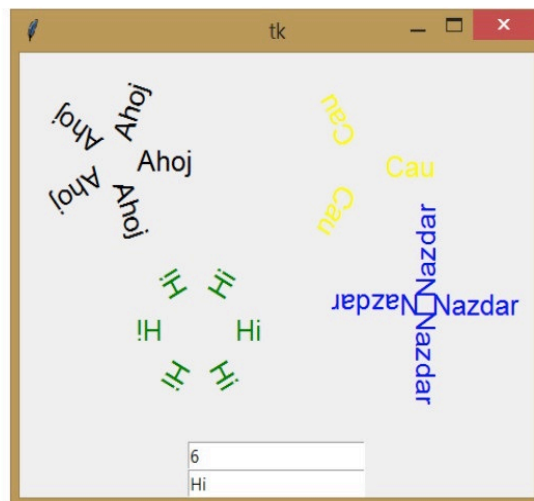
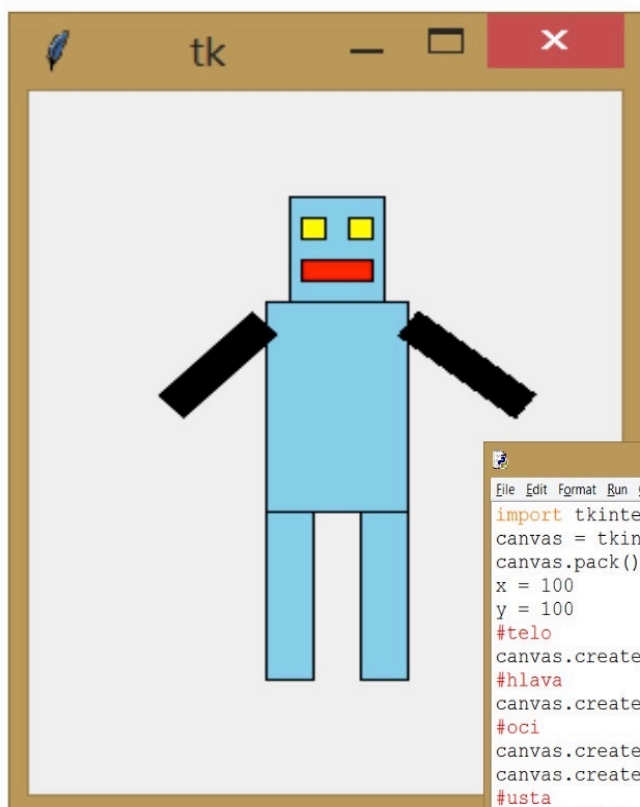


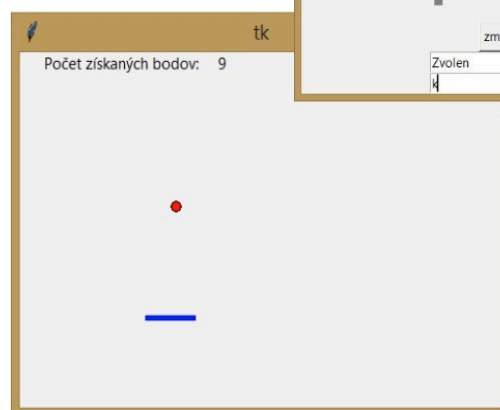
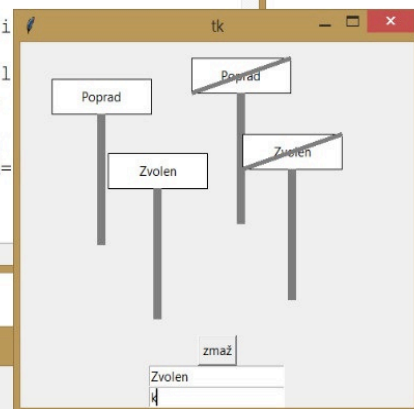
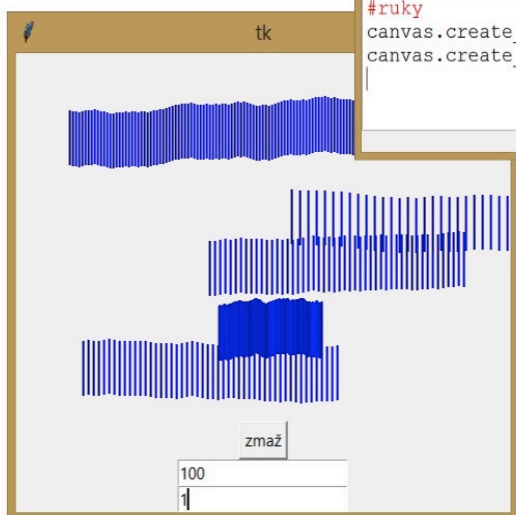
# Programujeme v Pythone

učebnica informatiky pre stredné školy

Peter Kučera



```
robot.py - D:/robot.py (3.5.1)
File Edit Format Run Options Window Help
import tkinter
canvas = tkinter.Canvas()
canvas.pack()
x = 100
y = 100
#telo
canvas.create_rectangle(x, y, x+60, y+100, fill='skyblue')
#hlava
canvas.create_rectangle(x+10, y-50, x+50, y, fill='skyblue')
#oci
canvas.create_rectangle(x+15, y-40, x+25, y-30, fill='yellow')
canvas.create_rectangle(x+35, y-40, x+45, y-30, fill='yellow')
#usta
canvas.create_rectangle(x+15, y-20, x+45, y-10, fill='red')
#nohy
canvas.create_rectangle(x, y+100, x+20, y+180, fill='skyblue')
canvas.create_rectangle(x+40, y+100, x+60, y+180, fill='skyblue')
#ruky
canvas.create_line(x, y+10, x-40, y+50, width=15)
canvas.create_line(x+60, y+10, x+110, y+50, width=15)
```



## **Programujeme v Pythone**

učebnica informatiky pre stredné školy

Autor © Mgr. Peter Kučera

Design © Mgr. Peter Kučera

Jazyková korektúra: Mgr. Katarína Kučerová

Prvé vydanie, 2016

Verzia číslo: 6. 5. 2016

Vydavateľ: Mgr. Peter Kučera

Upozorňujeme, že elektronická kniha je dielom chráneným podľa autorského zákona a je určená len pre osobnú potrebu kupujúceho. Kniha ako celok ani žiadna jej časť nesmie byť voľne šírená na internete, ani nijako ďalej zverejňovaná. V prípade ďalšieho šírenia neoprávnene zasiahnete do autorského práva s dôsledkami podľa platného autorského zákona a trestného zákonníka.

Veľmi si vážime, že e-knihu ďalej nešírite. Len vďaka vašim nákupom dostanú autori, vydavatelia a kníhkupci odmenu za svoju prácu. Ďakujeme, že tak prispievate k vzniku ďalších skvelých kníh.

stránka k učebnici: <https://www.facebook.com/programujemevpythone>

ISBN 978-80-972320-4-7 (pdf)

ISBN 978-80-972320-5-4 (epub)

ISBN 978-80-972320-6-1 (mobi)

# Obsah

## [1 Úvod](#)

## [2 Inštalácia jazyka Python a spustenie prostredia IDLE](#)

## [3 Grafické príkazy](#)

### [3.1 Vytvorenie grafickej plochy](#)

### [3.2 Súradnice bodov a kreslenie čiar](#)

### [3.3 Kreslenie obdĺžnikov](#)

### [3.4 Kreslenie elíps](#)

### [3.5 Písanie textu do grafickej plochy](#)

## [4 Premenné a náhodné hodnoty](#)

### [4.1 Náhodné čísla a príkazový režim \(shell\)](#)

### [4.2 Premenné a náhodná farba](#)

### [4.3 Kreslenie obrázkov na náhodnom mieste](#)

## [5 Opakovanie časti programu - for cyklus](#)

### [5.1 Opakujeme vykonávanie príkazov](#)

### [5.2 For cyklom kreslíme obrázky s pravidelnosťou](#)

## [6 Vytvárame podprogramy](#)

## [7 Úlohy na opakovanie I](#)

## [8 Klikanie myšou a ovládanie klávesnicou](#)

### [8.1 Reakcia na ľavé tlačidlo myši](#)

### [8.2 Reakcia na pravé tlačidlo myši](#)

### [8.3 Reakcia na stláčanie klávesnice](#)

## [9 Podmienené príkazy](#)

## [10 Časovač](#)

## [11 Tlačidlá a vstupné pole](#)

## [12 Posúvanie objektov canvasu](#)

## [13 Vytvárame jednoduché hry](#)

### [13.1 Testovač pozornosti](#)

### [13.2 Chytanie loptičiek](#)

### [13.3 Hľadaj percento](#)

## [14 Úlohy na opakovanie II](#)

## [15 Pracujeme s textom](#)

## [16 Test - Preteky rytierov](#)

## [17 Poznámky](#)

### [17.1 Používanie príkazu import](#)

### [17.2 Náhodné farby](#)

### [17.3 Zadávanie vstupu - entry](#)

## [18 Použitá literatúra](#)

# 1 Úvod

## Niečo o jazyku Python

Autorom jazyka Python je Guido van Rossum - počítačový programátor, ktorý sa narodil a vyrástol v Holandsku. Python vznikol v roku 1991 a názov získal podľa Monty Pythonovho Lietajúceho cirkusu (autor bol jeho fanúšikom). V súčasnosti je Python veľmi populárny a jeho popularita ešte rastie. Je to moderný programovací jazyk, ktorý podporuje rôzne programovacie paradigmy. Python je freeware a open source. Používa ho CERN, Google, Facebook, YouTube, Mozilla a iní. Beží na rôznych platformách, napr. Linux, Windows, Mac. Python sa učia študenti na mnohých špičkových univerzitách ako svoj úvodný programovací jazyk (MIT, Berkeley, ...).

## Komu je určená táto učebnica?

Učebnica je určená pre študentov strednej školy pre predmet informatika, je koncipovaná aj pre samoukov. Tiež môže pomôcť učiteľom, ktorí sa rozhodujú pre vyučovanie jazyka Python.

## Podrobnejšie o učebnici

Podľa Štátneho vzdelávacieho programu (ŠVP) pre gymnáziá programovanie tvorí jednu z piatich oblastí ŠVP - Algoritmické riešenie problémov. V ŠVP sa zámerne neurčuje konkrétny softvér a programovací jazyk, ktorý má škola vyučovať. Výber programovacieho jazyka je v kompetencii školy a učiteľa. V našom školskom prostredí má zatiaľ najväčšiu tradíciu jazyk Pascal, resp. objektový Pascal v prostrediach Delphi a Lazarus. Dnes je už tento jazyk starý a v praxi sa takmer nikde nepoužíva. Učebnica vychádza z našich niekoľkoročných skúseností z vyučovania programovania na gymnáziu v jazykoch Pascal, Object Pascal a z ročnej skúsenosti v jazyku Python. Aj u nás v škole sme riešili neaktuálnosť Delphi a Lazarusu a aj vďaka inšpiráciám od RNDr. Andreja Blaha PhD. sme prešli na jazyk Python. Keďže na tento jazyk v tom čase neexistovala vhodná učebnica, rozhodli sme sa ju napísať. Bude nás tešiť, keď pomôžeme aj pomocou tejto učebnice, aj pomocou vzdelávania v Klube učiteľov informatiky spopularizovať tento jazyk aj na slovenských stredných školách.

Učebnica reflektuje obsahový a výkonový štandard ŠVP. Obsahuje kapitoly, ktoré podľa nás majú zvládnuť aj študenti, ktorí sa nebudú pripravovať na maturitu z informatiky. Obsah učebnice pokrýva základný kurz programovania v rozsahu približne 33 vyučovacích hodín, pričom niektoré z úloh odporúčame zadávať na domácu prípravu. Prvých 16 vyučovacích hodín je odčlenených siedmou kapitolou - Úlohy na opakovanie I. Ďalších 16 hodín končí štrnástou kapitolou - Úlohy na opakovanie II. Kapitola 15 - pracujeme s textom je doplňujúca kapitola, ktorá je vhodná pre šikovnejšie skupiny alebo šikovnejších študentov v skupine. Nie však pre svoju náročnosť, ale len z časového hľadiska.

Učebnica je názorná, obsahuje množstvo praktických úloh, využíva grafické prostredie knižnice tkinter. Študenti sa na začiatku naučia kresliť grafické útvary, používať cyklus a vetvenie, ovládať program klávesnicou a myšou. V závere budú tvoriť animácie a jednoduché hry. V závere učebnice je jeden záverečný test z nášho školského prostredia.

## Dávame do pozornosti

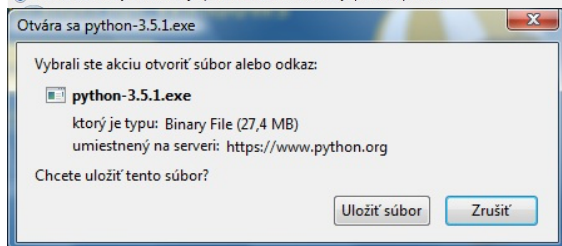
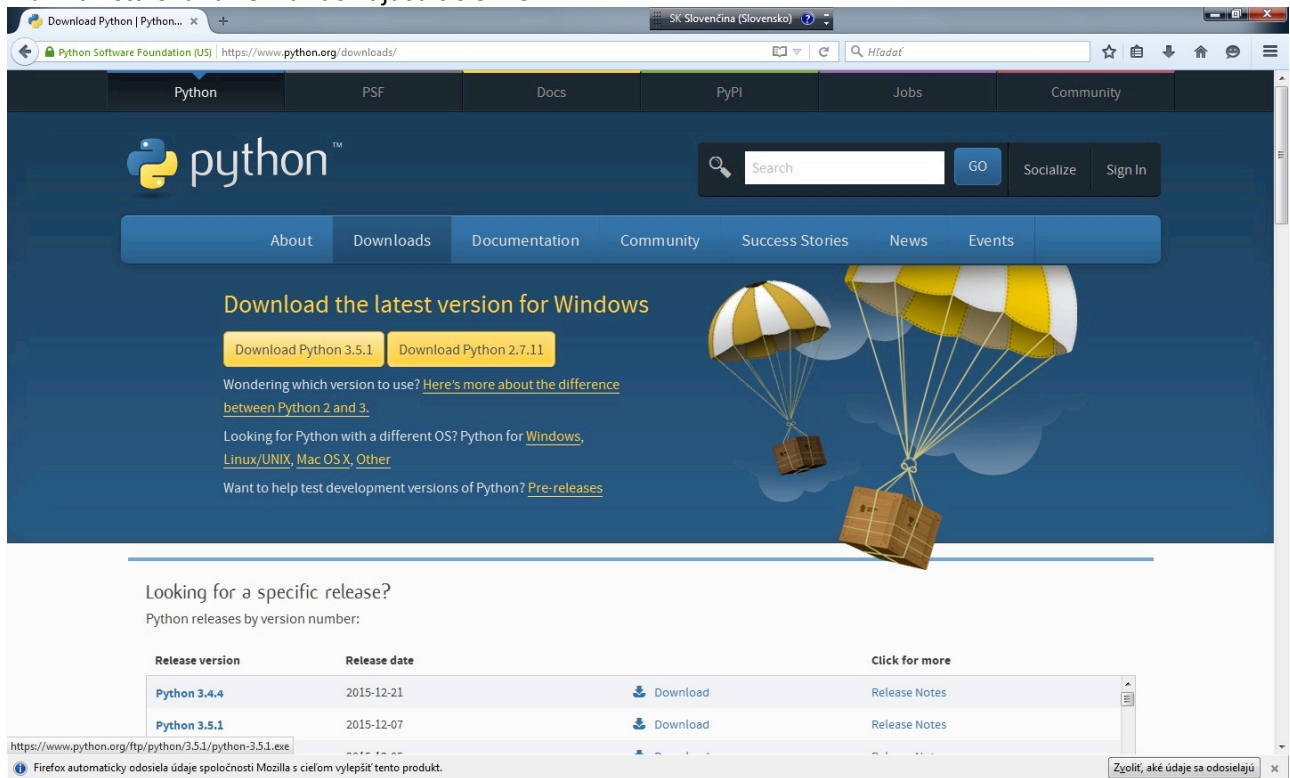
Stránku <https://www.facebook.com/programujemevpythone>, kde môžete písať otázky alebo odozvy na učebnicu.

Učiteľom dávame do pozornosti aj Klub učiteľov informatiky v Bratislave na stránke: <http://www.1sg.sk/itklub>

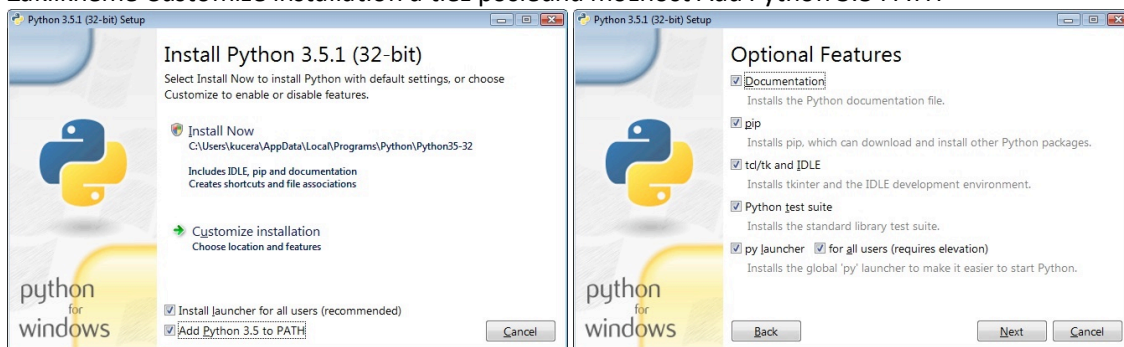
Prajeme vám príjemné chvíle pri učení sa Pythonu :)  
autor

## 2 Inštalácia jazyka Python a spustenie prostredia IDLE

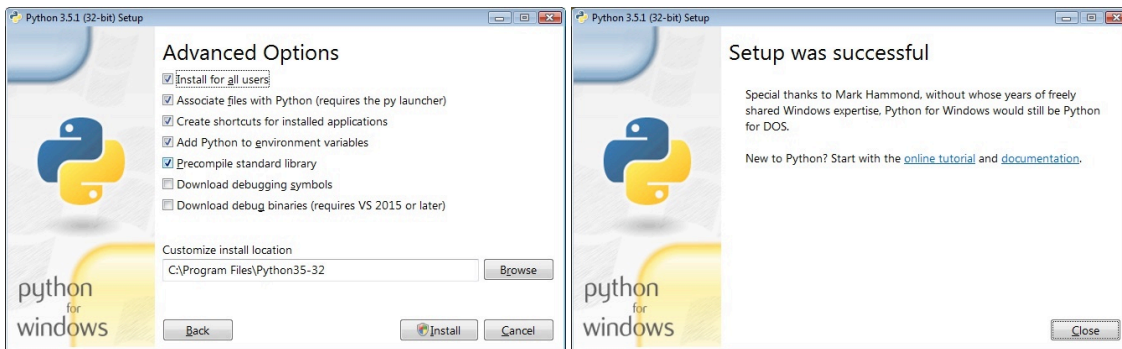
Na stránke [www.python.org](http://www.python.org) si stiahneme Python 3.5.1 (číslo verzie) vhodný pre náš počítač. Ak máme 64-bitovú verziu operačného systému, stiahneme si Python pre 64-bitový operačný systém. Dôležité je, aby sme mali nainštalovanú verziu začínajúcu číslom 3.



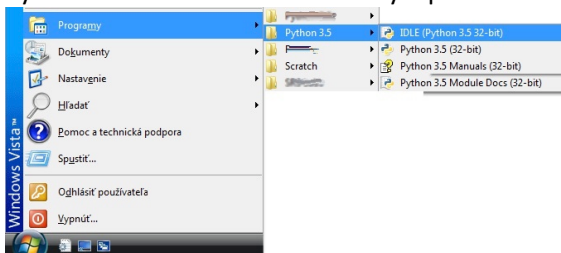
Zaklikneme Customize installation a tiež poslednú možnosť Add Python 3.5 PATH



V okne Advanced Options potvrdíme možnosť Install for all users.



Python 3.5.1 máme nainštalovaný. Spúšťame ho ikonou pre IDLE (Python 3.5 ....)



Po spustení sa otvorí okno s názvom Python Shell. V tomto okne sa dajú priamo písať príkazy a po potvrdení enterom sa automaticky vykonajú. Je to interaktívne prostredie, označujeme ho aj názvom príkazový režim alebo len shell. My budeme najprv zapisovať program v programovacom režime. To znamená, že v okne shell vyberieme z hlavného menu File a v ňom ponuku New File. Otvorilo sa nám nové okno, kde zapíšeme program. Program uložíme. Ak ho chceme spustiť, stlačíme F5. Po spustení sa nám do shell-u napíše informácia, ktorý program sme spustili, a tiež sa reštartujú všetky nastavenia Pythonu v príkazovom režime.


