

<계산기 만들기>

1) 계산기 외형을 만든다

- 표시창(Display)
- 숫자 버튼(Number)
- 연산자 버튼(Operator)

2) 계산하는 기능을 만든다

- 버튼과 표시창 연결
- 특수 버튼(=C, AC)기능
- 수식 계산

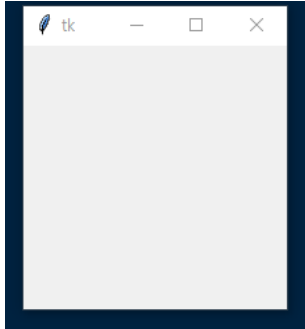


사용할 패키지 이름은 Tkinter(TK interface)이다. 파이썬의 Tkinter는 TCL/TK라는 범용 GUI 라이브러리 기반으로 만들어졌다. 표시창(Display), 숫자버튼(Number), 연산자버튼(Operator) 화면으로 구성할 것이다.

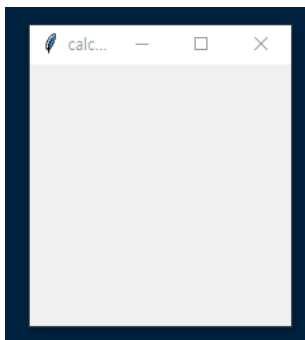
1) 먼저 코드 상단에 `tkinter import` 한다. `tk.TK()`는 윈도우를 생성해주는 함수이다.

```
# Import tkinter as tk
```

-tkinter 라이브러리를 호출: tkinter 함수를 쓸 수 있게 된다.



2) 윈도우 타이틀 이름을 바꿔준다.



3) 디스플레이를 생성한다.

```
# Entry 만들기: 한 줄 짜리 입력창
```

-어디에 만들 것인지, 넓이, 높이는 얼마로 할 것인지, 색상은 무엇으로 할지

-만들어진 Entry를 어디에 배치할지도 중요: `grid()`

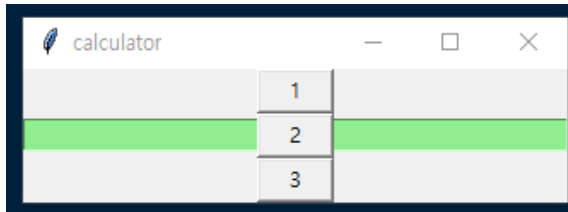


4) 프레임과 숫자 버튼을 생성한다.

tk.Button() 만들기

-어디에 만들 것인지, 넓이, 높이는 얼마로 할 것인지, text는 무엇으로 할지

-만들어진 Button을 어디에 배치할지도 고려



숫자 버튼 넣기(너무 길기 때문에 반복문 사용)

```
tk.Button(num_frame, text="1", width=5).grid(row=0,column=0)
```

```
tk.Button(num_frame, text="2", width=5).grid(row=1,column=0)
```

```
tk.Button(num_frame, text="3", width=5).grid(row=2,column=0)
```

```
tk.Button(num_frame, text="4", width=5).grid(row=0,column=1)
```

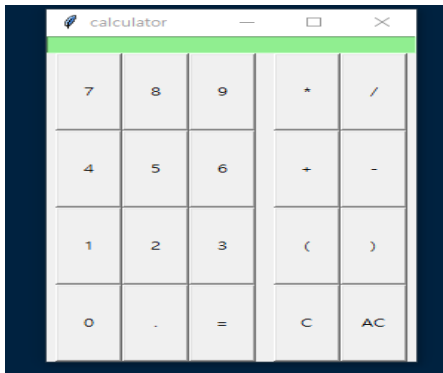
```
tk.Button(num_frame, text="5", width=5).grid(row=1,column=1)
```

```
tk.Button(num_frame, text="6", width=5).grid(row=2,column=1)
```

5) For 반복문 이용하여 버튼 생성



6) 연산자까지 넣어서 완성 (width=5, height=5 바꿈, colspan=2 넣음)



7) Command(cmd) 함수를 숫자, 연산자 버튼에 정의 해주고 click 함수도 만들어주기

```
#연산자 버튼 넣어보기
op_list=['*', '/', '+', '-', '(', ')', 'C', 'AC']
r=0
c=0

for btn_text in op_list:
    def cmd(x=btn_text):
        click(x)
    tk.Button(op_frame, text=btn_text, width=5, height=5, command=cmd).grid(row=r, column=c)
```

```
#숫자 버튼을 넣어봄
num_list=['7','8','9','4','5','6','1','2','3','0','.','=']
r=0
c=0
for btn_text in num_list:
    def cmd(x=btn_text):
        click(x)
    tk.Button(num_frame, text=btn_text, width=5, height=5, command=cmd).grid(row=r, column=c)
```

def click(text):

display.insert(tk.END, text)

Display에 추가한 후 삽입(insert)하고 마지막(END)에 추가하는 값은 text이다.

entry에 글씨 넣기, 지우기

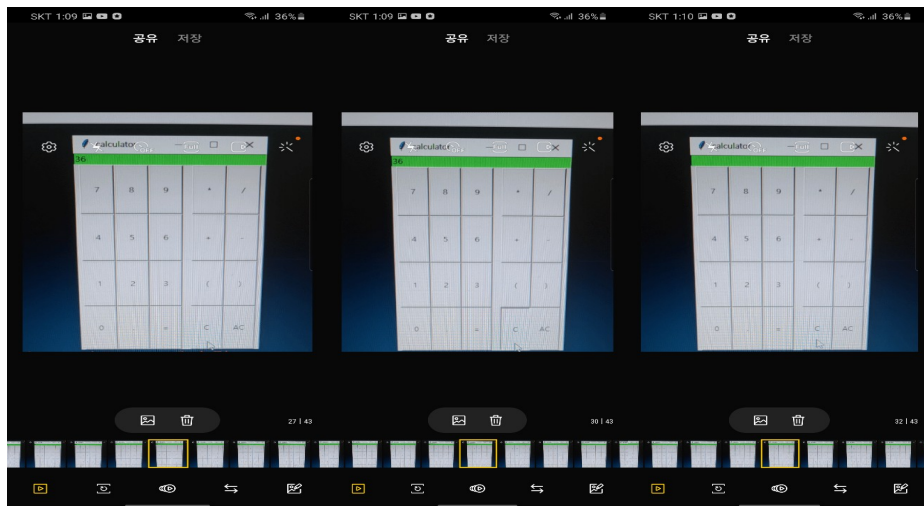
-display.insert(tk.END, result)

display의 가장 마지막에 result값을 추가

-display.delete(0, tk.END)

display의 처음부터 끝까지 삭제

8) elif문으로 display에서 delete되는 기능을 만든다.



9) if, elif, else문으로 결과값 계산 기능을 추가한다.

>>display에 있는 글자를 가져와서 값을 계산, 자리수를 2자리로 끊고 문자열 상태로 변환

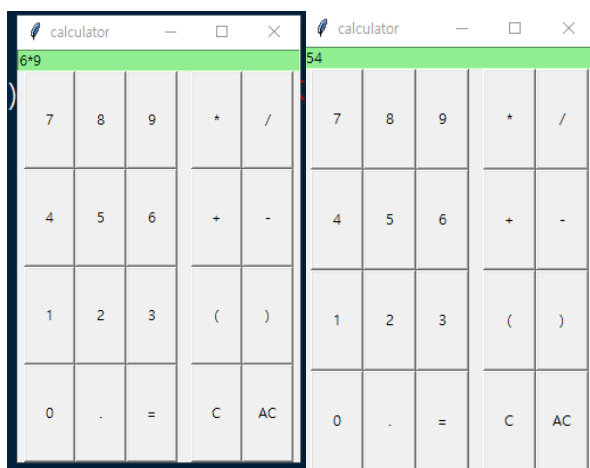
수식계산

-display.get(): display에 있는 글씨

-eval(입력값): 입력값을 수식으로 보고 계산

-round(값,2): 값을 소수점 2자리로 변환

-string(숫자값): 숫자값을 문자로 전환



10) 계산할 수 없는 값이 오면 오류라고 쓰도록 try, except문 넣기

try~except문

-try문으로 시도할 문장을 지정할 수 있다

-프로그램을 실행하는 중 시도할 문장에서 에러가 발생하면 예비문장을 수행한다

try:

 시도할 문장1

 시도할 문장2

except:

 예비문장1

