

13. Programování - Tkinter - základní komponenty

knihovna Tkinter

Tkinter je modul (knihovna) pro programovací jazyk Python, která umožňuje vytvářet grafické uživatelské rozhraní. Tkinter se dodává společně s instalací Pythonu a je tak dostupný všude, kde je Python. Funguje např. stejně na Linuxu i Windows.

Abyste mohli používat Tkinter, je nutné ho naimportovat.

```
import tkinter as tk
```

Nebo, častěji:

```
from tkinter import *
```

Vytvoření okna

```
hlavni=Tk() # založí hlavní okno programu
```

```
hlavni.mainloop() # spustí hlavní smyčku událostí, která probíhá tak dlouho, dokud se nezavře program
```

Úprava vlastností okna

```
hlavni.title("pokus") # titulek okna
```

```
hlavni.config(width=300,height=200,bg="navy") # šířka, výška, barva pozadí
```

```
hlavni.minsize(200,100) # minimální velikost
```

```
hlavni.maxsize(400,200) # maximální velikost
```

```
hlavni.geometry("300x150+500+200") # velikost okna s odsazením od levého horního rohu obr.
```

```
hlavni.resizable(False, False) # nelze měnit velikost (v tomto případě nastavíme jen minsize, nebo width a height)
```

Ke změně **vlastnosti komponent** obecně používáme metodu **.config** nebo index s názvem vlastnosti jako u slovníku. Většina komponent má tyto obecné možnosti formátu:

- background | bg - barvy se zapisují slovně nebo hexadecimálně v RGB
- foreground | fg
- activebackground
- activeforeground
- font - ("Helvetica", 10, "bold italic")
- width
- height
- padx
- pady
- relief - FLAT, RAISED, SUNKEN, GROOVE, RIDGE
- borderwidth | bd
- highlightbackground
- highlightcolor
- highlightthickness
- cursor

základní komponenty - vlastnosti, použití (Label, Entry, Button, Frame, LabelFrame)

Všechny komponenty mají minimálně jeden povinný parametr - rodičovskou komponentu *master* (okno, frame...).

Label (štítek) je určen k zobrazení textu nebo obrázku. Štítek může zobrazovat text pouze jednoho formátu, ale dá se rozdělit do více řádku pomocí \n nebo parametru wraplength.

```
stitek=Label(hlavni, text="Ahoj světe!", font="Arial 12", fg="red", bg="#00FF00")
```

Když neurčíte velikost, štítek se přizpůsobí na co nejmenší velikost úměrnou jeho obsahu. Pro toto nastavení se hodí použít parametr `height` a `width` k určení přesné velikosti. Pro text se velikost definuje v textových jednotkách, u obrázku v pixelech.

Entry je vstupní pole k zadávání textu do jednoho řádku.

Parametry:

- `show` = Určuje, jak se bude zobrazovat obsah; používá se u polí s hesly (=""")
- `state` = NORMAL, DISABLED, READONLY (jako DISABLED, avšak obsah se dá vybírat a kopírovat)
- `textvariable` = Spojuje tkProměnnou (obvykle StringVar) s obsahem vstupního pole
- `validate` = Určuje, kdy se má konat ověřování; "none", "focus", "focusin", "focusout", "key", "all"
- `validatecommand` = ověřovací funkce - vrací True/False

Metody:

- **get()** - vrací současný obsah vstupního pole jako řetězec
- **insert**(index, řetězec) - vkládá text na danou pozici; insert(INSERT, text) na pozici kurzoru, na konec pak insert(END, text)
- **delete**(od, do=None) - maže znak na pozici nebo v zadaném rozsahu; delete(0, END) na smazání všeho

Ve stavu **DISABLED/READONLY** jsou všechna volání insert a delete **ignorována**!

Button je tlačítko, které může obsahovat text nebo obrázek a funkci, která se zavolá při zmáčknutí.

```
def kliknuti():
```

```
    print ("Kliknul jsi!")
```

```
b = Button(hlavni, text="OK", command=kliknuti)
```

Funkce nesmí obsahovat povinné parametry, to můžeme obejít lambda funkcí.

Frame je v základu neviditelný rám, který se používá k seskupení a organizaci jiných komponent. Výrazně zvyšuje použitelnost rozmístění **pack**. Může být také použit jako vycpávka nebo okrasa:

```
Frame(height=2, bd=1, relief=SUNKEN).pack(fill=X, padx=5, pady=5)
```

K vytvoření komponenty v nějakém Framu musíme použít jeho jméno v povinném parametru `master` dané komponenty:

```
ramecek=Frame(hlavni, bd=1, relief="ridge")
```

```
ramecek.pack()
```

```
napis=Label(ramecek, text="...")
```

LabelFrame je rámeček s zvýrazněným okrajem a popiskem. Používá se pro seskupení a popis skupiny souvisejících komponent. Pro popisek se používá parametr **text** nebo **labelwidget**. Parametr **labelanchor** ovlivňuje pozici popisku.

metody pro rozmístění komponent - srovnání

Pack

Správce rozmístění pack umožňuje vytvářet layouty "balením" komponent do rodičovských komponent **v řadě nad sebou** nebo **vedle sebe**. Také je užitečný pro **vyplnění celého okna**:

```
Text(hlavni).pack(fill=BOTH, expand=1)
```

Parametry:

- **fill** říká správci, že udělátka chce vyplnit celý prostor; hodnoty X, Y, BOTH
- **expand** říká správci, aby přiděloval dodatečný prostor udělátku, pokud se rodičovský widget zvětší; nadbytečný prostor se rozdělí mezi všechny udělátka, které mají parametr expand nastaven na nenulovou hodnotu
- **anchor** = kam umístit komponentu uvnitř rodičovské komponenty; výchozí CENTER
- **side** = kam se má komponenta zapakovat; LEFT/RIGHT pro umístění vedle sebe

Potřebujete-li vytvořit sofistikovanější design, budete muset obvykle seskupovat skupiny komponent do dalších Framů nebo použít správce grid.

Grid a pack v jednom rodičovském okně **nejsou kompatibilní!**

Grid

Grid je nejflexibilnější ze správců v Tkinter. Umožňuje vytvářet layouty podobné tabulkám umístováním komponent do dvourozměrné mřížky.

Parametry:

- **row** = vloží komponentu do tohoto řádku; číslování od 0; výchozí je první další prázdný řádek v mřížce.
- **rowspan** = do kolika řádků má buňka přesahovat
- **column** = vloží komponentu do tohoto sloupce; číslování od 0; výchozí je 0
- **columnspan** = do kolika sloupců má buňka přesahovat
- **sticky** = kde bude komponenta v buňce umístěna (pokud je menší než buňka); S, N, E, W nebo NW, NE, SW, SE; W+E roztáhne horizontálně na celou buňku; W+E+N+S roztáhne v obou směrech; Standard je udělátka vycentrováno v buňce bez automatického rozšiřování
- **ipadx, ipady** = dobrovolné vnitřní výplně, i v **packu**
- **padx, pady** = dobrovolné vnější výplně, i v **packu**

Place

umožňuje umístit komponentu na konkrétní souřadnice, relativně od levého horního rohu rodičovské komponenty.

a) Absolutní - v pixelech

```
place(x= , y= )
```

b) Relativní - v desetinných číslech od 0 do 1

```
place(relx= , rely= )
```

Parametr **anchor** - kotevní bod komponenty: N, NE, E, SE, S, SW, W, NW (*výchozí*), CENTER.

Umožňuje rovnou nastavit absolutní rozměry (**width a height**) nebo rel. rozměry (**relwidth a relheight**).

proměnné knihovny Tkinter

tkProměnné se dají použít u většiny komponent (parametr **textvariable, variable**) a mít tak možnost sledovat změny zadávaných hodnot a dynamicky je měnit. **Checkbutton, Radiobutton a OptionMenu** dokonce použití těchto proměnných vyžadují.

Vytvoření:

```
pr = StringVar() # BooleanVar, IntVar, DoubleVar
```

Metoda **.get()** vrací aktuální hodnotu proměnné.

Metoda **.set(hodnota)** aktualizuje proměnnou a vyrozumívá všechny sledovatele. Parametr musí mít správný typ podle typu proměnné.

Metoda **.trace(mód, funkce)** se používá na spojení proměnné s funkcí. Funkce je volána vždy, když dojde např. ke změně proměnné (při módu "w"):

```
def funkce(*args):  
    print ("změna!")  
pr = StringVar()  
pr.trace("w", funkce)  
pr.set("ahoj")
```

metoda after()

.after je metoda oken v Tkinteru sloužící pro časovou prodlevu. Parametr čas je interval v milisekundách, po jehož uplynutí se spustí nepovinná funkce.

```
okno.after(čas, funkce)
```

Prodleva běží paralelně - umožňuje chod okna (program nezamrzne).

Pokud metodu použijeme ve funkci a zároveň do ní tuto funkci dosadíme, vytvoříme tak užitečný **časovač** (funkce se bude spouštět každých x milisekund).

```
def vypis():  
    print("Proběhl výpis")  
    hlavni.after(1000, vypis)  
vypis()
```