## 3 Grafické príkazy

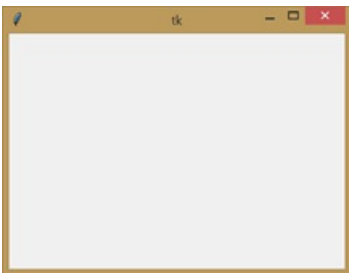
### 3.1 Vytvorenie grafickej plochy

Máme spustené prostredie IDLE a vytvorený nový súbor - sme v programovacom režime. Postup spustenia a vytvorenia súboru sme si ukázali v predchádzajúcej kapitole. Keď chceme pracovať s grafickou plochou, musíme najprv do programu nainportovať modul na prácu s grafickou plochou (je to nejaký program, v ktorom už niekto naprogramoval príkazy, ktoré budeme používať). My budeme na prácu s grafikou používať modul `tkinter`. Tento modul nainportujeme na začiatku programu zápisom `import tkinter`.

Po nainportovaní modulu vytvoríme grafickú plochu (plátno), do ktorej budeme neskôr kresliť `canvas = tkinter.Canvas()`. Príkaz `canvas.pack()` zabezpečí zobrazenie nového okna aj s vytvorenou grafickou plochou. Celý program zatiaľ vyzerá takto:

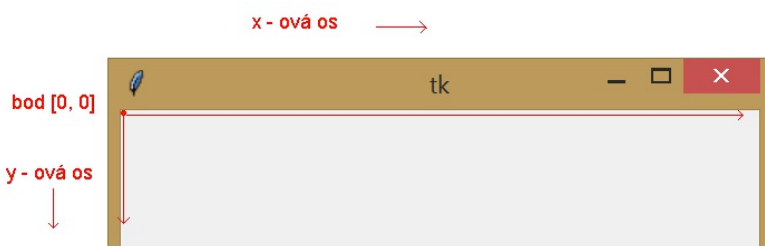
```
import tkinter
canvas = tkinter.Canvas()
canvas.pack()
```

Keď spustíme tento program (F5), vytvorí sa okno s grafickou plochou. V okne zatiaľ nič nevidíme, lebo sme doň nič nenakreslili. Okno sa zatvorí klasicky .

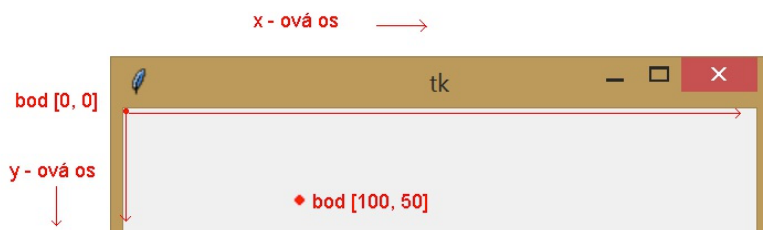


### 3.2 Súradnice bodov a kreslenie čiar

Do grafickej plochy môžeme kresliť napríklad čiary, obdĺžniky, elipsy. Pri kreslení potrebujeme príkazom zadávať súradnice bodov.



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 y-ová súradnica.



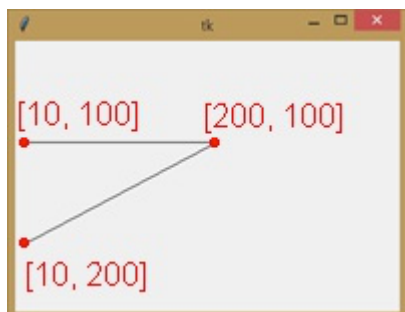
### Otázky:

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

Príkazom `canvas.create_line(10, 100, 200, 100)` nakreslíme čiaru, ktorá začína v bode so súradnicami [10, 100] a končí v bode [200, 100]. 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]. Po spustení programu:

```
import tkinter
canvas = tkinter.Canvas()
canvas.pack()
canvas.create_line(10, 100, 200, 100, 10, 200)
```

budeme vidieť čiaru, ktorá začína v bode [10, 100], pokračuje do bodu [200, 10] a končí v bode [10, 200].



### Úloha:

1

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

Pri kreslení môžeme čiaram nastaviť aj hrúbku, aj farbu.

```
canvas.create_line(10, 100, 200, 100, width=5)
```

Nakreslí čiaru s hrúbkou 5 bodov.

```
canvas.create_line(10, 100, 200, 100, fill='red')
```

Nakreslí červenú čiaru.

```
canvas.create_line(10, 100, 200, 100, fill='red', width=5)
```

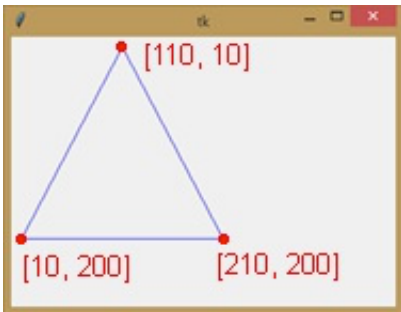
alebo



```
canvas.create_line(10, 100, 200, 100, width=5, fill='red')
```

nakreslí červenú čiaru s hrúbkou 5 bodov. Hrúbku nastavujeme parametrom `width` a uvedieme mu číselnú hodnotu. Zafarbenie nastavujeme parametrom `fill` a v apostrofoch zapíšeme názov farby (môžeme použiť rôzne farby, napríklad: `'white'`, `'black'`, `'red'`, `'blue'`, `'yellow'`, `'green'`, `'maroon'`, `'orange'`, `'gray'`, `'skyblue'`, `'violet'`, `'fuchsia'`, `'olive'`).

Znak `'` môžeme napísať na SK klávesnici stlačením `Ctrl+Alt+P`. Krok späť urobíme stlačením `Ctrl+Z`



Tento trojuholník môžeme nakresliť rôznymi postupmi:

```
import tkinter
canvas = tkinter.Canvas()
canvas.pack()
canvas.create_line(110, 10, 10, 200, fill='blue')
canvas.create_line(10, 200, 210, 200, fill='blue')
canvas.create_line(210, 200, 110, 10, fill='blue')
```

alebo

```
import tkinter
canvas = tkinter.Canvas()
canvas.pack()
canvas.create_line(110, 10, 10, 200, 210, 200, 110, 10, fill='blue')
```

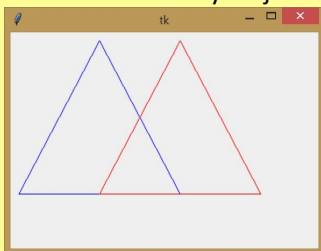
### Úlohy:

2

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

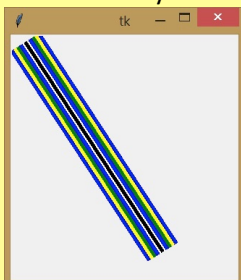
3

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.

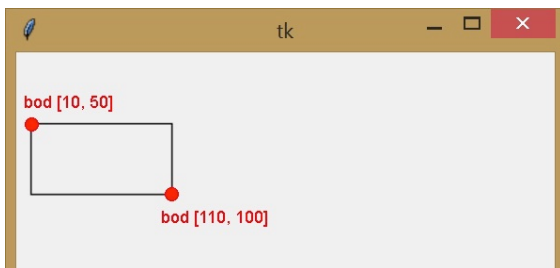
6 Pomocou čiar nakreslite obdĺžnik.

## 3.3 Kreslenie obdĺžnikov

### Otázky:

6. Obdĺžnik už vieme nakresliť pomocou príkazu na kreslenie čiar. Ak kreslíme obdĺžnik pomocou čiar, najmenej koľko bodov (súradníc týchto bodov) musíme v príkaze na kreslenie čiar použiť?
7. Vieme jednoznačne určiť obdĺžnik aj pomocou menšieho počtu bodov? Najmenej koľko ich potrebujeme? Svoju odpoveď zdôvodnite.

Príkaz `canvas.create_rectangle(10, 50, 110, 100)` nakreslí obdĺžnik, ktorý je zadaný pomocou bodu so súradnicou [10, 50] a [110, 100].



### Otázky:

8. Akú šírku v bodoch má obdĺžnik na obrázku?
9. Akú výšku v bodoch má obdĺžnik na obrázku?
10. Môžeme tento istý obdĺžnik nakresliť aj týmto príkazom `canvas.create_rectangle(110, 100, 10, 50)`?
11. Môžeme tento istý obdĺžnik (na tom istom mieste) nakresliť ešte pomocou iných súradníc?
12. Čo sa nakreslí po zadaní tohto príkazu `canvas.create_rectangle(10, 50, 110, 50)`?
13. Čo sa nakreslí po zadaní tohto príkazu `canvas.create_rectangle(10, 50, 10, 50)`?
14. Akú výšku a šírku majú predchádzajúce dva obdĺžniky?

### Úlohy:

7 Nakreslite hneď vpravo vedľa obdĺžnika `canvas.create_rectangle(10, 50, 110, 100)` rovnako veľký obdĺžnik, ktorý sa ho bude dotýkať.

8 Nakreslite hneď pod ľavým obdĺžnikom (z predchádzajúcej úlohy) rovnako veľký obdĺžnik.



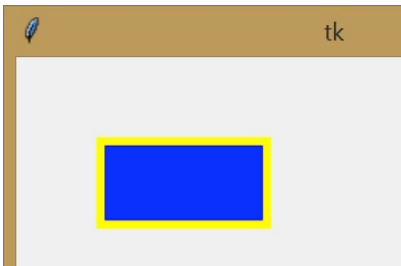
### Otázky:

15. Musíme pri zadávaní príkazov na kreslenie obdĺžnikov v úlohách číslo 7 a 8 vymýšľať alebo vypočítavať súradnice všetkých bodov?
16. Majú nové body, ktoré zadávame, niečo spoločné s bodmi, ktoré sme už zadali?

Aj pri kreslení obdĺžnika môžeme použiť parameter na nastavenie farby rovnakým spôsobom ako pri nastavení farby pri kreslení čiar. Parameter `fill` pri kreslení obdĺžnika nastavuje vnútornú farbu, čiže rovnako výplň kresleného útvaru. Neovplyvňuje ale farbu obrysu obdĺžnika, ako by sme si mohli myslieť. Na nastavenie farby strán obdĺžnika môžeme použiť parameter `outline`. Čiže príkaz:

```
canvas.create_rectangle(50, 50, 150, 100, fill='blue', outline='yellow', width=5)
```

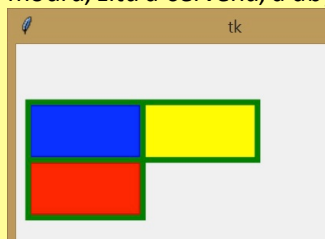
nakreslí obdĺžnik vyplnený modrou farbou so žltým obrysom, ktorý má hrúbku 5 bodov. Pripomeňme si, že tieto parametre môžu byť v rôznom poradí.



### Úlohy:

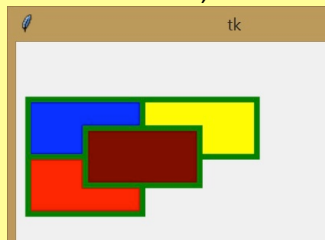
9

Trom obdĺžnikom z úlohy č. 8 nastavte parametre tak, aby jednotlivé obdĺžniky mali tieto výplne: modrú, žltú a červenú, a aby farba ich obrysu bola zelená a obrys mal hrúbku 5 bodov.



10

Doplňte do tohto obrázku štvrtý, rovnako veľký obdĺžnik, ktorého stred je v spoločnom bode všetkých troch obdĺžnikov, a nastavte mu hnedú výplň.



### Otázky:

17. Čo sa stane, keď parametre na nastavenie farby pomiešame medzi súradnice bodov `canvas.create_rectangle(outline='yellow', 50, 50, 150, 100, fill='blue')`?
18. V úlohe číslo 10 sme nakreslili štyri obdĺžniky štyrmi príkazmi. Záleží na poradí týchto príkazov? Čo sa stane, keď hnedý obdĺžnik budeme kresliť ako prvý? Poradie ktorých príkazov môžeme zameniť, aby sa nám nakreslil rovnaký obrázok?
19. Čo sa stane, keď parameter `outline` nastavíme takto: `outline=' '`?

20. Čo sa stane, keď parameter fill nastavíme takto: `fill=' '`?
21. Čo sa stane, keď parameter fill a súčasne parameter outline nastavíme takto: `fill='',outline=''`?
22. Koľko obdĺžnikov budeme vidieť po spustení týchto príkazov?
- a) `canvas.create_rectangle(120, 100, 170, 150)`  
`canvas.create_rectangle(100, 100, 150, 150)`  
`canvas.create_rectangle(150, 100, 200, 150)`  
`canvas.create_rectangle(100, 150, 150, 200)`
- b) `canvas.create_rectangle(120, 100, 170, 150, fill='white')`  
`canvas.create_rectangle(100, 100, 150, 150, fill='')`  
`canvas.create_rectangle(150, 100, 200, 150, fill='green')`  
`canvas.create_rectangle(100, 150, 150, 200, fill='', outline='')`

### Úlohy:

**11** Nakreslite vlajku: a) Poľska, b) Francúzska, c) Nemecka, d) Maďarska, e) Švajčiarska, f) Izraela.

**12** Iba dvomi príkazmi nakreslite vlajku Lotyšska.



**13** Nakreslite vlajku Grécka.



**14** Dano chcel nakresliť vlajku Francúzska a napísal tento program. Nájdite chybu v tomto programe.

```
import tkinter
canvas = tkinter.Canvas()
canvas.pack()
canvas.create_rectangle(10,50,110,300, fill='blue', outline='')
canvas.create_rectangle(60,50,160,300, fill='white', outline='')
canvas.create_rectangle(110,50,210,300, fill='red', outline='')
```

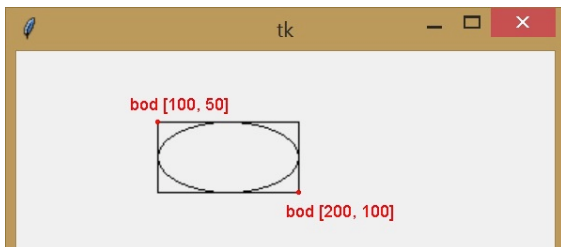
## 3.4 Kreslenie elíps

Už vieme, že príkaz `canvas.create_rectangle(100, 50, 200, 100)` nakreslí obdĺžnik, ktorého ľavý horný bod má súradnicu [100, 50] a pravý dolný bod má súradnicu [200, 100]. Ak použijeme rovnaké súradnice, ale zmeníme príkaz `canvas.create_rectangle` na `canvas.create_oval`, nakreslí sa elipsa.

Tento program s rovnakými súradnicami, ale rôznymi príkazmi:

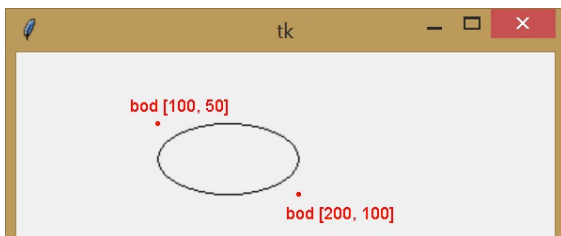
```
import tkinter
canvas = tkinter.Canvas()
canvas.pack()
canvas.create_rectangle(100, 50, 200, 100)
canvas.create_oval(100, 50, 200, 100)
```

nakreslí aj obdĺžnik, aj elipsu. Ako vidíme na obrázku, súradnice bodov, ktoré určujú elipsu (ovál), sa nenachádzajú na elipse, ale mimo nej. Tieto body sú súradnicami obdĺžnika, v ktorom sa nakreslí vpísaná elipsa (čiže najväčšia možná elipsa, ktorá sa zmestí do tohto pomyselného obdĺžnika).



Program len s príkazom na kreslenie elipsy nakreslí len samotnú elipsu:

```
import tkinter
canvas = tkinter.Canvas()
canvas.pack()
canvas.create_oval(100, 50, 200, 100)
```



Pri kreslení elipsy nám zo začiatku môže pri vymýšľaní súradníc pomôcť, keď si najprv nakreslíme správny obdĺžnik (v ňom si predstavujeme najväčšiu elipsu) a až potom, keď máme správne súradnice, zmeníme tento príkaz na elipsu.

### Otázky:

23. Vieme nakresliť obdĺžnik jedným príkazom. Potrebujeme špeciálny príkaz na kreslenie štvorca? Podľa čoho rozpoznáme, že súradnice v príkaze pre obdĺžnik nakreslia štvorec?
24. Môžeme pomocou príkazu `canvas.create_oval` kresliť kruhy? Potrebujeme špeciálny príkaz na kreslenie kruhov? Podľa čoho rozpoznáme, že súradnice v príkaze pre ovál nakreslia kruh?

V jednom programe môžeme kresliť aj viacero obrázkov vedľa seba. Ak nám nestačí veľkosť canvasu, alebo chceme zmeniť jeho farbu pozadia, môžeme tak urobiť už na začiatku programu pri vytváraní canvasu. Parametrom `height` nastavíme výšku, parametrom `width` nastavíme šírku a farbu pozadia nastavíme parametrom `bg`.

```
import tkinter
canvas = tkinter.Canvas(bg='white', width=800, height=600)
canvas.pack()
```

### Úlohy:

**15** K elipse `canvas.create_oval(100, 50, 200, 100)` nakreslite tesne vedľa nej rovnako veľkú elipsu.

**16** Pre obe elipsy nastavte rôzne farby výplne, obrysov a tiež hrúbky čiar.

**17** Nakreslite vlajku Japonska.

**18** Nakreslite snehuliaka.

**19** Nakreslite symetrickú tvár s očami, nosom a ústami:

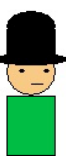


**20** Doplnite k tvári okuliare.

**21** Nakreslite vlajku červeného polmesiaca:



**22** Nakreslite postavičku s klobúkom:

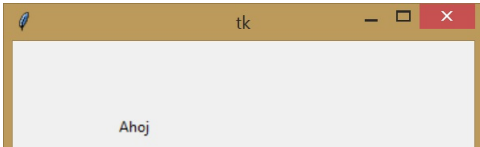


**23** Nakreslite farebný terč.

## 3.5 Písanie textu do grafickej plochy

Príkaz `canvas.create_text(100, 70, text='Ahoj')` napíše na súradnicu [100, 70] text Ahoj. Prvé dva parametre tohto príkazu sú súradnice stredu vypisovaného textu. Text sa napíše tak, aby aj v zvislom, aj vo vodorovnom smere boli súradnice presne vycentrované v strede napísaného textu. Parametrom `text` zadávame samotný text, ktorý chceme napísať.

```
import tkinter
canvas = tkinter.Canvas()
canvas.pack()
canvas.create_text(100, 70, text='Ahoj')
```



V príkaze môžeme použiť aj parameter `font`, ktorým určíme typ, veľkosť a rez písma. Napríklad `font='Arial 70 bold'`. Takto môžeme zadávať len jednoslovné názvy fontov. Viacslovný názov sa zadáva komplikovanejším spôsobom.

```
import tkinter
canvas = tkinter.Canvas()
canvas.pack()
canvas.create_text(100, 70, text='Ahoj')
canvas.create_text(200, 50, text='Python', font='Arial 70 bold')
```

Program napíše tieto texty:



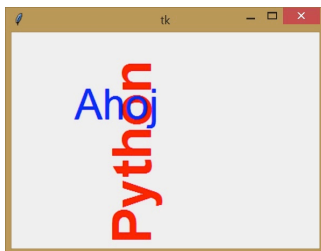
Farbu výplne písma určujeme parametrom `fill`, rovnako ako v príkazoch, ktoré už poznáme. Písmo môžeme aj otáčať parametrom `angle` (nefunguje to na počítačoch Mac).

```
canvas.create_text(100, 70, text='Ahoj', fill='blue')
canvas.create_text(200, 50, text='Python', font='Arial 70 bold', fill='red')
```



```
import tkinter
canvas = tkinter.Canvas()
canvas.pack()
canvas.create_text(150, 150, text='Python', font='Arial 50 bold', fill='red',
angle=90)
canvas.create_text(130, 90, text='Ahoj', fill='blue', font='Arial 40')
```

Nápis Python sme otočili o 90 stupňov. Najprv sa nakreslí tento nápis a potom sa modrou farbou napíše Ahoj. Pri nastavení uhla sa text otáča okolo zadanej súradnice, čiže presného stredu textu (fiktívneho opísaného obdĺžnika textu). Otáčanie určuje uhol v protismere hodinových ručičiek (ako sme zvyknutí na matematike). Uhol 180 stupňov nám text otočí dole hlavou. Ak chceme text otáčať v smere hodinových ručičiek, môžeme použiť záporný uhol.



Úlohy:

24 Nakreslite tieto dopravné značky:



25 Nakreslite tieto dopravné značky:

