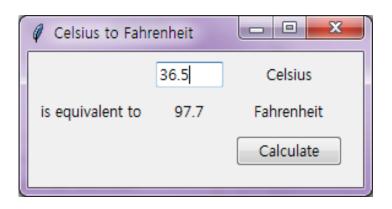
Real World Practices of Tkinter



Practice 1: C-F converter [1/2]



```
from tkinter import *
from tkinter import ttk

def calculate(*args):
    try:
       value = float(celsius.get())
       fahrenheit.set(value * 9 / 5 + 32)
    except ValueError:
       pass

root = Tk()
root.title("Celsius to Fahrenheit")
```



Practice 1: C-F converter [2/2]

root.bind('<Return>', calculate)

```
Celsius to Fahrenheit
#ttk.Frame()
mainframe = ttk.Frame(root, padding="3 3 12 12")
                                                                          36.5
                                                                                      Celsius
mainframe.grid(column=0, row=0)
                                                              is equivalent to
                                                                            97.7
                                                                                    Fahrenheit
celsius = StringVar()
                                                                                     Calculate
fahrenheit = StringVar()
#ttk.Entry
cel_entry = ttk.Entry(mainframe, width=7, textvariable=celsius)
cel entry.grid(column=2, row=1)
ttk.Label(mainframe, text="Celsius").grid(column=3, row=1)
ttk.Label(mainframe, text="is equivalent to").grid(column=1, row=2)
ttk.Label(mainframe, textvariable=fahrenheit).grid(column=2, row=2)
ttk.Label(mainframe, text="Fahrenheit").grid(column=3, row=2)
ttk.Button(mainframe, text="Calculate", command=calculate).grid(column=3, row=3)
for child in mainframe.winfo children():
   child.grid configure(padx=5, pady=5)
#entry and run calculate
cel entry.focus()
```

3/65

Practice 2 [1/2]

```
from tkinter import *
from tkinter import ttk
from tkinter.messagebox import *
#Frame
frame = ttk.Frame()
frame['padding'] = (5,10)
frame['borderwidth'] = 2
frame['relief']='sunken'
frame.pack()
#Label
label = ttk.Label(text='Full name;')
resultsContents = StringVar()
label['textvariable'] = resultsContents
resultsContents.set('New value to display')
label.pack()
#image
image = PhotoImage(file='gif.gif')
label['image'] = image
label.pack()
```

button = ttk.Button(text='자기소개', command=hello)

showinfo('자기소개', '안녕하세요~ {0}입니다'.format(name.get()))

4/65

def hello():

#button

button.pack()





Practice 2 [2/2]

```
#Entry
username = StringVar()
name = ttk.Entry(textvariable=username)
name.insert(0, "이름")
name.pack(pady = 5)
#radiobutton
location = StringVar()
home = ttk.Radiobutton(text='집', variable=location, value='집')
school = ttk.Radiobutton(text='학교', variable=location, value='학교')
library = ttk.Radiobutton(text='도서관', variable=location, value='도서관')
home.pack()
school.pack()
library.pack()
#combobox
countryvar = StringVar()
country = ttk.Combobox(textvariable=countryvar)
country.bind('<<ComboboxSelected>>')
country.set("거주 지역")
country['values'] = ('서울', '경기', '경상', '전라', '충청', '강원', '제주')
country.pack()
```



Practice 3 [1/6]

- 회원 등록을 위해 아래의 기능을 포함하는 화면을 구성한다
 - 이름을 입력한다
 - 성별을 선택한다
 - 생년월일을 선택한다
 - 전화번호를 입력한다
 - 이메일을 입력한다
 - 관심 언어를 최대 3개까지 선택한다





Practice 3 [2/6]

```
from tkinter import * from tkinter import ttk
```

```
root = Tk()
root.title("회원 등록")
```

```
main = Frame(root, width=400, height=400, bd=1, relief=SUNKEN)
main.pack(fill=X, padx=5, pady=5)
#이름
name label = Label(main, text="이름")
name_label.grid(row=1, column=1, padx = 5, pady = 5, sticky=W)
name_entry = Entry(main, bd = 3, width = 10)
name entry.grid(row=1, column=2, padx = 5, pady = 5, sticky=W)
#성별
gender_label = Label(main, text="성별")
gender label.grid(row=2, column=1, padx = 5, pady = 5, sticky=W)
#성별 입력
gender frame = Frame(main)
gender_frame.grid(row=2, column=2, padx = 5, pady = 5, sticky=W)
varG = IntVar()
gender_rb1 = Radiobutton(gender_frame, text="남성", variable=varG, value=1)
gender_rb1.grid(row=1, column=1, padx = 5)
```



gender rb2 = Radiobutton(gender frame, text="여성", variable=varG, value=2)

 $gender_rb2.grid(row=1, column=2, padx = 5)$

```
∅ 회원 등록
 Practice 3 [3/6]
                                                                               이름
#생년월일
birth_label = Label(main, text="생년월일")
birth label.grid(row=3, column=1, padx = 5, pady = 5, sticky=W)
                                                                               전화번호 010 ▼ - 2222 - 1111
#생년월일 입력
                                                                               이메일

    □ Python

birth frame = Frame(main)

▼ C#
                                                                               관심언어
birth frame.grid(row=3, column=2, padx = 5, pady = 5, sticky=W)

▼ PHP

                                                                                      최대 3개만 선택 가능합니다.
varY = IntVar()
birth combo1 = ttk.Combobox(birth frame, textvariable=varY, state='readonly', width=4)
birth combo1['values'] = list(range(2015, 1899, -1))
birth combo1.set("년")
birth_combo1.grid(row=1, column=1)
year_label = Label(birth_frame, text="년")
vear label.grid(row=1, column=2)
varM = IntVar()
birth combo2 = ttk.Combobox(birth frame, textvariable=varM, state='readonly', width=2)
birth combo2['values'] = list(range(1, 13))
birth_combo2.set("월")
birth combo2.grid(row=1, column=3)
month label = Label(birth frame, text="월")
month label.grid(row=1, column=4)
varD = IntVar()
birth combo3 = ttk.Combobox(birth frame, textvariable=varD, state='readonly', width=2)
birth combo3['values'] = list(range(1, 32))
birth combo3.set("일")
birth_combo3.grid(row=1, column=5)
day_label = Label(birth_frame, text="일")
                                             8/65
day label.grid(row=1, column=6)
```

_ D X

Practice 3 [4/6]

phone label3 = Label(phone frame, text="-")

phone entry2 = Entry(phone frame, bd = 3, width=6)

phone_label3.grid(row=1, column=4)

phone_entry2.grid(row=1, column=5)

```
#전화번호
phone_label1 = Label(main, text="전화번호")
phone_label1.grid(row=4, column=1, padx = 5, pady = 5, sticky=W)
#전화번호 입력
phone_frame = Frame(main)
phone frame.grid(row=4, column=2, padx = 5, pady = 5, sticky=W)
varP = IntVar()
phone_combo = ttk.Combobox(phone_frame, textvariable=varP, state='readonly', width=3)
phone combo['values'] = ['010', '02', '031', '032', '033', '041', '042', '043', '044', '051', '052',
'053', '054', '055', '061', '062', '063', '064']
phone combo.current(0)
phone combo.grid(row=1, column=1)
                                                                                  회원 등록
phone_label2 = Label(phone_frame, text="-")
phone label2.grid(row=1, column=2)
phone_entry1 = Entry(phone_frame, bd = 3, width=6)
                                                                            phone_entry1.grid(row=1, column=3)
```

Practice 3 [5/6]

```
#이메일
mail label1 = Label(main, text="이메일")
mail_label1.grid(row=5, column=1, padx = 5, pady = 5, sticky=W)
                                                                            이메일
#이메일 입력
mail frame = Frame(main)
                                                                            관심언어
mail frame.grid(row=5, column=2, padx = 5, pady = 5, sticky=W)
mail entry1 = Entry(mail frame, bd = 3, width=10)
mail_entry1.grid(row=1, column=1)
mail_label2 = Label(mail_frame, text="@")
mail label2.grid(row=1, column=2)
mail_entry2 = Entry(mail_frame, bd = 3, width=10)
mail_entry2.grid(row=1, column=3)
#관심언어
interest label1 = Label(main, text="관심언어")
interest_label1.grid(row=6, column=1, padx = 5, pady = 5, sticky=W)
#관심언어 입력
interest frame = Frame(main)
interest_frame.grid(row=6, column=2, padx = 5, sticky=W)
def check():
  if varCB1.get()+varCB2.get()+varCB3.get()+varCB4.get()+varCB5.get()+varCB6.get()>3:
     varCB1.set(0)
     varCB2.set(0)
     varCB3.set(0)
     varCB4.set(0)
     varCB5.set(0)
```



