

Віджети в Tkinter. Налаштування вікна. Label, entry, button.

Головне вікно

Виклик головного вікна не має додаткових параметрів для налаштування зовнішнього вигляду, але має ряд методів, які дозволяють змінити вигляд вікна. До них відносяться зміна розміру вікна, його відстань від країв монітору, колір фону та інш.

- `.geometry()` – зміна розміру вікна.

Його параметр – це строка типу ‘ширина x висота + відступ зверху + відступ зліва’.

```
win = tk.Tk()
win.geometry('300x300+200+100')
```

Якщо висота та ширина екрана невідома, то варто додати дві нові змінні:

```
w = win.winfo_screenwidth() #Визначаємо ширину екрана
h = win.winfo_screenheight() #Визначаємо висоту екрана
```

Це є число. Ми можемо його поділити або відняти якусь кількість пікселів. Додавати або множити це число не варто, бо тоді вікно вийде за границю екрану.

Далі ці змінні ми вставляємо до методу розмірів:

```
win.geometry('%dx%d+0+0' % (w,h))
```

- `.title()` – зміна заголовку вікна.

```
win.title('Моя програма')
```

- `.overrideredirect()` – прибирає строку назви вікна.
- `.destroy()` – закриває вікно, може використовуватися в функціях.
- `.configure()` – параметри вікна.

До параметрів цього метода можуть відноситися background. Він змінює колір фону всього вікна.

```
win.configure(background='pink')
```

Також є багато інших параметрів.

- `.resizable()` – дозволяє або забороняє розширення вікна.

Віджети label, entry, button

Створення елементів відбувається після створення головного вікна. Всі вони створюються методом вікна з прив'язкою до змінної. Наприклад:

```
lab = tk.Label(width='50', text='Введіть текст') #Створюємо напис
ent = tk.Entry(width=50) #Поле для вводу
but = tk.Button(text='Переробити') #Кнопку
rez = tk.Label(width='50', text='') #Та пусте поле для напису
```

Віджети зачасту мають такі ж параметри, як і вікно.

Поле Entry має кілька власних методів:

- `.get()` – отримати введений текст. У випадку, якщо аргументи не приведені, то метод зчитує весь текст. Якщо приведені 2 числових аргументи, то метод зчитує зріз з введеної строки.
- `.delete()` – видаляє введений текст. Аргументи аналогічні до методу `get`.
- `.insert()` – вставляє строку в поле Entry. Перший аргумент – позиція (індекс) вставлення строки (якщо текст повинен бути в кінці – END), другий аргумент – строка, що вставляється.

Для додання функціоналу віджетів треба додати функцію та прив'язати її до події.

Для редагування якоїсь властивості елемента вказується сам елемент, назва властивості у квадратних дужках та цьому виразу присвоюється значення.

Наприклад, якщо при натисканні ми хочемо зчитувати строку з поля вводу та виводити її у поле виводу тексту, то треба виконати наступне:

```
def vyvid(event):
    text = ent.get()
    rez['text'] = str(text)
```

Властивість елемента обов'язково повинна бути строкового типу.

Для того, щоб указати до чого прив'язана функція та при якій події вона виконується, використовують метод `bind()`. У всередині дужок ми пишемо подію та функцію, яка виконується. Наприклад:

```
but.bind('<Button-1>', vyvid)
```

У данному випадку ми прив'язуємо виконання функції до натискання лівою кнопкою миші на кнопку.

Основні події в Python:

<Button-1> – натискання лівої кнопки миші.

<Button-2> – настикання середньої кнопки миші (коліщатко).

<Button-3> – натискання правої кнопки миші.

<Enter> – введення курсору на зону елемента.

<Leave> – виведення курсору з зони елемента.

<Key> – натискання на будь-яку клавішу.

<Return> – натискання на Enter.

<BackSpace> – натискання пробілу.

<Shift_L> – натискання будь-якого Shift.

<Control_L> – натискання будь-якого Ctrl.

Також у «скобках» < > можна написати будь-яку клавішу – на цю клавішу і буде прив’язана функція. Звертайте увагу на регістр: <r> та <R> – це різні клавіші. Найбільш ефективно це прив’язувати до полей вводу тексту.

Контрольні питання:

- 1) Перечислити основні методи налаштування вікна.
- 2) Як створити однострочкове поле вводу?
- 3) Як зробити функцію і прив’язати її виконання до натискання кнопки?
- 4) Перечислити основні події у мові Python.