



Grafika v PYTHONĚ

Ing.Drgo Pavel, 27. október
2017, piatok 11:49

Blaho 1.diel

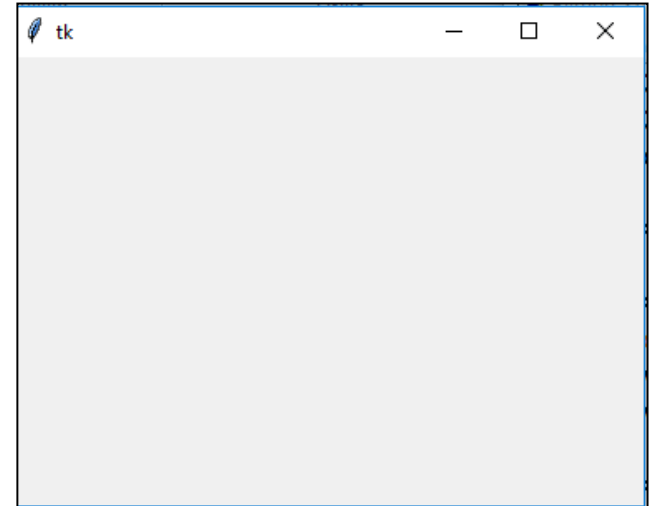
GRAFIKA PYTHON-a

- Doteraz sme pracovali bud' v priamom (shell) alebo v programovom režime.
- Programy sme spúšťali priamo z editora
- Teraz použijeme program, ktorý pracuje s grafickým oknom (v grafickom režime).
- Už vieme, že príkaz (funkcia) `print()` vypisuje priamo do textového okna (shell).
- Existuje iná skupina príkazov(funkcií),ktorá nevypisuje do textového okna ale do grafického okna
- Takéto okno sa ale nevytvorí samo, musíme zadať špeciálne príkazy.

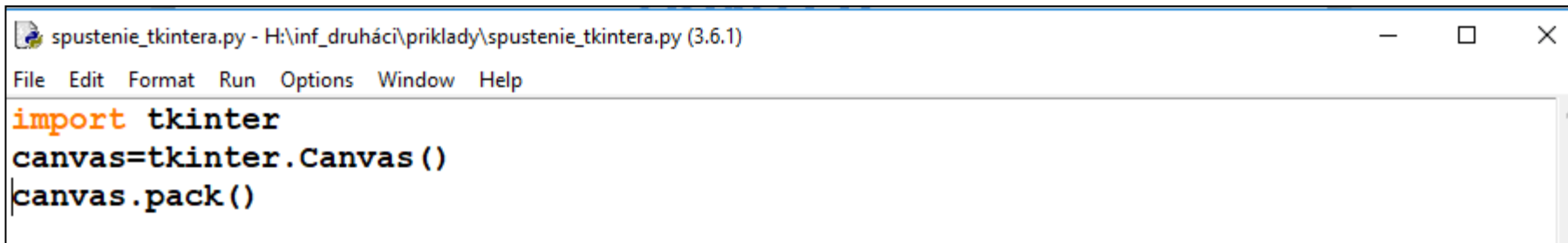
Grafické prostredie TKINTER

1) Do príkazového riadka postupne zadáme toto::

```
>>> import tkinter
>>> canvas=tkinter.Canvas ()
>>> canvas.pack ()
>>> |
```



2) Alebo Do súboru uložíme toto:



A spustíme to stlačením F5

V oboch prípadoch sa spustí grafické prostredie TKINTER

Čo znamenajú tie tri príkazy?

```
>>> import tkinter
>>> canvas=tkinter.Canvas()
>>> canvas.pack()
>>> |
```

