

Python講座

2017/5/25(木)

canvasとは

- canvasとは図形や文字を描くことができるスペースのこと
- turtleもcanvasを用いて描画している
⇒turtle.Pen()の呼び出しでcanvasを起動している
- 本来canvasは独自に起動させなければならない

描画ツールtkinter

- turtleと同じく canvas上に描画するためのツール
- 描画が一瞬
⇒ 何回も描いて消してを繰り返すことでアニメーションが可能
- turtleよりも幾何図形の描画が楽

まず canvas の起動

書き方:

```
from tkinter import*  
canvas = Canvas(Tk(), width = 500, height = 400, bg = "white")  
canvas.pack()
```

留意点

- 今回のimportはtkinterを省略形(Tk())で使えるようにするため
- Canvas()でcanvasの指定
⇒widthで横幅、heightで縦幅、bgでバックグラウンドカラーの指定
- pack()でcanvasの起動

線の描画

書き方:

x0 = 100

y0 = 300

x1 = 400

y1 = 200

x2 = 250

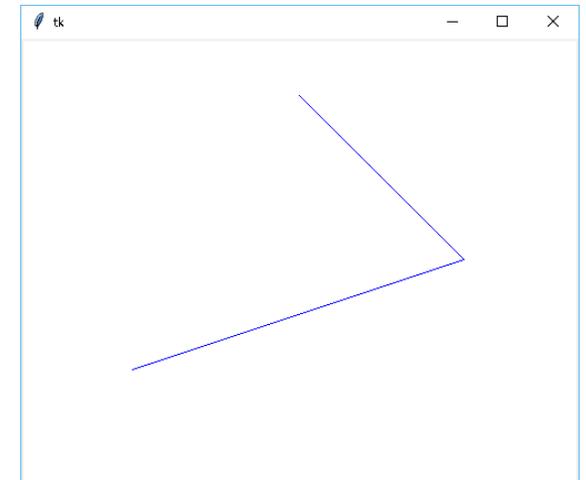
y2 = 50

```
canvas.create_line(x0,y0,x1,y1,x2,y2,fill = "blue")
```

留意点

- create_line()で描画
- 左から順に(x0,y0)と(x1,y1)を結ぶ線、(x1,y1)と(x2,y2)を結ぶ線を描画
⇒x,yはいくらでも続く
⇒xが右向きで正、yが下向きで正をとることに注意
- fillで色の指定が可能(省略可、黒色になる)

結果:



四角形の描画

書き方:

$x_0 = 100$

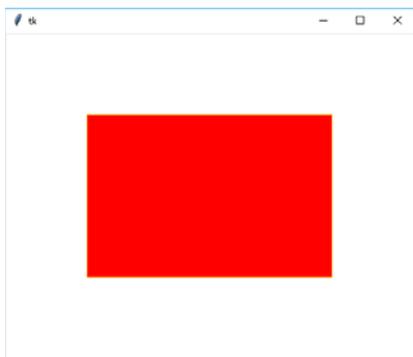
$y_0 = 300$

$x_1 = 400$

$y_1 = 100$

```
canvas.create_rectangle(x0,y0,x1,y1,fill = "red",outline = "orange")
```

結果:



留意点

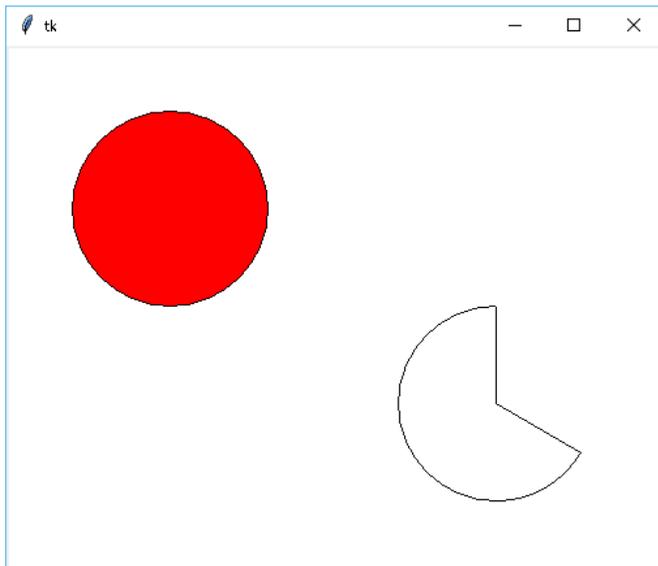
- (x_0, y_0) と (x_1, y_1) で囲まれる四角形を描画
⇒イメージとしては (x_0, y_0) , (x_1, y_0) , (x_1, y_1) , (x_0, y_1) を頂点に持つ四角形
- fillで中身、outlineで枠線の色を指定可能

円、弧の描画

書き方:

```
canvas.create_oval(50,50,200,200,fill = "red")  
canvas.create_arc(450,350,300,200,start = 90,extent = 240)
```

結果:



留意点

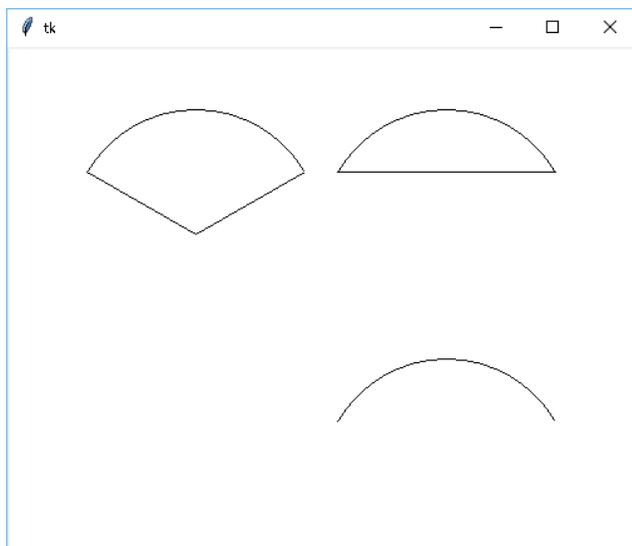
- ovalは(50,50)と(200,200)で囲まれる四角形の内接円を描画
- arcは(450,350)と(300,200)で囲まれる四角形の内接円上の弧を描画
⇒x軸並行を0度とし、
startの角度から始めextentの角度分だけ描画
- fillで中身、outlineで枠線の色を指定可能

さまざまな弧

書き方:

```
canvas1.create_arc(50,50,250,250,start = 30,extent = 120,style = PIESLICE)  
canvas1.create_arc(250,50,450,250,start = 30,extent = 120,style = CHORD)  
canvas1.create_arc(250,250,450,450,start = 30,extent = 120,style = ARC)
```

結果:



留意点

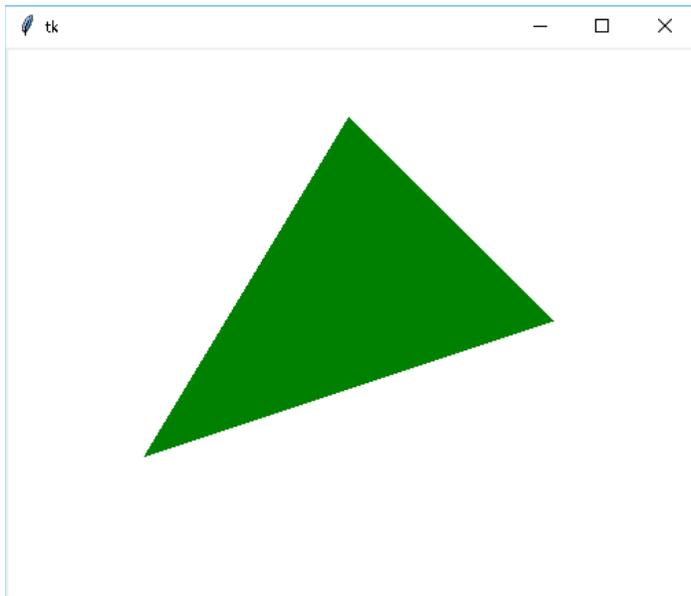
- styleによって弧の種類を指定
- PIESLICEで扇形
- CHORDで弓型
- ARCで弧の部分のみ

多角形の描画

書き方:

```
canvas.create_polygon(100,300,400,200,250,50,fill = "green")
```

結果:



留意点

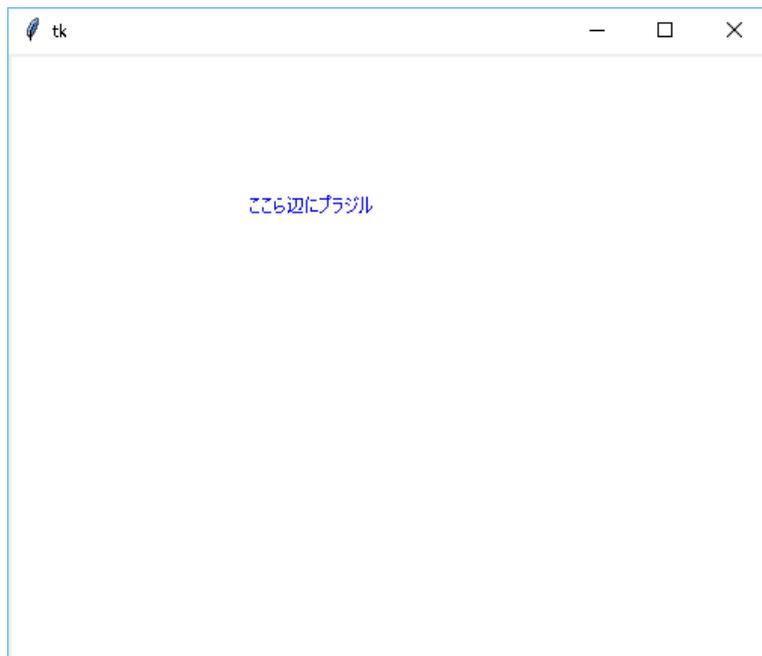
- 書き方はほぼlineと一緒に
- 最後の二点と最初の二点(ここでは(250,50)と(100,300))をつなげてくれる

文字の描画

書き方:

```
canvas.create_text(200,100,text = “こちら辺にブラジル”,fill = “blue”)
```

結果:



留意点

- (200,100)は文字の中央の座標
- textで描画したい文字列を入れる

来期のアニメを作ろう

書き方:

```
import time
```

```
x = 0
```

```
y = 0
```

```
While True:
```

```
    canvas.reset()
```

```
    canvas.create_line(x,y,x,y-1)
```

```
    x += 1
```

```
    y += 2
```

```
    time.sleep(0.05)
```

留意点

- ループに入れることでアニメっぽく
- reset()で直前に描画したものを消去
- sleepは処理を一定時間(今回は0.05秒)止める(使わなくてもよいが調節必須)

演習問題

- テーマ：複数の円をバラバラに動かす
 1. 一つの円を描画し、移動させる
 2. 画面外に行ったとき、その反対側から出てくるようにする
 3. 数字を入力しその個数分、円を描画（初期座標は同じで可）
 4. それぞれがランダムに移動しまくる（互いの接触は不問）