

Uživatelská rozhraní cvičení 2

Na minulém cvičení jste si vyzkoušeli rozvrhnout prvky pomocí systému rozvržení **pack()** a umístění prvků jste ovlivnili pomocí parametru **side**. Dnes si představíme další parametry, jejichž funčnost budeme opět ilustrovat pomocí tlačítek. Většina vlastností ale platí obecně na jakékoliv prvky včetně framů (tedy kontejnerů, které mohou obsahovat další prvky).

Uvažme následující rozložení:

```
b1 = Button(root, text="Button1", bg="yellow")
```

```
b2 = Button(root, text="Button2", bg="red")
```

```
b3 = Button(root, text="Button3", bg="green")
```

```
b1.pack(side="left")
```

```
b2.pack()
```

```
b3.pack()
```



Fill

Často budeme v rozhraní chtít, aby výška žluté komponenty vyplnila volný prostor, který byl pro komponentu alokovaný (asi ne tak často v případě tlačítek, ale právě v případě framů). To lze provést pomocí parametru **fill**, který lze nastavit na **X**, **Y** nebo **BOTH**. V tomto případě chceme, aby se komponenta rozšířila ve vertikálním směru, je to tedy směr **Y**.

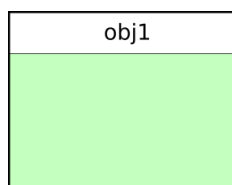
```
b1.pack(side="left", fill=Y)
```

```
b2.pack()
```

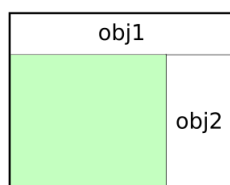
```
b3.pack()
```



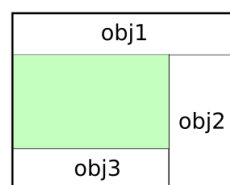
U parametru **fill** si je třeba uvědomit, že se zaplňuje prostor, který byl pro komponentu alokován. Tento prostor je znázorněn v následujícím obrázku z minulého cvičení. Pro obj1, který je vykreslen pomocí parametru **side="top"**, je alokován pouze horizontální pruh o výšce komponenty. Tzn. že **fill=X** vyplní pruh, ale **fill=Y** neudělá nic, protože komponenta nemá alokováno nic navíc ve vertikálním směru.



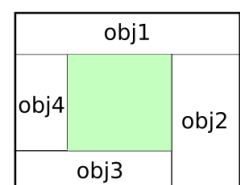
top



top
right



top
right
bottom



top
right
bottom
left

Toto lze ilustrovat, pokud zvětšíme vykreslené okno. Mějme následující rozložení, pouze s parametrem `side`, a mírně zvětšíme vykreslené okno (v tuto chvíli se rovnoměrně zvětšují prostory alokované pro komponenty a komponenty jsou zarovnané na střed alokované plochy).

```
b1.pack(side="bottom")
b2.pack(side="left")
b3.pack()
```



Tzn že b3, který má **side="top"** můžeme roztáhnout ve směru **X**, ale pokud nastavíme **BOTH**, neznamená to, že zaplní volný prostor pod tímto objektem. To stejné u b2, ale ve druhém směru.

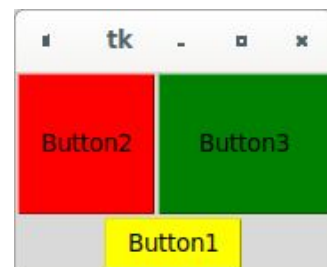
```
b1.pack(side="bottom")
b2.pack(side="left", fill=BOTH)
b3.pack(fill=BOTH)
```



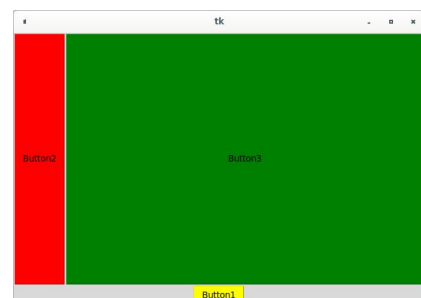
Expand

Pokud chceme, aby komponenta alokovala i volný prostor svého rodiče, který nebyl alokován pro tuto komponentu, je nutné použít parametr **expand**, který nabývá hodnot **1** nebo **0** (defaultně 0).

```
b1.pack(side="bottom")
b2.pack(side="left", fill=BOTH)
b3.pack(fill=BOTH, expand=1)
```



