

# Python – lekce 14

## 14.7 Metody rozmísťování

### Pack

Správce rozmístění *pack* vám umožňuje vytvářet layouty "balení" komponent do rodičovských komponent tím, že jsou považovány za čtvercové oblasti umísťované do rámečků.

Je trochu limitovaný, ale snadněji se používá v několika běžných situacích:

1. Komponenta uvnitř rámečku, nebo nějaké komponenty typu kontejner, která má vyplnit celý rám
2. Umístění několika komponent v **řadě nad sebou**
3. Umístění několika komponent **vedle sebe**

Potřebujete-li vytvořit sofistikovanější design, budete muset obvykle seskupovat skupiny komponent do dalších Framů. Také můžete v těchto případech používat správce grid.

---

**Poznámka:** Nikdy nemíchejte *grid* a *pack* ve stejném rodičovském okně. Tkinter po zbytek vašeho života bude hledat řešení, které uspokojí oba dva správce. Namísto čekání aplikaci raději shodte a znovu se podívejte na svůj kód. Běžná chyba je použití chybného rodiče pro některé komponenty.

---

### parametry

*anchor* = Kam umístit komponentu uvnitř rodičovské komponenty. Standard je CENTER.

*expand* = Určuje, jestli má komponenta vyplňovat nadbytečný prostor v rodiči. Pokud se rovná nepravdě (default), komponenta se nebude rozšiřovat.

*fill* = Určuje, jestli má komponenta okupovat veškerý prostor poskytnutý mu rodičem. Pokud je **NONE**(default), komponenta si zachovává svoji původní velikost. Pokud je **X** (vyplnit vodorovně), **Y** (vyplnit svisle) nebo **BOTH**, vyplní daný prostor v patřičném směru. Chcete-li, aby komponenta zcela a vždy vyplňovala prostor rodiče, nastavte **fill=BOTH** a **expand** na nějakou nenulovou hodnotu.

*in* = Zapakuj tuto komponentu uvnitř daného widgetu. Komponenta se dá packovat jen uvnitř svého rodiče nebo potomka rodiče. Tento parametr se většinou vynechává a v tom případě se komponenta zapakuje uvnitř svého rodiče. Všimněte si, že **in** je v Pythonu klíčové slovo. Abyste ho mohli použít jako pojmenovaný parametr, přidejte podtržítko (**in\_**).

*ipadx* = Vnitřní výplň. Standard je 0.

*ipady* = Vnitřní výplň. Standard je 0.

*padx* = Vnější výplň. Standard je 0.

*pady* = Vnější výplň. Standard je 0.

*side* = Určuje, u které strany se má komponenta zapakovat. Použijte **TOP**, chcete-li pakovat svisle. Chcete-li pakovat vodorovně, použijte **LEFT**. Můžete také pakovat ke spodní nebo pravé straně (**BOTTOM** a **RIGHT**). Strany můžete klidně kombinovat, ale nemusíte vždy dostat to, co očekáváte. Ačkoliv použitím Rámečků můžete vytvářet vcelku komplikovaná rozložení komponent, možná dáte u netriviálních layoutů přednost správci rozmístění grid.

## Place

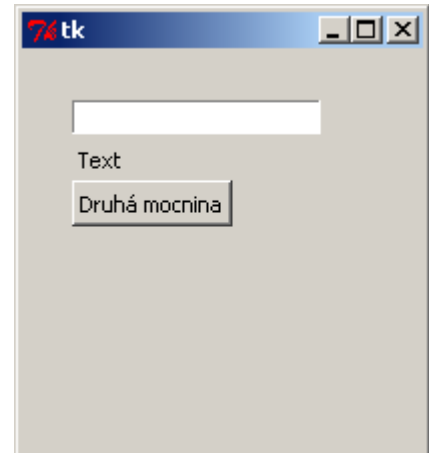
Správce rozmístění *place* vám umožňuje umístit komponentu na přesně stanovené místo. Chcete-li tento typ rozmísťování, použijte metodu **place**.

```
hlavni = Tk()
hlavni.minsize(150,100)
hlavni.maxsize(300,200)

vstup=Entry(hlavni,width=20)
vstup.place(x=25, y=25)

napis=Label(hlavni,text="")
napis.place(x=25, y=45)

tlacitko=Button(hlavni, text=u"Druhá mocnina")
tlacitko.place(x=25, y=65)
```



## Grid

Grid je nejflexibilnější ze správců v Tkinter.

