



Pythonの道

GUIの基礎を学ぶ①

もくじ

- ・ GUIの基礎①
- ・ おみくじゲーム (GUI版) を作る



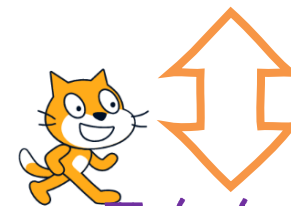
GUIの基礎①

本格的なソフトウェア開発にはグラフィカル・ユーザ・インターフェース（GUI）の知識が欠かせない。GUIやグラフィックを用いたプログラムを作ろう。

GUI（グラフィカル・ユーザ・インターフェース）の例

Scratchのインターフェース。左側のメニューは「操作するためのアイコン」として、右側のメニューは「メニューバー」として注釈されている。

操作方法が直感的にわかるインターフェースをGUIという



CUI（キャラクター・ベースド・ユーザ・インタフェース）

```
コマンドプロンプト
Microsoft Windows [Version 10.0.18363.900]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\sator>ipconfig

windows IP 構成

Wireless LAN adapter ローカル エリア接続* 4:

メディアの状態 . . . . . メディアは接続されていません
接続固有の DNS サフィックス . . . . .

Wireless LAN adapter ローカル エリア接続* 5:
```

tkinterモジュールを使う

PythonでGUIを扱うにはtkinter(ティーキンター)モジュールを使用する。まずは画面にウィンドウを表示してみよう。

```
import tkinter . . . . tkinterモジュールの呼出し  
ウィンドウ = tkinter.Tk() . . . . 変数「ウィンドウ」にtkinter.Tk()  
関数を代入する  
ウィンドウ.mainloop() . . . . mainloop()関数を実行してウィンド  
ウを表示する。
```

※変数は通常、アルファベットで定義する。理解が追いつくまで、かな文字で変数を定義してもOK。ただし、他のプログラミング言語ではアルファベットでしか変数を定義できないことは知っておこう。

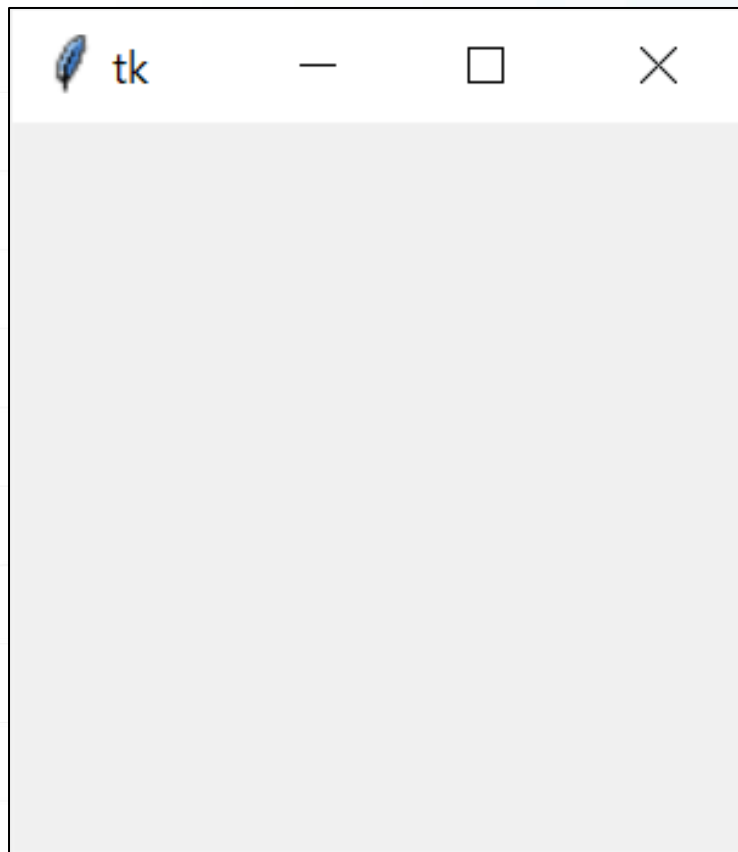
単語帳 ～英語も一緒に覚えてしまおう！～

import:取り込む

※tkinter、mainloopは英語ではなく、メソッドの名前

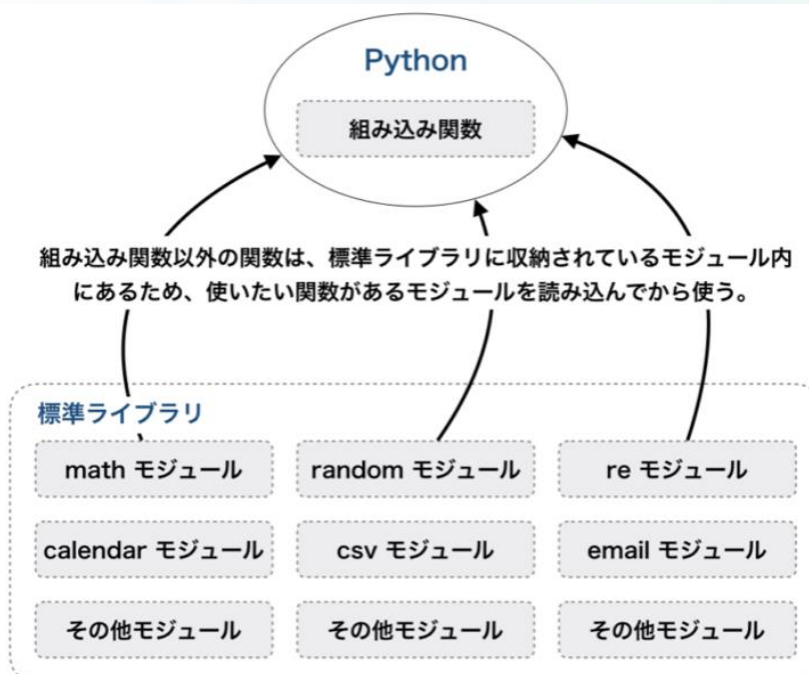
tkinterモジュールを使う

「Run Module」またはF5キーを押してプログラムを実行する。



tkinterモジュールを理解する

tkinterモジュールとはグラフィカルな画面で操作するための部品を集めたもの。まずは、モジュールについて理解しよう。



モジュールとは「部品」のことを意味する。元々は、家を建てたり、家具を作ったりする時の、一つ一つの部分（パーツ）のこと。建築の際は、あらかじめ書いている図面の通りに、パーツを作っていく、最後に、それらのパーツを組み合わせて完成させる。プログラミングもこれと同じで、パーツ毎にファイルを分けて、一つずつ完成させていく。1つのファイルにすべてのプログラムを作ることもできるけど、バグが出たときにどこのプログラムが間違っているかを探すのが大変になる。そこでパーツごとにプログラムを作るというのが今のシステム開発における主流の考え方になっている。

tkinter.Tk()関数を理解する

tkinter.Tk()関数はウィンドウを作るための関数。書き方にルールがあることを覚えておこう。

- ・ import tkinterを一回だけ記述する
- ・ tkinter.Tk() を変数に代入する
- ・ 「変数.メソッド名」の形でメソッドの関数が実行できる
- ・ 変数.mainloop()は最後に記述する

✓ tkinter.Tk()関数の書き方

```
import tkinter
```

```
変数 = tkinter.Tk()
```

```
変数.geometry("横幅x高さ")
```

```
変数.title("タイトル名")
```

```
変数.mainloop()
```

メソッド名	役割
mainloop()	ウィンドウを表示する
geometry()	ウィンドウを指定したサイズにする
title()	ウィンドウにタイトルを付ける

ウィンドウを作る

タイトルとサイズを指定して、ウィンドウを作ってみよう。

```
import tkinter          . . . . tkinterモジュールの呼出し
ウィンドウ = tkinter.Tk()
                        . . . . 変数「ウィンドウ」にtkinter.Tk()関数を代入する
ウィンドウ.title(" 初めてのウィンドウ" )
                        . . . . ウィンドウのタイトルに「初めてのウィンドウ」と表示させる
ウィンドウ.geometry(" 800x600" ) . . . . ウィンドウのサイズを指定
ウィンドウ.mainloop() . . . . ウィンドウを表示する関数
```

単語帳 ～英語も一緒に覚えてしまおう！～

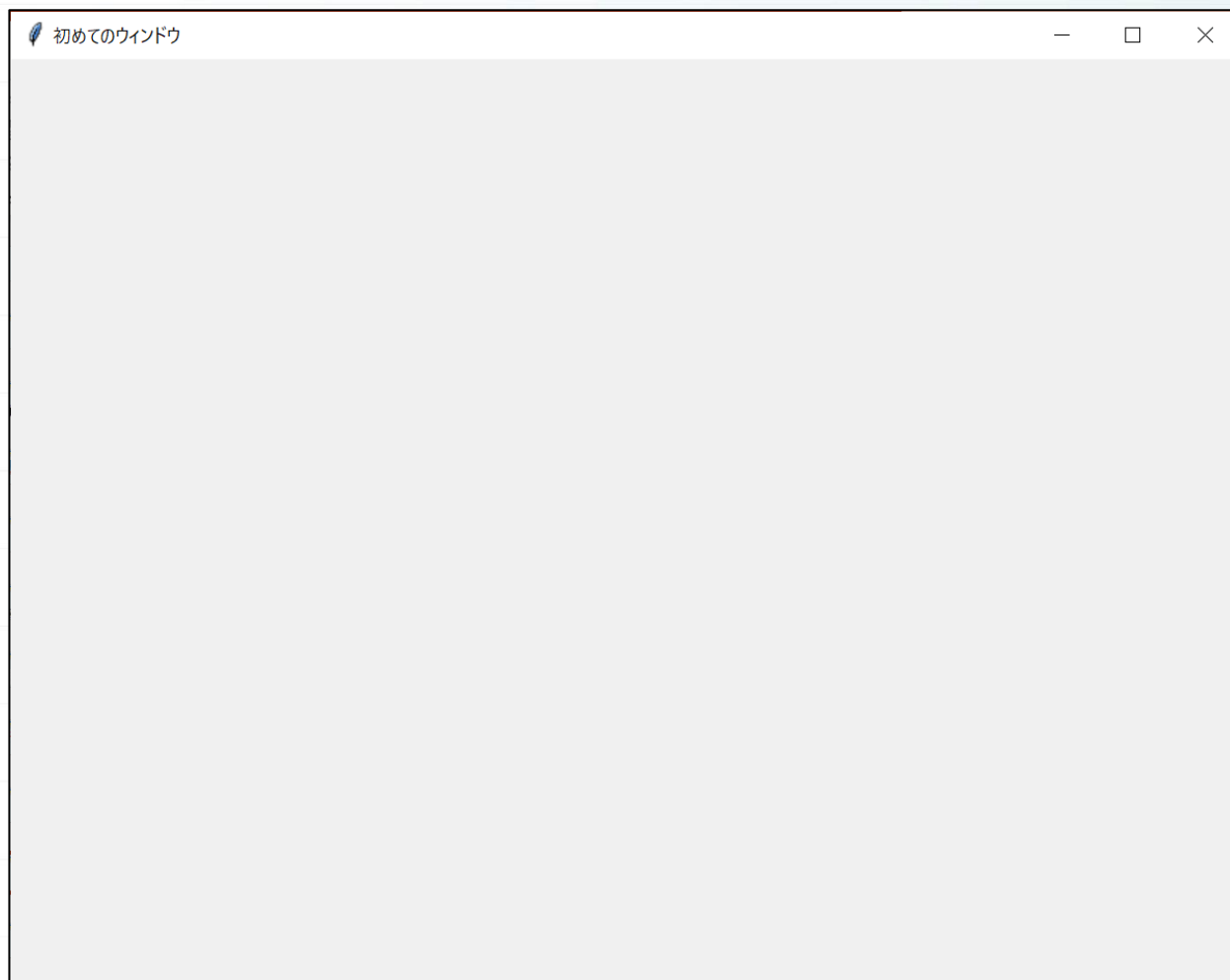
import:取り込む、title:題名、geometry:幾何学

ウィンドウを作る

「Run Module」またはF5キーを押してプログラムを実行する。



800ドット



600ドット

ラベルを作る

ウィンドウにラベルを配置しよう。最初は文字列を表示するラベルを配置してみよう。

```
import tkinter . . . . tkinterモジュールの呼出し
ウィンドウ = tkinter.Tk()
. . . . 変数「ウィンドウ」にtkinter.Tk()関数を代入する
ウィンドウ.title(" 初めてのラベル" )
. . . . ウィンドウのタイトルに「初めてのウィンドウ」と表示させる
ウィンドウ.geometry(" 800x600" ) . . . . ウィンドウのサイズを指定
ラベル = tkinter.Label(ウィンドウ,text=" ラベルの文字列" ,
font = (" system" ,24))
. . . . 変数「ラベル」にフォントサイズ24のテキストを代入する
ラベル.place(x=200,y=100)
. . . . x座標=200,y座標=100の位置にラベルを配置する
ウィンドウ.mainloop() . . . . ウィンドウを表示する関数
```

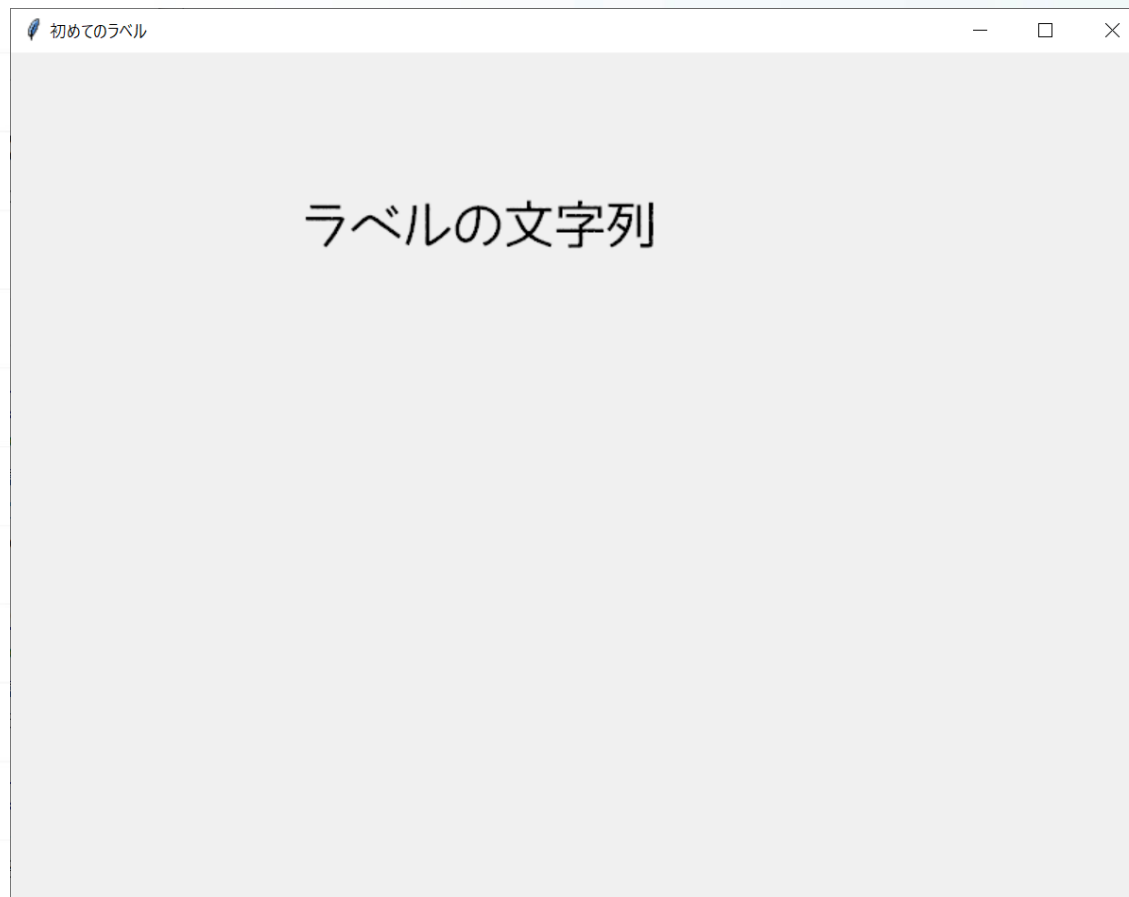
ラベルを作る

単語帳 ～英語も一緒に覚えてしまおう！～

import:取り込む、title:題名、geometry:幾何学、label:張り紙

font:書体のデザイン、place:場所、text:テキスト

「Run Module」またはF5キーを押してプログラムを実行する。



Label()命令、place()命令を理解する

画面に表示するラベルはLabel()命令で作って、place()命令で配置する。

・ Label()命令の書き方

ラベルの変数名 = tkinter.Label(ラベルの中身)
ラベルの中身⇒(ウィンドウ名,text=" ラベルの文字列"
,font(" フォント名" ,フォントサイズ))

