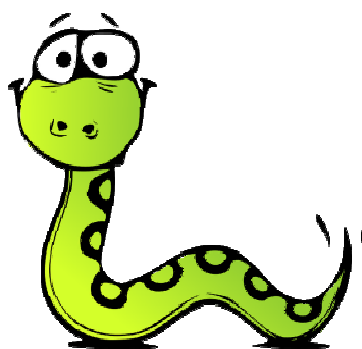




# Programování v jazyce Python pro střední školy

Metodický list pro učitele  
Lekce 6 – Barvy



Andrej Blaho  
Lubomír Salanci  
Václav Šimandl

## Cíle lekce

- Seznámit se s konceptem barev, jejich pojmenováním a použitím v grafických příkazech
- Pochopit důležitost pořadí, v jakém jsou kresleny překrývající se grafické objekty

## Dovednosti

- Zakreslování souřadnic obdélníků na papír při řešení grafických úloh
- Orientace v seznamu názvů barev

## Osvojená syntaktická pravidla

- Způsob zápisu parametru, který nastavuje barvu výplně, v grafickém příkazu `create_rectangle`

## Průběh výuky

První úloha slouží k opakování a žáci se v ní zároveň seznámí s rozměry grafické plochy:

1. Vytvoř program `nejvetsi_obdelnik.py`, který nakreslí co největší obdélník tak, aby byly vidět jeho strany (souřadnice zvol metodou pokus-omyl):



Řešení:

```
import tkinter

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

canvas.create_rectangle(2, 2, 379, 265)
```

Žáci mohou k řešení úlohy přistoupit s různou pečlivostí. Je na našem uvážení, která řešení akceptujeme jako dostatečně kvalitní a u kterých budeme po žácích požadovat jejich úpravu. V každém případě bychom však měli trvat na tom, aby byly všechny strany obdélníku viditelné. V opačném případě by nebylo zřejmé, jak velký obdélník vlastně žák nakreslil.

Někteří žáci mohli od poslední výuky zapomenout nejen příkazy využívané pro kreslení na plátno, ale také způsob fungování souřadnicové soustavy v Pythonu. V tom případě mohou nahlédnout do svých poznámek, dříve vytvořených programů nebo pracovního listu 5. lekce. Jak uvádíme v Úvodu metodických listů, je možné žákům doporučit průběžně si vytvářet

„tahák“ obsahující používané příkazy. Když žák zápis některého příkazu zapomene, snadno si jej vyhledá.

V další úloze žákům prozradíme ideu barvení a necháme žáky, aby program vyzkoušeli:

2. Zatím jsi kreslil jednoduché prázdné obdélníky. Vytvoř nový program `vybarveny.py` a pomocí následujícího kódu nakresli vybarvený obdélník:

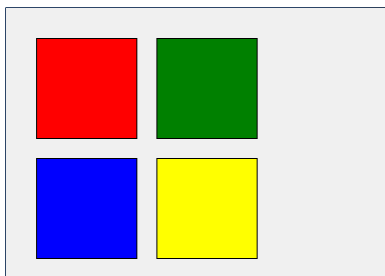
```
import tkinter

canvas = tkinter.Canvas()
canvas.pack()
canvas.create_rectangle(30, 30, 130, 130, fill='red')
```

V jazyce Python zápis `fill='...'` nazýváme pojmenovaným parametrem. Tuto terminologii není nutné žákům prozrazovat.

Následují úlohy na seznámení se s barvami:

3. Přidej do programu `vybarveny.py` další 3 příkazy na kreslení obdélníků, abys dostal následující obrázek:



Další barvy získáš, když místo slova `'red'` napíšeš `'green'`, `'blue'` nebo `'yellow'`.

Řešení:

```
import tkinter

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

canvas.create_rectangle(30, 30, 130, 130, fill='red')
canvas.create_rectangle(150, 30, 250, 130, fill='green')
canvas.create_rectangle(30, 150, 130, 250, fill='blue')
canvas.create_rectangle(150, 150, 250, 250, fill='yellow')
```

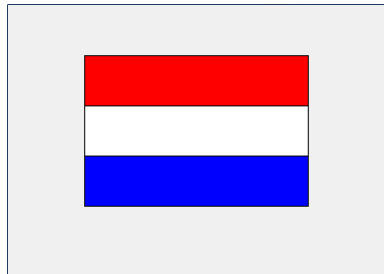
Žáci pravděpodobně nebudou schopni souřadnice určit z paměti, ale budou si vytvářet náskres obrázku na papír a na jeho základě teprve souřadnice určit. Pokud to bude potřeba, je vhodné se žáky individuálně diskutovat o tom, jak spolu souvisí souřadnice jednotlivých čtverců:

- Červený čtverec je již hotový z předchozí úlohy
- Chceme nakreslit zelený čtverec, a tak se žáka zeptáme, co má společného s červeným čtvercem. Žák by měl odpovědět, že má shodné y-ové souřadnice
- Chceme nakreslit modrý čtverec, a tak se žáka zeptáme, co má společného s již nakreslenými čtverci. Žák by měl odpovědět, že má s červeným čtvercem shodné x-ové souřadnice
- Chceme nakreslit žlutý čtverec, a tak se žáka zeptáme, co má společného s již nakreslenými čtverci. Žák by měl odpovědět, že má se zeleným čtvercem shodné x-ové souřadnice a s modrým čtvercem má shodné y-ové souřadnice

Po realizaci každého kroku by měl žák program vyzkoušet pro ověření, zda výsledek odpovídá jeho očekávání.

Následující úlohy slouží k trénování kreslení s barvami:

4. Vytvoř nový program `nizozemi.py`, který nakreslí nizozemskou vlajku:



Jaký název má bílá barva?

Řešení:

```
import tkinter

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

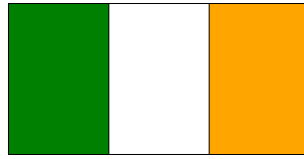
canvas.create_rectangle(50, 50, 300, 100, fill='red')
canvas.create_rectangle(50, 100, 300, 150, fill='white')
canvas.create_rectangle(50, 150, 300, 200, fill='blue')
```

Pokud by některý žák v příkazu pro kreslení prostředního obdélníku vynechal parametr `fill='white'` a napsal pouze:

```
canvas.create_rectangle(50, 100, 300, 150)
```

nakreslí se nevybarvený obdélník, skrze nějž bude vidět světle šedá barva podkladové grafické plochy. Je na našem zvážení, zda takové řešení akceptujeme, nebo zda budeme požadovat vybarvení obdélníku bílou barvou.

5. Vytvoř program `irsko.py`, který nakreslí irskou vlajku s barvou `'orange'`:



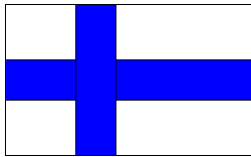
Řešení:

```
import tkinter

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

canvas.create_rectangle(50, 50, 150, 200, fill='green')
canvas.create_rectangle(150, 50, 250, 200, fill='white')
canvas.create_rectangle(250, 50, 350, 200, fill='orange')
```

6. Vytvoř nový program `vlajka.py` a nakresli takovýto obrázek:



Vlajce kterého státu se obrázek podobá?

Obrázek se podobá finské vlajce.

Řešení:

```
import tkinter

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

canvas.create_rectangle(50, 50, 300, 200, fill='white')
canvas.create_rectangle(50, 105, 300, 145, fill='blue')
canvas.create_rectangle(120, 50, 160, 200, fill='blue')
```

Někteří žáci mohou přijít i na řešení, ve kterém se v modré části vlajky nebudou vyskytovat černé čáry. Toto řešení je však náročnější, neboť spočívá v nakreslení modrého podkladového obdélníku a čtyř oddělených bílých obdélníků:

```
import tkinter

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

canvas.create_rectangle(50, 50, 300, 200, fill='blue')
canvas.create_rectangle(50, 50, 120, 105, fill='white')
canvas.create_rectangle(50, 145, 120, 200, fill='white')
canvas.create_rectangle(160, 50, 300, 105, fill='white')
canvas.create_rectangle(160, 145, 300, 200, fill='white')
```

7. Vytvoř nový program `ramecek.py`, ve kterém ze čtyř úzkých obdélníků nakreslí takovýto rámeček:



Řešení:

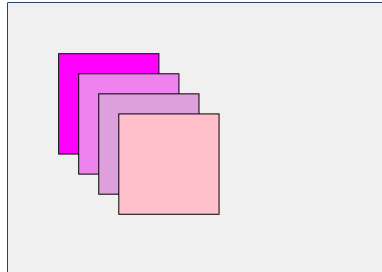
```
import tkinter

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

canvas.create_rectangle(10, 10, 360, 20, fill='red')
canvas.create_rectangle(10, 20, 20, 260, fill='blue')
canvas.create_rectangle(20, 250, 370, 260, fill='red')
canvas.create_rectangle(360, 10, 370, 250, fill='blue')
```

V následující úloze jsou uvedeny názvy některých dalších barev. Když je žáci později budou potřebovat, zde rychle najdou jejich názvy a ukázky. Žáky ze znalosti názvů barev nezkoušíme.

8. Následující obrázek vznikl ze čtyř čtverců. První z nich má souřadnice levého horního vrcholu `[50, 50]`. Napiš program `pres_sebe.py`, který obrázek nakreslí – zvol si libovolné čtyři různé barvy (mohou být jiné než na vzorovém obrázku):



Při kreslení v Pythonu můžeš využít mnoha barev, zde je výběr některých z nich:

Blue	LightBlue	Cyan	SkyBlue	CornFlowerBlue	DeepSkyBlue	DodgerBlue
RoyalBlue	SlateBlue	SteelBlue	MediumBlue	Navy	Red	SandyBrown
Salmon	Coral	Tomato	Orange	DarkOrange	OrangeRed	IndianRed
Chocolate	Tan	Maroon	Sienna	Brown	SaddleBrown	Pink
Plum	Violet	Orchid	Magenta	Purple	DarkMagenta	Green
PaleGreen	YellowGreen	MediumSeaGreen	LawnGreen	LimeGreen	ForestGreen	DarkGreen
Yellow	Khaki	Gold	Gray	LightGray	Black	White

Řešení:

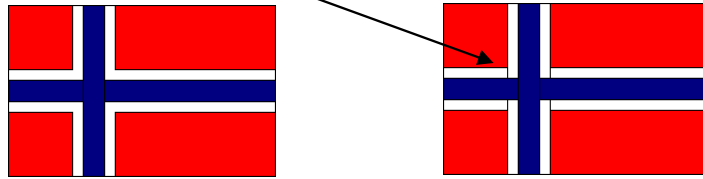
```
import tkinter

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

canvas.create_rectangle(50, 50, 150, 150, fill='magenta')
canvas.create_rectangle(70, 70, 170, 170, fill='violet')
canvas.create_rectangle(90, 90, 190, 190, fill='plum')
canvas.create_rectangle(110, 110, 210, 210, fill='pink')
```

Množina barev je výrazně rozsáhlejší. Všechny názvy lze najít v adresáři, ve kterém je nainstalovaný Python, v souboru `rgb.txt`. Podobné množiny názvů barev lze nalézt taktéž na internetu, například na stránce [HTML Color Names](https://www.w3schools.com/colors/colors_names.asp).

9\* Vytvoř nový program `norsko.py`, ve kterém nakresli norskou vlajku jako na levém vzorovém obrázku. Udělej to tak, aby se (na rozdíl od pravého vzorového obrázku) v bílých částech nekřížily černé čáry:



Řešení s křížícími se čarami (tj. jako na pravém vzorovém obrázku) spočívá v postupném nakreslení červeného podkladového obdélníku, bílého širšího kříže (tvořeného dvěma obdélníky) a poté modrého užšího kříže (tvořeného opět dvěma obdélníky):

```
import tkinter

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

canvas.create_rectangle(50, 50, 300, 210, fill='red')
canvas.create_rectangle(50, 110, 300, 150, fill='white')
canvas.create_rectangle(110, 50, 150, 210, fill='white')
canvas.create_rectangle(50, 120, 300, 140, fill='navy')
canvas.create_rectangle(120, 50, 140, 210, fill='navy')
```

Řešení bez křížících se čar (tj. jako na levém vzorovém obrázku) spočívá v nakreslení bílého podkladového obdélníku, čtyř oddělených červených obdélníků a poté modrého kříže (tvořeného dvěma obdélníky):

```
import tkinter

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

canvas.create_rectangle(50, 50, 300, 210, fill='white')
canvas.create_rectangle(50, 50, 110, 110, fill='red')
canvas.create_rectangle(150, 50, 300, 110, fill='red')
canvas.create_rectangle(50, 150, 110, 210, fill='red')
canvas.create_rectangle(150, 150, 300, 210, fill='red')
canvas.create_rectangle(50, 120, 300, 140, fill='navy')
canvas.create_rectangle(120, 50, 140, 210, fill='navy')
```

Někteří žáci mohou přijít i na řešení, ve kterém se černé čáry nebudou křížit v bílých ani v modrých částech vlajky. Toto řešení spočívá v nakreslení modrého podkladového obdélníku, čtyř oddělených bílých obdélníků a poté čtyř oddělených červených obdélníků:



```
import tkinter

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

canvas.create_rectangle(50, 50, 300, 210, fill='navy')
canvas.create_rectangle(50, 50, 120, 120, fill='white')
canvas.create_rectangle(140, 50, 300, 120, fill='white')
canvas.create_rectangle(50, 140, 120, 210, fill='white')
canvas.create_rectangle(140, 140, 300, 210, fill='white')
canvas.create_rectangle(50, 50, 110, 110, fill='red')
canvas.create_rectangle(150, 50, 300, 110, fill='red')
canvas.create_rectangle(50, 150, 110, 210, fill='red')
canvas.create_rectangle(150, 150, 300, 210, fill='red')
```

10\* Vytvoř nový program 5\_misto\_3.py, ve kterém uprav následující kód tak, aby nakreslil stejný obrázek, ale aby program obsahoval jen 3 příkazy pro kreslení obdélníků:

```
canvas.create_rectangle(90, 90, 150, 150, fill='yellow')
canvas.create_rectangle(150, 90, 210, 150, fill='red')
canvas.create_rectangle(90, 150, 150, 210, fill='green')
canvas.create_rectangle(30, 90, 90, 150, fill='red')
canvas.create_rectangle(90, 30, 150, 90, fill='green')
```

V této úloze předpokládáme, že žáci si nejdříve daný program spustí a výsledný obrázek analyzují. Měli by objevit, že dva oddělené červené čtverce (podobně jako dva zelené čtverce) lze nakreslit jako jeden obdélník, který je na závěr překryt žlutým čtvercem.

Řešení:

```
import tkinter

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

canvas.create_rectangle(30, 90, 210, 150, fill='red')
canvas.create_rectangle(90, 30, 150, 210, fill='green')
canvas.create_rectangle(90, 90, 150, 150, fill='yellow')
```