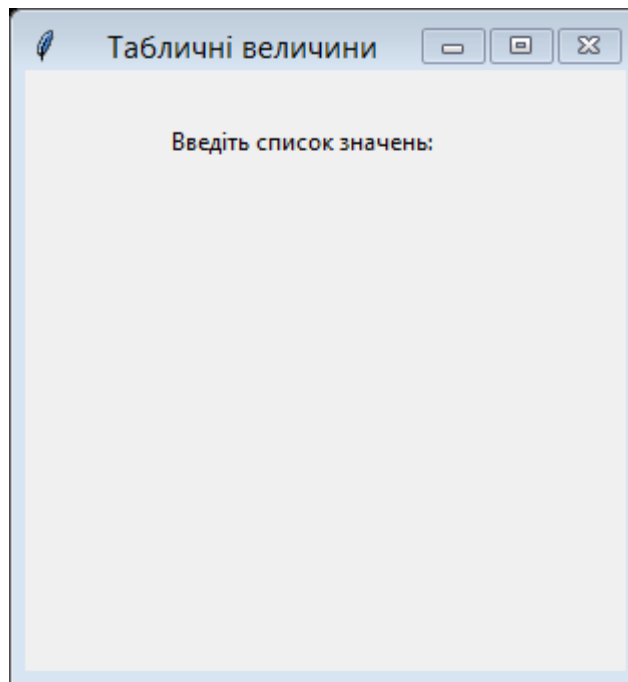
Як для роботи з масивами на формах проектів використовують візуальні елементи?

Для початку роботи пригадаймо, яким чином в Python ми працюємо з графічним інтерфейсом (модуль tkinter):

Отже, створимо вікно розміром 300х300 пікселів, назвою – Табличні величини, та написом – Введіть список значень:

```
1 from tkinter import*
2 root = Tk()
3 root.geometry("300x300")
4 root.title("Табличні величини")
5 Label(root, text="Введіть список значень:").place(x=70, y=25)
6 root.mainloop()
```

Збережемо та запустимо отриманий проект:



Ми вже знайомилися з текстовим полем *Entry*, але його використання в даному випадку недоречно, оскільки *Entry* – однорядкове текстове поле. Тому в цих умовах набагато краще використовувати багаторядкове текстове поле *Text*. Створити його можна за допомогою функції *Text()* наступним чином:

назва_поля=Text(атрибут1, атрибут2,...)

До атрибутів поля *Text* належать:

<i>Атрибут</i>	<i>Пояснення</i>
<i>bg</i> ="колір"	Колір поля
<i>fg</i> ="колір"	Колір тексту поля
<i>font</i> ="шрифт та розмір шрифту"	Шрифт тексту
<i>width</i> =число	Ширина, зазначається у кількості знаків
<i>height</i> =число	Висота, зазначається у кількості знаків
<i>bd</i> =число	Ширина контуру поля (у пік селях)

До методів поля *Text* належать:

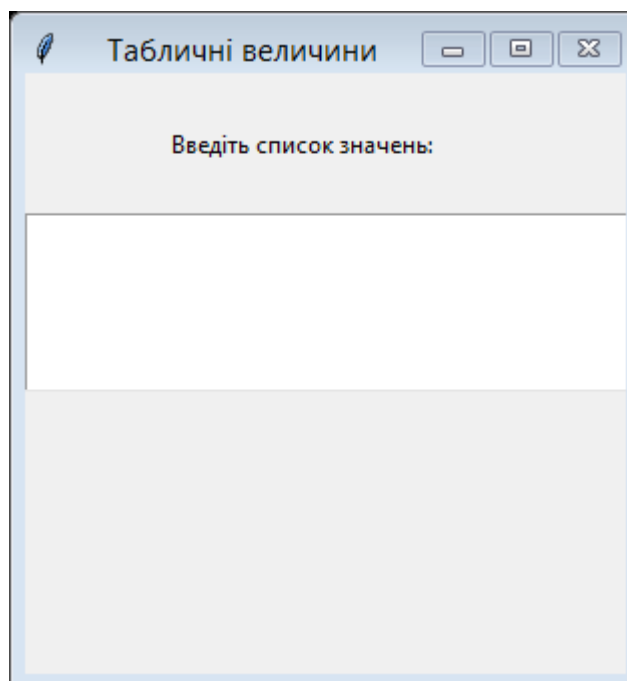
Метод	Опис
-------	------

<i>insert()</i>	Метод додавання даних. Має атрибути – початкова позиція та текст який додається.
<i>delete()</i>	Метод видалення даних. Має атрибути – початкова та кінцева позиції у форматі “x.y” (x – номер рядка, y – номер стовпчика)
<i>get()</i>	Метод отримання даних/ Має атрибути – початкова та кінцева позиції у форматі “x.y” (x – номер рядка, y – номер стовпчика)

Тепер додамо до нашого прикладу багаторядкове текстове поле та розмістимо його під написом із запрошення ввести дані:

```
txt=Text(root, width=42, height=5, bg="white", font="Arial 11")
txt.place(x=0, y=70)
```

Внаслідок чого отримаємо:



Для введення даних поля **Text** достатньо. А ось для виведення потрібно використовувати додаткові можливості. Зокрема в Python існує метод розміщення об'єктів у вигляді таблиці. Цей метод називається **grid**.

Але перед розміщення елементів таблиці створимо рамку для неї. Використаємо в цих цілях функцію **Frame()**, яка має наступні атрибути:

Атрибут	Пояснення
<i>bg</i> ="колір"	Колір поля
<i>width</i> =число	Ширина, зазначається у кількості знаків
<i>height</i> =число	Висота, зазначається у кількості знаків
<i>bd</i> =число	Ширина контуру поля (у пік селях)
<i>relief</i> ="тип"	Вказує на тип лінії рамки

Рамку також потрібно розміщувати у вікні. Для цього:

```
frame=Frame(root, bd=0.5, relief="solid")
frame.place(x=5, y=200)
```

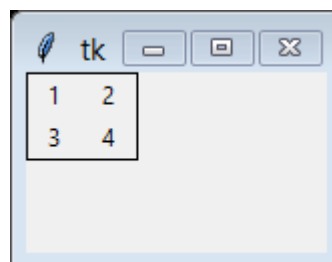
Тепер потрібно створити таблицю за допомогою метода `grid()`, наступним чином:

об'єкт.grid(row=число, column=число)

Тут `row` – номер рядка, `column` – номер стовпця.

Для прикладу:

```
from tkinter import *
root=Tk()
frame=Frame(root, bd=0.5, relief = "solid")
frame.place(x=0, y=0)
label1=Label(frame, text="1", width=3).grid(row=0, column=0)
label2=Label(frame, text="2", width=3).grid(row=0, column=1)
label3=Label(frame, text="3", width=3).grid(row=1, column=0)
label4=Label(frame, text="4", width=3).grid(row=1, column=1)
root.mainloop()
```



А тепер спробуємо переписати нашу програму:

```
from tkinter import*

def fun1(event):
    list=[int(i) for i in txt.get("1.0", END).split()]
    c=0
    for i in list:
        label=Label(frame, text=i, bd=0.5, relief="solid", width=3)
        label.grid(row=0, column=c)
        c=c+1

root=Tk()
root.geometry("300x300")
root.title("Табличні величини")
Label(root, text="Введіть список значень:").place(x=70, y=25)

txt=Text(root, width=42, height=5, bg="white", font="Arial 11")
txt.place(x=0, y=70)

frame=Frame(root, bd=0.5, relief="solid")
frame.place(x=5, y=200)

root.bind("<Return>", fun1)

root.mainloop()
```

Ось що ми отримуємо:

