Python講座

2016/5/26(木)

今回の内容

- canvas
- tkinter
- 準備
- ボタン
- 座標
- 配色
- 描画する関数
- 線の描画
- 四角形の描画
- 円の描画①,②,③
- 多角形の描画
- テキストの描画
- 図形の移動
- アニメーション
- 演習問題

canvas

- 図形や文字を描くことができるスペースのこと
- turtleでもcanvasは使用している
- t = turtle.Pen()
  turtleモジュールのPen()という関数を実行
  →canvasの作成
- t.forward(x)
  作られたcanvasにおいて、turtleをx進ませる

tkinter

- 同じくcanvasを作り、描画できるもの
- turtleとの違い

1. 描画するときに待ち時間がない
2. アニメーションのような表現ができる
3. 図形を書く際にそれぞれに対応した関数を使用することで面倒な手間を省ける

準備

- canvasを開くための必須文
- from tkinter import*
- tk=Tk()
- canvas
  =Canvas(tk,width=値1,height=値2,bg=“”)
- canvas.pack()
- 値1:横幅、値2:縦幅、bg:背景の色

ボタン

- ボタンの設置が可能
- 変数名=Button(tk,text=“ ",command =)
- 変数名.pack()
- text:ボタン本体に書きこまれるテキスト
- command:押されたときに実行する関数を指定

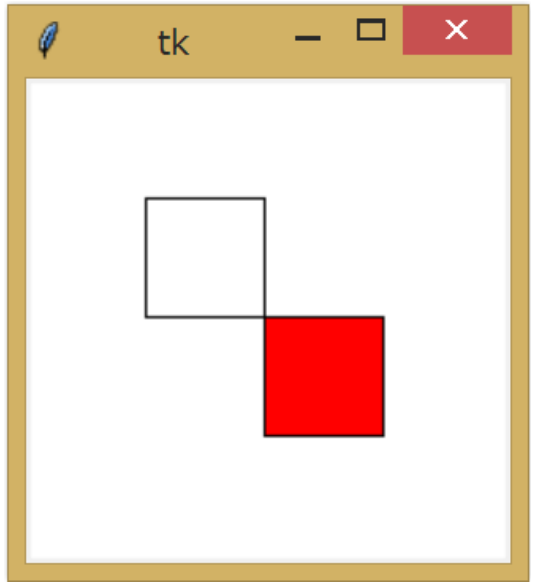
座標

- turtleは中心が(0, 0)
- tkinterの場合

  左上=(0, 0)

  右下=(width, height)

  中心=(width / 2, height / 2)
- y座標が下が＋、上が－であることに注意



配色

- fill=“ ”,outline=“ ”のような形で色の指定が可能
  “red”, ”green”, “blue”..
- 文字ではなくRGBで指定も可能
- ただし16進数で指定する必要がある
  “#ff0000” = R:255, G:0, B:0
- 必ず書く必要はない、その場合は黒もしくは透明

描画する関数

- canvas.をすべての先頭に付ける
- create_line(x0,y0,x1,y1,fill=“”)
- create_rectangle(x0,y0,x1,y1,fill="",outline="")
- create_arc(x0,y0,x1,y1,start=,extent=,style=,fill="")
- create_polygon(x0,y0,x1,y1..,fill="",outline="")
- create_text(x0,y0,text=“”,fill=“”)

参考：http://www.not-enough.org/abe/manual/program-aa08/pythontk2.html

線の描画

- canvas.create_line(x0,y0,x1,y1..,fill=“”)
- 座標(x0,y0)と(x1,y1)を結んだ直線を描画
- 続けて(x2,y2)(x3,y3)..を書くと、折れ線グラフのような線が描画できる
- fillで直線の色の指定、省略した場合は黒
- 省略した場合の例

  canvas.create_line(x0,y0,x1,y1)

四角形の描画

- canvas.create_rectangle(x0,y0,x1,y1,fill=“ ”, outline=“ ”)
- 左上の座標(x0,y0)と右下の座標(x1,y1)を指定し、そこに囲まれた四角形を描画
- イメージとして、四角形の頂点(座標)は(x0,y0),(x0,y1),(x1,y0),(x1,y1)
- fill:塗りつぶし、初期値は透明
- outline:辺の色の指定、初期値は黒

円の描画①

- canvas.create_oval(x0,y0,x1,y1,fill=“”..)
- 完全な円を描画する
- 座標は同じく、左上の座標(x0,y0)と右下の座標(x1,y1)に囲まれた四角形の内接円

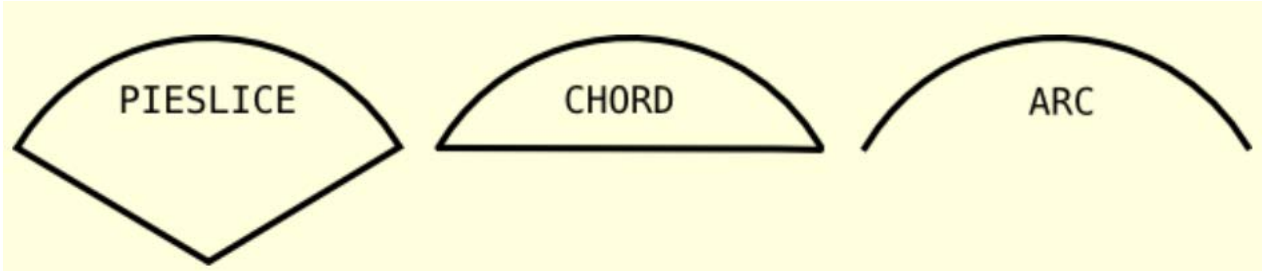
- fill,outline

円の描画②

- canvas.create_arc(x0,y0,x1,y1,start=,..)
- 左上の座標(x0,y0)と右下の座標(x1,y1)を指定し、そこに囲まれた四角形の内接円を描画
- start:始点(書き始め)の角度を決められる
- extent:始点から0~359までの角度を指定、初期値は90
- fill,outline

円の描画③

- styleについて



- 塗りつぶしの範囲などの指定に利用
- 初期値はstyle=PIESLISE

多角形の描画

- canvas.create_polygon(x0,y0,..,fill="",outline="")
- 任意の座標(x0,y0)(x1,y1)(x2,y2)…を順に直線で結んだ多角形を描画
- fill,outline
- 三角形を描画する際に使用

テキストの描画

- canvas.create_text(x0,y0,text="",fill="")
- (x0,y0)はテキストの中心座標
- text:表示するテキスト
- fill

- fontの指定なども可能、方法は知りません

図形の移動

- 描画された図形を移動させる
- canvas.move(ID, dx, dy)
- tk.update()

- ID:描画された順番に1,2..と割り振られる
- dx,dy:軸ごとの移動させたい値

アニメーション

- 図形の移動をアニメーションのように描画させる
- import time
- time.sleep(t)
- t秒間、実行処理を止めることができる
- 使用例:

```
canvas.create_rectangle(50,50,100,100)
for x in range (100):
    canvas.move(1,1,0)
    tk.update()
    time.sleep(0.05)
```

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)