Practice 3 [6/6]

```
이름 이름 성별 ⓒ 남성 ⓒ 여성 생년월일 1991 ▼ 년 7 ▼ 월 3 ▼ 일 전화번호 010 ▼ - 2222 - 1111 이메일 ② □ □ C □ Python 관심언어 □ Java ▼ C# □ C++ ▼ PHP 최대 3개만 선택 가능합니다. 다시 하세요.
```

```
varCB1 = IntVar()
varCB2 = IntVar()
varCB3 = IntVar()
varCB4 = IntVar()
varCB5 = IntVar()
varCB6 = IntVar()
CB1 = Checkbutton(interest_frame, text="C", variable=varCB1, onvalue=1, offvalue=0, command=check)
CB1.grid(row=1, column=1, sticky=W)
CB2 = Checkbutton(interest_frame, text="Python", variable=varCB2, onvalue=1, offvalue=0, command=check)
CB2.grid(row=1, column=2, padx= 20, sticky=W)
CB3 = Checkbutton(interest_frame, text="Java", variable=varCB3, onvalue=1, offvalue=0, command=check)
CB3.grid(row=2, column=1, sticky=W)
CB4 = Checkbutton(interest frame, text="C#", variable=varCB4, onvalue=1, offvalue=0, command=check)
CB4.grid(row=2, column=2, padx= 20, sticky=W)
CB5 = Checkbutton(interest frame, text="C++", variable=varCB5, onvalue=1, offvalue=0, command=check)
CB5.grid(row=3, column=1, sticky=W)
```

interest_label2 = Label(main)
interest_label2.grid(row=7, column=2, padx = 5, pady = 5)

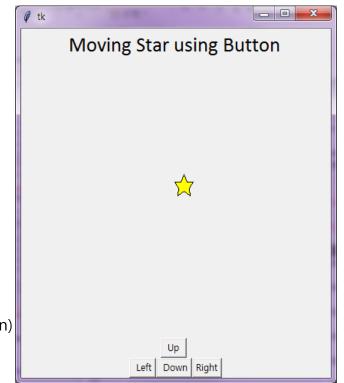
CB6.grid(row=3, column=2, padx= 20, sticky=W)

root.mainloop()

CB6 = Checkbutton(interest frame, text="PHP", variable=varCB6, onvalue=1, offvalue=0, command=check)

Practice 4: Moving picture on clicking arrow

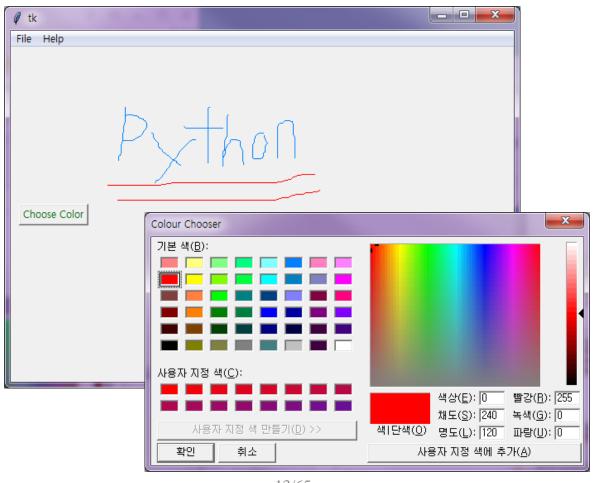
```
from tkinter import *
tk = Tk()
canvas = Canvas(tk, width=400, height=400)
canvas.pack()
canvas.create text(200,20, font=('Calibri', 20), text='Moving Star using Button')
fill='yellow', outline='black')
def keyup():
  canvas.move(star,0,-5)
  canvas.itemconfig(star, fill='yellow')
def keydown():
  canvas.move(star,0,5)
  canvas.itemconfig(star, fill='pink')
def keyleft():
  canvas.move(star,-5, 0)
  canvas.itemconfig(star, fill='lightblue')
def keyright():
  canvas.move(star,5,0)
  canvas.itemconfig(star, fill='white')
btn frame = Frame(tk)
btn frame.pack()
btn_up = Button(btn_frame, text=" Up ", command=keyup)
btn down = Button(btn frame, text=" Down", command=keydown)
btn left = Button(btn frame, text=" Left", command=keyleft)
btn right = Button(btn frame, text="Right", command=keyright)
btn up.grid(row=1, column=2)
btn_left.grid(row=2, column=1)
btn down.grid(row=2, column=2)
btn right.grid(row=2, column=3)
```





Practice 5: drawing board [1/3]

