



# Pythonの道

GUIの基礎を学ぶ②

# もくじ

- ・ GUIの基礎②
- ・ 診断ゲームを作る



# GUIの基礎②

テキスト入力欄などのGUIの使い方を学習しよう。  
テキスト入力欄は1行のみの入力欄と複数行の入力欄があることを知っておこう。

初めてのテキスト入力欄

※文字や数字を入力できる  
エリアのことをテキスト入  
力欄という。

初めてのテキスト入力欄

あああ

# GUIの基礎②

1行のテキスト入力欄はEntry()命令で作る。  
テキスト入力欄を配置するときはplace()命令を使う。

```
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("初めてのテキスト入力欄")
. . . . ウィンドウのタイトルに「初めてのテキスト入力欄」と表示させる
ウィンドウ.geometry("400x200") . . . . ウィンドウのサイズを指定
テキスト入力 = tkinter.Entry(width = 20) ※width=20は横幅サイズが20をあらわす
. . . . 変数「テキスト入力」にtkinter.Entry()命令を代入する
テキスト入力.place(x=10,y=10) . . . . 作成したテキスト入力欄を配置する
ウィンドウ.mainloop() . . . . mainloop()命令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。

# GUIの基礎②

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、entry:記入

width:幅、place:場所

「Run Module」またはF5キーを押してプログラムを実行する。



# Entry()命令を理解する

1行のテキスト入力欄を作るときは、Entry()命令を使う。テキスト入力欄もplace()命令で配置する。

・ Entry()命令の書き方

テキスト入力欄の名前 = tkinter.Entry(テキスト入力欄の中身)

テキスト入力欄の中身 ⇒ (width = 20)

【例】

テキスト入力欄 = tkinter.Entry(width = 20)

⇒ 横幅20のテキスト入力欄が作成される

・ Place()命令の書き方 ⇒ Entry()命令で作成したラベルを設定する

テキスト入力欄.place(x=X座標, y=Y座標)

⇒ (X座標, Y座標)の位置にテキスト入力欄を配置する

# GUIの基礎②



入力欄に文字を入力し、ボタンを押すとその文字列が取得できるプログラムを作ってみよう。Entry内の文字列はget()命令で取得できる。

```
import tkinter . . . . tkinter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時に実行する関数を定義する
    テキスト = テキスト入力.get()
    . . . . 変数「テキスト」にテキスト入力欄に入力された文字を代入する
    ボタン["text"] = テキスト
    . . ボタンに表示されている文字をテキスト入力欄に入力された文字に変更する
ウィンドウ = tkinter.Tk()
    . . . . 変数「ウィンドウ」にtkinter.Tk()命令を代入する
ウィンドウ.title("初めてのテキスト入力欄")
    . . . . ウィンドウのタイトルに「初めてのテキスト入力欄」と表示させる
ウィンドウ.geometry("400x200") . . . . ウィンドウのサイズを指定
```

次のページに続く

# GUIの基礎②

テキスト入力 = tkinter.Entry(width = 20)

・・・変数「テキスト入力」にtkinter.Entry()命令を代入する

テキスト入力.place(x=20,y=20)

・・・作成したテキスト入力欄を配置する

ボタン= tkinter.Button(text = "文字列の取得",command=クリックボタン) ・・・変数「ボタン」にtkinter.Button()命令を代入する

ボタン.place(x=20,y=100) ・・・作成したボタンを配置する

ウィンドウ.mainloop()

・・・mainloop()命令を実行してウィンドウを表示する

単語帳 ～英語も一緒に覚えてしまおう！～

import:取り込む、define:定義する※defと略されている、

get:手に入れる、txt:テキスト、title:題名、geometry:幾何学、

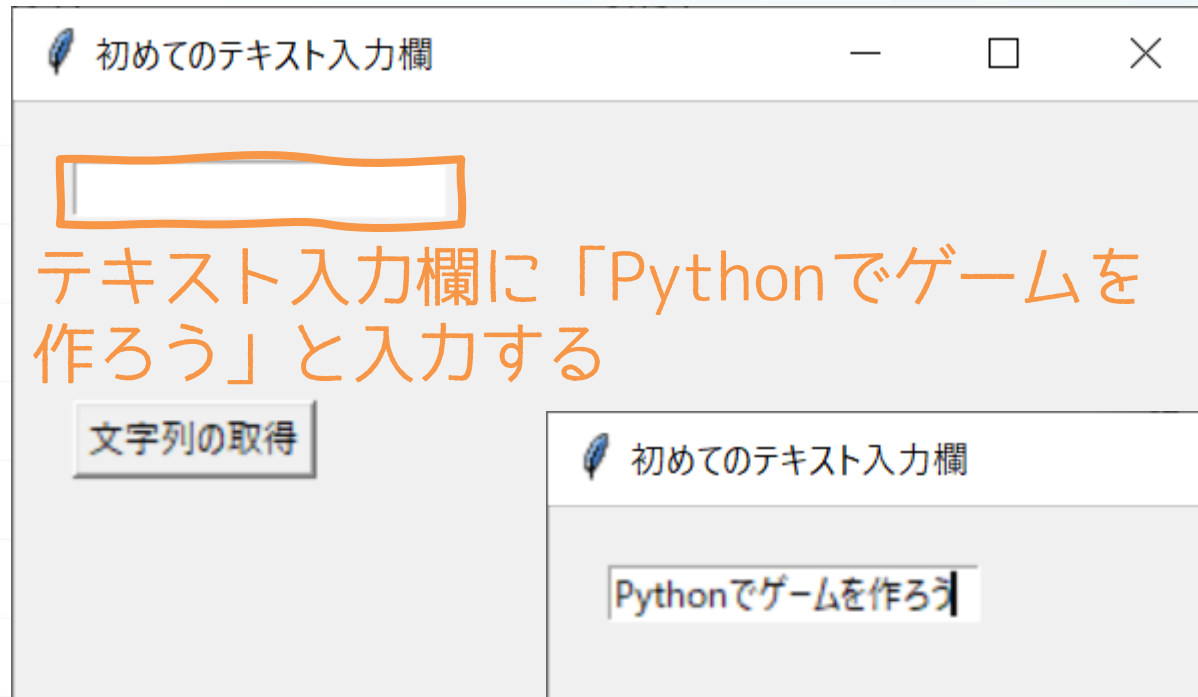
entry:記入、width:幅、place:場所、button:ボタン、

command:命令する

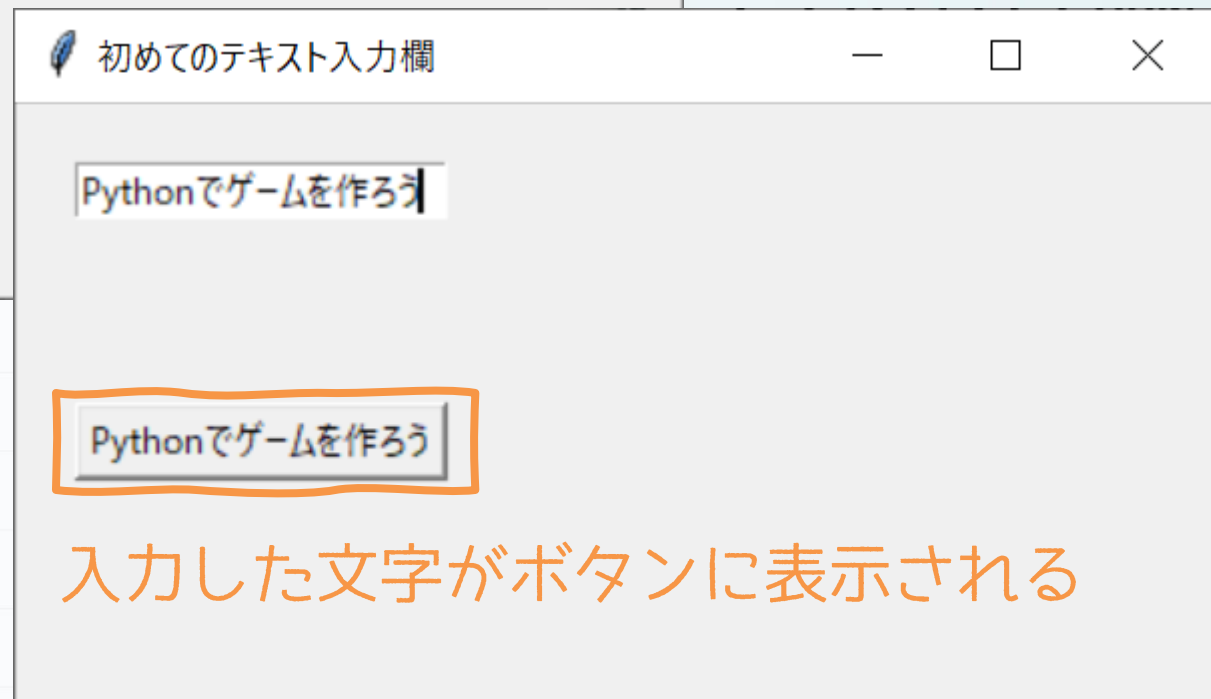


# GUIの基礎②

「Run Module」またはF5キーを押してプログラムを実行する。



テキスト入力欄に「Pythonでゲームを作ろう」と入力する



入力した文字がボタンに表示される

# get()命令を理解する

Entry()命令で作成したテキスト入力欄の値を取得する場合、get()命令を使用する。

・ get()命令の書き方

テキスト入力 = tkinter.Entry(width = 20)

・・・テキスト入力欄を最初に作成する

テキスト入力.insert(tkinter.END, "初期値を入力")

・・・insert()命令を使うとテキスト入力欄の初期値を設定することができる

※tkinter.ENDは最終行に文字を入力するという意味

入力値 = テキスト入力.get()

・・・変数「入力値」にget()命令で取得した値を代入する

print(入力値)

・・・変数「入力値」の値を表示する

# 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

※command命令を使うと関数を呼び出すことができる。

# GUIの基礎②

Text()命令を使って複数行のテキスト入力欄をつくってみよう。

```
import tkinter . . . . tkinterモジュールの呼出し
def クリックボタン():
    . . . . クリックボタン関数を呼び出した時に実行する関数を定義
    テキスト入力.insert(tkinter.END,"モンスターが現れた!")
    . . . . テキスト入力欄に「モンスターが現れた」という文字列を挿入する
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("複数行のテキスト入力")
    . . . . ウィンドウのタイトルに「複数行のテキスト入力」と表示させる
ウィンドウ.geometry("400x200") . . . . ウィンドウのサイズを指定
ボタン = tkinter.Button(text="メッセージ",command = クリック
ボタン) . . . . 変数「ボタン」にtkinter.Button()命令を代入する
```

※ボタンに表示される文字は「メッセージ」、ボタンをクリックするとクリックボタン関数が呼び出される

次のページに続く

# GUIの基礎②

ボタン.pack() . . . ボタンを配置する

テキスト入力 = tkinter.Text()

. . . 変数「テキスト入力」にtkinter.Text命令を代入する

テキスト入力.pack() . . . テキスト入力欄を配置する

ウィンドウ.mainloop() . . . ウィンドウを表示する

単語帳 ～英語も一緒に覚えてしまおう！～

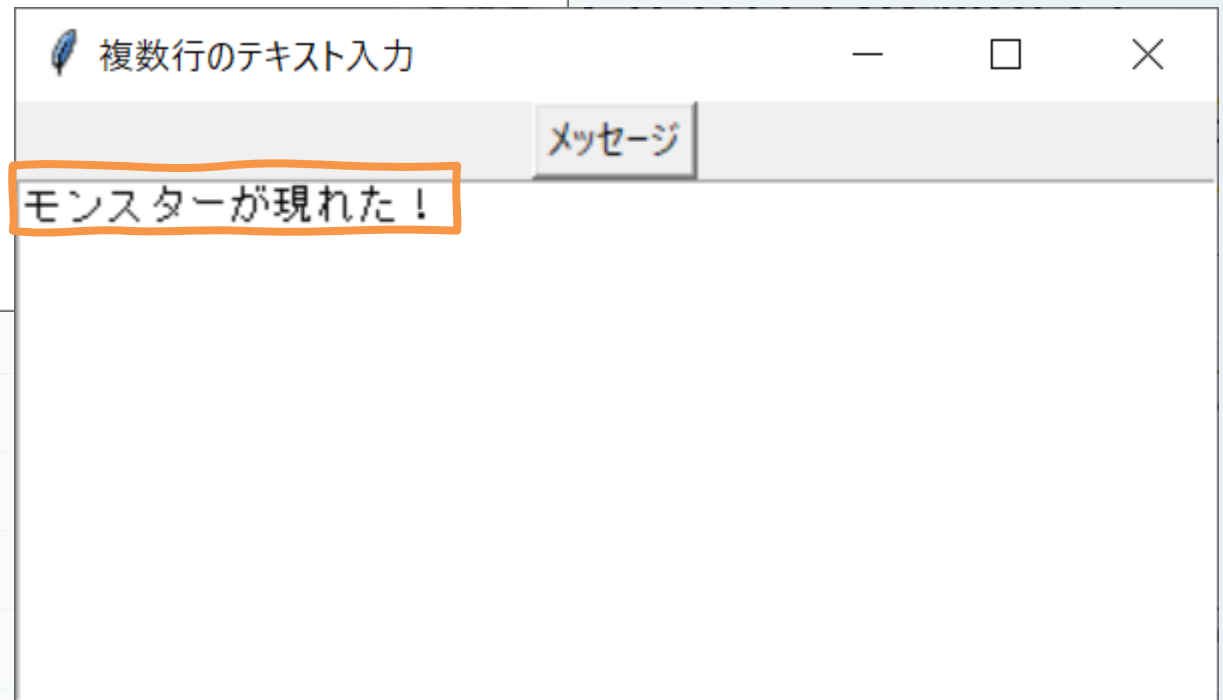
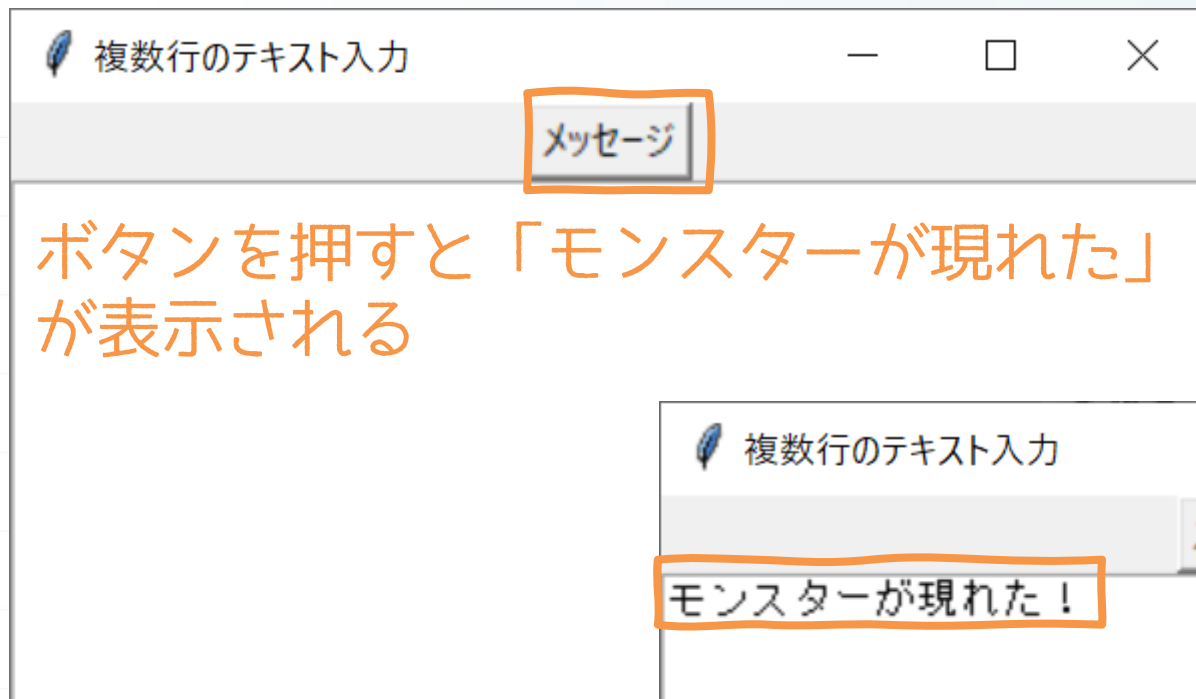
import:取り込む、define:定義する※defと略されている、

insert:挿入する、title:題名、 geometry:幾何学、 button:ボタン

、txt:テキスト、command:命令する、pack:包み

# GUIの基礎②

「Run Module」またはF5キーを押してプログラムを実行する。



# Text()命令を理解する

複数行のテキスト入力欄を作る場合、Text()命令を使用する。Text()命令で作ったテキスト入力欄はpack()命令または、place()命令で配置する。

- ・ Text()命令の書き方

テキスト = tkinter.Text()

テキスト.pack()

⇒pack()命令でテキスト入力欄を配置する

- ・ place()命令を使用する場合の書き方

テキスト = tkinter.Text()

テキスト.place(x=20 , y= 50 , width = 360 ,height = 120)

⇒place()命令を使うとテキスト入力欄の位置や大きさを自由に決めることができる

## GUIの基礎②

テキスト入力欄の次はチェックボックスを作ってみよう。  
チェックボックスはCheckbutton()命令で作る。

