

본 콘텐츠는 과학기술정보통신부정보통신기획평가원의
SW중심대학지원사업단의 연구결과로 수행되었습니다.
(2019 - 0 - 01838)



과학기술정보통신부



정보통신기획평가원



배재대학교 AI·SW중심대학사업단

AI·SW중심대학사업단

파이썬으로 AI게임 만들기

tkinter (GUI) 기초

이태준

배재대학교 컴퓨터공학과 박사과정

목차

1. 사라진 문자 찾기 게임
2. tkinter 기초 - 라벨과 버튼
3. tkinter 기초 - 캔버스
4. tkinter 기초 - 뿔기 프로그램
5. tkinter 기초 - 텍스트 입력 필드
6. tkinter 기초 - 체크 버튼과 메시지 박스

1. 사라진 문자 찾기 게임

- 타임 어택(time attack) 게임
 - 시간을 다투는 게임 장르의 예시: 레이싱 게임



<https://www.motorgraph.com/news/articleView.html?idxno=27140>



<https://www.ea.com/ko-kr/games/need-for-speed>

- 퍼즐, 틀린그림 찾기 등 얼마나 빠르게 게임을 끝낼 수 있는지 판정하는 게임이 있음
- 우리가 만들 게임: 알파벳 중에서 빠진 문자를 찾는 게임

1. 사라진 문자 찾기 게임

- 1단계: 사라진 알파벳 만들기

```
alpha = ['A', 'B', 'C', 'D', 'E', 'F', 'G']
```

```
for i in alpha:
    print(i)
```

```
import random    # or from random import choice
```

```
alpha = ['A', 'B', 'C', 'D', 'E', 'F', 'G']
```

```
answer = random.choice(alpha)    # choice(alpha)
```

```
question = ''    # 처음에 아무것도 없는 빈 문자열 생성
```

```
for i in alpha:
```

```
    if i != answer:    # 정답이 아니면 (빠진 문자를 빼고 넣기)
```

```
        question += i    # question = question + i
```

```
print(question)
```


1. 사라진 문자 찾기 게임

- 2단계: 문자 찾기 판정

```
import random    # or from random import choice

alpha = ['A', 'B', 'C', 'D', 'E', 'F', 'G']
answer = random.choice(alpha)    # choice(alpha)
question = ''    # 처음에 아무것도 없는 빈 문자열 생성

for i in alpha:
    if i != answer:    # 정답이 아니면 (빠진 문자를 빼고 넣기)
        question += i    # question = question + i

print(question)

my = input('빠진 알파벳은?')
if answer == my:
    print('정답!')
else:
    print('오답!')
```

1. 사라진 문자 찾기 게임

- 3단계: 시간 측정하기

```
import random    # or from random import choice
import datetime

# 이 부분은 이전 코드와 같음

start = datetime.datetime.now() # 현재 시간 측정
my = input('빠진 알파벳은?')
if answer == my:
    print('정답!')
    end = datetime.datetime.now() # 이 때 현재 시간 측정
    print((end - start).seconds)  # end - start를 빼서 초로 바꾸기
else:
    print('오답!')
```

1. 사라진 문자 찾기 게임

- 예제 1. "A ~ G"까지에서 "A ~ Z"까지로 변경하기
 - 모듈을 사용할 수도 있음
- 예제 2. 초를 출력하는 부분에서 아래와 같이 변경하기
출력 예시: 5초 걸렸습니다.

목차

1. 사라진 문자 찾기 게임
2. **tkinter 기초 - 라벨과 버튼**
3. tkinter 기초 - 캔버스
4. tkinter 기초 - 뿔기 프로그램
5. tkinter 기초 - 텍스트 입력 필드
6. tkinter 기초 - 체크 버튼과 메시지 박스

2. tkinter 기초 - 라벨과 버튼

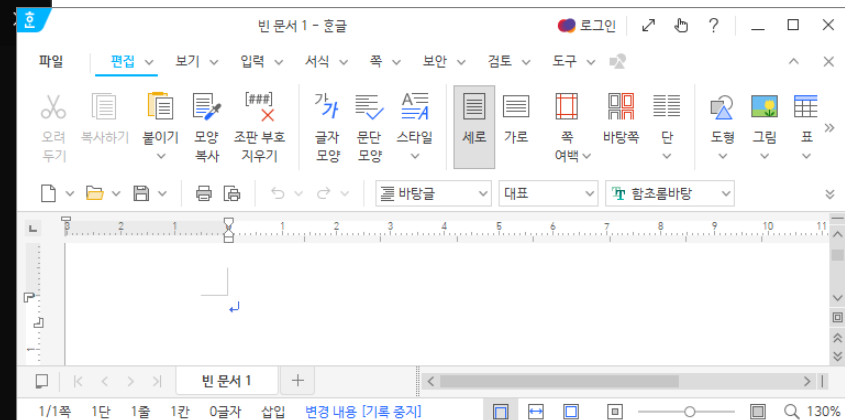
- CUI(Character(or Command Line) User Interface)
vs. GUI(Graphic ~) - UI: 사용자와 시스템의 접점

```
C:\Windows\system32\cmd
Microsoft Windows [Version 10.0.22000.795]
Microsoft Windows [Version 10.0.22000.795]
(c) Microsoft Corporation. All rights reserved.

C:\Users\marlrero>dir
c 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: 6A38-169A

C:\Users\marlrero 디렉터리

2022-07-13 오후 04:57 <DIR> .
2021-10-13 오후 04:13 <DIR> ..
2022-07-13 오후 04:29 <DIR> .cache
2021-10-15 오후 04:21 <DIR> .conda
2022-01-17 오후 01:05      27 .condarc
2022-07-13 오후 04:29 <DIR> .eclipse
2022-05-20 오후 07:22     53 .git-for-windows-updater
2022-05-19 오후 07:22    301 .gitconfig
```



```
Python 3.6.0 Shell
File Edit Shell Debug Options Window Help
Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 07:18:10) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> print('Hello, world!')
Hello, world!
>>>
```



2. tkinter 기초 - 라벨과 버튼

- 게임 개발 과정 (게임 회사)
 1. 기획 수립 (게임 규칙, 설정, 계획 등) → 기획서
 2. 예산, 프로그램 인원 검토 → 개발비 산정
 3. 알파 버전 게임 개발
 4. 베타 버전 게임 개발 (알파 버전보다 더 개선된 버전)
 5. 마스터 버전 게임 개발 (디버그, 완성도)
 6. 릴리즈
- 게임 개발 과정 (개인, 동아리)
 1. 아이디어 도출 및 소재 수집
 2. 프로그래밍
 3. 테스트 플레이 및 동료 의견 교환
 4. 프로그래밍 (개선)
 5. 릴리즈

2. tkinter 기초 - 라벨과 버튼

- 게임 제작에 필요한 것
 - 게임에 필요한 리소스: 그래픽(이미지) 및 사운드 관련
 - 직접 그리기, 무료 리소스 사용, 크리에이터 의뢰 (사운드 크리에이터, 그래픽 디자이너)
 - 프로그래머: 게임 규칙과 리소스를 활용한 개발
 - 디버거: 게임에서 발생하는 버그와 관련된 일 담당
 - 디렉터: 게임 제작 총괄 관리 담당자, 프로그램 구상 작업
 - 크리에이티브 디렉터, 테크니컬 디렉터 등 다양함
 - 기획/프로듀서: 게임의 기획 업무 담당

2. tkinter 기초 - 라벨과 버튼

- 게임 제작 툴, 엔진
 - Unity, Unreal과 같은 엔진을 사용해 개발
 - 게임 엔진을 직접 제작하는 사례도 있음

- 수업 진행 방향
 - tkinter를 활용한 GUI 프로그램 맛보기
 - 간단한 게임이나 프로그램 제작을 통해 맛보기
 - Tkinter로 게임을 제작하는 사례는 많이 없음

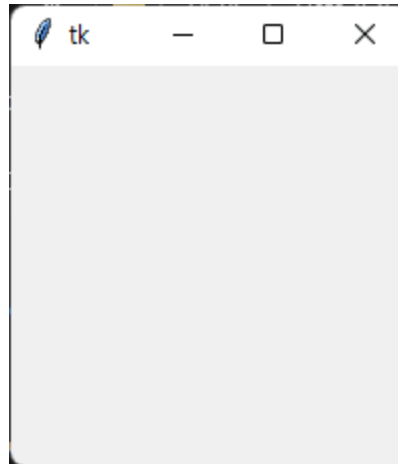
 - Pygame이라는 오픈 소스 엔진을 사용해서 제작
 - 배운 내용을 가지고 (개인 혹은 팀별) 프로젝트 진행

2. tkinter 기초 - 라벨과 버튼

- tkinter: 파이썬에서 제공하는 모듈로, GUI 제작에 사용
 - 윈도우(window): tkinter에서 하나의 창을 의미

```
import tkinter

root = tkinter.Tk()    # 윈도우 객체 생성
root.mainloop()        # 윈도우 객체를 표시하라(띄워라)
```



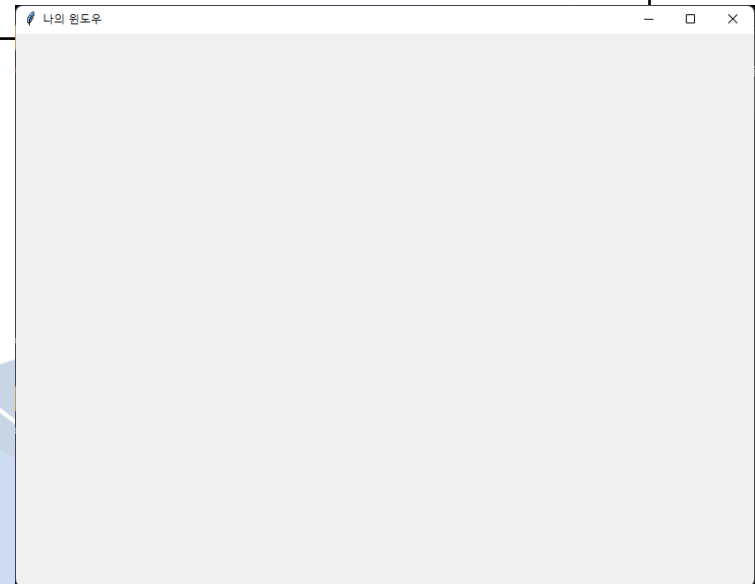
2. tkinter 기초 - 라벨과 버튼

- tkinter: 파이썬에서 제공하는 모듈로, GUI 제작에 사용
 - 윈도우(window)의 제목과 사이즈 지정

```
import tkinter

root = tkinter.Tk()
root.title("나의 윈도우") # root의 제목(title)
root.geometry("800x600") # root의 크기
root.mainloop()
```

- 제목(title)
- 사이즈(geometry): 가로x세로
픽셀(pixel)
- 픽셀: picture + element
→ 화면을 구성하는 기본 단위



2. tkinter 기초 - 라벨과 버튼

- tkinter로 라벨 배치
 - 라벨(label): 문자열을 표시할 때 사용
 - 라벨 변수명 = tkinter.Label(윈도우 객체, text="라벨문자열", font=("폰트명", 폰트크기))
 - {라벨 변수명}.place(x=x좌표, y=y좌표)

```
import tkinter

root = tkinter.Tk()
root.title("나의 윈도우") # root의 제목(title)
root.geometry("800x600") # root의 크기

label = tkinter.Label(root, text="Hello~", font("System", 24))
label.place(x=200, y=100)

root.mainloop()
```

2. tkinter 기초 - 라벨과 버튼

- tkinter로 라벨 배치

- 라벨(label): 문자열을 표시할 때 사용

- tkinter에서 사용 가능한 폰트

- 시스템에 해당 폰트가 없으면 파이썬 기본값으로 출력

- 폰트에 민감하다면, 폰트를 설치할 수 있도록 해야 함

```
import tkinter
import tkinter.font

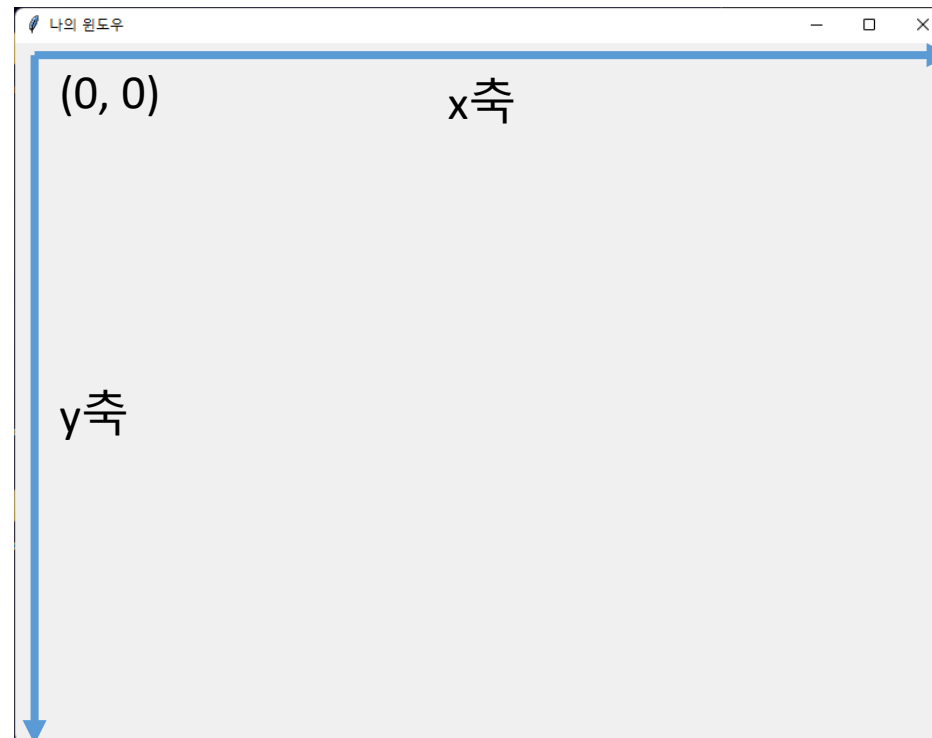
print(tkinter.font.families())
```

- 예제 1. 위에서 출력된 폰트로 출력할 라벨의 폰트 바꾸기

- 예제 2. 폰트의 크기 변경해보기

2. tkinter 기초 - 라벨과 버튼

- tkinter로 라벨 배치
 - tkinter의 좌표계: 왼쪽 상단 모서리가 원점(0, 0)
→ 대부분의 GUI에서 이를 원점으로 택함
 - 다른 GUI에서는 좌표계의 축이 다를 수 있으므로 항상 확인해야 함



2. tkinter 기초 - 라벨과 버튼

- tkinter로 버튼 배치

- 버튼 변수명 = tkinter.Button(윈도우객체, text="버튼문자열", font=("폰트명", 폰트크기))
- {버튼 변수명}.place(x=x좌표, y=y좌표)

```
import tkinter as tk # tkinter 대신, tk라는 별명 사용

root = tk.Tk()
root.title("버튼 만들기")
root.geometry("800x600")

button = tk.Button(root, text="Button", \
                    font=("Time New Roman", 24))
button.place(x=200, y=100)

root.mainloop()
```

2. tkinter 기초 - 라벨과 버튼

- tkinter로 버튼 배치

- 버튼을 클릭했을 때 처리 → 버튼 이벤트 처리
 - 함수를 만들어주고 이를 command로 넘겨줌

```
import tkinter as tk # tkinter 대신, tk라는 별명 사용

def click_btn():
    button["text"] = "Click!!"

root = tk.Tk()
root.title("버튼 만들기")
root.geometry("800x600")

button = tk.Button(root, text="Button", \
                    font=("Time New Roman", 24), \
                    command=click_btn)
button.place(x=200, y=100)

root.mainloop()
```


목차

1. 사라진 문자 찾기 게임
2. tkinter 기초 - 라벨과 버튼
3. **tkinter 기초 - 캔버스**
4. tkinter 기초 - 뽑기 프로그램
5. tkinter 기초 - 텍스트 입력 필드
6. tkinter 기초 - 체크 버튼과 메시지 박스

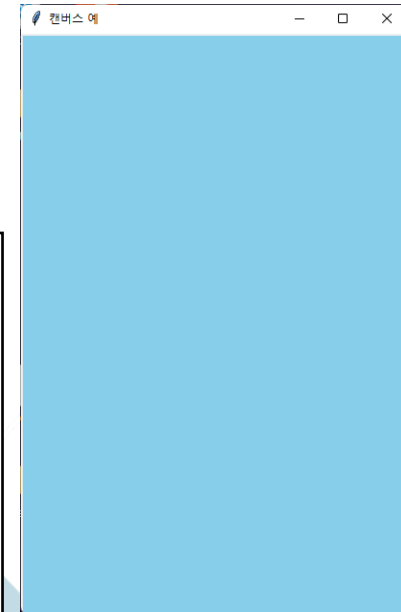
3. tkinter 기초 - 캔버스



- tkinter의 캔버스(canvas)
 - 이미지나 도형을 그릴 때 사용함
 - 게임 개발에서 이미지, 도형은 필수적임
 - Canvas()로 캔버스를 생성하고, pack()과 place()로 배치함
 - 캔버스 변수명 = tkinter.Canvas(윈도우 객체, width=폭, height=높이, bg=배경색)
 - 캔버스 만을 배치하면 윈도우 크기가 pack()에 의해 결정되므로 root.geometry() 생략 가능

```
import tkinter as tk

root = tk.Tk()
root.title("캔버스 예")
canvas = tk.Canvas(root, width=400, height=600, bg="skyblue")
canvas.pack()
root.mainloop()
```



<https://hwabang.net/category/%EB%A9%B4%EC%B2%9C%EB%AF%B8%EB%8B%88%EC%BA%94%EB%B2%84%EC%8A%A4/6625/>

3. tkinter 기초 - 캔버스

- tkinter의 캔버스(canvas)

- 배경색 설정 시,

"red", "green", "blue", "yellow" 등과 같이 영어 단어 사용 가능

- 배경색 설정 시, RGB(Red, Green, Blue)의 조합으로도 설정 가능

- 빨강, 초록, 파랑의 빛의 세기를 조합하여 색을 지정함

- 빨강, 초록, 파랑 각각의 빛의 세기는 0(00) ~ 255(ff)까지 임

- 지정 시 앞에 #을 붙여서 표현하고 #RRGGBB 방식임

- #000000은 검은색, #ffffff는 하얀색임

- 정확한 값 지정이 어렵다면, 프로그램이나 사이트 이용
예) <https://www.color-hex.com/>

- 예제 1. 캔버스의 색상을 변경 (RGB 방식으로)

3. tkinter 기초 - 캔버스

- 캔버스의 이미지 표시

- 이미지를 구글에서 다운로드 (예: apple.gif)
주의: gif, png 파일을 지원 (jpg, jpeg, bmp는 지원하지 않음)
- 다운받은 이미지를 소스 파일과 같은 폴더 안에 넣기

```
import tkinter as tk

root = tk.Tk()
root.title("사진 넣기")
canvas = tk.Canvas(root, width=400, height=600)
canvas.pack()

img = tk.PhotoImage(file="apple.gif")
# file="이미지 경로"
canvas.create_image(200, 200, image=img)
# x좌표, y좌표, image=이미지를 로딩한 변수
# x좌표와 y좌표는 이미지의 중점을 캔버스 어디에 위치할 것인지

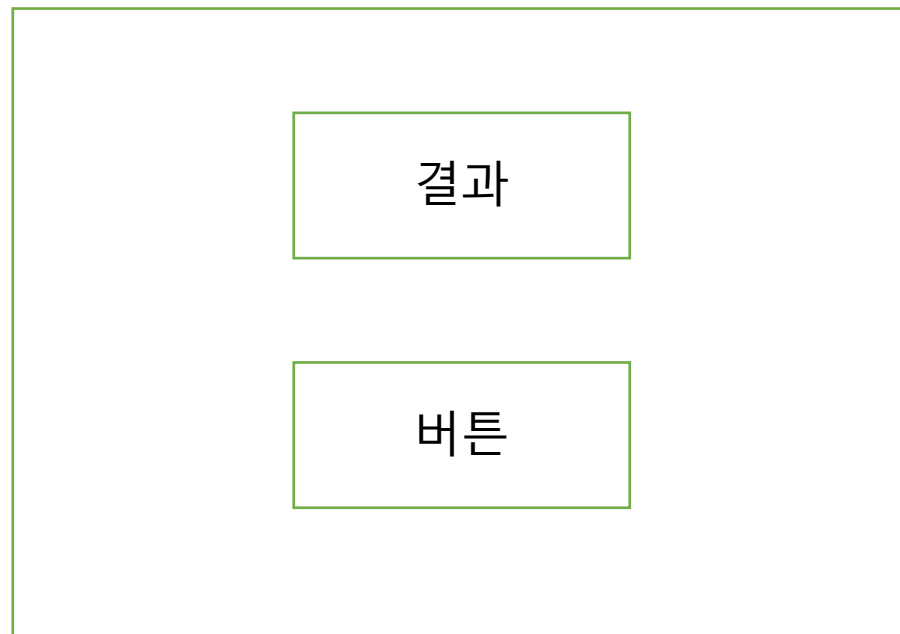
root.mainloop()
```

목차

1. 사라진 문자 찾기 게임
2. tkinter 기초 - 라벨과 버튼
3. tkinter 기초 - 캔버스
4. **tkinter 기초 - 뿔기 프로그램**
5. tkinter 기초 - 텍스트 입력 필드
6. tkinter 기초 - 체크 버튼과 메시지 박스

4. tkinter 기초 - 뽐기 프로그램

- 1단계: 화면 레이아웃 생각하기
 - 버튼을 어떻게 둘 것인가?
 - 배경은 무엇으로 할 것인가?



4. tkinter 기초 - 뽐기 프로그램

- 2단계: 화면 레이아웃에 맞게 구성하기
 - 배경 이미지를 넣고 싶은 경우
 - 단, 배경 이미지는 캔버스(윈도우) 크기보다 커야 함

```
import tkinter as tk

root = tk.Tk()
root.title("뽐기 프로그램")
root.resizable(False, False) # 윈도우 크기 고정
canvas = tk.Canvas(root, width=400, height=300)
canvas.pack()

img = tk.PhotoImage(file="background.png")
canvas.create_image(200, 150, image=img)

root.mainloop()
```

4. tkinter 기초 - 뽐기 프로그램

- 2단계: 화면 레이아웃에 맞게 구성하기
 - 버튼 배치

```
# .. 그대로 ..

label = tk.Label(root, text="??", font=("Time New Roman", 60), bg="white")
label.place(x=150, y=50)

btn = tk.Button(root, text="뽐기", font=("Time New Roman", 30), fg="red")
btn.place(x=140, y=200)

root.mainloop()
```

4. tkinter 기초 - 뽑기 프로그램

- 3단계: 버튼 이벤트 구성

- 1, 2, 3, 4 중에 1개를 뽑는 것을 예
(여러분들이 뽑고 싶은 것을 넣으면 됨)

```
import random

# .. 상단에 ..
def click_btn():
    label["text"] = random.choice(["1", "2", "3", "4"])
    label.update()

# 버튼 부분 수정
btn = tk.Button(root, text="뽑기", font=("Time New Roman", 30), \
                fg="red", command=click_btn)
btn.place(x=140, y=200)

root.mainloop()
```

목차

1. 사라진 문자 찾기 게임
2. tkinter 기초 - 라벨과 버튼
3. tkinter 기초 - 캔버스
4. tkinter 기초 - 뿔기 프로그램
5. **tkinter 기초 - 텍스트 입력 필드**
6. tkinter 기초 - 체크 버튼과 메시지 박스

5. tkinter 기초 - 텍스트 입력 필드

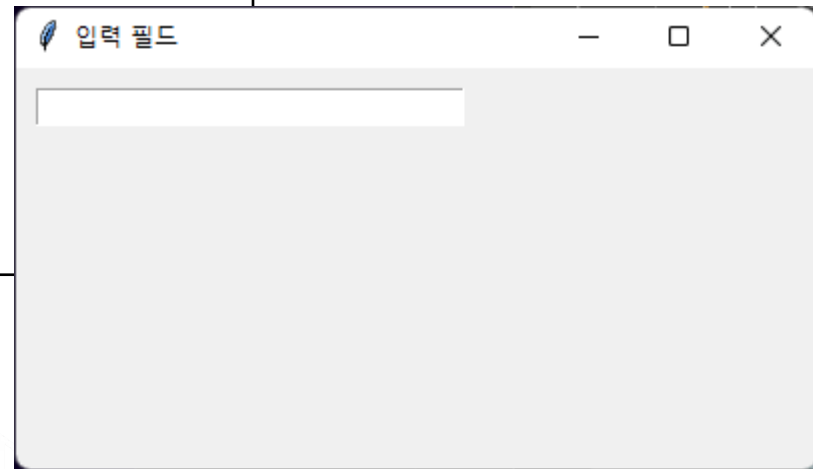
- 텍스트 입력 필드(Entry)

```
import tkinter as tk

root = tk.Tk()
root.title("입력 필드")
root.geometry("400x200")

entry = tk.Entry(width=30)
entry.place(x=10, y=10)

root.mainloop()
```



5. tkinter 기초 - 텍스트 입력 필드

- 텍스트 입력 필드(Entry)

```
import tkinter as tk

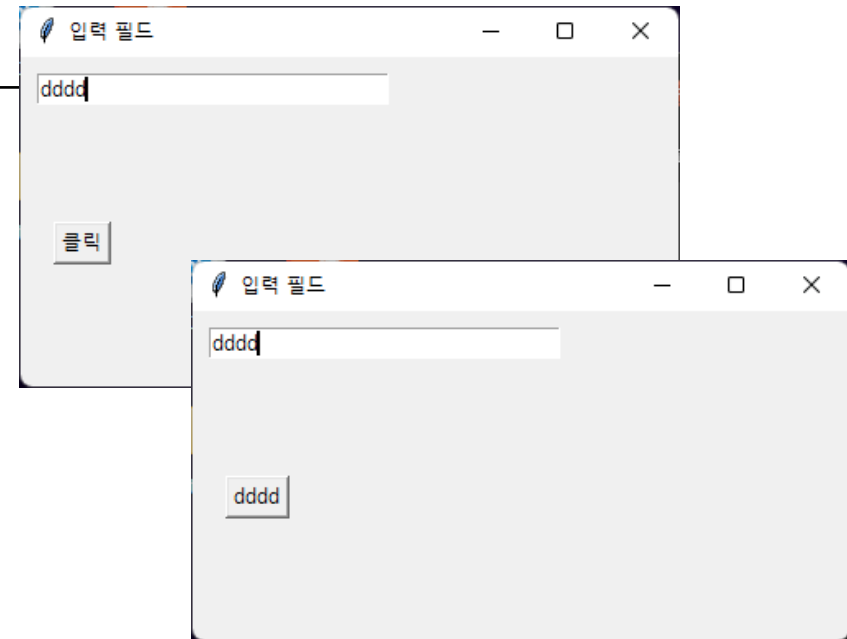
def click_btn():
    txt = entry.get()
    btn["text"] = txt

root = tk.Tk()
root.title("입력 필드")
root.geometry("400x200")

entry = tk.Entry(width=30)
entry.place(x=10, y=10)

btn = tk.Button(text="클릭", command=click_btn)
btn.place(x=20, y=100)

root.mainloop()
```



이 밖에도 여러 개인 경우
Text(),
스크롤 형태는
ScrolledText() 등이 있음

6. tkinter 기초 - 체크 박스

- 체크 박스(Checkbutton)과 체크 박스의 설정 여부 확인

```
import tkinter as tk

def check():
    if check_val.get() == True:
        print("체크되어 있음")
    else:
        print("체크되어 있지 않음")
```

```
root = tk.Tk()
root.title("체크박스")
root.geometry("400x200")

check_val = tk.BooleanVar()
check_val.set(False)
```

```
check_btn = tk.Checkbutton(text="체크버튼", variable=check_val,
                           command=check)
```

```
check_btn.pack()
```

```
root.mainloop()
```



6. tkinter 기초 - 메시지 박스

- 메시지 박스(message box)

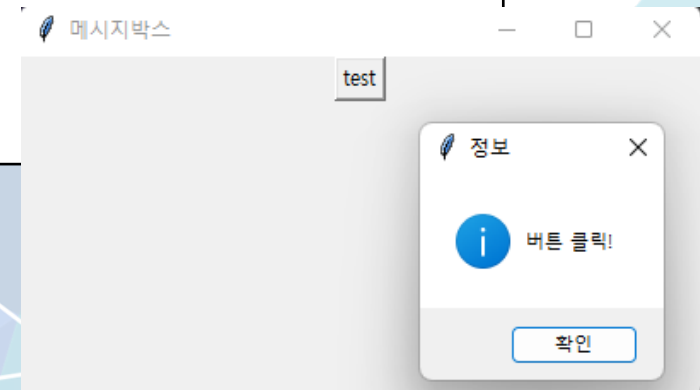
```
import tkinter as tk
import tkinter.messagebox

def click_btn():
    tkinter.messagebox.showinfo("정보", "버튼 클릭!")

root = tk.Tk()
root.title("메시지박스")
root.geometry("400x200")

btn = tk.Button(text="test", command=click_btn)
btn.pack()

root.mainloop()
```



6. tkinter 기초 - 메시지 박스

- 메시지 박스(message box)
 - showinfo() : 정보 표시 박스
 - showwarning(): 경고 표시 박스
 - showerror(): 에러 표시 박스
 - askyesno(): 네, 아니오 버튼이 있는 박스
 - askokcandle(): OK와 취소 버튼이 있는 박스

- 예제 1. 위에서 언급한 박스를 한 번 실행

다음 시간

1. tkinter 기초 - 실시간 처리
2. tkinter 기초 - 키 입력
3. tkinter 기초 - 미로 게임
4. Pygame

문의사항 및 질문

- 혹시 질문있나요?
- 모르는 문제나 더 알고 싶은 사항이 있으면 언제든지 연락 가능
 - 배재대학교 컴퓨터공학과 박사과정 이태준
 - Tel: 010-5223-2912
 - Email: marlrero@kakao.com

문의사항 및 질문

고생하셨습니다!