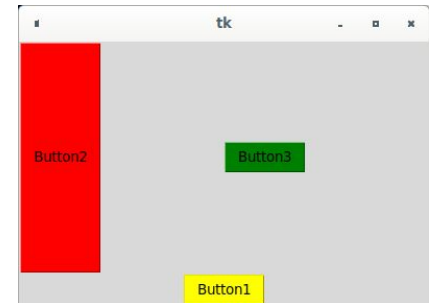
Pokud okno zvětšíme víc (klidně fullscreen), komponenta, která má `expand = 1`, bude zabírat celou volnou plochu.



Anchor

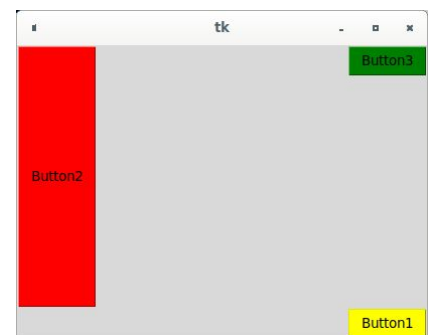
Dalším parametrem je **anchor**, který ukotví komponentu na jednu ze stran (**W**, **E**, **N**, **S** = západ, východ, sever, jih, strany lze i kombinovat). Zrušme u b3 **fill**, ale nechme **expand=1**. Co se stane? Volný prostor se bude alokovat pro tuto komponentu, ale komponenta samotná se nebude vyplňovat do volného prostoru.

```
b1.pack(side="bottom")
b2.pack(side="left", fill=BOTH)
b3.pack(side="right", expand=1)
```



V tuto chvíli lze komponentu posunout v rámci volného prostoru (to lze samozřejmě provést i bez **expand=1**, ale jen v rámci prostoru alokovaném pro komponentu, viz b1).

```
b1.pack(side="bottom", anchor=E)
b2.pack(side="left", fill=BOTH)
b3.pack(side="right", expand=1, anchor=N+E)
```



Padx a pady

Poslední parametry, které si představíme, už nebudou souviset s umístěním, ale s mezerami mezi komponentami. Pokud budeme mít vedle sebe např. dvě tlačítka, není dobře, aby byly vedle sebe bez nějaké mezery.

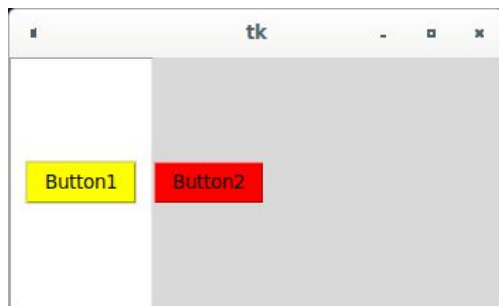
Mezery lze nastavit pomocí **padx** a **pady** (vnější mezery) nebo **ipadx** a **ipady** (vnitřní mezery). Pro ilustraci si vytvoříme frame, do tohoto framu vložíme tlačítko a vedle framu vložíme druhé tlačítko. Kromě mezer si na tomto příkladu taky ukážeme chování komponent uvnitř kontejnerů.

```
fr = Frame(root, relief=SUNKEN, bg="white", bd=1)
# relief je styl ohraničení, bd je šířka ohraničení
b1 = Button(fr, text="Button1", bg="yellow")
b2 = Button(root, text="Button2", bg="red")

fr.pack(side="left", fill=Y)
b1.pack(expand=1)
b2.pack(side="left")
```



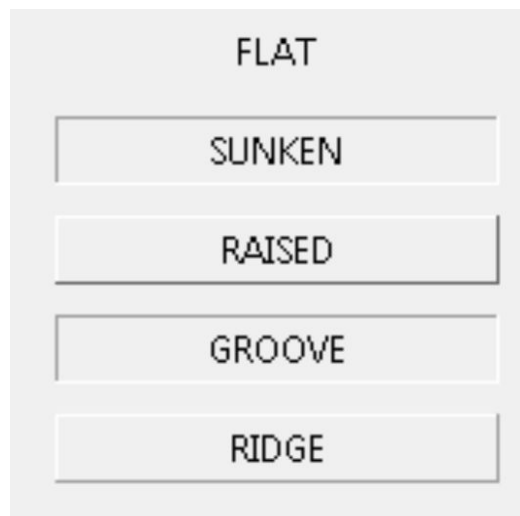
Prvním parametrem u každé komponenty je rodič. Dosud jsme používali **root**, což bylo vykreslovací okno. Teď máme u **b1** rodiče **fr**, to znamená, že komponenta se objeví uvnitř **fr** a stejně tak její chování (anchor, expand, fill) se projeví v rámci **fr**. Tedy, i když jsme použili u **b1** **expand=1**, u **fr** jsme **expand** nepoužili, pouze **fill=Y**, takže **fr** se roztáhlo vertikálně do prostoru alokovaném pro **fr**, **b1** se expandovalo do plochy **fr**, nikoliv do plochy celého okna. Nastavíme u **fr** mezery pomocí **ipadx** (obrázek vlevo - vnitřní mezera, tzn. mezera od komponent ve framu k okraji framu) a **padx** (obrázek vpravo - vnější mezera, tzn. mezera mezi framen a ostatními komponentami a okrajem okna).



To je tedy vše k funkci `pack()` a jejím parametrům. Vyzkoušejte si s nimi experimentovat, ať pochopíte vliv jednotlivých parametrů na vzhled grafického rozhraní.

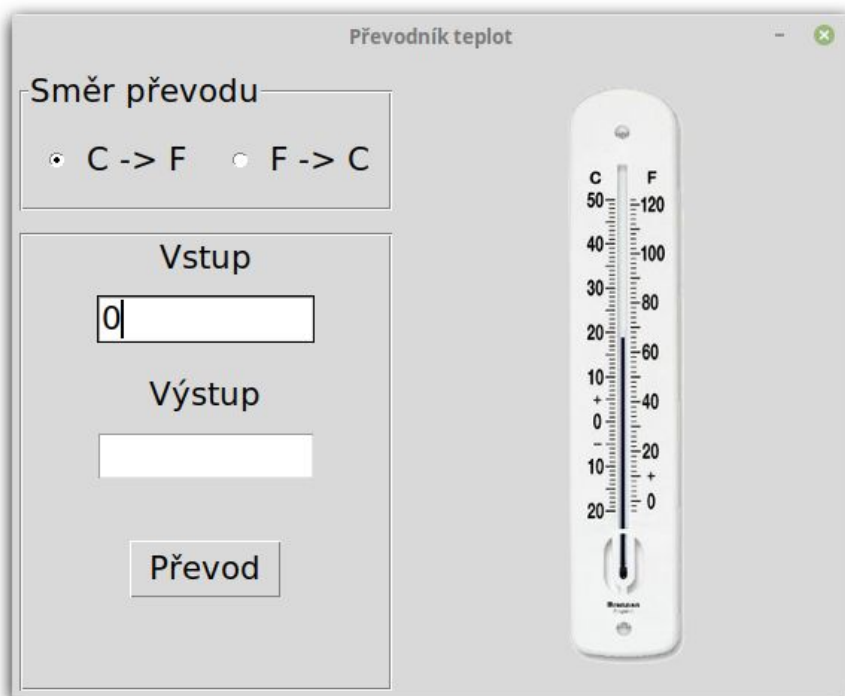
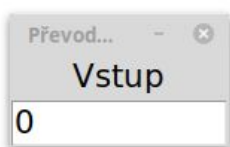
Frame

Již v předchozím textu jsme použili frame, tedy rámeček, kontejner obsahující další komponenty. Tento kontejner budete využívat velmi často. Frame samotný není vidět, pokud v něm nevykreslíme nějaké prvky. U frame lze nastavit také šířku okraje (`bd`) a styl okraje (`relief`) - možnosti jsou v následujícím obrázku. Více zde https://www.tutorialspoint.com/python/tk_frame.htm



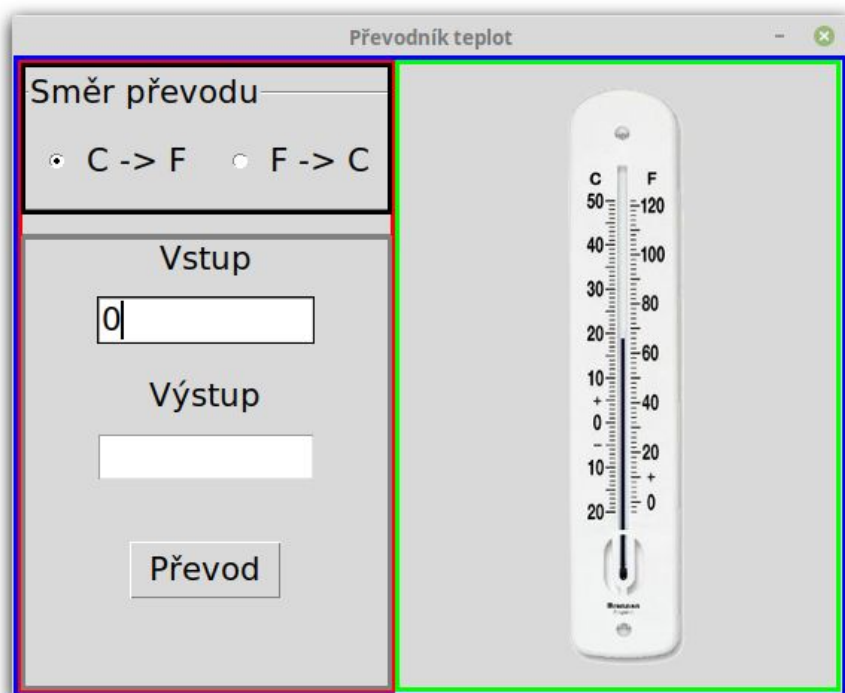
Úkol na cvičení

Vytvořte aplikaci pro převod teplot. Vyjděte ze šablony (vlevo), výsledná aplikace by mohla vypadat např. takto. Zkuste i experimentovat a aplikaci graficky “vytunit”.



Rozložení framů v šabloně

- root
- left_frame
- right_frame
- dir_frame
- ent_frame



Co se vám bude hodit v úkolu

Proměnné v tkinter

U některých komponent chceme, aby byly svázány s nějakou proměnnou. Např. u množiny radiobuttonů chceme, aby bylo možné vybrat jen jeden z nich a zároveň budeme chtít i v nějaké funkci zjistit, který z nich je momentálně zatržený. Možné řešení:

```
var = IntVar() # vytvoříme si proměnnou var celočíselného typu (pro řetězec lze StringVar)
var.set(1) # nastavíme hodnotu na 1
```

```
# radiobuttony spojíme s touto proměnnou a taky nastavíme, jakou hodnotu má mít proměnná, když se
radiobutton vybere
```

```
Radiobutton(variable=var, value=1)
```

```
Radiobutton(variable=var, value=2)
```

```
# takhle získáme hodnotu proměnné
```

```
print(var.get())
```

https://www.tutorialspoint.com/python/tk_radiobutton.htm

Zachytávání výjimek

Co když uživatel zadá něco co nemá?

Vstup očekává číslo, když je zadáno něco jiného, měl by program uživatele informovat, ať uživatel ví, co udělal špatně a čemu se příště vyhnout.

```
a="text"
```

```
try:
```

```
    b=float(a) # zkusíme přetypovat, pokud nepůjde, je vyvolána výjimka ValueError
```

```
    print(b)
```

```
except ValueError:
```

```
    print("Error")
```

Vykreslení obrázku pomocí Canvasu

https://www.python-course.eu/tkinter_canvas.php

```
ca = Canvas(frame, width=300, height=400)
```

```
photo = PhotoImage(file="th.png")
```

```
ca.create_image(150, 200, image=photo) # pozice obrázku
```

```
# lze vykreslit i jiné prvky (úsečky, čtverce)
```

```
r = ca.create_rectangle(146, 292, 152, 80, fill="blue")
```

a lze je editovat

ca.coords(r, 146, 292, 152, 100)

Můžete vykreslit i rtuťový sloupek (v tom případě použijte obrázek th_empty.png).

HINT: $x = [146, 152]$, $y(50\text{ °C}) = 80$, $y(-20\text{ °C}) = 292$

Nastavení vlastností okna

root.title('Převodník teplot') # popis okna

root.resizable(False, False) # povolení změny velikosti okna ve směru x a y

root.bind('<Return>', funkce) # mapování událostí (klávesnice, myš)

def_font = tkinter.font.nametofont("TkDefaultFont")

def_font.config(size=16) # změna defaultního fontu