```
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("チェックボタンを扱う")  
                . . . . ウィンドウのタイトルに「チェックボタンを扱う」と表示させる  
ウィンドウ.geometry("400x200") . . . . ウィンドウのサイズを指定  
チェックボックス = tkinter.Checkbutton(text = "チェックボタ  
ン") . . . . 変数「チェックボックス」にtkinter.Checkbutton()命令を代入する  
チェックボックス.pack() . . . . チェックボックスを配置する  
ウィンドウ.mainloop() . . . . ウィンドウを表示する
```

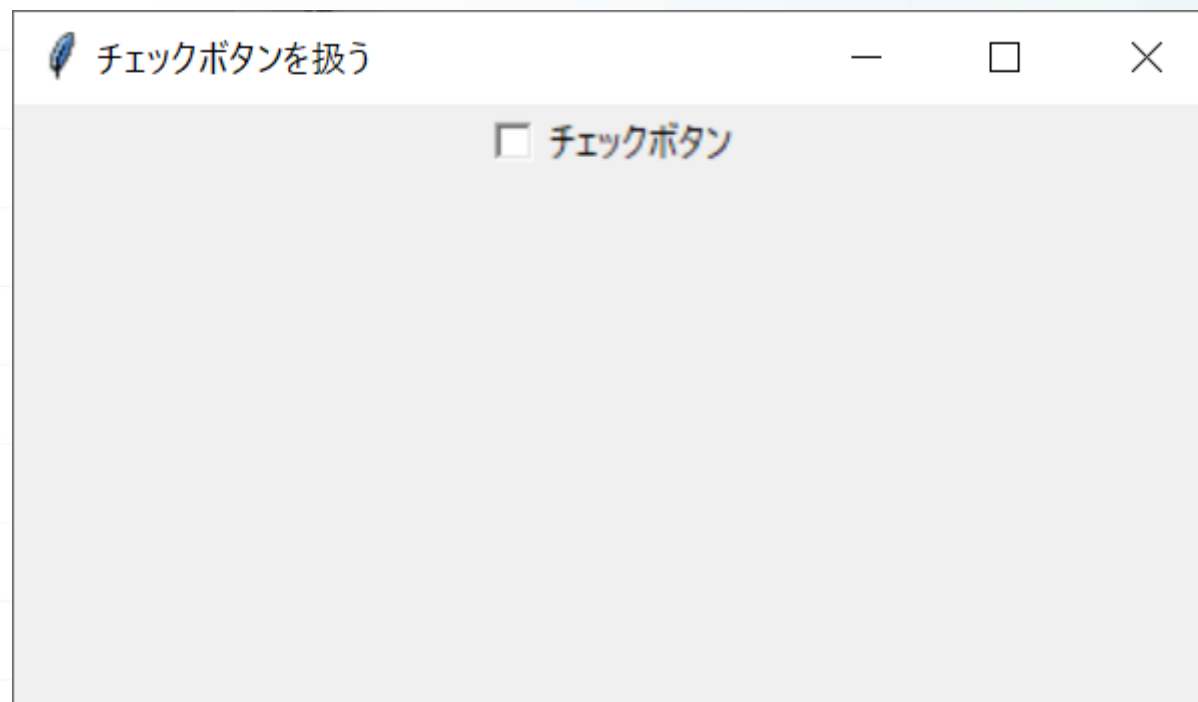


# GUIの基礎②

単語帳 ~英語も一緒に覚えてしまおう!~

import:取り込む、check:点検する、title:題名、 geometry:幾何学、  
button:ボタン、txt:テキスト、 pack:包み

「Run Module」またはF5キーを押してプログラムを実行する。



# Checkbox()命令を理解する

画面にチェックボックスを作る場合、Checkbox()命令を使用する。Checkbox()命令で作ったチェックボックスはpack()命令で配置する。

- ・ Checkbox()命令の書き方

```
チェックボックス = tkinter.Checkbutton()
```

```
チェックボックス.pack()
```

⇒pack()命令でチェックボックスを配置する

## 【例】

```
チェックボックス = tkinter.Checkbutton(text = "チェックボタン")
```

```
チェックボックス.pack()
```

# GUIの基礎②

チェックボックスにチェックがついているかを調べるプログラムをつくる。まずはチェックがついた状態を作ってみよう。

```
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("チェックボタンを扱う")
. . . . ウィンドウのタイトルに「チェックボタンを扱う」と表示させる
ウィンドウ.geometry("400x200") . . . . ウィンドウのサイズを指定
チェック = tkinter.BooleanVar()
. . . . 変数「チェック」にtkinter.BooleanVar()命令を代入する
チェック.set(True) . . . . tkinter.BooleanVar()命令にTrueをセットする
※Trueをセットするとチェックボックスにチェックが入る。チェックを外すときはFalseをセットする
チェックボックス = tkinter.Checkbutton(text = "チェックボタ
ン",variable = チェック)
. . . . 変数「チェックボックス」にtkinter.Checkbutton()命令を代入する
※variable = にチェックの中身 (True) をセットしている
```

次のページに続く

# GUIの基礎②

チェックボックス.pack() . . . . チェックボックスを配置する  
ウィンドウ.mainloop() . . . . ウィンドウを表示する

単語帳 ～英語も一緒に覚えてしまおう！～

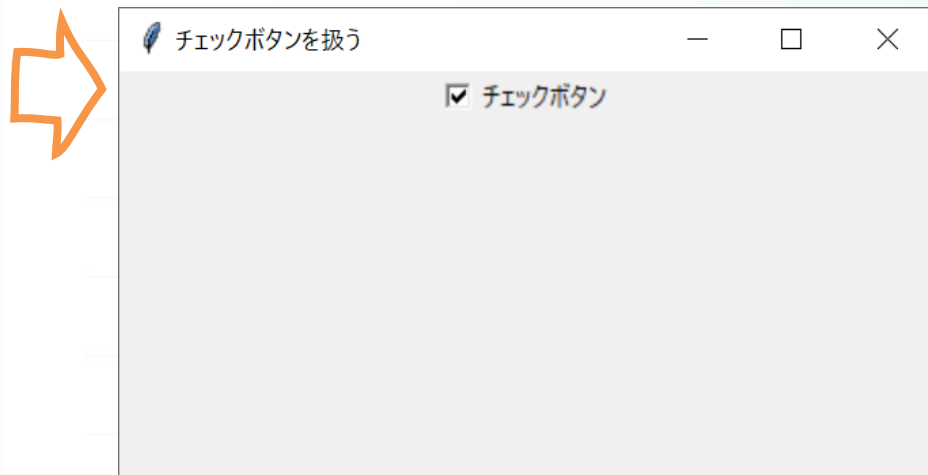
import:取り込む、title:題名、 geometry:幾何学、

Boolean:ブールの ※ジョージブール（イギリスの数学者）、 True : 真実の

varchar:可変長の文字列 ※varと略されている

button:ボタン、 txt: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。



# GUIの基礎②



次にBooleanVar()命令を使って、チェックボックスにチェックがついているかを調べてみよう。

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

```
def チェック(): . . . . 関数「チェック」を定義する
```

```
    if チェック結果.get() == True:
```

```
        print("チェックされています")
```

```
    else:
```

```
        print("チェックされていません")
```

．．．if文を定義する  
もし、「チェック結果」から取り出した値が「True」なら、「チェックされています」と表示する。そうでなければ、「チェックされていません」と表示する。

```
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("チェックの状態を知る")
```

．．．ウィンドウのタイトルに「チェックの状態を知る」と表示させる

次のページに続く

# GUIの基礎②

ウィンドウ.geometry("400x200") . . . ウィンドウのサイズを指定

チェック結果 = tkinter.BooleanVar()

. . . . 変数「チェック結果」にtkinter.BooleanVar()命令を代入する

チェック結果.set(False)

. . . . 変数「チェック結果」の初期値に「False」を代入する

チェックボックス = tkinter.Checkbutton(text = "チェック  
ボタン",variable = チェック結果,command = チェック)

. . . . 変数「チェックボックス」にtkinter.Checkbutton()命令を代入する

※variable = に変数「チェック結果」をセットしている

※command = 関数「チェック」をセットしている

チェックボックス.pack() . . . . チェックボックスを配置する

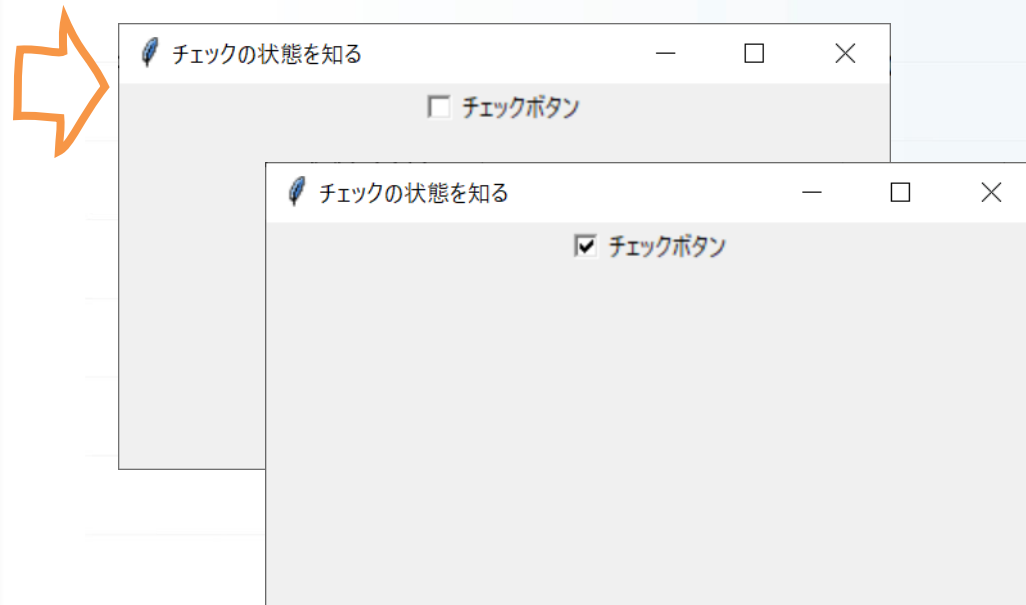
ウィンドウ.mainloop() . . . . ウィンドウを表示する

# GUIの基礎②

単語帳 ～英語も一緒に覚えてしまおう！～

import:取り込む、 define:定義する※defと略されている、 if:もし～なら、  
get:手に入れる、 True : 真実の、 print:印刷する、 else:その他の、  
title:題名、 geometry:幾何学、 button:ボタン、 txt:テキスト  
Boolean:ブールの ※ジョージブール（イギリスの数学者）、  
variable:変わりやすい、 varchar:可変長の文字列 ※varと略されている

「Run Module」またはF5キーを押してプログラムを実行する。



=====  
チェックされています  
チェックされていません

# BooleanVar()命令を理解する

チェックボックスにチェックがついているかを調べる場合、BooleanVar()命令を使用する。

- ・ BooleanVar()命令の書き方

```
チェック結果 = tkinter.BooleanVar()
```

```
チェック結果.set(False)
```

BooleanVar()命令を使うことで真偽を扱うことができるようになる。

真：True、偽：False



scratchの調べるブロックのように真偽を扱うときに使う。



## GUIの基礎②

メッセージボックスを使って画面にメッセージを表示させる。  
メッセージボックスを使うときは、tkinter.messageboxモジュールをインポートする必要がある。

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

def クリックボタン(): . . . . クリックボタン関数を定義する
    tkinter.messagebox.showinfo("情報","ボタンを押しました")
    . . . . メッセージボックスを表示する
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("初めてのメッセージボックス")
    . . . . ウィンドウのタイトルに「初めてのメッセージボックス」と表示させる
```

次のページに続く

# GUIの基礎②

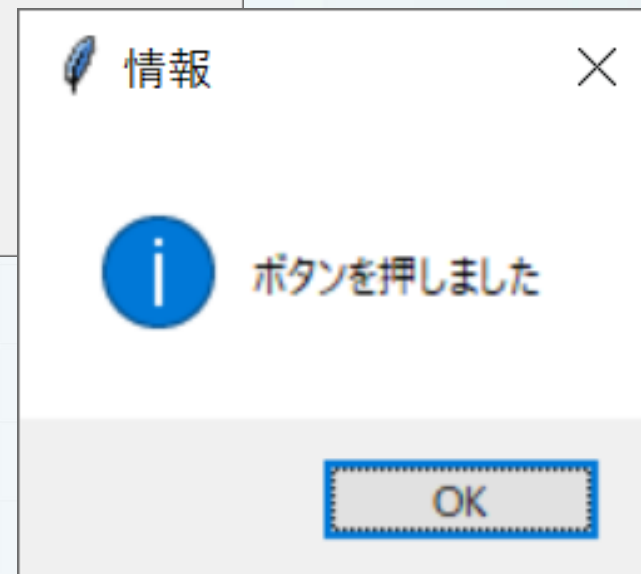
ウィンドウ.geometry("400x200") . . . ウィンドウのサイズを指定  
ボタン = tkinter.Button(text = "テスト",command=クリックボタ  
ン) . . . . 変数「ボタン」にtkinter.Button()命令を代入する  
ボタン.pack() . . . . ボタンを配置する  
ウィンドウ.mainloop() . . . . ウィンドウを表示する

単語帳 ～英語も一緒に覚えてしまおう！～

import:取り込む、 messagebox:メッセージボックス、  
define:定義する※defと略されている、 show:見せる、  
information:情報 ※infoと略されている、 title:題名、  
geometry:幾何学、 button:ボタン、 txt:テキスト、 command:命令、  
pack:包み

# GUIの基礎②

「Run Module」またはF5キーを押してプログラムを実行する。



# tkinter.messageboxを理解する

tkinter.messageboxモジュールを使うと様々な種類のメッセージボックスを使うことができる。

・ tkinter.messageboxモジュールの使い方

①import tkinter.messageboxでモジュールをインポートする。

②tkinter.messagebox.showinfo()命令でメッセージボックスを作成する。

【例】

○○：メッセージボックスのラベル名

△△：メッセージボックスに表示するメッセージ

tkinter.messeagebox.showinfo (" ○○" ," △△" )

## メッセージボックスの種類

showinfo()	情報を表示するメッセージボックス
showwarning()	警告を表示するメッセージボックス
showerror()	エラーを表示するメッセージボックス
askyesno()	「はい」「いいえ」のボタンがあるメッセージボックス
askokcancel()	「OK」「キャンセル」のボタンがあるメッセージボックス

③ボタン命令などに組み込む。

# ネコ度診断ゲームを作る

ここまでに学習した「テキスト入力欄」、「チェックボックス」をネコ度診断ゲームを作るよ。

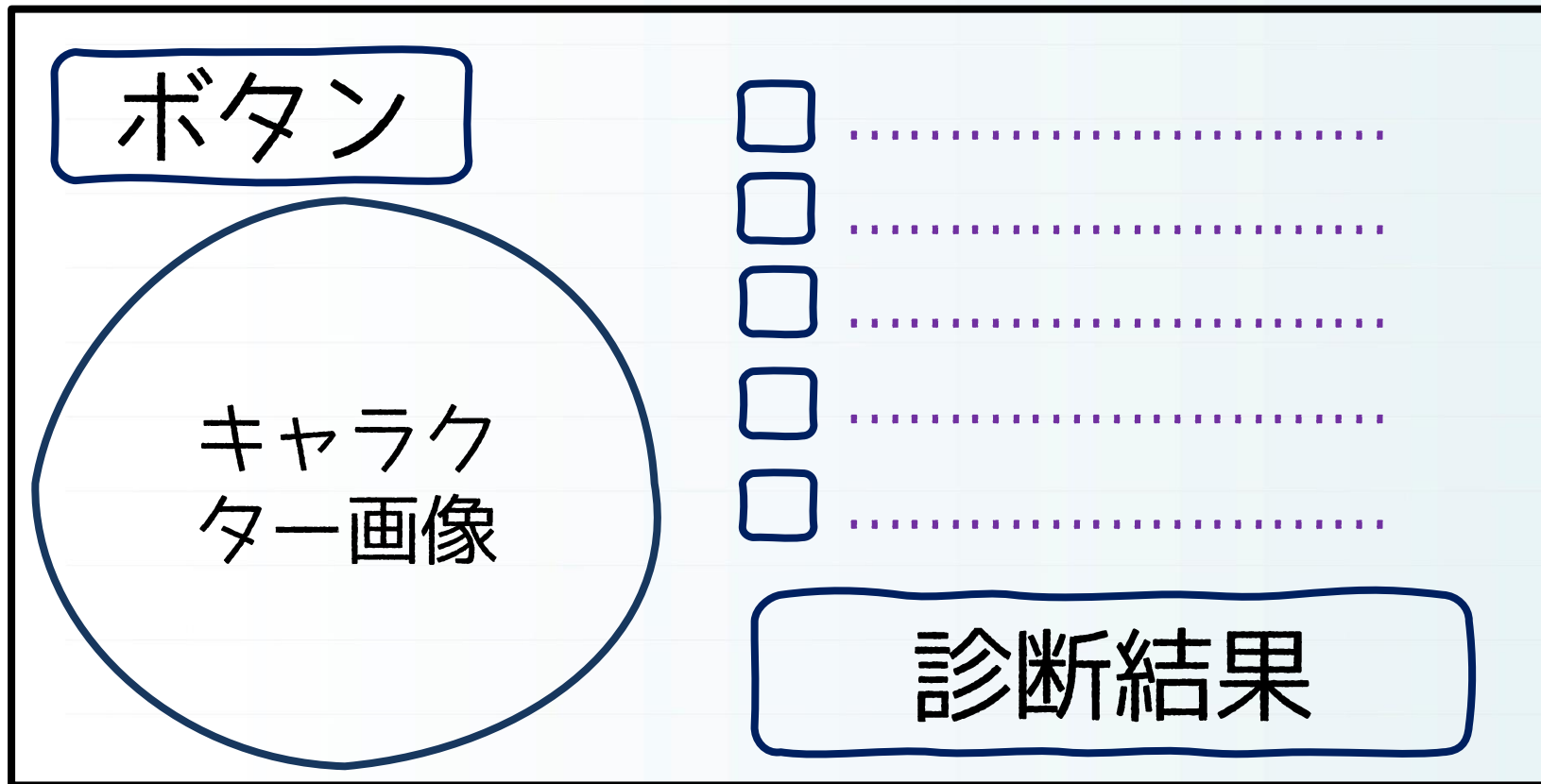
## ネコ度診断ゲーム

質問に当てはまるものをチェックしてチェックした数に応じてネコ度を診断するゲーム



# ネコ度診断ゲームを作る

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。

# ネコ度診断ゲームを作る

## ステップ1：GUIの配置

まずはキャンバスに画像を表示し、チェックボタン以外のGUI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("ネコ度診断ゲーム")
```

. . . . ウィンドウのタイトルに「ネコ度診断ゲーム」と表示させる

```
ウィンドウ.resizable(False,False) . . . . ウィンドウを固定する
```

```
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

※ウィンドウに横幅 800 × 高さ 600のキャンバスを作成する

```
キャンバス.pack() . . . . キャンバスを配置する
```

```
画像 = tkinter.PhotoImage(file = "neko.png")
```

. . . . 変数「画像」にtkinter.PhotoImage()命令を代入する

※画像ファイルに「neko.png」を使用する

次のページに続く

# ネコ度診断ゲームを作る

キャンバス.create\_image(400, 300, image = 画像)

・・・ X座標=400, Y座標=300の位置にキャンバスを配置する

ボタン = tkinter.Button(text = "診断する", font = ("Times New Roman", 32), bg = "orange")

・・・ 変数「ボタン」にtkinter.Button()命令を代入する

※ボタンの名前「診断する」、フォント「Times New Roman」、フォントサイズ「32」、ボタンの背景色「オレンジ」

ボタン.place(x=10, y=10)

・・・ X座標=10, Y座標=10の位置にボタンを配置する

テキスト入力 = tkinter.Text(width = 35, height = 5, font = ("Times New Roman", 16))

・・・ 変数「テキスト入力」にtkinter.Text()命令を代入する

※テキスト入力欄のサイズ「横幅35、縦幅5」、フォント「Times New Roman」、フォントサイズ「16」

テキスト入力.place(x=375, y=400)

・・・ X座標=375, Y座標=400の位置にテキスト入力欄を配置する

ウィンドウ.mainloop() ・・・ ウィンドウを表示する



# GUIの基礎②

単語帳 ～英語も一緒に覚えてしまおう！～

import:取り込む、title:題名、resize:サイズ変更、  
cncvas:キャンバス、photo:写真、image:イメージ、file:ファイル、  
create:創造する、button:ボタン、txt:テキスト、font : フォント  
、place:場所、width:横、height:縦

「Run Module」またはF5キーを押してプログラムを実行する。



# ネコ度診断ゲームを作る

ステップ2：複数のチェックボタンの配置  
次は複数のチェックボタンをfor文で配置する。

```
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("ネコ度診断ゲーム") . . . . ウィンドウのタイトルに「ネコ度診断ゲーム」と表示させる
ウィンドウ.resizable(False,False) . . . . ウィンドウを固定する
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
※ウィンドウに横幅 800 × 高さ 600のキャンバスを作成する
キャンバス.pack() . . . . キャンバスを配置する
画像 = tkinter.PhotoImage(file = "neko.png") . 変数「画像」にtkinter.PhotoImage()命令を代入する
. . . . ※画像ファイルに「neko.png」を使用する
キャンバス.create_image(400, 300,image =画像)
. . . . X座標=400,Y座標=300の位置にキャンバスを配置する
ボタン = tkinter.Button(text = "診断する",font =("Times New Roman",32),bg = "orange")
. . . . 変数「ボタン」にtkinter.Button()命令を代入する
※ボタンの名前「診断する」、フォント「Times New Roman」、フォントサイズ「32」、ボタンの背景色「オレンジ」
ボタン.place(x=10 ,y=10) . . . . X座標=10,Y座標=10の位置にボタンを配置する
```

次のページに続く

# ネコ度診断ゲームを作る

テキスト入力 = tkinter.Text(width =35,height =5,font =("Times New Roman",16))

・・・変数「テキスト入力」にtkinter.Text()命令を代入する

※テキスト入力欄のサイズ「横幅35、縦幅5」、フォント「Times New Roman」、フォントサイズ「16」

テキスト入力.place(x=375 , y=400) ・・・X座標=375,Y座標=400の位置にテキスト入力欄を配置する

判定 = [None]\*7 ・・・リスト「判定」に空っぽの箱を7つ作成する ※Noneは何もないことを意味している

チェックボックス = [None]\*7 ・・・リスト「チェックボックス」に空っぽの箱を7つ作成する

※Noneは何もないことを意味している

アイテム = [

・・・リスト「アイテム」に以下の値を代入する

"高いところが好き",

「高いところが好き」

"ボールを見ると転がしたくなる",

「ボールを見ると転がしたくなる」

"びっくりすると髪の毛が逆立つ",

「びっくりすると髪の毛が逆立つ」

"ネズミの玩具が気になる",

「ネズミの玩具が気になる」

"匂いに敏感",

「匂いに敏感」

"魚の骨をしゃぶりたくなる",

「魚の骨をしゃぶりたくなる」

"夜、元気になる"

「夜、元気になる」

]

次のページに続く

# ネコ度診断ゲームを作る

for 結果 in range(7): . . . . 変数「結果」の値が0,1,2,3,4,5,6と変化しながら下の処理を繰り返す

```
0000判定[結果] = tkinter.BooleanVar()
```

1つの変数と考える

. . . . 変数「判定[結果]」にtkinter.BooleanVar()命令を代入する

```
0000判定[結果].set(False) . . . . 変数「判定[結果]」の初期値にFalseを代入する
```

```
0000チェックボックス[結果] = tkinter.Checkbutton(text = アイテム[結果] ,
```

```
0000font = (" Time New Roman" ,12) ,variable = 判定[結果] ,bg="orange")
```

. . . . 変数「チェックボックス[結果]」にtkinter.Checkbutton()命令を代入する

```
0000チェックボックス[結果].place(x=400, y=35+52*結果)
```

. . . . X座標=400,Y座標=35+52× 結果 の位置にチェックボックスを配置する

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

for文の中で  
7回繰り返す  
処理

# ネコ度診断ゲームを作る



単語帳 ～英語も一緒に覚えてしまおう！～

import:取り込む、title:題名、resize:サイズ変更

、pack:包み、photo:写真、image:イメージ、file:ファイル、

create:創造する、button:ボタン、place:場所、txt:テキスト

、font:フォント、background:背景、none:どれも～ない、

for:～のために、in:～の中に、range:範囲、

Boolean:ブールの ※ジョージブール(イギリスの数学者)

、variable:変わりやすい、varchar:可変長の文字列 ※varと略されている

# GUIの基礎②

「Run Module」またはF5キーを押してプログラムを実行する。



# for文のおさらい

For文を使えば繰り返し処理を実行することができる。



繰り返し処理とはscratchの「〇回繰り返す」ブロックと同じ

5回繰り返す処理を書くコード（変数を使わない場合）

```
for 変数 in range(5): . . . . 関数を使うときは最後に「:」をつける  
    forward(100) } . . . . 繰り返しの中で行う処理  
    left(90)
```

※forとinの間には変数が入る。変数を使用しない場合でもforとinの間に変数を書く必要がある。

5回繰り返す処理を書くコード（変数を使う場合）

```
for 変数 in range(5):  
    print(変数)
```



```
0  
1  
2  
3  
4  
>>> |
```

# ネコ度診断ゲームを作る

ステップ3：チェックされたボタンを数える  
チェックされている項目を数える処理を追加しよう。

```
import tkinter . . . . tkinterモジュールの呼出し
def クリックボタン(): . . . . クリックボタン関数を定義する
    カウント = 0 . . . . 変数「カウント」に0を代入する
    for 結果 in range(7): . . . . 変数「結果」の値が0,1,2,3,4,5,6と変化しながら下の処理を繰り返す
        if 判定[結果].get() == True:
            . . . . 変数「判定[結果]」の値を取得し、Trueの場合、下の処理に進む
            カウント = カウント + 1 . . . . 変数「カウント」にカウント+1を代入する
        テキスト入力.delete("1.0" , tkinter.END) . . . . 入力欄の文字を削除する
        ※delete(" 1.0" ,tkinter.END)は1行目の0文字目から最後まで消すという意味
        テキスト入力.insert("1.0" ,"チェックの数は" + str(カウント))
            . . . . 入力欄に変数を代入
            どの位置に文字を挿入するか  どんな文字を挿入するか
    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("ネコ度診断ゲーム") . . . . ウィンドウのタイトルに「ネコ度診断ゲーム」と表示させる
    ウィンドウ.resizable(False,False) . . . . ウィンドウを固定する
```

for文の中で  
7回繰り返す  
処理

次のページに続く



# ネコ度診断ゲームを作る

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

※ウィンドウに横幅 800 × 高さ 600のキャンバスを作成する

キャンバス.pack() .....キャンバスを配置する

画像 = tkinter.PhotoImage(file = "neko.png")

.....変数「画像」にtkinter.PhotoImage()命令を代入する

※画像ファイルに「neko.png」を使用する

キャンバス.create\_image(400, 300,image =画像)

.....X座標=400,Y座標=300の位置にキャンバスを配置する

※キャンバスに設置する画像は変数「画像で」定義したファイルを使用する

ボタン = tkinter.Button(text = "診断する",font =("Times New Roman",32),bg = "orange",command = クリックボタン) .....変数「ボタン」にtkinter.Button()命令を代入する

※ボタンの名前「診断する」、フォント「Times New Roman」、フォントサイズ「32」、ボタンの背景色「オレンジ」

※command = クリックボタン はボタンをクリックしたときに「クリックボタン」関数を呼び出す

ボタン.place(x=10 ,y=10) ..... X座標=10,Y座標=10の位置にボタンを配置する

テキスト入力 = tkinter.Text(width =35,height =5,font =("Times New Roman",16))

.....変数「テキスト入力」にtkinter.Text()命令を代入する

※テキスト入力欄のサイズ「横幅35、縦幅5」、フォント「Times New Roman」、フォントサイズ「16」

テキスト入力.place(x=375 , y=400) ..... X座標=375,Y座標=400の位置にテキスト入力欄を配置する

次のページに続く

# ネコ度診断ゲームを作る

判定 = [None]\*7 . . . リスト「判定」に空っぽの箱を7つ作成する

※Noneは何もないことを意味している

チェックボックス = [None]\*7 . . . リスト「チェックボックス」に空っぽの箱を7つ作成する

※Noneは何もないことを意味している

アイテム = [ . . . . . リスト「アイテム」に以下の値を代入する

"高いところが好き",

「高いところが好き」

"ボールを見ると転がしたくなる",

「ボールを見ると転がしたくなる」

"びっくりすると髪の毛が逆立つ",

「びっくりすると髪の毛が逆立つ」

"ネズミの玩具が気になる",

「ネズミの玩具が気になる」

"匂いに敏感",

「匂いに敏感」

"魚の骨をしゃぶりたくなる",

「魚の骨をしゃぶりたくなる」

"夜、元気になる"

「夜、元気になる」

]

次のページに続く

# ネコ度診断ゲームを作る

for 結果 in range(7): . . . . 変数「結果」の値が0,1,2,3,4,5,6と変化しながら下の処理を繰り返す

```
0000判定[結果] = tkinter.BooleanVar()
```

1つの変数と考える

. . . . 変数「判定[結果]」にtkinter.BooleanVar()命令を代入する

```
0000判定[結果].set(False) . . . . 変数「判定[結果]」の初期値にFalseを代入する
```

```
0000チェックボックス[結果] = tkinter.Checkbutton(text = アイテム[結果] ,
```

```
0000font = (" Time New Roman" ,12) ,variable = 判定[結果] ,bg="orange")
```

. . . . 変数「チェックボックス[結果]」にtkinter.Checkbutton()命令を代入する

```
0000チェックボックス[結果].place(x=400, y=35+52*結果)
```

. . . . X座標=400,Y座標=35+52× 結果 の位置にチェックボックスを配置する

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

for文の中で  
7回繰り返す  
処理

# ネコ度診断ゲームを作る

単語帳 ～英語も一緒に覚えてしまおう！～

import:取り込む、title:題名、resize:サイズ変更

、pack:包み、photo:写真、image:イメージ、file:ファイル、

create:創造する、button:ボタン、place:場所、txt:テキスト

、font:フォント、background:背景、none:どれも～ない、

for:～のために、in:～の中に、range:範囲、get:手に入れる

Boolean:ブールの ※ジョージブール(イギリスの数学者)

、variable:変わりやすい、varchar:可変長の文字列 ※varと略されている

delete:削除する、insert:挿入する、command:命令する



# GUIの基礎②

「Run Module」またはF5キーを押してプログラムを実行する。



# ネコ度診断ゲームを作る

ステップ4：コメントを出力する  
ボタンを押すとチェックした数に応じたコメントを出力させる。

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

```
診断結果 = [ . . . .リスト「診断結果」に以下の値を代入する
```

```
" 前世がネコだった可能性は極めて薄いです。",      「前世がネコだった可能性は極めて薄いです。」
```

```
" いたって普通の人間です。",                          「いたって普通の人間です。」
```

```
" 特別、おかしなところはありません。",              「特別、おかしなところはありません。」
```

```
" やや、ネコっぽいところがあります。",              「やや、ネコっぽいところがあります。」
```

```
" ネコに近い性格のようです。",                        「ネコに近い性格のようです。」
```

```
" ネコにかなり近い性格です。",                        「ネコにかなり近い性格です。」
```

```
" 前世はネコだったかもしれません。",                「前世はネコだったかもしれません。」
```

```
" 見た目は人間、中身はネコの可能性があります。"    「前世はネコだったかもしれません。」
```

```
]                                                       「見た目は人間、中身はネコの可能性があります。」
```

次のページに続く



# ネコ度診断ゲームを作る

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

※ウィンドウに横幅 800 × 高さ 600のキャンバスを作成する

キャンバス.pack() .....キャンバスを配置する

画像 = tkinter.PhotoImage(file = "neko.png")

.....変数「画像」にtkinter.PhotoImage()命令を代入する

※画像ファイルに「neko.png」を使用する

キャンバス.create\_image(400, 300,image =画像)

.....X座標=400,Y座標=300の位置にキャンバスを配置する

※キャンバスに設置する画像は変数「画像で」定義したファイルを使用する

ボタン = tkinter.Button(text = "診断する",font =("Times New Roman",32),bg = "orange",command = クリックボタン) .....変数「ボタン」にtkinter.Button()命令を代入する

※ボタンの名前「診断する」、フォント「Times New Roman」、フォントサイズ「32」、ボタンの背景色「オレンジ」

※command = クリックボタン はボタンをクリックしたときに「クリックボタン」関数を呼び出す

ボタン.place(x=10 ,y=10) ..... X座標=10,Y座標=10の位置にボタンを配置する

テキスト入力 = tkinter.Text(width =35,height =5,font =("Times New Roman",16))

.....変数「テキスト入力」にtkinter.Text()命令を代入する

※テキスト入力欄のサイズ「横幅35、縦幅5」、フォント「Times New Roman」、フォントサイズ「16」

テキスト入力.place(x=375 , y=400) ..... X座標=375,Y座標=400の位置にテキスト入力欄を配置する

次のページに続く



# ネコ度診断ゲームを作る

判定 = [None]\*7 . . . リスト「判定」に空っぽの箱を7つ作成する

※Noneは何もないことを意味している

チェックボックス = [None]\*7 . . . リスト「チェックボックス」に空っぽの箱を7つ作成する

※Noneは何もないことを意味している

アイテム = [ . . . . . リスト「アイテム」に以下の値を代入する

"高いところが好き",

「高いところが好き」

"ボールを見ると転がしたくなる",

「ボールを見ると転がしたくなる」

"びっくりすると髪の毛が逆立つ",

「びっくりすると髪の毛が逆立つ」

"ネズミの玩具が気になる",

「ネズミの玩具が気になる」

"匂いに敏感",

「匂いに敏感」

"魚の骨をしゃぶりたくなる",

「魚の骨をしゃぶりたくなる」

"夜、元気になる"

「夜、元気になる」

]

次のページに続く

# ネコ度診断ゲームを作る

for 結果 in range(7): . . . . 変数「結果」の値が0,1,2,3,4,5,6と変化しながら下の処理を繰り返す

```
0000判定[結果] = tkinter.BooleanVar()
```

1つの変数と考える

. . . . 変数「判定[結果]」にtkinter.BooleanVar()命令を代入する

```
0000判定[結果].set(False) . . . . 変数「判定[結果]」の初期値にFalseを代入する
```

```
0000チェックボックス[結果] = tkinter.Checkbutton(text = アイテム[結果] ,
```

```
0000font = (" Time New Roman" ,12) ,variable = 判定[結果] ,bg="orange")
```

. . . . 変数「チェックボックス[結果]」にtkinter.Checkbutton()命令を代入する

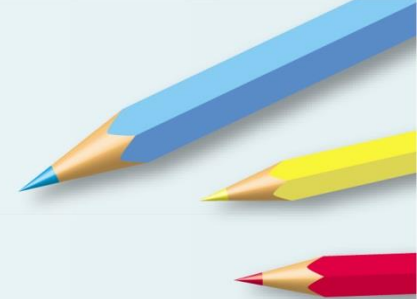
```
0000チェックボックス[結果].place(x=400, y=35+52*結果)
```

. . . . X座標=400,Y座標=35+52× 結果 の位置にチェックボックスを配置する

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

for文の中で  
7回繰り返す  
処理

# ネコ度診断ゲームを作る



単語帳 ～英語も一緒に覚えてしまおう！～

import:取り込む、title:題名、resize:サイズ変更

、pack:包み、photo:写真、image:イメージ、file:ファイル、

create:創造する、button:ボタン、place:場所、txt:テキスト

、font:フォント、background:背景、none:どれも～ない、

for:～のために、in:～の中に、range:範囲、get:手に入れる

Boolean:ブールの ※ジョージブール(イギリスの数学者)

、variable:変わりやすい、varchar:可変長の文字列 ※varと略されている

delete:削除する、insert:挿入する、command:命令する

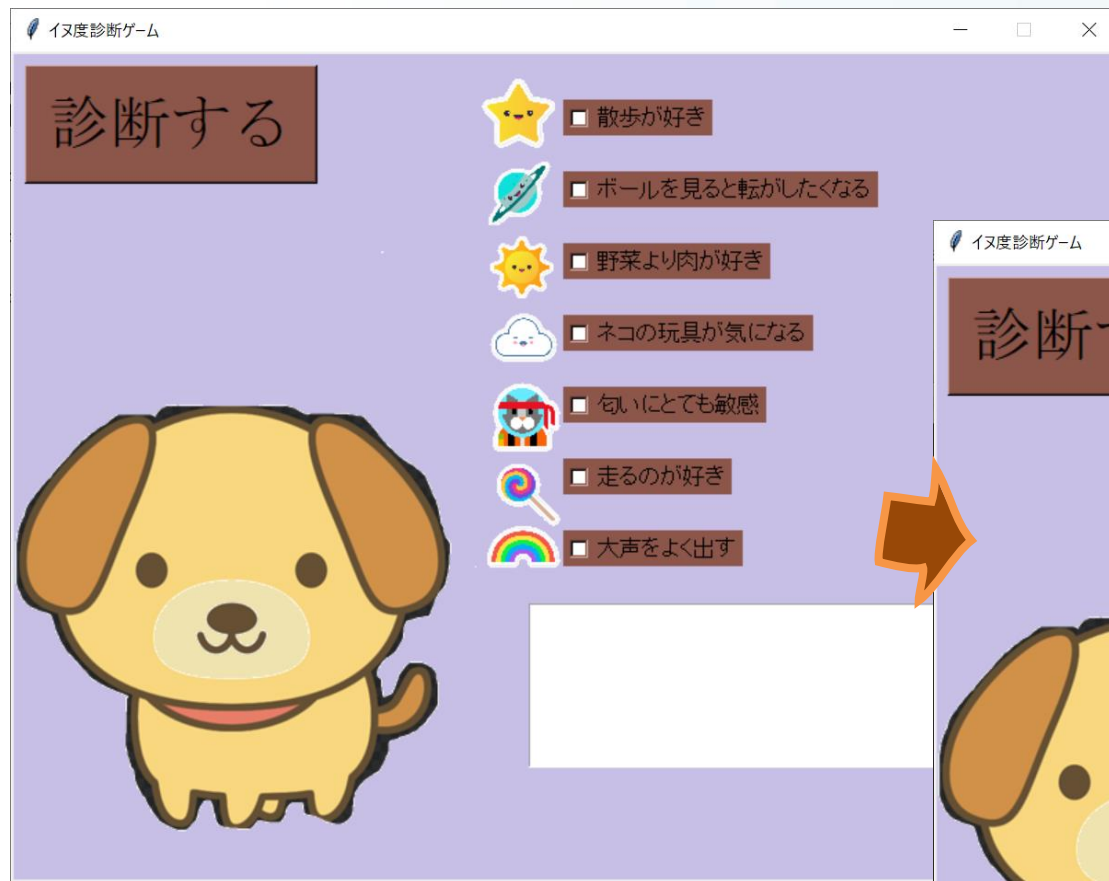
# GUIの基礎②

「Run Module」またはF5キーを押してプログラムを実行する。



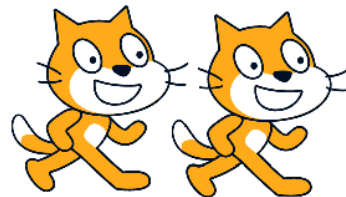
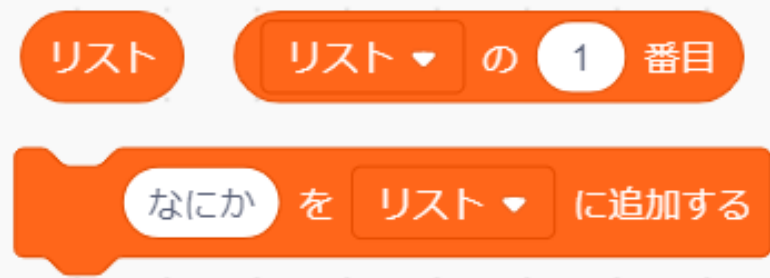
# 復習 & チャレンジ

ここまで習ったこと知識を利用して「イヌ度診断ゲーム」を作ってみよう。画像も自分で用意しよう。



# 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：