【例】

ラベル = tkinter.Label(ウィンドウ,text=" ラベルの文字列" ,font = (" system" ,24))

・ Place()命令の書き方⇒Label()命令で作成したラベルを設定する
ラベルの変数名.place(x=X座標,y=Y座標)

ボタンを作る

次はウィンドウにボタンを配置しよう。最初は「ボタンの文字列」と表示するボタンを作ってみよう。

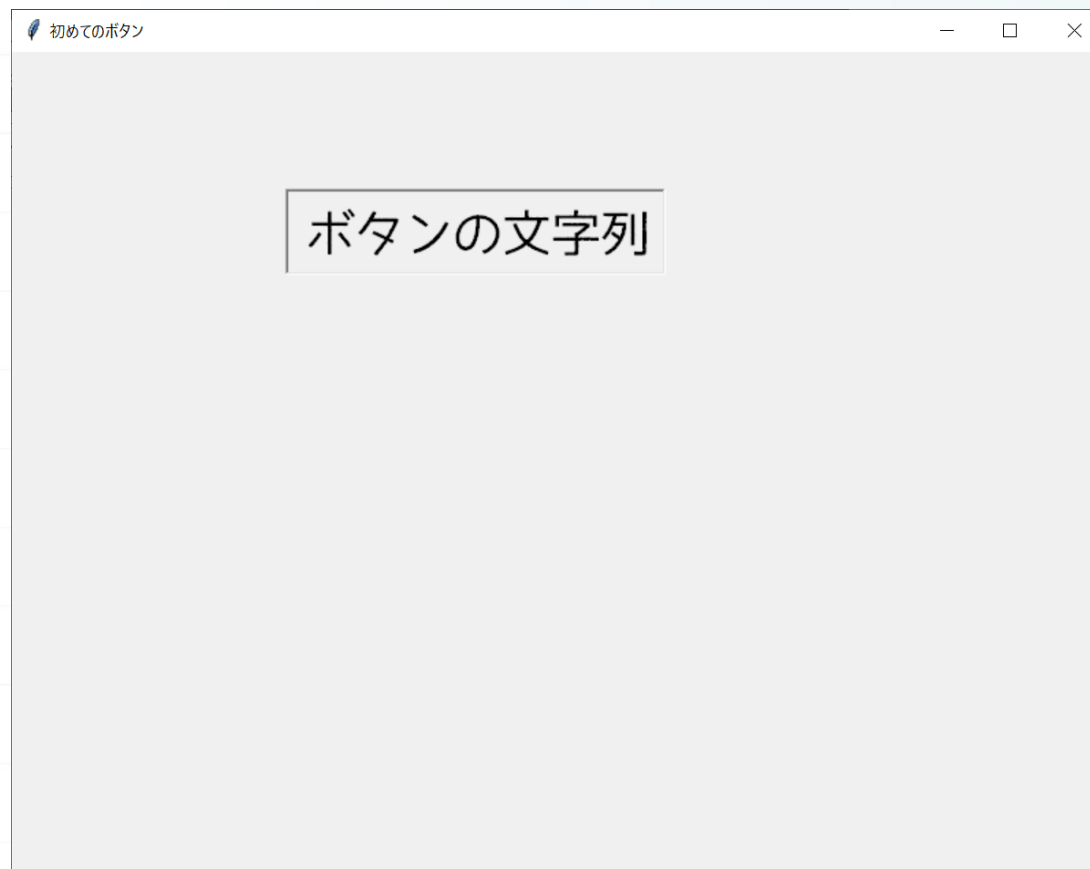
```
import tkinter . . . . tkinterモジュールの呼出し
ウィンドウ = tkinter.Tk()
. . . . 変数「ウィンドウ」にtkinter.Tk()関数を代入する
ウィンドウ.title(" 初めてのボタン" )
. . . . ウィンドウのタイトルに「初めてのボタン」と表示させる
ウィンドウ.geometry(" 800x600" ) . . . . ウィンドウのサイズを指定
ボタン = tkinter.Button(ウィンドウ,text=" ボタンの文字列" ,
font = (" system" ,24))
. . . . 変数「ボタン」にフォントサイズ24のテキストを代入する
ボタン.place(x=200,y=100)
. . . . x座標=200,y座標=100の位置にボタンを配置する
ウィンドウ.mainloop() . . . . ウィンドウを表示する関数
```

ボタンを作る

単語帳 ～英語も一緒に覚えてしまおう！～

import:取り込む、title:題名、geometry:幾何学、button:ボタン
font:書体のデザイン、place:場所、text:テキスト

「Run Module」またはF5キーを押してプログラムを実行する。



Button()命令、place()命令を理解する

画面に表示するボタンはButton()命令で作って、place()命令で配置する。

・ Button()命令の書き方

ボタンの変数名 = tkinter.Button(ボタンの中身)

ボタンの中身⇒(ウィンドウ名 , text=" ボタンの文字列"
,font(" フォント名" ,フォントサイズ))

【例】

```
ボタン = tkinter.Button(ウィンドウ,text=" ボタンの文字列" ,font = (" system" ,24))
```

・ Place()命令の書き方⇒Button()命令で作成したボタンを設定する
ボタンの変数名.place(x=X座標,y=Y座標)

※Button () 命令はLabel () 命令と基本的に同じ書式をつかう。
違いはButtonと記載するかLabelと記載するかどうかだけ！

ボタンを動かしてみよう

ボタンを設置した後はボタンをクリックした時の動作を加えてみよう。ボタンをクリックした時の処理を関数で定義する。

```
import tkinter . . . . tkinterモジュールの呼出し
ウィンドウ = tkinter.Tk()
. . . . 変数「ウィンドウ」にtkinter.Tk()関数を代入する
def クリックボタン(): . . . . クリックボタン関数を定義する
    ボタン[" text" ] = " クリックしました。 "
    . . . . ボタン「text=" クリックしてください" 」で定義されている文字
        に" クリックしました" を代入する
ウィンドウ.title(" 初めてのボタン" )
. . . . ウィンドウのタイトルに「初めてのボタン」と表示させる
ウィンドウ.geometry(" 800x600" ) . . . . ウィンドウのサイズを指定
```

次のページに続く

ボタンを動かしてみよう



```
ボタン = tkinter.Button(ウィンドウ, text=" クリックしてく  
ださい", font = (" system", 24), command = クリックボタ  
ン)
```

・・・変数「ボタン」にフォントサイズ24のテキスト（クリックしてください）を代入する さらにボタンをクリックすると関数を実行させる

```
ボタン.place(x=200,y=100)
```

・・・x座標=200,y座標=100の位置にボタンを配置する

```
ウィンドウ.mainloop() ・・・ウィンドウを表示する関数
```

単語帳 ～英語も一緒に覚えてしまおう！～

import:取り込む、definition:定義（※defと略されている）、

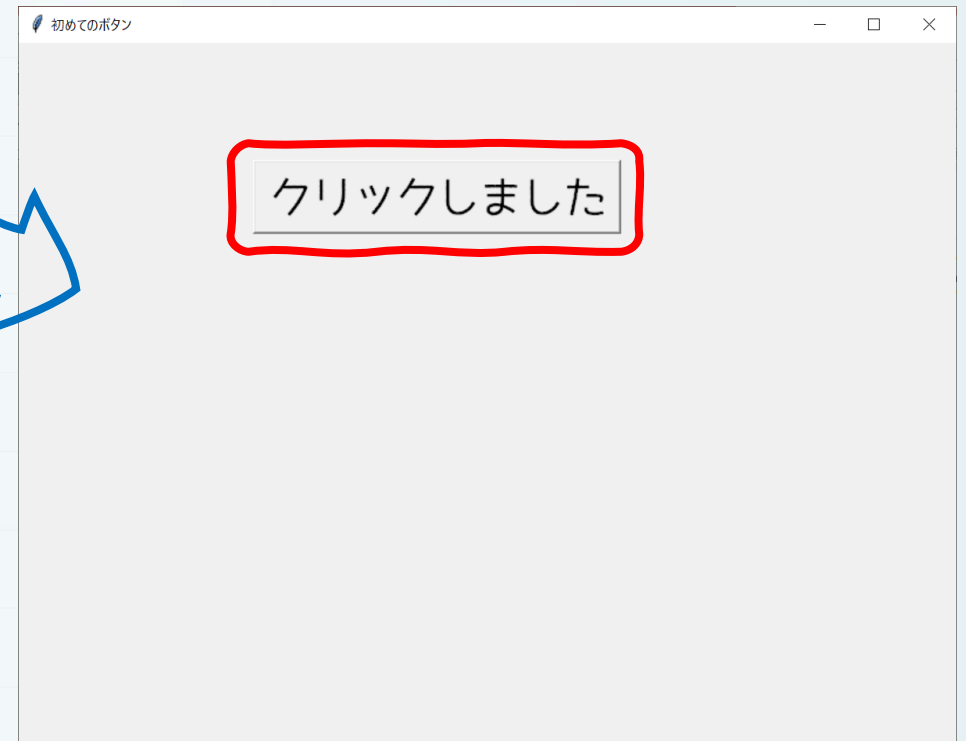
click:カチツという音、title:題名、geometry:幾何学、

button:ボタン（※btnと略されている）、font:書体のデザイン、

command:命令する、place:場所、text:テキスト

ボタンを動かしてみよう

「Run Module」またはF5キーを押してプログラムを実行する。



command = 関数名を理解する

command = 関数名はdef 関数名():で定義した関数を呼び出す命令

クリックボタン関数を呼び出したときの処理を定義

```
def クリックボタン():  
    ボタン[" text" ] = " クリックしました"
```

```
ボタン = tkinter.Button(ウィンドウ, text = " クリックしてください", font = (system , 24), command = クリックボタン)
```

ボタンをクリックすると
クリックボタン関数を実行

キャンバスを使う

画像や図形を描くGUIをキャンバスという。キャンバスの上にラベルやボタンを配置して画面を作っていくよ。

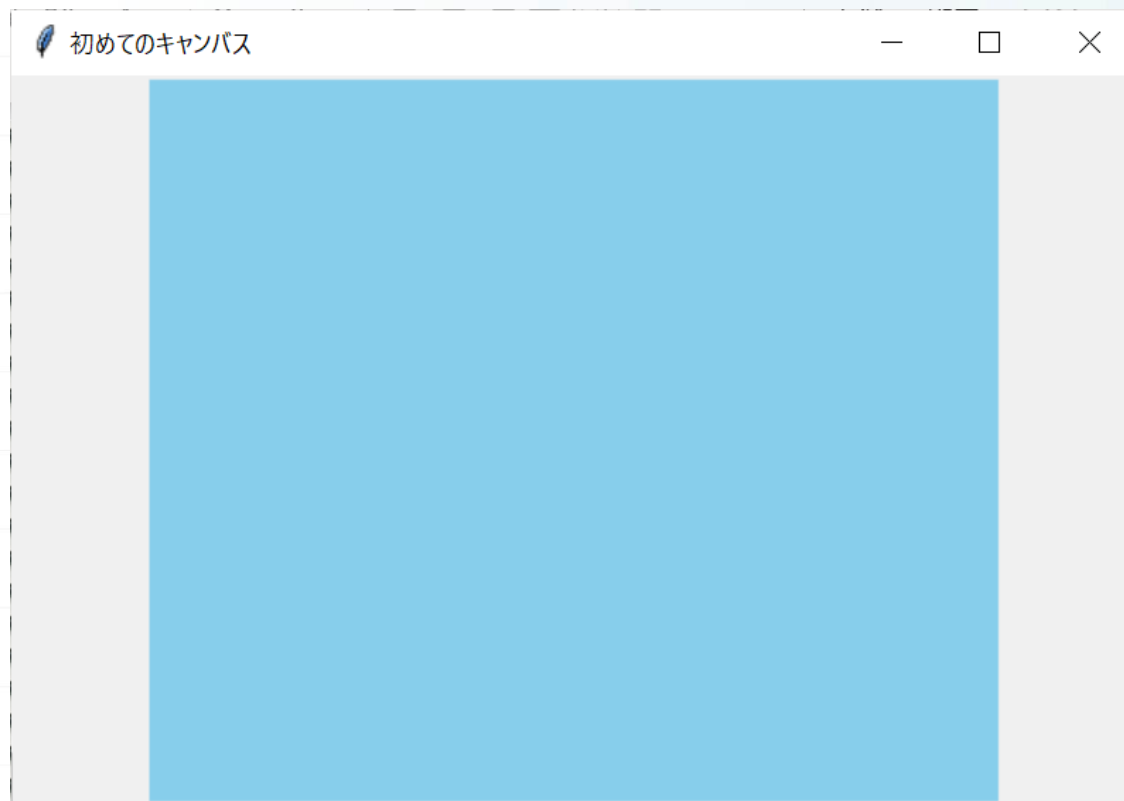
```
import tkinter . . . . tkinterモジュールの呼出し
ウィンドウ = tkinter.Tk()
. . . . 変数「ウィンドウ」にtkinter.Tk()関数を代入する
ウィンドウ.title(" 初めてのキャンバス" )
. . . . ウィンドウのタイトルに「初めてのキャンバス」と表示させる
キャンバス = tkinter.Canvas(ウィンドウ,width = 400 ,
height = 600 , bg = " skyblue" )
. . . . 変数「キャンバス」にtkinter.Canvas()命令を代入する
キャンバス.pack() . . . . ウィンドウにキャンバスを配置する
ウィンドウ.mainloop() . . . . ウィンドウを表示する関数
```

キャンバスを使う

単語帳 ～英語も一緒に覚えてしまおう！～

import:取り込む、title:題名、 canvas:キャンバス、 width:幅
height:高さ、 background:背景 ※bgと略されている
skyblue:スカイブルー、 pack:梱包する

「Run Module」またはF5キーを押してプログラムを実行する。



他の色も試してみよう。

red:赤色:

green:緑色

blue:青色

yellow:黄色

black:黒色

white:白色

Canvas()命令を理解する

キャンバスはCanvas()命令で作り、pack()命令やplace()命令で配置する。

・Canvas()命令の書き方

キャンバスの変数名 = tkinter.Canvas(キャンバスの中身)

キャンバスの中身⇒(ウィンドウ名, width = 幅, height = 高さ, bg = 背景色)

【例】

キャンバス = tkinter.Canvas(ウィンドウ, width = 600, height = 600, bg = "red")

・Pack()命令の書き方

⇒Canvas()命令で作成したキャンバスを設定する

キャンバスの変数名.pack()

キャンバスに画像を表示する

キャンバスに画像を表示するには、PhotoImage()命令で画像ファイルを読み込み、create_image()命令で表示させる。

```
import tkinter . . . . tkinterモジュールの呼出し
ウィンドウ = tkinter.Tk()
. . . . 変数「ウィンドウ」にtkinter.Tk()関数を代入する
ウィンドウ.title(" 初めての画像表示" )
. . . . ウィンドウのタイトルに「初めての画像表示」と表示させる
キャンバス = tkinter.Canvas(ウィンドウ,width = 800 ,
height = 600) . . . 変数「キャンバス」にtkinter.Canvas()命令を代入する
キャンバス.pack() . . . . ウィンドウにキャンバスを配置する
画像 = tkinter.PhotoImage(file = " おみくじ.png" )
. . . . 変数「画像」に画像ファイルを読み込む
キャンバス.create_image(400 , 300 , image = 画像)
. . . . キャンバスに画像を表示する
ウィンドウ.mainloop() . . . . ウィンドウを表示する関数
```

キャンバスを使う

単語帳 ～英語も一緒に覚えてしまおう！～

import:取り込む、title:題名、 canvas:キャンバス、 width:幅
height:高さ、 photo image:写真画像、 file: ファイル、 create:作る
、 image:画像

「Run Module」またはF5キーを押してプログラムを実行する。

初めての画像表示



PhotoImage()命令を理解する

画像の表示はPhotoImage()命令で作り、create_image()命令で描画する。

・ PhotoImage()命令の書き方

画像の変数名 = tkinter.PhotoImage(画像のファイル名)

画像のファイル名⇒(file = " ファイル名.png")

【例】 パスを指定する場合、(file = r" ¥xx¥xx¥ファイル名.png")と記載する

画像 = tkinter.PhotoImage(file = " おみくじ.png")

・ create_image()命令の書き方

⇒ create_image()命令で画像を表示させることができる

キャンバスの変数名.create_image(横幅 , 高さ , image = 画像の変数名)

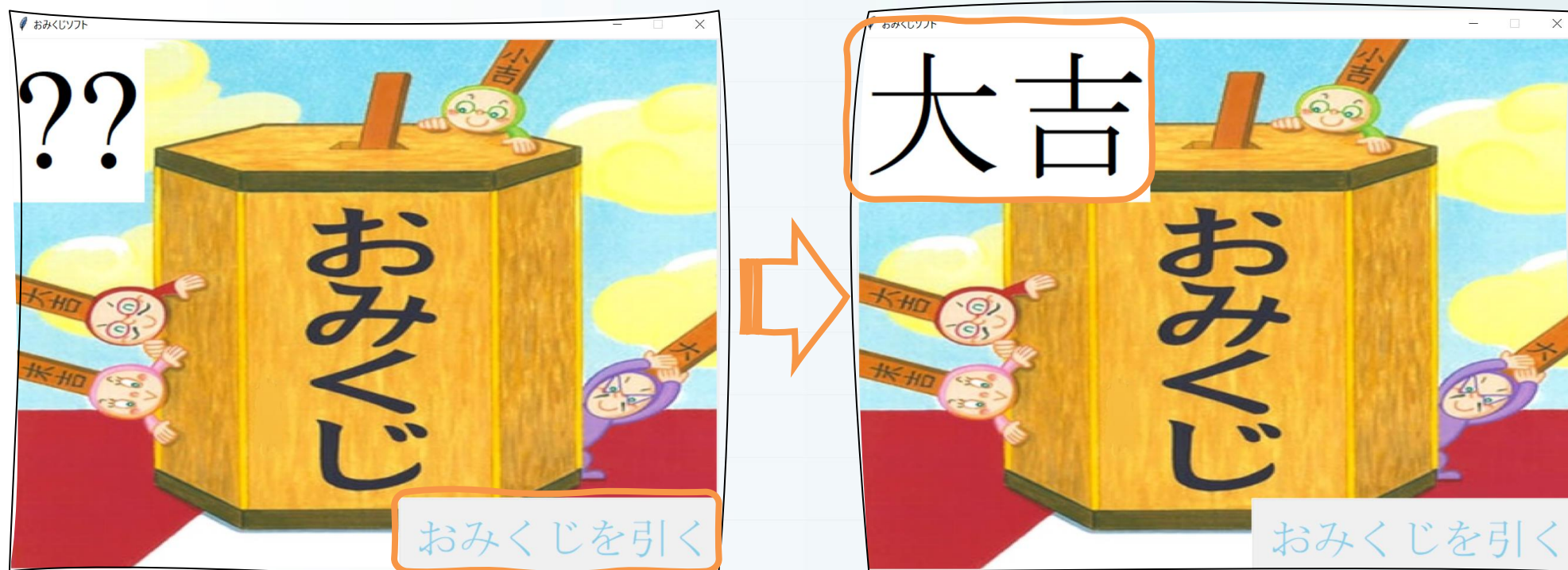
【例】

キャンバス.create_image(400 , 300 , image = 画像)

おみくじゲーム (GUI版) を作る

ここまでに学習した「ラベル」、「ボタン」、「キャンバス」を使っておみくじゲームを作るよ。

おみくじゲーム



ボタンをクリックするとおみくじがひける！

おみくじゲーム (GUI版) を作る

画面のレイアウトを考えよう。ゲームに限らずソフトウェアを開発するときには最初に画面のレイアウトの作成を行う。



画像ファイルを利用する場合、プログラムファイルと同じフォルダに画像ファイルをセットすることを覚えておこう。

おみくじゲーム (GUI版) を作る

ステップ1：画像の表示

まずはキャンバスに画像を表示するところから始めるよ。

```
import tkinter          . . . . tkinterモジュールの呼出し
ウィンドウ = tkinter.Tk()
                    . . . . 変数「ウィンドウ」にtkinter.Tk()関数を代入する
ウィンドウ.title(" おみくじゲーム" )
                    . . . . ウィンドウのタイトルに「おみくじゲーム」と表示させる
ウィンドウ.resizable(False , False) . . . . ウィンドウサイズを固定する
キャンバス = tkinter.Canvas(ウィンドウ,width = 800 , height = 600)
                    . . . . 変数「キャンバス」にtkinter.Canvas()命令を代入する
キャンバス.pack()     . . . . ウィンドウにキャンバスを配置する
画像 = tkinter.PhotoImage(file = " おみくじ.png" )
                    . . . . 変数「画像」に画像ファイルを読み込む
キャンバス.create_image(400 , 300 , image = 画像)
                    . . . . キャンバスに画像を表示する
ウィンドウ.mainloop() . . . . ウィンドウを表示する関数
```

おみくじゲーム (GUI版) を作る

単語帳 ~英語も一緒に覚えてしまおう!~

import:取り込む、title:題名、resizable:サイズ変更可能な
、False:間違った、canvas:キャンバス、width:幅
height:高さ、photo image:写真画像、file:ファイル、create:作る
image:画像

「Run Module」またはF5キーを押してプログラムを実行する。



resizable()命令を理解する

resizable()命令を使うとウィンドウのサイズを固定することができる。

・ resizable()命令の書き方

resizable(横サイズの固定, 縦サイズの固定)

【例1】

ウィンドウ.resizable(False,False)

- ・ 横と縦のウィンドウサイズが固定される



【例2】

ウィンドウ.resizable(True,False)

- ・ 縦のウィンドウサイズが固定される
- ※横はウィンドウサイズを変更できる



【例3】

ウィンドウ.resizable(False,True)

- ・ 横のウィンドウサイズが固定される
- ※縦はウィンドウサイズを変更できる



おみくじゲーム (GUI版) を作る

ステップ2：GUIの配置

キャンバスの上にラベルとボタンを配置する。

```
import tkinter . . . . tkinterモジュールの呼出し
```

```
ウィンドウ = tkinter.Tk() . . 変数「ウィンドウ」にtkinter.Tk()関数を代入する
```

```
ウィンドウ.title(" おみくじゲーム" )
```

. . . ウィンドウのタイトルに「おみくじゲーム」と表示させる

```
ウィンドウ.resizable(False , False) . . . ウィンドウサイズを固定する
```

```
キャンバス = tkinter.Canvas(ウィンドウ,width = 800 , height = 600)
```

. . . 変数「キャンバス」にtkinter.Canvas()命令を代入する

```
キャンバス.pack() . . . . ウィンドウにキャンバスを配置する
```

```
画像 = tkinter.PhotoImage(file = " おみくじ.png" )
```

. . . 変数「画像」に画像ファイルを読み込む

```
キャンバス.create_image(400 , 300 , image = 画像)
```

. . . キャンバスに画像を表示する

次のページに続く

おみくじゲーム (GUI版) を作る

ラベル = tkinter.Label(ウィンドウ,text = " ??",font = (" Times New Roman",120) , bg = " white") . . . 変数「ラベル」にラベル関数を代入する

ラベル.place(x=1,y=1) . . . x座標=1,y座標=1の位置にラベルを配置する

ボタン = tkinter.Button(ウィンドウ,text=" おみくじを引く" , font = (" Times New Roman" ,24) ,fg = " skyblue")

. . . 変数「ボタン」にボタン関数を代入する

ボタン.place(x=550,y=540). . . x座標=550,y座標=540の位置にボタンを配置する

ウィンドウ.mainloop() . . . ウィンドウを表示する関数

単語帳 ～英語も一緒に覚えてしまおう！～

import:取り込む、title:題名、resizable:サイズ変更可能な

、False:間違った、canvas:キャンバス、width:幅

height:高さ、photo image:写真画像、file: ファイル、create:作る

image:画像、font:書体のデザイン、white:白色、place:場所

、foreground:最前面 ※fgと略されている、skyblue:スカイブルー

おみくじゲーム (GUI版) を作る

「Run Module」またはF5キーを押してプログラムを実行する。



おみくじゲーム (GUI版) を作る

ステップ3: ボタンを反応させる
ボタンをクリックした時におみくじの結果を表示させる。

```
import tkinter . . . . tkinterモジュールの呼出し
import random . . . . randomモジュールの呼出し
def クリックボタン(): . . . ボタンをクリックしたときの動きを関数で定義する
    ラベル["text"]=random.choice(["大吉","中吉","小吉","凶"])
    . . . ラベルの文字をリストからランダムに選択した値を代入する
    ラベル.update() . . . ラベルの文字の更新をすぐに行う
ウィンドウ = tkinter.Tk() . . 変数「ウィンドウ」にtkinter.Tk()関数を代入する
ウィンドウ.title("おみくじゲーム")
    . . . ウィンドウのタイトルに「おみくじゲーム」と表示させる
ウィンドウ.resizable(False , False) . . . ウィンドウサイズを固定する
キャンバス = tkinter.Canvas(ウィンドウ,width = 800 , height = 600)
    . . . キャンバスに画像を表示する
    次のページに続く
```

おみくじゲーム (GUI版) を作る

キャンバス.pack() ウィンドウにキャンバスを配置する

画像 = tkinter.PhotoImage(file = "おみくじ.png")

. . . . 変数「画像」に画像ファイルを読み込む

キャンバス.create_image(400 , 300 , image = 画像)

. . . . キャンバスに画像を表示する

ラベル = tkinter.Label(ウィンドウ,text = "??",font = ("Times New Roman",120),bg = "white") 変数「ラベル」にラベル関数を代入する

ラベル.place(x=1,y=1) x座標=1,y座標=1の位置にラベルを配置する

ボタン = tkinter.Button(ウィンドウ,text="おみくじを引く",font =("Times New Roman",24),fg="skyblue",command = クリックボタン)

. . . . 変数「ボタン」にボタン関数を代入する

ボタン.place(x=550,y=540)

. . . . x座標=550,y座標=540の位置にボタンを配置する

ウィンドウ.mainloop() ウィンドウを表示する関数

おみくじゲーム (GUI版) を作る



単語帳 ～英語も一緒に覚えてしまおう！～

import:取り込む、random:でたらしめの、choice:選択する、title:題名、
resizable:サイズ変更可能な、update:～を最新のものにする、
False:間違った、canvas:キャンバス、width:幅、height:高さ、
photo image:写真画像、file:ファイル、create:作る
image:画像、font:書体のデザイン、white:白色、place:場所
、foreground:最前面 ※fgと略されている、skyblue:スカイブルー
command:命令する

おみくじゲーム (GUI版) を作る

「Run Module」またはF5キーを押してプログラムを実行する。



おさらい&発展学習

これまで習った方法を使って、キャンバスにいろいろな図形を描いてみよう。

- ・キャンバスに直線を引く

`create_line(x1,y1,x2,y2 fill = 色 , width = 線の太さ)`

(x1,y1)



(x2,y2)

```
import tkinter
```

```
ウィンドウ = tkinter.Tk()
```

```
ウィンドウ.geometry("400x400")
```

```
ウィンドウ.title("いろんな図形を描こう")
```

```
キャンバス = tkinter.Canvas(ウィンドウ,width = 800 , height = 800 , bg="white")
```

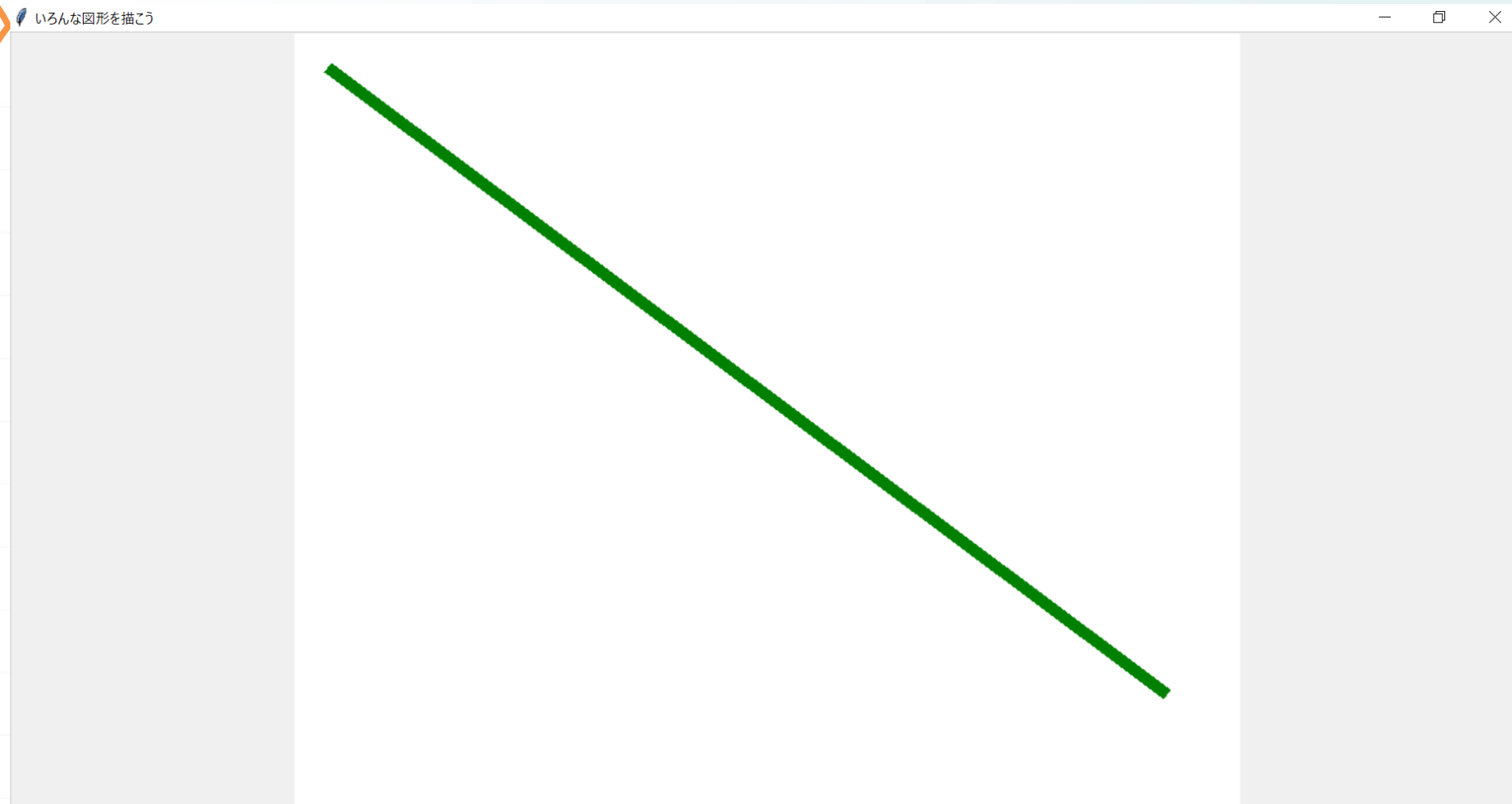
```
キャンバス.pack()
```

```
キャンバス.create_line(30 , 30 , 740 , 560 ,fill="green" ,width = 10 )
```

```
ウィンドウ.mainloop()
```

おさらい&発展学習

「Run Module」またはF5キーを押してプログラムを実行する。



おさらい&発展学習



・ 矩形(くけい)を描く

※正方形や長方形のことをプログラミングの世界では矩形と呼ぶ

`create_rectangle(x1,y1,x2,y2 fill = 色 , width = 線の太さ)`

(x1,y1)



(x2,y2)

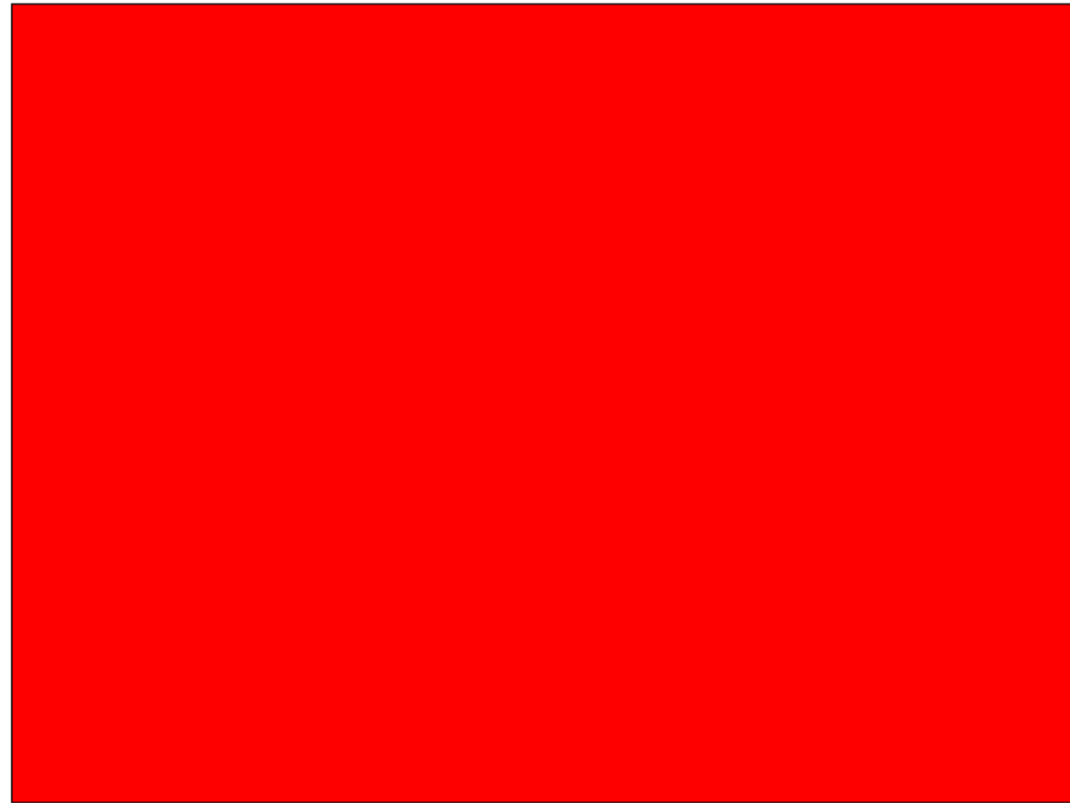
```
import tkinter
ウィンドウ = tkinter.Tk()
ウィンドウ.geometry("400x400")
ウィンドウ.title("いろんな図形を描こう")
キャンバス = tkinter.Canvas(ウィンドウ,width = 800 , height = 800 , bg="white")
キャンバス.pack()
キャンバス.create_rectangle(30 , 30 , 740 , 560 ,fill=" red" ,width = 1 )
ウィンドウ.mainloop()
```


おさらい&発展学習

「Run Module」またはF5キーを押してプログラムを実行する。



いろんな図形を描こう

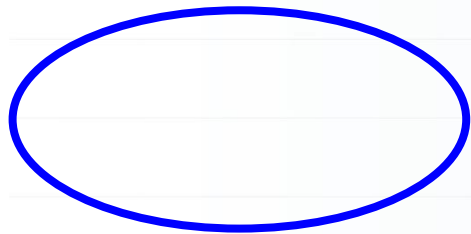


おさらい&発展学習

- ・ 楕円(だえん)を描く

`create_oval(x1,y1,x2,y2 fill = 色 , width = 線の太さ)`

(x1,y1)



(x2,y2)

```
import tkinter
```

```
ウィンドウ = tkinter.Tk()
```

```
ウィンドウ.geometry("400x400")
```

```
ウィンドウ.title("いろんな図形を描こう")
```

```
キャンバス = tkinter.Canvas(ウィンドウ,width = 800 , height = 800 , bg="white")
```

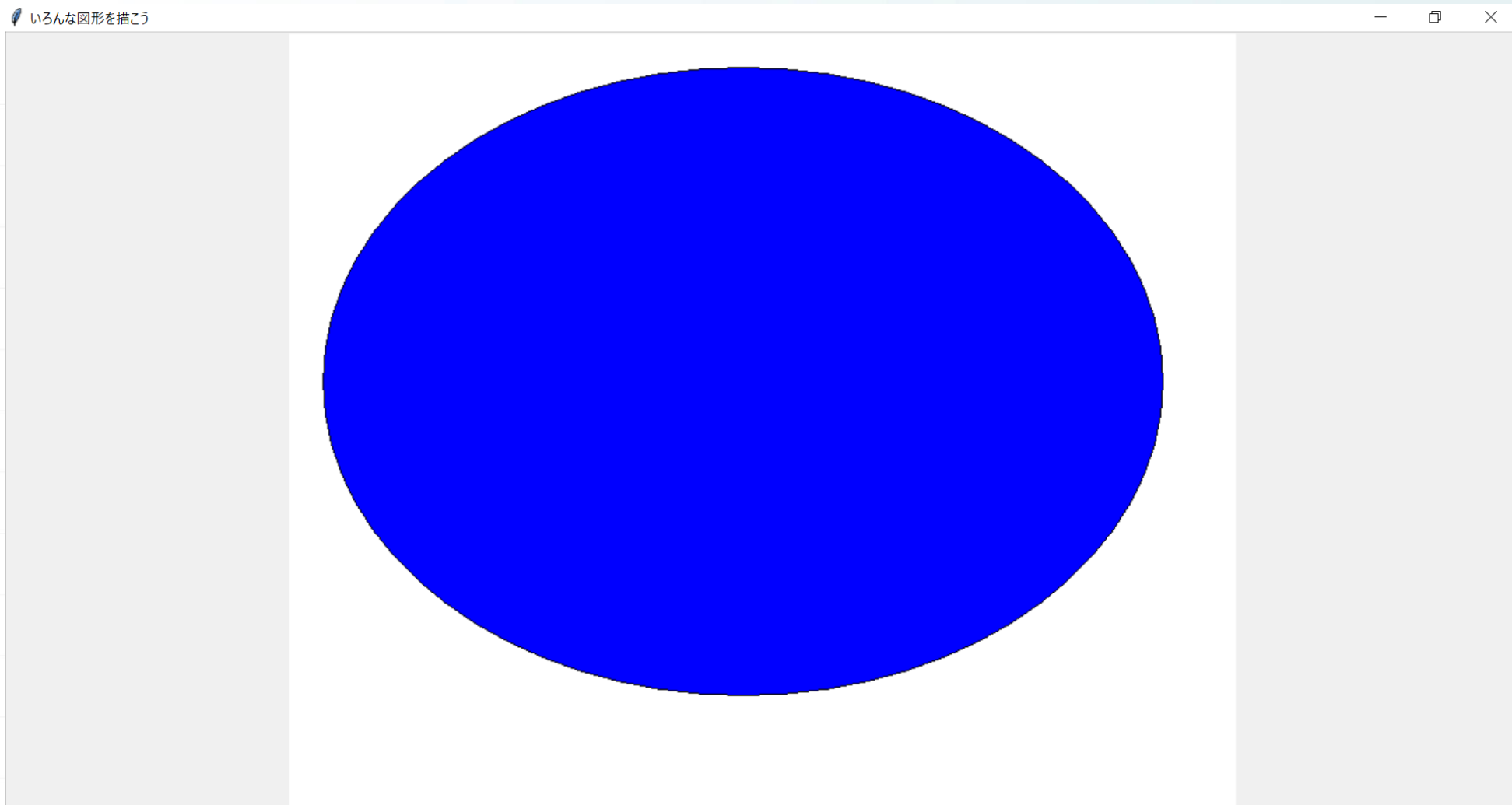
```
キャンバス.pack()
```

```
キャンバス.create_oval(30 , 30 , 740 , 560 ,fill=" blue" ,width = 1 )
```

```
ウィンドウ.mainloop()
```

おさらい&発展学習

「Run Module」またはF5キーを押してプログラムを実行する。

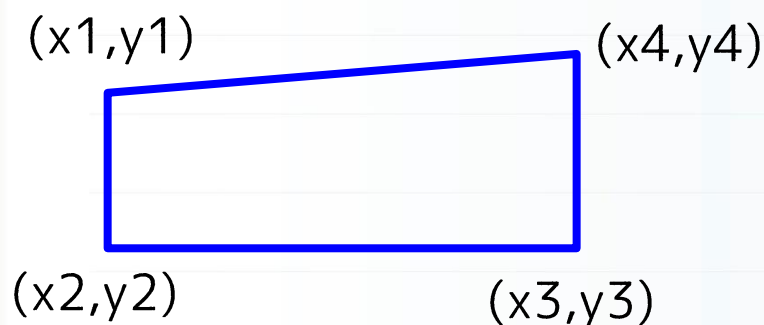


おさらい&発展学習



- ・ 多角形を描く

`create_polygon(x1,y1,x2,y2,x3,y3,x4,y4 fill = 色 ,
width = 線の太さ)`



```
import tkinter
```

```
ウィンドウ = tkinter.Tk()
```

```
ウィンドウ.geometry("400x400")
```

```
ウィンドウ.title("いろんな図形を描こう")
```

```
キャンバス = tkinter.Canvas(ウィンドウ,width = 800 , height = 800,bg="white")
```

```
キャンバス.pack()
```

```
キャンバス.create_polygon(250,250,150,350,380,400,350,300,fill = "blue",width = 1 )
```

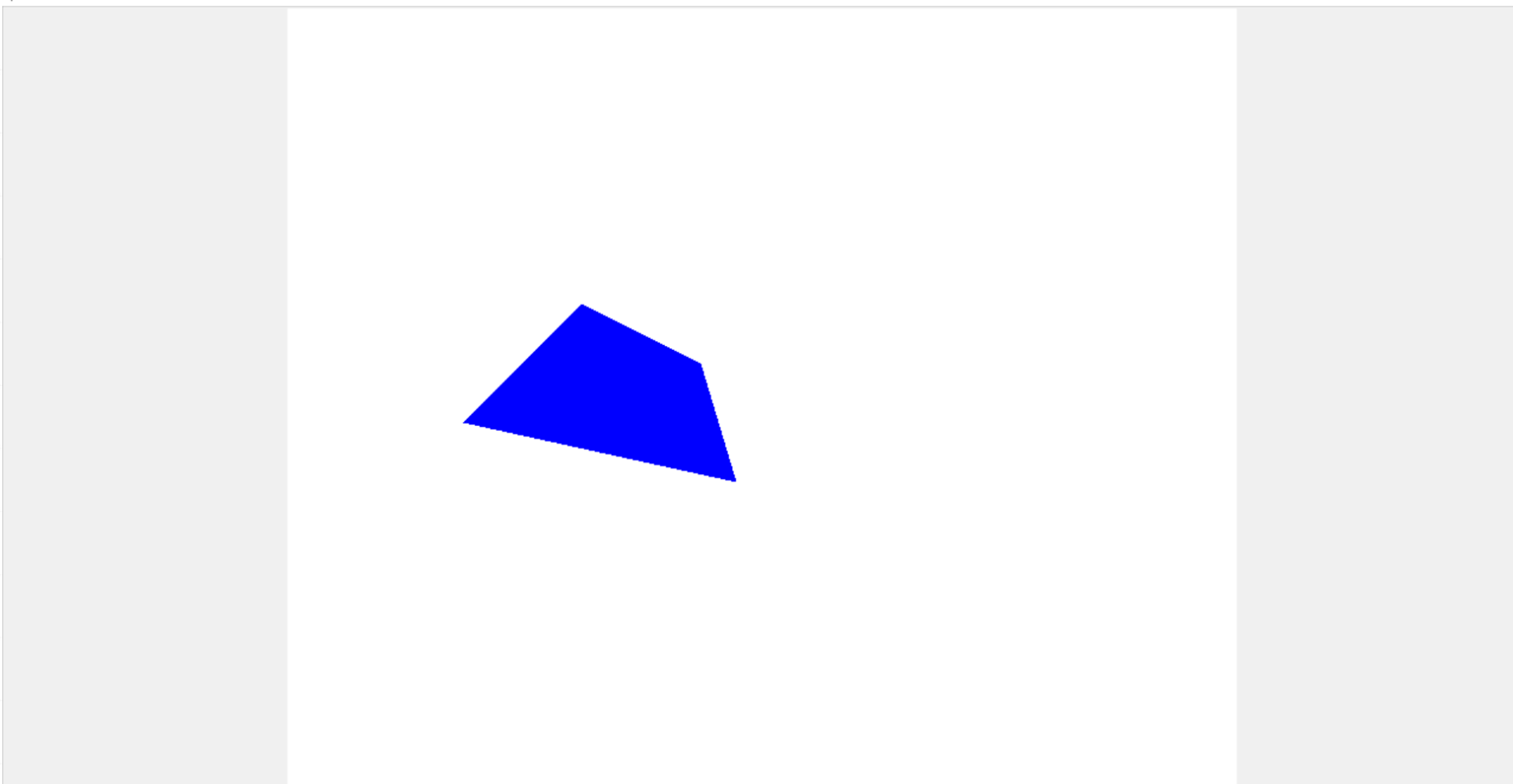
```
ウィンドウ.mainloop()
```

おさらい&発展学習

「Run Module」またはF5キーを押してプログラムを実行する。

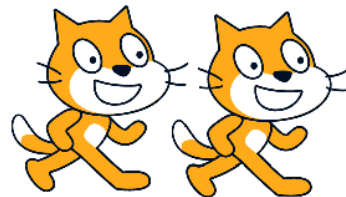


いろんな図形を描こう



復習 & チャレンジ

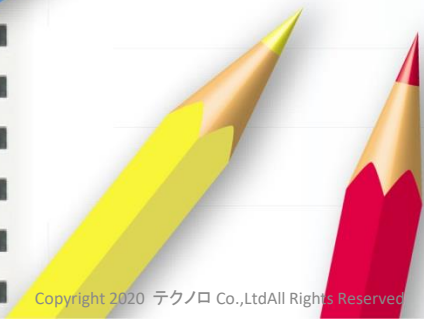
ここまで習ったことをScratchでもできるかチャレンジしてみよう。
その過程でScratchでできること、Pythonでないといけないことを整理してみよう。



メモ



プログラミング教室の テクノロ



なまえ：