- príkaz [import tkinter](#) oznámi, že budeme pracovať s modulom tkinter, ktorý obsahuje grafické príkazy
- týmto príkazom vznikla nová premenná [tkinter](#), ktorá obsahuje referenciu na tento modul,
- t.j. všetky funkcie a premenné, ktoré sú definované v tomto module, sú prístupné pomocou tejto premennej
- Budeme k nim pristupovať tzv. bodkovou notáciou, t.j. vždy uvidíme meno premennej [tkinter](#), za tým bodku a meno funkcie alebo premennej, napr. [tkinter.Canvas](#) •
- [tkinter.Canvas\(\)](#) vytvorí grafickú plochu
- Aby sme s touto plochou mohli ďalej pracovať, uložíme si jej referenciu do premennej [canvas](#)
- kým ne zadáme aj príkaz [canvas.pack\(\)](#), grafická plocha sa ešte nezobrazí - volanie [canvas.pack\(\)](#) zabezpečí zobrazenie nového okna aj s novovytvorenou grafickou plochou

Súradnicová sústava

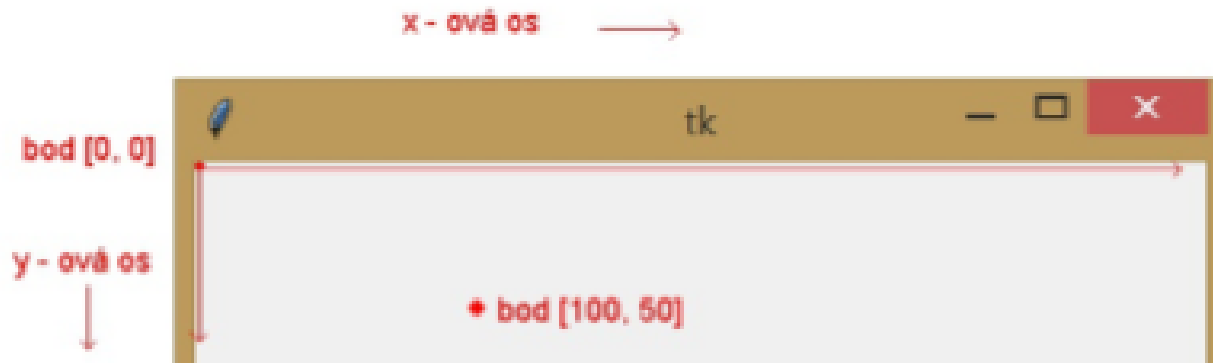
- Do grafickej plochy môžeme kresliť napríklad čiary, obdĺžniky, elipsy.
- Pri kreslení potrebujeme príkazom zadávať súradnice bodov.
- Veľkosť grafickej plochy je zatiaľ 378x264 bodov (pixelov), ale neskôr uvidíme, ako môžeme túto veľkosť

...



Súradnicová sústava

- Každý bod má dve súradnice: x-ovú a y-ovú.
- X-ová súradnica určuje vodorovnú pozíciu a y-ová určuje zvislú pozíciu.
- Ale pozor, y-ová súradnica rastie smerom dole (nie ako sme zvyknutí v matematike, kde rastie smerom hore).
 - Súradnice zapisujeme ako dvojicu čísel, pričom x-ová súradnica je vždy v tejto dvojici prvá a druhá je yová súradnica.



Otázky:

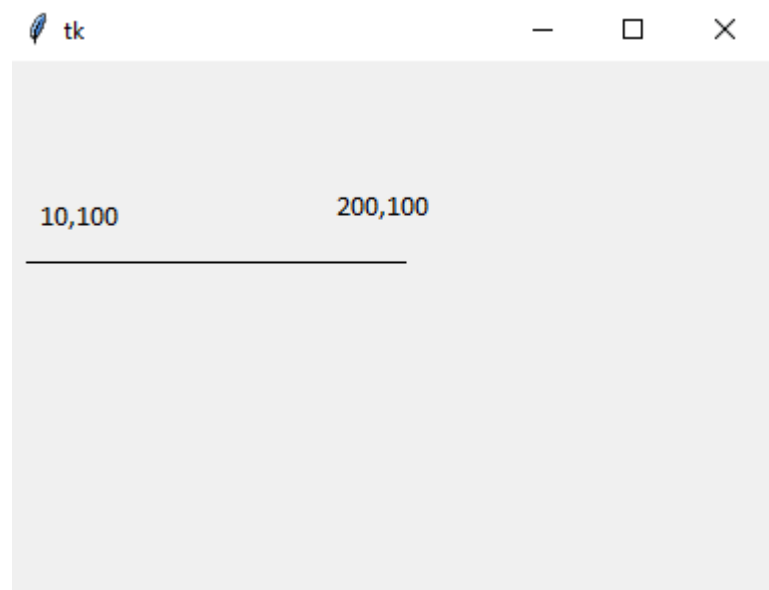
1. Aké súradnice bude mať bod, ktorý je o 20 bodov vľavo od bodu so súradnicami [100, 50]? [80, 50]
2. Aké súradnice bude mať bod, ktorý je o 20 bodov vpravo od bodu so súradnicami [100, 50]? [120, 50]
3. Aké súradnice bude mať bod, ktorý je o 20 bodov hore od bodu so súradnicami [100, 50]? [100, 30]
4. Aké súradnice bude mať bod, ktorý je o 20 bodov dole od bodu so súradnicami [100, 50]? [100, 70]
5. Čo majú spoločné všetky body, ktoré sú vľavo alebo vpravo od bodu [100, 50]?
majú rovnakú y-ovú súradnicu

Kreslenie čiary

- `canvas.create_line(x1, y1, x2,y2)`
- `x1,y1` –súradnice začiatku čiary
- `x2,y2`-súradnice konca čiary

Napr. `canvas.create_line(10, 100, 200, 100)`

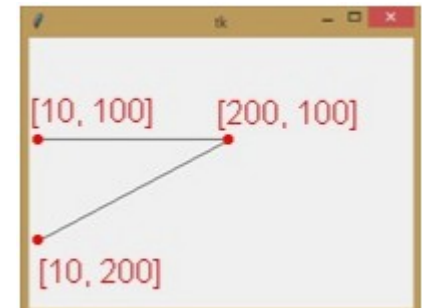
nakreslíme čiaru, ktorá začína v bode so súradnicami `[10, 100]` a končí v bode `[200, 100]`



Kreslenie čiary

- Týmto príkazom môžeme kresliť aj čiary, ktoré prechádzajú cez viacero bodov.
- Jednotlivé súradnice bodov pridávame do príkazu.
- Napríklad: `canvas.create_line(10, 100, 200, 100, 10, 200)` k pôvodnej čiare sme pridali ďalší bod so súradnicou `[10, 200]`
- `canvas.create_line(x1, y1, x2,y2,x3,y3)`
- `x1,y1` -súradnice začiatku čiary
- `x2,y2`-súradnice konca čiary
- `X3,y3`-súradnice bodu do ktorého smeruje druhá úsečka

```
ciara2.py x
1 import tkinter
2 canvas=tkinter.Canvas()
3 canvas.pack()
4 canvas.create_line(10,100,200,100,10,200)
5
```

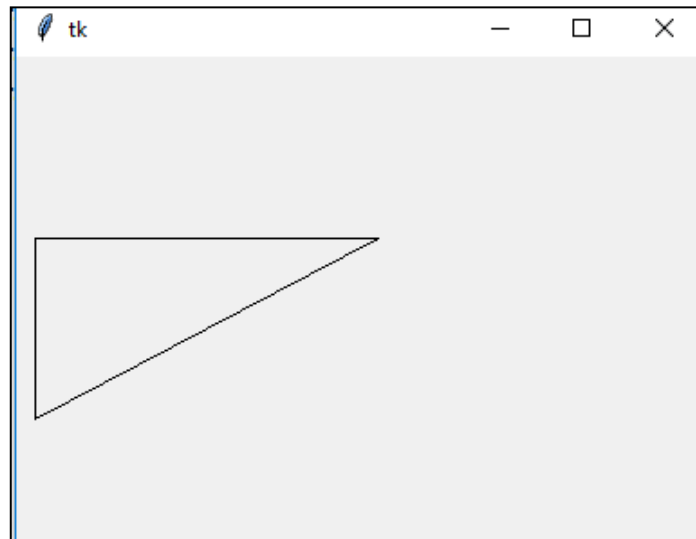


Kreslenie čiar

Úloha:

1 Upravte tento program tak, aby dokreslil tieto dve čiarly do trojuholníka.

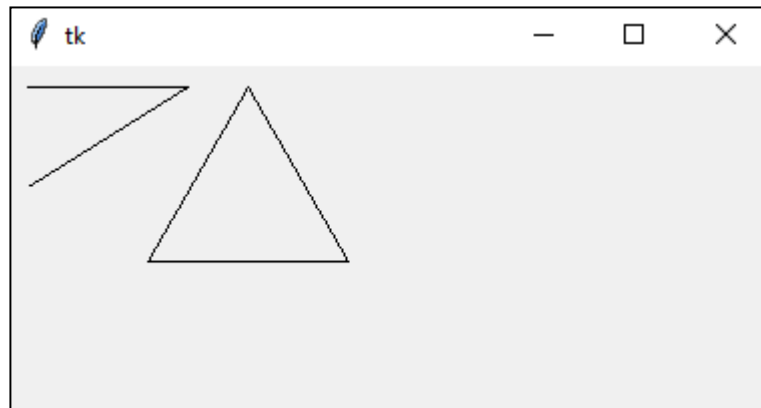
```
ciara3.py x
1 import tkinter
2 canvas=tkinter.Canvas()
3 canvas.pack()
4 canvas.create_line(10,100,200,100,10,200,10,100)
5
```



Kreslenie čiar

- Pri práci s grafikou budeme často experimentovať v priamom režime a keď budeme s výsledkom spokojní, prekopírujeme príkazy zo shellu do programu v editovacom okne a prípadne znovu spustíme pomocou F5.
- V prípade opätovného uvedenia viacerých príkazov `canvas.create_line()` sa vytvoria samostatné krivky

```
ciara4.py x
1 import tkinter
2 canvas=tkinter.Canvas()
3 canvas.pack()
4 canvas.create_line(10, 10, 90, 10, 10, 60)
5 canvas.create_line(120, 10, 70, 97, 170, 97, 120, 10)
6
```



Ďalšie parametre čiar

```
funkcia canvas.create_line()
```

```
canvas.create_line(x, y, x, y, ...)
```

```
canvas.create_line(x, y, x, y, ..., width=číslo, fill=farba)
```

Zrejme, keď tieto dva parametre nešpecifikujeme, kreslí sa čierna čiara hrúbky 1.

Parametre

- **x, y** – dvojica súradníc jedného bodu lomenej čiary
- **width=číslo** – nastavenie hrúbky čiary, ak tento parameter chýba, predpokladá sa hrúbka 1
- **fill=farba** – nastavenie farby čiary, ak tento parameter chýba, predpokladá sa farba 'black'

Funkcia `canvas.create_line()` kreslí lomenú čiaru, ktorá sa môže skladať aj z niekoľkých navzájom napojených úsečiek. Postupnosť dvojíc súradníc by mala obsahovať aspoň dva body (teda aspoň jednu úsečku). Číselné hodnoty pre súradnice môžu byť aj desatinné čísla.

Farby v TKINTERE

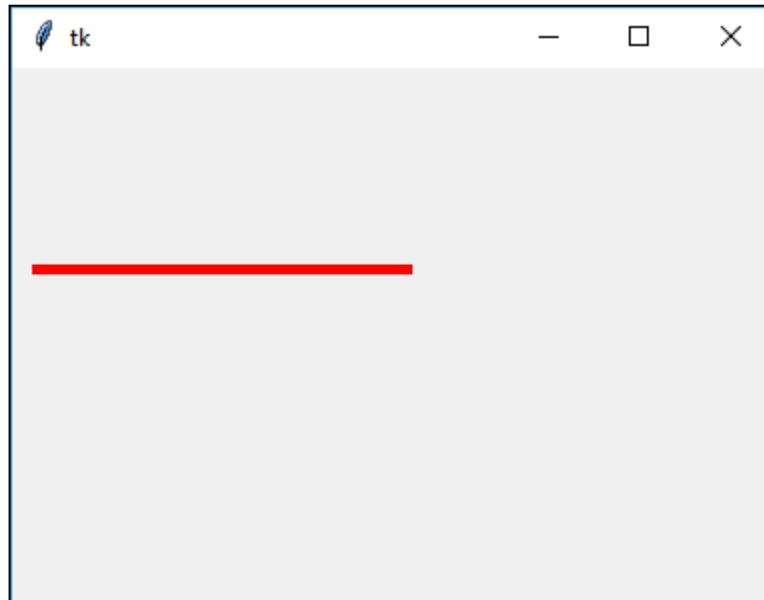
Farby v tkintere zadávame ako znakové reťazce názvov farieb v angličtine.

Najčastejšie budeme používať tieto farby: 'white' ,

- 'black' ,
- 'red' ,
- 'blue' ,
- 'green' ,
- 'yellow'

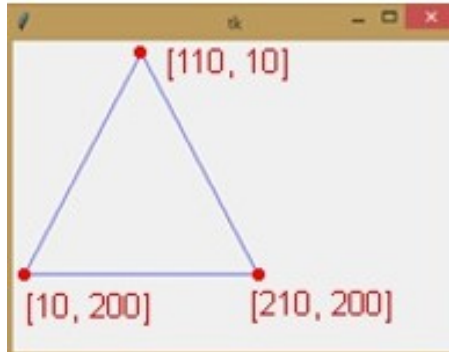
Priklady použitia hrúbky a farby čiary

```
ciara5.py
1 import tkinter
2 canvas=tkinter.Canvas()
3 canvas.pack()
4 canvas.create_line(10, 100, 200, 100, width=5)#čierna hrubá čiara
5 canvas.create_line(10, 100, 200, 100, fill='red')#červená čiara hrubá
6 canvas.create_line(10, 100, 200, 100, fill='red', width=5)#hrubá červená čiara
7 canvas.create_line(10, 100, 200, 100, width=5, fill='red')#zase na tom istom mieste hruba červená čiara
```



Príklady na čiary

ZADANIE

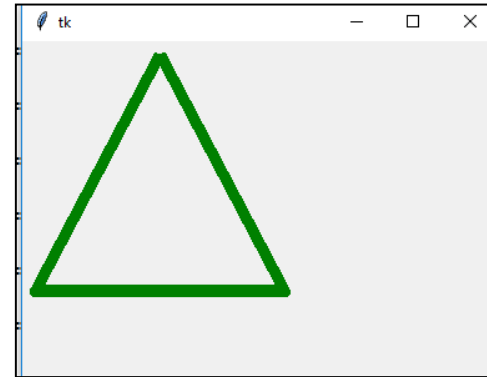
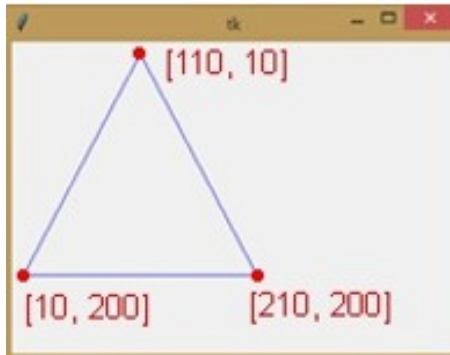


RIEŠENIE 1

```
ciara6.py
1
2 import tkinter
3 canvas = tkinter.Canvas()
4 canvas.pack()
5 canvas.create_line(110, 10, 10, 200, fill='blue')
6 canvas.create_line(10, 200, 210, 200, fill='blue')
7 canvas.create_line(210, 200, 110, 10, fill='blue')
8
```

Príklady na čiaru

ZADANIE



RIEŠENIE 2

```
ciara7.py x
1
2 import tkinter
3 canvas = tkinter.Canvas()
4 canvas.pack()
5 canvas.create_line(110, 10, 10, 200, 210, 200, 110, 10, fill='green', width=10)
6
```


ULOHY NA ČIARY

Úlohy:

2 Zamyslite sa. Môžeme trojuholník v predchádzajúcom programe nakresliť ešte inými spôsobmi?

3 Nakreslite rovnaký trojuholník červenou farbou, posunutý podľa ukážky na obrázku.



4 Zmenou farby a hrúbky čiar nakreslite takýto obrázok:



5 Pomocou čiar nakreslite písmená L, T, H, Z.

6 Pomocou čiar nakreslite obdĺžnik.

Kreslenie obdlžnika

Kreslenie čiar

Príručka učiteľa str.22

Blaho 1.diel str.34