Umožňuje vytvářet layouty podobné tabulkám, umísťováním komponent do dvourozměrné mřížky.

<label 1>	<entry 2>	<image>	
<label 1>	<entry 2>		
<checkboxbutton>		<button 1>	<button 2>

Vytvoření takového layoutu pomocí správce pack je možné, ale vyžaduje vytvoření několika extra komponent Frame a mnoho starostí, aby věci vypadaly tak, jak mají. Když místo něho použijete správce grid, budete volat pouze jednu metodu na každou komponentu a všechno bude, jak má být.

## parametry

*column* = Vloží komponentu do tohoto sloupce. Číslování sloupců začíná na 0. Pokud je vynecháno, je 0.

*columnspan* = Pokud je uvedeno, indikuje, že buňka komponenty má přesahovat do několika sloupců. Standard je 1.

*in* = Zapakuj tuto komponentu uvnitř daného widgetu. Komponenta se dá packovat jen uvnitř svého rodiče nebo potomka rodiče. Tento parametr se většinou vynechává a v tom případě se komponenta zapakuje uvnitř svého rodiče. Všimněte si, že **in** je v Pythonu klíčové slovo. Abyste ho mohli použít jako pojmenovaný parametr, přidejte podtržítko (**in\_**).

*ipadx* = Dobrovolná vodorovná vnitřní výplň. Funguje jako **padx**, ale výplň je přidána *dovnitř* hranic komponenty. Standard je 0.

*ipady* = Dobrovolná svislá vnitřní výplň. Funguje jako **pady**, ale výplň je přidána *dovnitř* hranic komponenty. Standard je 0.

*padx* = Dobrovolná vodorovná výplň, která zaplní prostor kolem buňky komponenty. Stand. je 0.

*pady* = Dobrovolná svislá výplň, která zaplní prostor kolem buňky komponenty. Standard je 0.

*row* = Vloží komponentu do tohoto řádku. Čísla řádků začínají na 0. Pokud je vynecháno, vezme se první další prázdný řádek v mřížce.

*rowspan* = Pokud je uvedeno, indikuje, že buňka komponenty má přesahovat do několika řádek. Standard je 1.

*sticky* = Určuje, jak se bude komponenta rozšiřovat, když bude buňka větší než komponenta sama. Může nabývat hodnot **S**, **N**, **E** a **W** nebo **NW**, **NE**, **SW** a **SE**.

Například **W** (west) znamená, že udělátko bude zarovnáno k levému okraji buňky.

**W+E** znamená, že udělátko bude roztahováno tak, aby vždy vyplnilo celou buňku.

**W+E+N+S** znamená, že udělátko bude roztahováno v obou směrech. Standard je udělátko vycentrováno v buňce bez automatického rozšiřování.

def Zkontroluj():

```
    kontrola["text"]= vstup1.get()+" "+vstup2.get()
```

```
jmeno=Label(hl_okno, text="Jméno:")
```

```
jmeno.grid(row=0,sticky=E)
```

```
prijmeni=Label(hl_okno, text="Příjmení:")
```

```
prijmeni.grid(row=1,sticky=E)
```

```
vstup1=Entry(hl_okno)
```

```
vstup1.grid(row=0,column=1)
```

```
vstup2=Entry(hl_okno)
```

```
vstup2.grid(row=1,column=1)
```

```
kontrola=Label(hl_okno,text="")
```

```
kontrola.grid(row=2,columnspan=2)
```

```
tlacitko1=Button(hl_okno,text="Kontrola",command=Zkontroluj)
```

```
tlacitko1.grid(row=3,sticky=W)
```

```
tlacitko2=Button(hl_okno,text="Konec",command=hl_okno.quit)
```

```
tlacitko2.grid(row=3,column=1,sticky=E)
```