- 색상 선택 가능
- 파일 열기, 저장





Practice 5: Drawing Board [2/3]

```
from tkinter import *
from tkinter.colorchooser import *
def NewFile() :
  print("New File!")
def OpenFile() :
   print("Open File!")
def About() :
  print("This is a simple example of a menu")
#menu
root = Tk()
menu = Menu(root)
root.config(menu=menu)
filemenu = Menu(menu)
menu.add_cascade(label="File", menu=filemenu)
filemenu.add_command(label="New", command=NewFile)
filemenu.add_command(label="Open...", command=OpenFile)
filemenu.add_separator()
filemenu.add_command(label="Exit", command=root.quit)
helpmenu = Menu(menu)
menu.add_cascade(label="Help", menu=helpmenu)
helpmenu.add_command(label="About...", command=About)
#canvas
canvas = Canvas(root, width=1000, height=500)
color = "red"
result = '#476042'
```



Practice 5: Drawing Board [3/3]

```
def callback() :
   global result
   result = askcolor(title = "Colour Chooser")
   result = result[1]
button = Button(root, text='Choose Color', fg="darkgreen", command=callback)
button.pack(side=LEFT, padx=10)
lastx, lasty = 0, 0
def xy(event):
   global lastx, lasty
   lastx, lasty = event.x, event.y
def addLine(event) :
   global lastx, lasty
   canvas.create_line((lastx, lasty, event.x, event.y), fill=result)
   lastx, lasty = event.x, event.y
root.columnconfigure(0, weight=1)
root.rowconfigure(0, weight=1)
canvas.pack()
canvas.bind("<Button-1>", xy)
canvas.bind("<B1-Motion>", addLine)
root.mainloop()
```

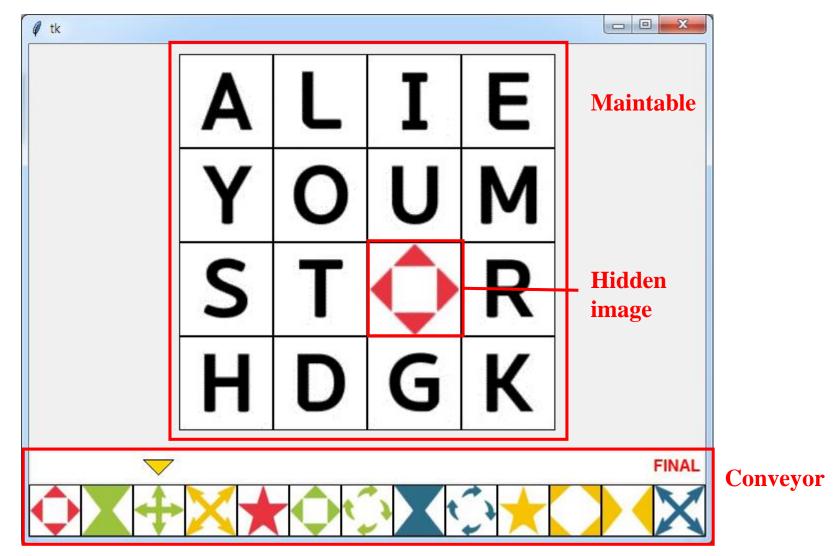


Team Project 6 (Deadline: June 15 Noon):

- •Tkinter를 이용하여 같은 그림 찾기
- 목표
 - 같은 그림 찾기 프로그램을 구현하라
- 요구 사항
 - Python 3 사용
 - tkinter 모듈 사용
- 비고
 - 뼈대 코드 제공
 - 이미지 제공
 - 원하는 library, module, method 및 parameter 추가 가능
- 같은 그림 찾기 설명
 - 후면 참조

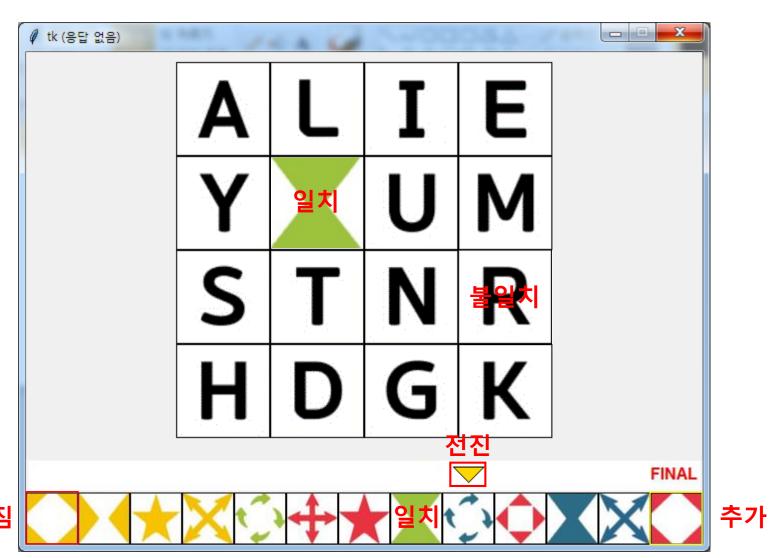


같은 그림 찾기 설명 - GUI 예제





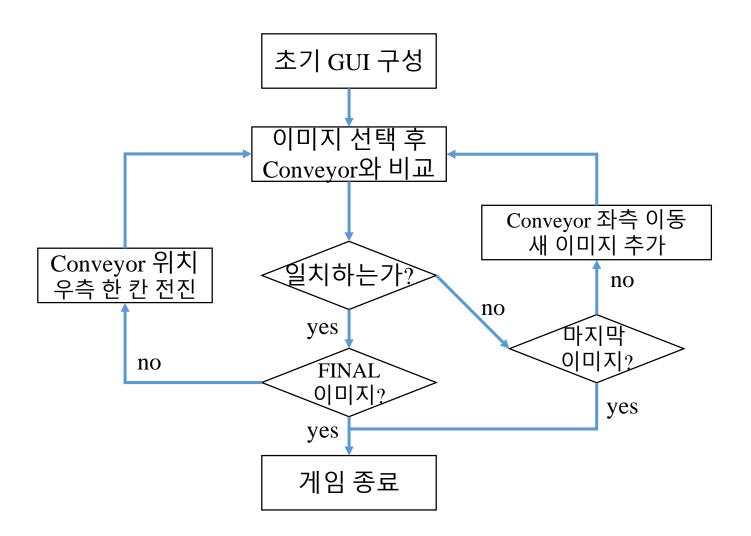
같은 그림 찾기 설명 _ 실행 예제



버려짐



게임 실행 순서

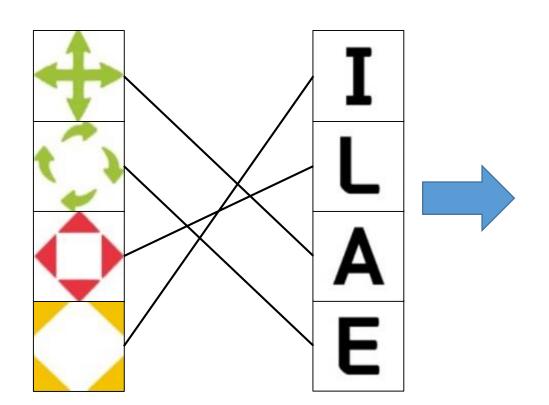


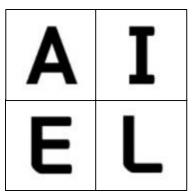


프로젝트 세부사항

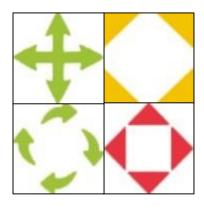
[1/3]

- Maintable 구성하기
 - 4x4 크기로 구성
 - 앞면 이미지들과 뒷면 이미지들간 랜덤 매칭
 - 앞면 이미지 클릭 시 뒷면 이미지 3초간 공개





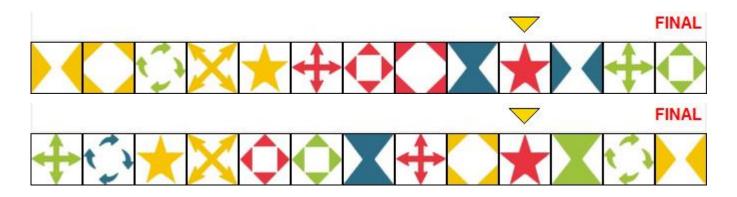
<앞>





프로젝트 세부사항 [2/3]

- Conveyor belt 구성
 - 그림이미지 shuffle
 - 총 이미지 16개 중 임의로 13개 선택해 구성
 - 시작 위치 10번째 이미지
 - 이미지 중복 X
 - 새로운 이미지 추가 시 기존 이미지 제외하고 추가
 - 화살표가 FINAL에 도착한 후 일치하는 도형을 찾으면 게임 종료
 - 화살표가 좌측 끝에 도착한 후 일치하는 도형을 찾지 못하면 게임 종료





프로젝트 세부사항 [3/3]

- 사용 모듈 설명
 - tkinter
 - Python의 GUI 구성을 위한 모듈
 - Frame, Button, Label 등 다양한 widget 포함
 - random
 - Random number 생성을 위한 모듈
 - PIL
 - 이미지 데이터 처리를 위한 모듈
 - JPG 파일을 읽어오기 위해 필요
 - Time
 - Time interval을 위한 모듈

