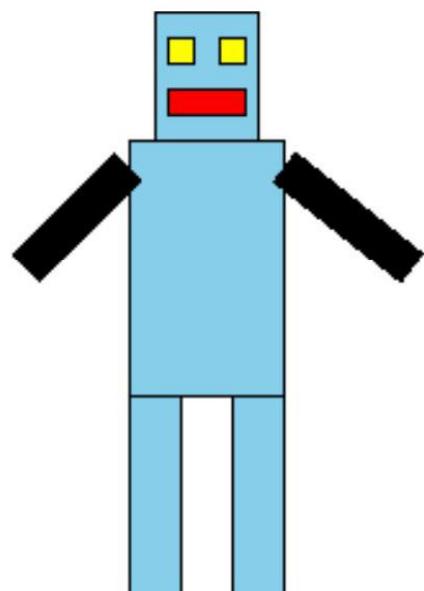


Programujeme v Pythonu

Peter Kučera



import tkinter
canvas.create_line

randrange
canvas.create_oval
canvas.create_text
x=200

canvas.create_rectangle
x=i*10

min_cislo = int(entry1.get())
button1=tkinter.Button(start,command='start')
canvas.bind('<Button-1>', klik)
canvas.bind_all('<Up>', nahoru)
if y<200:
 canvas.after(100, ukaz)
def mic(x,y):

for i in range(10):
 print(x)

www.programujemevpythonu.cz

Programujeme v Pythonu

učebnice informatiky pro střední školy

Autor © Mgr. Peter Kučera

Design © Mgr. Peter Kučera

Překlad: Helena Murínová

Jazyková korektura: Ema Andrejsová

První vydání, 2017

Verze číslo: 20170111

Vydavatel: Mgr. Peter Kučera

Upozorňujeme, že elektronická kniha je chráněným dílem podle autorského zákona a je určena pouze pro osobní potřebu kupujícího. Kniha jako celek ani žádná její část nesmí být volně šířena na internetu, ani nijak dále publikována. V případě dalšího šíření neoprávněně zasáhnete do autorského práva s důsledky podle platného autorského zákona a trestního zákoníku.

Velmi si vážíme, že e-knihu dále nešíříte. Jen díky vašemu nákupu dostanou autoři, vydavatelé a knihkupci odměnu za svou práci. Děkujeme, že tak přispíváte ke vzniku dalších skvělých knih.

Učebnici a další materiály si můžete zakoupit i přímo na stránkách autora:

<http://www.programujemevpythonu.cz> alebo <https://www.facebook.com/programujemevpython>

ISBN 978-80-972580-0-9 (pdf)

ISBN 978-80-972580-1-6 (epub)

ISBN 978-80-972580-2-3 (mobi)

Obsah

[1 Úvod](#)

[2 Instalace jazyku Python a spuštění prostředí IDLE](#)

[3 Grafické příkazy](#)

[3.1 Vytvoření grafické plochy](#)

[3.2 Souřadnice bodů a kreslení čar](#)

[3.3 Kreslení obdélníků](#)

[3.4 Kreslení elips](#)

[3.5 Psaní textu do grafické plochy](#)

[4 Proměnné a náhodné hodnoty](#)

[4.1 Náhodná čísla a příkazový režim \(shell\)](#)

[4.2 Proměnné a náhodná barva](#)

[4.3 Kreslení obrázků na náhodném místě](#)

[5 Opakování části programu – for cyklus](#)

[5.1 Opakujeme vykonávání příkazů](#)

[5.2 For cyklem kreslíme obrázky s pravidelností](#)

[6 Vytváříme podprogramy](#)

[7 Úlohy na opakování I](#)

[8 Klikání myší a ovládání klávesnicí](#)

[8.1 Reakce na levé tlačítko myši](#)

[8.2 Reakce na pravé tlačítko myši](#)

[8.3 Reakce na mačkání klávesnice](#)

[9 Podmíněné příkazy](#)

[10 Časovač](#)

[11 Tlačítka a vstupní pole](#)

[12 Posouvání objektů canvasu](#)

[13 Vytváříme jednoduché hry](#)

[13.1 Testovač pozornosti](#)

[13.2 Chytání míčků](#)

[13.3 Hledej procento](#)

[14 Úlohy na opakování II](#)

[15 Pracujeme s textem](#)

[16 Test – Závody rytířů](#)

[17 Poznámky](#)

[17.1 Používání příkazu import](#)

[17.2 Náhodné barvy](#)

[17.3 Zadávání vstupu – entry](#)

[18 Bibliografie](#)

1 Úvod

Něco o jazyku Python

Autorem jazyka Python je Guido van Rossum – počítačový programátor, který se narodil a vyrostl v Nizozemí. Python vznikl v roce 1991 a název získal podle Monty Pythonova Létajícího cirkusu (autor byl jeho fanouškem). V současnosti je Python velmi populární a jeho popularita dále roste. Je to moderní programovací jazyk, který podporuje různá programovací paradigmata. Python je freeware a open source. Používá ho CERN, Google, Facebook, YouTube, Mozilla a jiní. Běží na různých platformách, např. Linux, Windows, Mac. Programovat v Pythonu se učí studenti na mnoha špičkových univerzitách, kde jej mají jako úvodní programovací jazyk (MIT, Berkeley, ...).

Pro koho je určena tato učebnice?

Učebnice je určena pro studenty středních škol k výuce předmětu informatika, ale i pro samouky, kteří se chtějí seznámit s Pythonem. Také může pomoci učitelům, kteří se rozhodli pro výuku jazyka Python.

Podrobněji o učebnici

V našem školním prostředí má zatím největší tradici jazyk Pascal, resp. objektový Pascal v prostředích Delphi a Lazarus. Dnes je tento jazyk již starý a v praxi se téměř nikde nepoužívá. Učebnice vychází z našich několikaletých zkušeností s výukou programování na gymnáziu v jazycích Pascal, Object Pascal a Python. I u nás ve škole jsme řešili neaktuálnost Delphi a Lazarus a rozhodli jsme se přejít na jazyk Python. Jelikož na tento jazyk v té době na Slovensku neexistovala vhodná učebnice, rozhodli jsme se ji napsat. Potěší nás, pokud pomůžeme i díky této učebnici zpopularizovat tento programovací jazyk i na českých středních školách.

Učebnice obsahuje kapitoly, které podle nás mají zvládnout i studenti, kteří se nebudou připravovat na maturitu z informatiky. Obsah učebnice pokrývá základní kurz programování v rozsahu přibližně 33 vyučovacích hodin, přičemž některé z úkolů doporučujeme zadávat na domácí přípravu. Prvních 16 vyučovacích hodin je oddělených sedmou kapitolou – Úkoly na opakování I. Dalších 16 hodin končí čtrnáctou kapitolou – Úkoly na opakování II. Kapitola 15 – Pracujeme s textem je doplňující kapitola, která je vhodná pro šikovnější skupiny nebo chytřejší studenty ve skupině. Ne však pro svou náročnost, ale jen z časového hlediska.

Učebnice je názorná, obsahuje řadu praktických úkolů, využívá grafické prostředí knihovny tkinter. Studenti se na začátku naučí kreslit grafické útvary, používat cyklus a větvení, ovládat program klávesnicí a myší. V závěru budou tvořit animace a jednoduché hry. V závěru učebnice je jeden závěrečný test z našeho školního prostředí.

Doufáme, že si tento překlad české učebnice najde své příznivce i v Čechách a na Moravě.

Aktuální nabídku e-knih najdete přímo na stránkách autora:

<http://www.programujemevpythonu.cz>
<https://www.facebook.com/programujemevpythone>
<http://www.programujemevpythone.sk>

Přejeme vám příjemné chvíle při učení se Pythonu :-)
autor

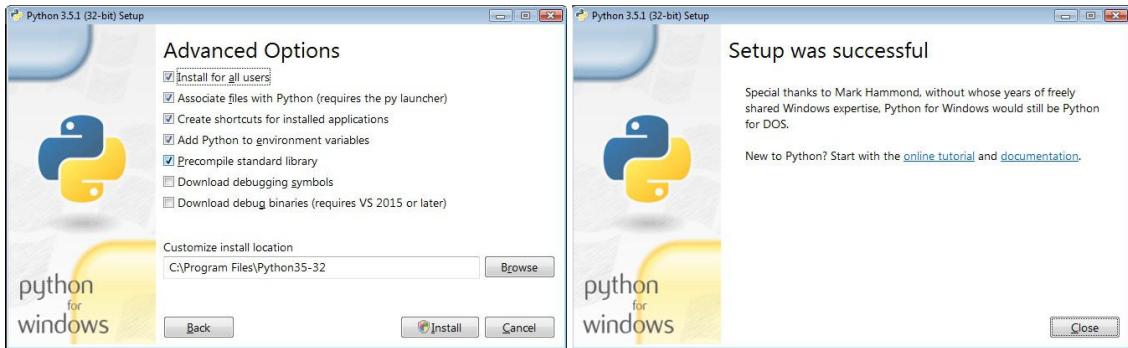
2 Instalace jazyku Python a spuštění prostředí IDLE

Na stránce [www.python.org](https://www.python.org/downloads/) si stáhneme Python 3.5.1 (číslo verze) vhodný pro náš počítač. Máme-li 64-bitovou verzi operačního systému, stáhneme si Python pro 64-bitový operační systém. Důležité je, abychom měli nainstalovanou verzi začínající číslem 3.

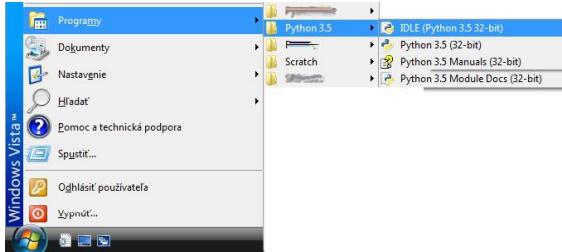
The screenshot shows the Python Software Foundation website. At the top, there's a navigation bar with links for Python, PSF, Docs, PyPI, Jobs, and Community. Below the navigation is a search bar and links for Socialize and Sign In. A large banner features the Python logo and the text "Download the latest version for Windows". It includes two prominent yellow buttons: "Download Python 3.5.1" and "Download Python 2.7.11". Below these buttons, there's a note about the difference between Python 2 and 3. Further down, there are links for other operating systems like Linux/UNIX, Mac OS X, and Other, and a link for Pre-releases. To the right of the banner is an illustration of two boxes descending from the sky on yellow and white parachutes. Below the banner, there's a section for specific releases, listing Python 3.4.4 and Python 3.5.1 with their release dates (2015-12-21 and 2015-12-07 respectively), download links, and release notes. A Firefox browser window is overlaid on the bottom left, showing a download dialog for "python-3.5.1.exe".

The screenshot shows the Python 3.5.1 (32-bit) Setup wizard. The first window, titled "Install Python 3.5.1 (32-bit)", offers to "Select Install Now" or "Customize installation". It includes options for "Install Now" (C:\Users\kucera\AppData\Local\Programs\Python\Python35-32), "Customize installation" (Choose location and features), and checkboxes for "Install launcher for all users (recommended)" and "Add Python 3.5 to PATH". The second window, titled "Optional Features", lists several checkboxes: "Documentation" (checked), "pip" (checked), "tk/tk and IDLE" (checked), "Python test suite" (checked), and "py launcher" (checked). Buttons for "Back", "Next", and "Cancel" are at the bottom of both windows.

V okně Advanced Options potvrdíme možnost Install for all users.



Python 3.5.1 máme nainstalovaný. Spouštěme ho ikonou pro IDLE (Python 3.5 ...).



Po spuštění se otevře okno s názvem Python Shell. V tomto okně se dají přímo psát příkazy a po potvrzení entrem se automaticky provedou. Je to interaktivní prostředí, označujeme ho i názvem příkazový režim nebo jen shell. My budeme nejprve zapisovat program v programovacím režimu. To znamená, že v okně shell vybereme z hlavního menu File a v něm nabídku New File. Otevřelo se nám nové okno, kde zapíšeme program. Program uložíme. Pokud ho chceme spustit, stiskneme F5. Po spuštění se nám do shell-u napíše informace, který program jsme spustili, a také se restartují všechna nastavení Pythonu v příkazovém režimu.

3 Grafické příkazy

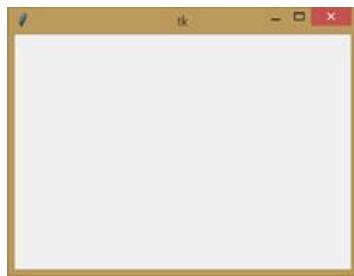
3.1 Vytvoření grafické plochy

Máme spuštěné prostředí IDLE a vytvořen nový soubor – jsme v programovacím režimu. Postup spuštění a vytvoření souboru jsme si ukázali v předchozí kapitole. Pokud chceme pracovat s grafickou plochou, musíme nejprve do programu nainstalovat modul pro práci s grafickou plochou (je to nějaký program, ve kterém již někdo naprogramoval příkazy, které budeme používat). My budeme na práci s grafikou používat modul `tkinter`. Tento modul nainstalujeme na začátku programu zápisem `import tkinter`.

Po nainstalování modulu vytvoříme grafickou plochu (plátno), na které budeme později kreslit `canvas = tkinter.Canvas()`. Příkaz `canvas.pack()` zabezpečí zobrazení nového okna i s vytvořenou grafickou plochou. Celý program zatím vypadá takto:

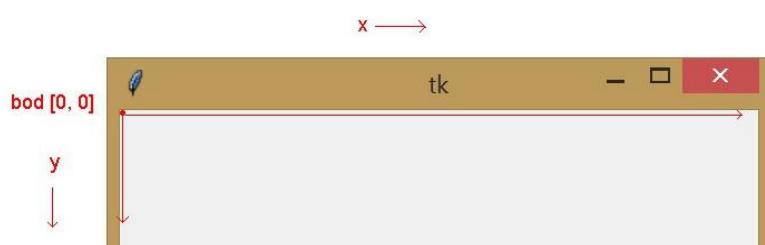
```
import tkinter  
canvas = tkinter.Canvas()  
canvas.pack()
```

Když spustíme tento program (F5), vytvoří se okno s grafickou plochou. V okně zatím nic nevidíme, protože jsme do něj nic nenakreslili. Okno se zavře klasicky .

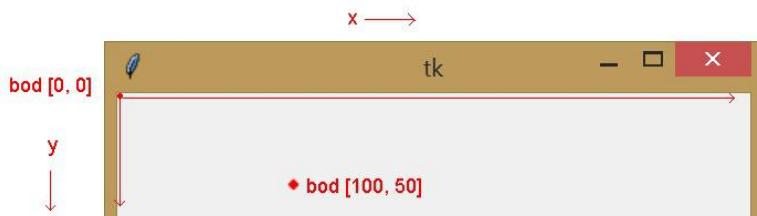


3.2 Souřadnice bodů a kreslení čar

Do grafické plochy můžeme kreslit například čáry, obdélníky, elipsy. Při kreslení potřebujeme příkazem zadávat souřadnice bodů.



Každý bod má dvě souřadnice: x-ovou a y-ovou. X-ová souřadnice určuje vodorovnou pozici a y-ová určuje svislou pozici. Ale pozor, y-ová souřadnice roste směrem dolů (ne jako v matematice, kde roste směrem nahoru). Souřadnice zapisujeme jako dvojici čísel, přičemž x-ová souřadnice je vždy v této dvojici první a druhá je y-ová souřadnice.



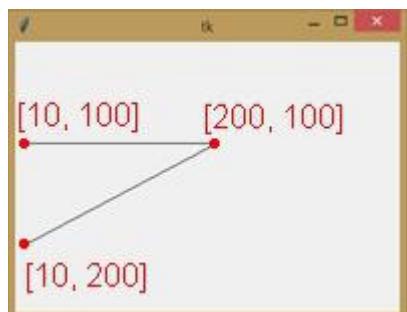
Otázky:

1. Jaké souřadnice bude mít bod, který je o 20 bodů vlevo od bodu se souřadnicemi [100, 50]?
2. Jaké souřadnice bude mít bod, který je o 20 bodů vpravo od bodu se souřadnicemi [100, 50]?
3. Jaké souřadnice bude mít bod, který je o 20 bodů nahoru od bodu se souřadnicemi [100, 50]?
4. Jaké souřadnice bude mít bod, který je o 20 bodů dolů od bodu se souřadnicemi [100, 50]?
5. Co mají společného všechny body, které jsou vlevo nebo vpravo od bodu [100, 50]?

Příkazem `canvas.create_line(10, 100, 200, 100)` nakreslíme čáru, která začíná v bodě se souřadnicemi [10, 100] a končí v bodě [200, 100]. Tímto příkazem můžeme kreslit i čáry, které procházejí přes několik bodů. Jednotlivé souřadnice bodů přidáváme do příkazu. Například `canvas.create_line(10, 100, 200, 100, 10, 200)` k původní čáře přidá další bod se souřadnicí [10, 200]. Po spuštění programu:

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

uvidíme čáru, která začíná v bodě [10, 100], pokračuje do bodu [200, 10] a končí v bodě [10, 200].



Úloha:

- 1** Upravte tento program tak, aby dokreslil tyto dvě čáry do trojúhelníku.

Při kreslení můžeme čáram na nastavit tloušťku i barvu.

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

Nakresli čáru s tloušťkou 5 bodů.

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

Nakresli červenou čáru.

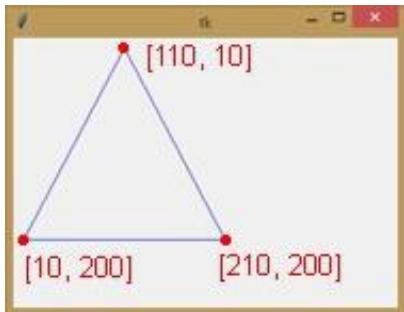
```
canvas.create_line(10, 100, 200, 100, fill='red', width=5)
```

nebo

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

nakresli červenou čáru s tloušťkou 5 bodů. Tloušťku nastavujeme parametrem width a číselnou hodnotou. Zabarvení nastavujeme parametrem fill a v apostrofech zapsaným názvem barvy (můžeme použít různé barvy, například: 'white', 'black', 'red', 'blue', 'yellow', 'green', 'maroon', 'orange', 'gray', 'skyblue', 'violet', 'fuchsia', 'olive').

Znak ' můžeme napsat na CZ klávesnici stisknutím Ctrl + Alt + P. Krok zpět uděláme stisknutím Ctrl + Z.



Tento trojúhelník můžeme nakreslit různými postupy:

```
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')
```

nebo

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

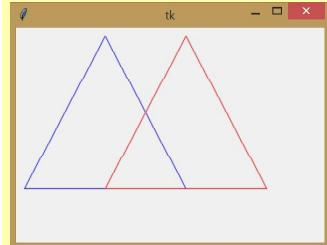
Úlohy:

2

Zamyslete se. Můžeme trojúhelník v předchozím programu nakreslit ještě jinými způsoby?

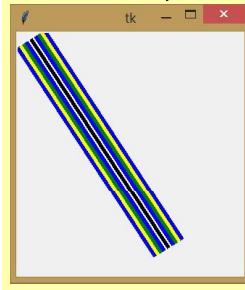
3

Nakreslete stejný trojúhelník červenou barvou posunutý podle ukázky na obrázku.



4

Změnou barvy a tloušťky čáry nakreslete tento obrázek:



5 Pomocí čar nakreslete písmena L, T, H, Z.

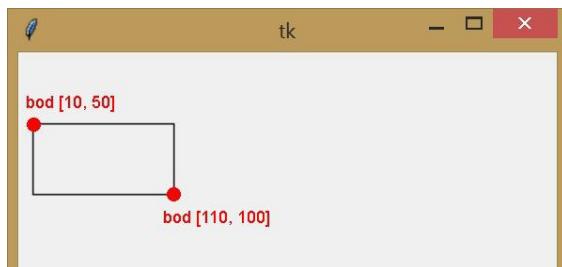
6 Pomocí čar nakreslete obdélník.

3.3 Kreslení obdélníků

Otázky:

6. Obdélník už umíme nakreslit pomocí příkazu na kreslení čar. Pokud kreslíme obdélník pomocí čar, kolik nejméně bodů (souřadnic těchto bodů), musíme v příkazu na kreslení čar použít?
7. Umíme jednoznačně určit obdélník i pomocí menšího počtu bodů? Kolik nejméně jich potřebujeme? Svou odpověď zdůvodněte.

Příkaz `canvas.create_rectangle(10, 50, 110, 100)` nakreslí obdélník, který je zadán pomocí bodů se souřadnicemi [10, 50] a [110, 100].



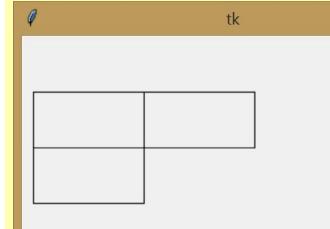
Otázky:

8. Jakou šířku v bodech má obdélník na obrázku?
9. Jakou výšku v bodech má obdélník na obrázku?
10. Můžeme stejný obdélník nakreslit i tímto příkazem `canvas.create_rectangle(110, 100, 10, 50)`?
11. Můžeme stejný obdélník (na stejném místě) nakreslit ještě i pomocí jiných souřadnic?
12. Co se nakreslí po zadání tohoto příkazu `canvas.create_rectangle(10, 50, 110, 50)`?
13. 6. Co se nakreslí po zadání tohoto příkazu `canvas.create_rectangle(10, 50, 10, 50)`?
14. Jakou výšku a šířku mají předchozí dva obdélníky?

Úlohy:

7 Nakreslete vpravo hned vedle obdélníku `canvas.create_rectangle(10, 50, 110, 100)` stejně velký obdélník, který se ho bude dotýkat.

8 Nakreslete hned pod levý obdélník (z předchozí úlohy) stejně velký obdélník.



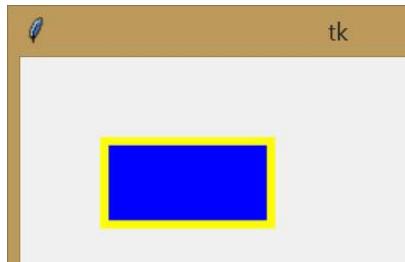
Otázky:

15. Musíme při zadávání příkazů na kreslení obdélníků v úlohách číslo 7 a 8 vymýšlet nebo vypočítávat souřadnice všech bodů?
16. Mají nové body, které zadáváme, něco společného s body, které jsme již zadali?

I při kreslení obdélníku můžeme použít parametr pro nastavení barvy, a to stejným způsobem jako u nastavení barvy při kreslení čar. Parametr `fill` při kreslení obdélníku nastavuje vnitřní barvu, takže také výplň kresleného útvaru. Neovlivňuje ale barvu obrysu obdélníku, jak bychom si mohli myslit. Pro nastavení barvy stran obdélníku můžeme použít parametr `outline`. Čili příkaz:

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

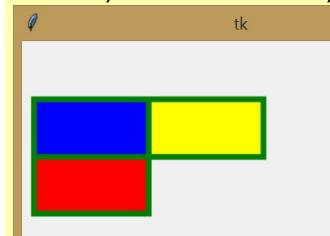
nakreslí obdélník vyplněný modrou barvou se žlutým obrysem, který má tloušťku 5 bodů. Připomeňme si, že tyto parametry mohou být v různém pořadí.



Úlohy:

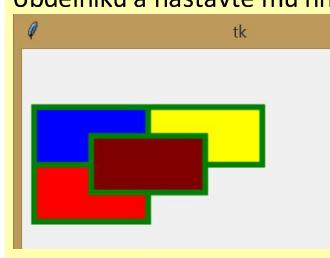
9

Třem obdélníkům z úkolu č. 8 nastavte parametry tak, aby jednotlivé obdélníky měly tyto výplně: modrou, žlutou a červenou, a aby barva jejich obrysů byla zelená a obrys měl tloušťku 5 bodů.



10

Doplňte do tohoto obrázku čtvrtý, stejně velký obdélník, jehož střed je ve společném bodě všech tří obdélníků a nastavte mu hnědou výplň.



Otázky:

17. Co se stane, když parametry na nastavení barvy zamícháme mezi souřadnice bodů
`canvas.create_rectangle(outline='yellow', 50, 50, 150, 100, fill='blue')?`
18. V úloze číslo 10 jsme nakreslili čtyři obdélníky čtyřmi příkazy. Záleží na pořadí těchto příkazů? Co se stane, když hnědý obdélník budeme kreslit jako první? Pořadí kterých příkazů můžeme změnit, aby se nám nakreslil stejný obrázek?
19. Co se stane, když parametr `outline` nastavíme takto: `outline=''`?

20. Co se stane, když parametr `fill` nastavíme takto: `fill=''`?
21. Co se stane, když parametr `fill` a současně parametr `outline` nastavíme takto: `fill='', outline=''`?
22. Kolik obdélníků uvidíme po spuštění těchto příkazů?
 - 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 Nakreslete vlajku: a) Polska, b) Francie, c) Německa, d) Maďarska, e) Švýcarska, f) Izraele.

12 Pouze dvěma příkazy nakreslete vlajku Lotyšska.



13 Nakreslete vlajku Řecka.



14 Dan chtěl nakreslit vlajku Francie a napsal tento program. Najděte v něm chybu.

```
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='')
```

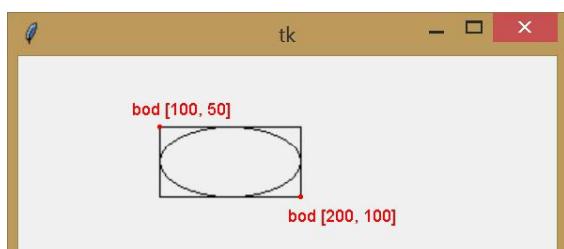
3.4 Kreslení elips

Už víme, že příkaz `canvas.create_rectangle(100, 50, 200, 100)` nakreslí obdélník, jehož levý horní bod má souřadnici [100, 50] a pravý dolní bod má souřadnici [200, 100]. Pokud použijeme stejné souřadnice, ale změníme příkaz `canvas.create_rectangle` na `canvas.create_oval`, nakreslí se elipsa.

Tento program se stejnými souřadnicemi, ale různými příkazy:

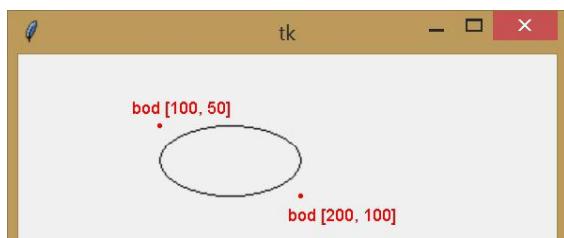
```
import tkinter
canvas = tkinter.Canvas()
canvas.pack()
canvas.create_rectangle(100, 50, 200, 100)
canvas.create_oval(100, 50, 200, 100)
```

nakreslí obdélník i elipsu. Jak vidíme na obrázku, souřadnice bodů, které určují elipsu (ovál), se nenacházejí na elipse, ale mimo ni. Tyto body jsou souřadnicemi obdélníku, ve kterém se nakreslí vepsaná elipsa (čili největší možná elipsa, která se vejde do tohoto pomyslného obdélníku).



Program pouze s příkazem na kreslení elipsy nakreslí jen samotnou elipsu:

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



Při kreslení elipsy nám ze začátku může při vymýšlení souřadnic pomoci, když si nejprve nakreslíme správný obdélník (v něm si představujeme největší elipsu) a až poté, kdy máme správné souřadnice, změníme tento příkaz na elipsu.

Otzázkы:

23. Umíme nakreslit obdélník jedním příkazem. Potřebujeme speciální příkaz na kreslení čtverce? Podle čeho rozpoznáme, že souřadnice v příkazu pro obdélník nakreslí čtverec?
24. Můžeme pomocí příkazu `canvas.create_oval` kreslit kruhy? Potřebujeme speciální příkaz na kreslení kruhů? Podle čeho rozpoznáme, že souřadnice v příkazu pro ovál nakreslí kruh?

V jednom programu můžeme kreslit i několik obrázků vedle sebe. Pokud nám nestačí velikost canvasu nebo chceme změnit barvu jeho pozadí, můžeme tak učinit již na začátku programu při vytváření canvasu. Parametrem `height` nastavíme výšku, parametrem `width` nastavíme šířku a barvu pozadí nastavíme parametrem `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)` nakreslete těsně vedle ní stejně velkou elipsu.

16 Pro obě elipsy nastavte různé barvy výplně, obrysů a také tloušťky čar.

17 Nakreslete vlajku Japonska.

18 Nakreslete sněhuláka.

19 Nakreslete symetrickou tvář s očima, nosem a ústy:

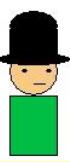


20 Doplňte k obličeji brýle.

21 Nakreslete vlajku červeného půlměsíce:



22 Nakreslete postavičku s kloboukem:

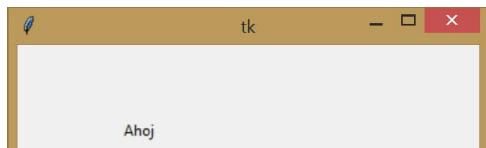


23 Nakreslete barevný terč.

3.5 Psaní textu do grafické plochy

Příkaz `canvas.create_text(100, 70, text='Ahoj')` napiše na souřadnici [100, 70] text Ahoj. První dva parametry tohoto příkazu jsou souřadnice středu vypisování textu. Text se napiše tak, aby ve svislém i ve vodorovném směru byly souřadnice přesně vycentrované uprostřed napsaného textu. Parametrem `text` zadáváme samotný text, který chceme napsat.

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



V příkazu můžeme použít i parametr `font`, kterým určíme typ, velikost a řez písma. Například `font='Arial 70 bold'`. Takto můžeme zadávat pouze jednoslovné názvy fontů. Víceslovny název se zadává komplikovanějším způsobem.

```
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 napiše tyto texty:



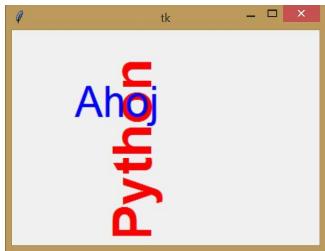
Barvu výplně písma určujeme parametrem `fill` stejně jako v příkazech, které již známe. Písmo můžeme i otáčet parametrem `angle` (nefunguje to na počítačích Mac).

```
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 jsme otočili o 90 stupňů. Nejprve se nakreslí tento nápis a pak se modrou barvou napiše Ahoj. Při nastavení úhlu se text otáčí kolem zadané souřadnice, čili přesného středu textu (fiktivního popsaného obdélníku textu). Otáčení určuje úhel v protisměru hodinových ručiček (jak jsme zvyklí v matematice). Úhel 180 stupňů nám text otočí hlavou dolů. Pokud chceme text otáčet ve směru hodinových ručiček, můžeme použít záporný úhel.



Úlohy:

24 Nakreslete následující dopravní značky:



25 Nakreslete tyto dopravní značky:

