Tkinter

Tkinter slouží vytváření grafických aplikací. Příkladem v něm vytvořené aplikace je i prostředí IDLE.

Minimální zápis grafické aplikace je:

import tkinter
canvas = tkinter.Canvas()
canvas.pack()
 # tu se bude kreslit do grafické plochy
tkinter.mainloop()

import tkinter + canvas = tkinter.Canvas() - vytvoří plátno grafické aplikace (zde je přidané do proměnné "canvas", což je nejčastěji používaný zápis). Vznikne malá grafická aplikace a plátno, které zatím nejsou vidět, dokud zde není něco umístěno.

canvas.pack() - tímto příkazem se umístí naše plátno do grafické aplikace a je připravené, abychom mohli do něho kreslit.

tkinter.mainloop() - tento příkaz "oživuje" grafickou aplikaci. Nyní reaguje na klikání, posouvání, změnu velikosti, překreslování, ...
Při spuštění z IDLE je možné tento příkaz vynechávat, protože je zde již obsažen.

Po spuštění tohoto okna se objeví grafická aplikace obsahující plátno (šedá plocha uvnitř), které dokud nezadáme žádný grafický příkaz, bude prázdné.

Souřadnicová soustava

V počítačových programech je bod (0,0) v levém horním rohu plátna. X-ová osa jde doprava a Y-ová osa směrem dolů:

Je možné psát i záporné souřadnice, pokud budeme chtít kreslit a vkládat objekty mimo viditelnou část plátna. Základní plátno má rozměry přibližně 380 x 260, které se ale dají měnit.

Grafické příkazy

Všechny grafické příkazy pracují s plátnem, a proto začínají slovem "canvas", za kterým bude následovat tečka a název příkazu: *canvas.create_<jmenoutvaru>(x,y,...,<dalšiparametre>)*

Do grafické plochy se vykreslí uvedený útvar na uvedené souřadnice (x, y) a dle dalších specifik.

canvas.create_text(...) - vypíše zadaný text
canvas.create_rectangle(...) - nakreslí obdélník
canvas.create_oval(...) - nakreslí elipsu
canvas.create_line(...) - nakreslí lomenou čáru
canvas.create_polygon(...) - nakreslí polygon
canvas.create_image(...) - nakreslí png obrázek

Text v grafické ploše

Nejjednodušší grafický příkaz, který do plochy plátna vypíše na určené souřadnice nějaký text.

Ukážeme nejjednodušší grafický příkaz, který do plochy (plátna) zapíše nějaký text. Do naší šablony na vytvořené grafické aplikace přidáme jeden příkaz na „nakresleni“ (vypsaní) nějakého textu:

canvas.create_text(150, 100, text='programujem v Pythone')

Souřadnice vyznačují střed textu:

Kreslení obdélníku

Na kreslení obdélníků slouží grafický příkaz, kde se buď definují dva protilehlé body (x1, y1) a (x2, y2), mezi kterýma se nakreslí obdélník:

canvas.create_rectangle(x1,y1,x2,y2)

Nebo zápis, kde se uvede jen jeden bod a velikost obdélníka (t*ímto zápisem můžeme snadno zakreslit stejně velký obdélník na různých bodech):*

canvas.create_rectangle(x, y, x + šířka, y + výška)

Kreslení elips

kreslení elips je shodné jako kreslení obdélníků, akorát na místo slova "***rectangel***" použijeme slovo "***oval***":

canvas.create_oval(80, 50, 300, 200, fill='white')

Elipsa je tak obdélníkem se zakulacenými rohy.

Pokud budeme chtít kreslit **kružnici**, pak můžeme použít její poloměr (r) a souřadnice středu (x, y).

x,y=150,100
r=80
canvas.create_oval(x-r, y-r, x+r, y+r)

Kreslení úseček a lomených čar

Dalším grafickým příkazem kreslíme lomené čáry, tj. čáry, které se skládají z navazujících úseček.

Tvar příkazu je:

canvas.create_line(x1,y1,x2,y2,x3,y3,...)

Parametrem je posloupnost souřadnic, které tvoří lomenou čáru. Tato posloupnost musí obsahovat alespoň 2 body (alespoň 4 čísla) - tehdy se nakreslí jedna úsečka.

Při přiřazování souřadnic do proměnných, můžeme pro obě hodnoty souřadnic (x, y) použít jednu proměnou.

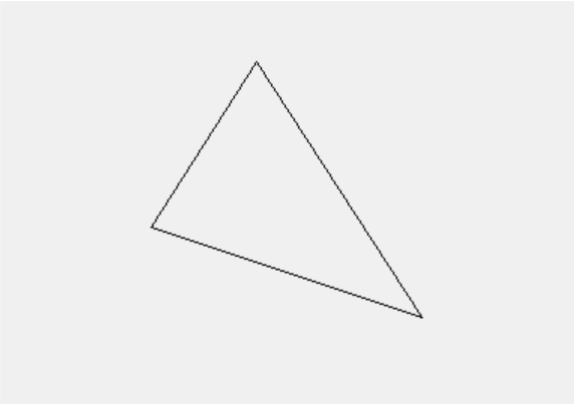
a=100,150

Pokud vytvoříme tři body, které spolu propojíme (a - b - c - a) nakreslíme trojúhelník pomocí tří příček.

a=100,150
b=280,210
c=170,40

canvas.create_line(a, b)
canvas.create_line(b, c)
canvas.create_line(c, a)

Tento zápis můžeme i zkrátit: ***canvas.create_line(a, b, c, a)***



Body na kružnici

Pokud potřebujeme ne celou kružnici, ale jen několik bodů na jejím obvodu. Využijeme goniometrické funkce "***sin***" a "***cos***". Potom každý bod na kružnici můžeme zapsat takovým vzorcem:

*x=x0+r*cos(*radians *(uhel))*
*y=y0+r*sin(*radians *(uhel))*

kde "***uhel***" je zřejmě nějaké číslo od 0 do 360 (nemusí být celé) a "***x0***" a "***y0***" jsou souřadnice středu kružnice a "radians" je zde pro přepočet radiánů s kterými python pracuje.

Při více bodech na kružnici a jejich pospojováním úsečkami se umíme přiblížit k tvaru kružnice.

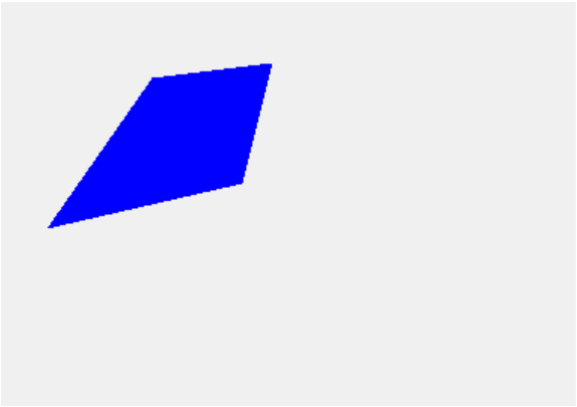
Kreslení polygonů

Polygonem voláme oblast grafické plochy, která je ohraničena zadanou lomenou čarou (alespoň se třemi vrcholy) a vyplní se nějakou barvou.

a=(100,50)
b=(30,150)
c=(160,120)
d=(180,40)
canvas.create_polygon(a, b, c, d, fill='blue')

což se dá zapsat i takto:
canvas.create_polygon(100,50,30,150,160,120,180,40,fill='blue')

nebo takto pro větší přehlednost:
canvas.create_polygon((100,50),(30,150),(160,120),(180,40),fill='blue')



Grafický objekt obrázek

Abychom mohli do plochy vložit nějaký obrázek, musíme nejprve vytvořit **obrázkový objekt** (pomocí ***tkinter.PhotoImage()*** načíst obrázek ze souboru), a až takto poslat jako parametr do příkazu na kreslení obrázků ***canvas.create_image().***

Obrázkový objekt vytvoříme příkazem: ***premenna=tkinter.PhotoImage(file='jmeno souboru')***, v kterém ***‚jmeno souboru‘*** je soubor s obrázkem ve formátu ***.png***, nebo ***.gif***.

Samotná funkce ***canvas.create_image()*** na vykreslení obrázku má tři parametry: první dva jsou souřadnice středu obrázku a další **pojmenovaný parametr** určuje obrázkový objekt.

canvas.create_image(x,y,image=premenna)

Barvy v grafických programech

Barevná plocha se v příkazu značí jako **pojmenovaný parametr** a zapisuje se do závorek příkazu, za čárkou za vyznačením souřadnic do uvozovek ve formátu fill='<označení barvy>'

canvas.create_rectangle(x,y,x+sirka,y+vyska,fill='red')

Tkinter akceptuje nejběžnější jména barev v angličtině odpovídajícím ve velké míře HTML zápisům.

K dohledání zde: HTML Color Names .

Je také možné zapisovat barvy i pomocí RGB modelu, kdy každá barva je namíchaná z kombinace tří barev **red** (červená), **green** (zelená) a **blue** (modrá) a jejich 255 "odstínů" saturace.

Takto definované barvy se ale musí zapsat ve speciálním formátu, který je přesně stejný jako v HTML .

Zapisuje se jako 7-znakový řetězec ve tvaru: ***#rrggbb***

Přičemž "***rr***" označuje číslo (od 0 do 255) pro červenou složku a je zapsáno v šestnáctkové (hexadecimální) soustavě jako dvouciferné číslo, podobně "***gg***" a "**bb**" vyjadřují zelenou a červenou složku, také jako dvouciferná šestnáctková čísla.

V pythonu je možné pro výpočet a zápis barev použít následující vzorec:

r,g,b=255,192,203 # růžová barva
f'#{r:02x}{g:02x}{b:02x}'
'#ffc0cb'

Formát '***{r:02x}***'označuje, že zapisujeme hodnotu proměnné "***r***" na šířku "***2***", "**0**" označuje, že číslo bude zleva doplněno nulami a specifikace "***x***" označuje výpis v šestnáctkové soustavě.

Při generování náhodné barvy se dá použít následující zápis:

r=random.randint(0,255)# anebo r = random.randrange(256)
g=random.randint(0,255)
b=random.randint(0,255)
barva=f'#{r:02x}{g:02x}{b:02x}'

Případně ve své zkrácené podobě, pokud nepotřebujeme zastoupení jednotlivých barev více specifikovat:
barva=f'#{random.randrange(256**3):06x}'

Velikost písma v grafických programech

K změně velikosti písma slouží **pojmenovaný parametr "*font*"** za který se píše znak rovná se a název a velikost fontu:

canvas.create_text(x,y,text=znak, font='arial 35')

Do volání create_textmůžeme přidat i otočení vypisovaného textu o nějaký úhel ve stupních. Například taková změna otočí každé písmeno textu tak, jako bychom se na něj dívali ze středu:

canvas.create_text(x,y,text=znak, font='arial 35',angle=270-uhol)

Velikost tloušťky čáry v grafických programech

K změně velikosti tloušťky čáry slouží **pojmenovaný parametr "*width*"** za který se píše znak rovná se a číslo značící její tloušťku:

canvas.create_line(x,y,x1,y1,fill=barva, width=30)

Barva obrysů v grafických programech

K zobrazení a změně barvy obrysů slouží **pojmenovaný parametr " *outline*"** za který se píše znak rovná se a číslo značící její tloušťku:

canvas.create_polygon(..., outline='red')

Parametry grafické plochy

Při vytváření grafické plochy (pomocí ***tkinter.Canvas()***) můžeme nastavit velikost plochy, ale i barvu pozadí grafické plochy. Můžeme uvést tyto parametry:

- ***bg*** = nastavuje baru pozadí (z anglického „background“)
- ***width*** = nastavuje šířku grafické plochy
- ***height*** = výšku plochy

Například: *canvas=tkinter.Canvas(bg='white',width=400,height=200)*

Vytvoří bílou grafickou plochu, která má šířku 400 a výšku 200.

Shrnutí parametrů v grafických příkazech

Texty:

canvas.create_text(x,y,...) # souřadnice jediného bodu

- text = vypisovaný text
- font = písmo a velikost
    - buď 'jméno a velikost' pro jednoslovná jména fontu
    - anebo ('jméno', velikost)
- fill = barva textu
- angle = úhel otočení v stupních
- anchor = ukotvení (pozice (x, y))
    - jedno z 'center', 'nw', 'n', 'ne', 'e', 'se', 's', 'sw', 'w'

Obdélník:

canvas.create_rectangle(x,y,x,y,...) # souřadnice dvou bodů

- width = tloušťka obrysu
    - hodnota 0 označuje bez obrysu
- outline = barva obrysu
    - hodnota '' označuje bez obrysu
- fill = barva výplně
    - hodnota '' označuje bez výplně

Elipsy:

canvas.create_oval(x,y,x,y,...) # souřadnice dvou bodů

- width = tloušťka obrysu
    - hodnota 0 označuje bez obrysu
- outline = barva obrysu
    - hodnota '' označuje bez obrysu
- fill = barva výplně
    - hodnota '' označuje bez výplně

Lomené čáry:

canvas.create_line(x,y,x,y,x,y,x,y,...) # souřadnice aspoň dvou bodů

- width = tloušťka čáry
- fill = barva čáry
- arrow = šipka na konci čáry
    - jedno z 'first', 'last', 'both'

Polygony:

canvas.create_polygon(x,y,x,y,x,y,x,y,...) # souřadnice aspoň dvou bodů

- width = tloušťka obrysu
    - hodnota 0 označuje bez obrysu
- outline = barva obrysu

    - hodnota '' označuje bez obrysu
- fill = barva výplně
    - hodnota '' označuje bez výplně

Změny nakreslených útvarů:

Každý grafický příkaz, např. *canvas.create_line()*, je funkcí která vrací celé číslo je **identifikátor nakresleného obrázku**. Všechny vložené útvary si Python pamatuje a ukládá v pořadí, jak jsme je zadali a dokáže je tak dodatečně posouvat a měnit.

Kromě identifikačního čísla můžeme grafickým objektům přidělit i štítek – pojmenovaný parametr ***tag=***, a ten pak používat namísto čísla identifikátoru. Výhodou je, že pod jedením takovýmto štítkem můžeme mít více objektů a tak můžeme najednou měnit modifikačními příkazy více objektů.

Zrušení nakresleného útvaru:

canvas.delete(označení)

kde parametr *(označení)* je jedno z

- číselný identifikátor
- přidělený štítek (tag)
- řetězec 'all' označuje všechny útvaru v ploše

Posováni útvarů:

canvas.move(označení,dx,dy)

kde

- označení je buď identifikátor, nebo štítek grafických útvarů (bude fungovat i řetězec 'all')
- dx a dy označují číselné hodnoty změny souřadnic útvarů, tj. posun v směru osy **x** a v směru osy **y**

Změna parametrů útvaru:

canvas.itemconfig(označení, parametry)

kde

- označení je buď identifikátor, nebo štítek grafických útvarů
- parametry jsou **pojmenované parametry** ve stejném formátu, jaký byl při jejich vytváření

Změna souřadnic

canvas.coords(označení, postupnost)

kde

- označení je buď identifikátor, nebo štítek grafických útvarů
- postupnost je libovolná posloupnost souřadnic, která je vhodná pro daný útvar - táto postupnost musí obsahovat sudý počet čísel (celých nebo desetinných)