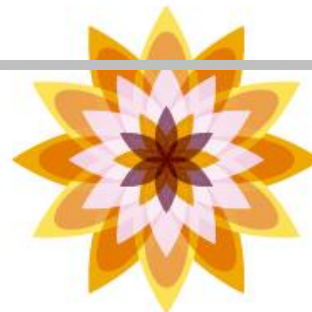
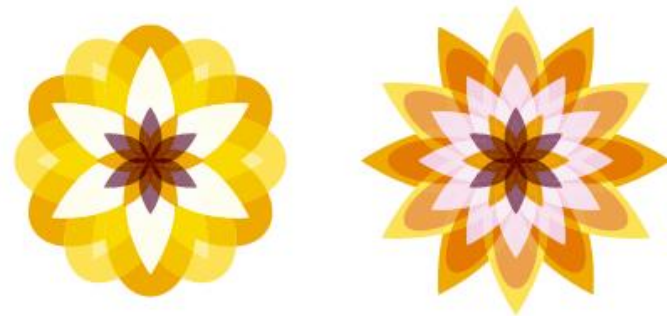


Chapter 06
GUI 기초(2)



1. 텍스트 입력 필드 배치하기

- 텍스트 입력을 수행하는 파이썬 GUI에는 Entry라는 1행 입력 필드와 Text라는 여러 행 입력 필드가 있습니다.
- 1절에서는 Entry 사용 방법, 2절에서는 Text 사용 방법을 설명합니다.

1. 텍스트 입력 필드 배치하기

■ 1행 텍스트 입력 필드

- 1행 텍스트 입력 필드는 `Entry()` 명령으로 만듭니다.
- 텍스트 입력 필드도 `place()` 명령으로 배치합니다.
- 다음 프로그램을 입력하고 파일 이름을 붙여 저장한 뒤 실행해 봅니다.
- 예제 `ex0601_1.py`

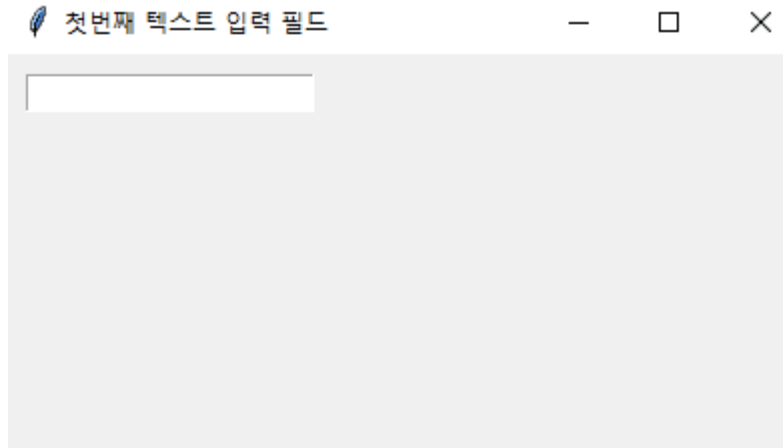
```
1 import tkinter
2 root = tkinter.Tk()
3 root.title("첫 번째 텍스트 입력 필드")
4 root.geometry("400x200")
5 entry = tkinter.Entry(width=20)
6 entry.place(x=10, y=10)
7 root.mainloop()
```

tkinter 모듈 임포트
윈도우 객체 생성
윈도우 제목 지정
윈도우 크기 고정
20 문자 크기 입력 필드 컴포넌트 생성
윈도우에 입력 필드 컴포넌트 배치
윈도우 표시

1. 텍스트 입력 필드 배치하기

■ 1행 텍스트 입력 필드

- 예제 ex0601_1.py
- 이 프로그램을 실행하면 윈도우에 다음과 같은 입력 필드가 배치됩니다.



- 파이썬에서 GUI 컴포넌트를 만드는 명령의 인수에 사용하는 root는 만들어지는 컴포넌트를 윈도우에 배치하는 경우에는 생략할 수 있으므로 이 프로그램의 Entry()에서는 root를 생략합니다.
- Entry() 인수 'width = '의 값으로 문자 입력 필드의 크기를 지정합니다.

1. 텍스트 입력 필드 배치하기

■ Entry 내 문자열 조작하기

- Entry 내 문자열은 `get()` 명령으로 얻을 수 있습니다.
- 입력 필드에 문자를 입력하고 버튼을 누르면 그 문자열을 얻는 프로그램을 확인합니다.
- 다음 프로그램을 입력하고 파일 이름을 붙여 저장한 뒤 실행해 봅니다.
- 예제 `ex0601_2.py` (텍스트 입력 필드의 문자열을 얻어 버튼에 표시하는 처리 관련 내용 굵게 표시)

```
1 import tkinter
2
3 def click_btn( ):
4     txt = entry.get( )
5     button["text"] = txt
6
7 root = tkinter.Tk( )
8 root.title("첫번째 텍스트 입력 필드")
9 root.geometry('400x200')
```

tkinter 모듈 импорт

버튼 클릭시 실행되는 함수 정의
입력 필드의 문자열을 변수 txt에 대입
버튼의 문자열에 txt의 값을 대입

윈도우 객체 생성

윈도우 제목 지정

윈도우 크기 고정

1. 텍스트 입력 필드 배치하기

■ Entry 내 문자열 조작하기

- 예제 ex0601_2.py (텍스트 입력 필드의 문자열을 얻어 버튼에 표시하는 처리 관련 내용 굵게 표시)

```
10 entry = tkinter.Entry(width=20)
11 entry.place(x=20, y=20)
12 button = tkinter.Button(text='문자
13 열 얻기', command=click_btn)
14 button.place(x=20, y=100)
15 root.mainloop()
```

20 문자 크기 입력 필드 컴포넌트 생성
윈도우에 입력 필드 컴포넌트 배치
버튼 컴포넌트 생성, 'command='로 클
릭시 실행할 함수 지정
버튼 배치
윈도우 표시

- 이 프로그램을 실행하고 텍스트 입력 필드에 임의의 문자열을 입력한 뒤, 버튼을 클릭하면 버튼에 그 문자열이 표시됩니다.

2. 여러 행 텍스트 입력 필드 배치하기

■ 여러 행 텍스트 입력 필드

- 여러 행 텍스트 입력 필드는 `Text()` 명령을 실행합니다.
- 다음 프로그램을 입력하고 파일 이름을 붙여 저장한 뒤 실행해 봅니다.
- 예제 `ex0602_1.py`

```
1 import tkinter
```

tkinter 모듈 импорт

```
2
```

```
3 def click_btn( ):
```

버튼 클릭시 실행되는 함수 정의
텍스트 입력 필드 마지막에 문자열
추가

```
4     text.insert(tkinter.END, "몬스터가 나  
5     타났다.")
```

```
5
```

```
6 root = tkinter.Tk( )
```

윈도우 객체 생성

```
7 root.title("여러 행 텍스트 입력")
```

윈도우 제목 지정

```
8 root.geometry('400x200')
```

윈도우 크기 고정

2. 여러 행 텍스트 입력 필드 배치하기

■ 여러 행 텍스트 입력 필드

- 여러 행 텍스트 입력 필드는 `Text()` 명령을 실행합니다.
- 다음 프로그램을 입력하고 파일 이름을 붙여 저장한 뒤 실행해 봅니다.
- 예제 `ex0602_1.py`

```
1 button = tkinter.Button(text='메시지',  
2   command=click_btn)  
3 text = tkinter.Text( )  
4 text.pack( )  
5 root.mainloop( )
```

버튼 컴포넌트 생성, 'command='로
클릭시 실행할 함수 지정

버튼 배치

여러 행 텍스트 입력 필드 컴포넌트
생성

입력 필드 컴포넌트 배치

윈도우 표시

2. 여러 행 텍스트 입력 필드 배치하기

■ 여러 행 텍스트 입력 필드

- 이 프로그램을 실행하면 버튼과 여러 행 텍스트 입력 필드가 표시됩니다.
- 버튼을 클릭하면 입력 필드에 '몬스터가 나타났다!'라는 문자열이 추가됩니다.
- 이 프로그램에서는 `Text()` 명령으로 만든 입력 필드를 `packO` 명령으로 배치했습니다.
- `place()` 명령을 사용하는 경우, 입력 필드의 배치 위치와 사이즈를 적절하게 지정합니다.
- 크기는 `place()` 명령의 인수인 '`width=`', '`height=`'로 지정할 수 있습니다.

```
text = tkinter.Text()
text.place(x=20, y=50, width=360, height=120)
```

- 텍스트 입력 필드에는 4번 행과 같이 `insert()` 명령으로 문자열을 추가할 수 있습니다.
- `insert()` 명령의 인수는 문자열을 추가할 위치 및 추가할 문자열입니다.
- 이번에 문자열을 추가할 위치는 `tkinter.END`로 입력 필드의 가장 마지막으로 지정했습니다.

2. 여러 행 텍스트 입력 필드 배치하기

■ 여러 행 텍스트 입력 필드

- Text에 입력된 문자열을 얻을 때는 Entry 프로그램에서 사용했던 `get()` 명령으로 'get(시작위치, 종료 위치)'를 사용합니다.
- 입력 필드의 문자열을 삭제하려면 'delete(시작 위치, 종료 위치)'를 사용합니다.
- 예를 들면, 입력 필드의 전체 문자열을 얻는 경우에는 `get("1.0", "end-1c")`로 합니다.
- "1.0"은 1행 0번째 문자(즉, 첫번째 문자라는 의미입니다. "end-1c"는 "end"만으로는 가장 마지막의 다음 위치가 되므로 거기에서 1 문자 앞이라는 의미입니다.
- Text의 문자 위치 지정은 상당히 복잡하기 때문에 지금 바로 이해하지 않더라도 괜찮습니다.
- 대량의 문자열을 다루는 소프트웨어에서 스크롤 바(scroll bar)가 포함된 텍스트 입력 필드를 사용해야 하는 경우에는 `ScrolledText()`라는 명령어로 컴포넌트를 배치할 수 있습니다.
- `ScrolledText()` 사용 방법은 기본적으로 앞에서 사용한 `Text()`와 같습니다.
- `ScrolledText()`를 사용하려면 `tkinter.scrolledtext` 모듈을 임포트해야 합니다.

3. 체크 버튼 배치하기

- 체크 버튼은 항목 선택 시 사용하는 작은 사각형으로 일반적으로 체크 박스(check box)라고 부르기도 합니다.
- 여기에 체크하면 v 표시가 나타납니다.
- 본 강좌에서는 파이썬 명령어에 맞춰 체크 박스가 아니라 체크 버튼으로 통일해서 표기하겠습니다.

3. 체크 버튼 배치하기

■ 체크 버튼 배치하기

- 체크 버튼은 `Checkbutton()` 명령으로 만듭니다.
- 다음 프로그램을 입력하고 파일 이름을 붙여 저장한 뒤 실행해 봅니다.
- 예제 `ex0603_1.py` (체크 버튼 생성과 배치 관련 내용은 굵게 표시)

1	<code>import tkinter</code>	tkinter 모듈 импорт
2	<code>root = tkinter.Tk()</code>	윈도우 객체 생성
3	<code>root.title("체크 버튼 다루기")</code>	윈도우 제목 지정
4	<code>root.geometry('400x200')</code>	윈도우 크기 고정
5	<code>cbtn = tkinter.Checkbutton(text='체크 버튼')</code>	체크 버튼 컴포넌트 생성
6	<code>cbtn.pack()</code>	체크 버튼 컴포넌트 배치
7	<code>root.mainloop()</code>	윈도우 표시

- 이 프로그램을 실행하면 캔버스에 체크 버튼이 배치됩니다.
- □를 클릭하면 v 표시가 되면서 선택되는 것을 확인할 수 있습니다.

3. 체크 버튼 배치하기

■ 체크 여부 확인하기

- 체크 버튼의 체크 여부를 확인하기 위해서는 다소 복잡한 코드를 작성해야 합니다.
- 체크 여부는 BooleanVar() 명령을 사용해 확인하므로 먼저 해당 명령의 사용 방법을 설명합니다.
- 체크 버튼을 체크한 상태로 만들어 봅시다.
- 다음 프로그램을 입력하고 파일 이름을 붙여 저장한 뒤 실행해 봅시다.

3. 체크 버튼 배치하기

■ 체크 여부 확인하기

- 예제 ex0603_2.py (BooleanVar()를 사용한 관련 내용은 굵게 표시)

```
1 import tkinter
2 root = tkinter.Tk( )
3 root.title("처음부터 체크된 상태 만들기")
4 root.geometry('400x200')
5 cval = tkinter.BooleanVar( )
6 cval.set(False)
7 cbtn = tkinter.Checkbutton(text='체크 버튼',
8 variable=cval)
9 cbtn.pack( )
root.mainloop( )
```

tkinter 모듈 импорт
윈도우 객체 생성
윈도우 제목 지정
윈도우 크기 고정
BooleanVar() 객체 준비
객체 True 설정
체크 버튼 컴포넌트 생성

체크 버튼 컴포넌트 배치
윈도우 표시

- 이 프로그램을 실행하면 체크된 상태의 체크 버튼이 표시됩니다.

3. 체크 버튼 배치하기

■ 체크 여부 확인하기

- 5번 행에서 BooleanVar() 명령으로 객체를 생성하고 6번 행에서 객체를 True로 설정합니다.
- True면 체크를 한 상태 False면 체크를 하지 않은 상태입니다.
- 7번 행에서 체크 버튼을 만들면서 'variable='로 BooleanVar()로 생성한 객체를 지정합니다.
- 이제 BooleanVar() 객체가 체크 버튼과 연결됩니다.

3. 체크 버튼 배치하기

■ 체크 여부 확인하기

- 그 다음 체크 여부를 확인합니다.
- 확인할 때는 BooleanVar() 객체에 대해 get() 메서드를 사용합니다.
- 예제 ex0603_3.py

```
1 import tkinter
2
3 def check( ):
4     if cval.get() == True:
5         print("체크되어 있습니다.")
6     else:
7         print("체크되어 있지 않습니다.")
8
9 root = tkinter.Tk( )
10 root.title("체크 상태 확인")
```

tkinter 모듈 импорт

체크 버튼 클릭 시 실행되는 함수 정의

체크되어 있다면

'체크되어 있습니다' 출력

그렇지 않다면

'체크되어 있지 않습니다' 출력

윈도우 객체 생성

윈도우 제목 지정

3. 체크 버튼 배치하기

■ 체크 여부 확인하기

▪ 예제 ex0603_3.py

11	<code>root.geometry('400x200')</code>	윈도우 크기 고정
12	<code>cval = tkinter.BooleanVar()</code>	BooleanVar() 객체 준비
13	<code>cval.set(False)</code>	객체 True 설정
14	<code>cbtn = tkinter.Checkbutton(text='체크 버튼', variable=cval, command=check)</code>	체크 버튼 컴포넌트 생성
15	<code>cbtn.pack()</code>	체크 버튼 컴포넌트 배치
16	<code>root.mainloop()</code>	윈도우 표시

- 이 프로그램을 실행하고 체크 버튼을 클릭하면 셸 윈도우에 체크 상태가 출력됩니다.
- 체크 및 해제를 반복하며 동작을 확인합니다.
- 14번 행에서 체크 버튼을 생성하면서 클릭 시 실행할 함수를 'command='로 지정했습니다.
- 이는 버튼 클릭 시 처리에서 학습한 것과 동일합니다.
- 체크 버튼의 체크 여부를 확인하기 위해서는 4번 행과 같이 BooleanVar() 객체의 get() 명령을 사용합니다.

4. 메시지 박스 표시하기

■ 메시지 박스 사용 방법

- 메시지 박스를 사용하기 위해서는 `tkinter.messagebox` 모듈을 임포트합니다.
- 예제 `ex0604_1.py`

1	<code>import tkinter</code>	tkinter 모듈 임포트
2	<code>import tkinter.messagebox</code>	tkinter.messagebox 모듈 임포트
3		
4	<code>def click_btn():</code>	함수 정의
5	<code> tkinter.messagebox.showinfo("정보", "</code>	메시지 박스 표시
6	<code>버튼을 눌렀습니다.")</code>	
7	<code>root = tkinter.Tk()</code>	윈도우 객체 생성
8	<code>root.title("체크 상태 확인")</code>	윈도우 제목 지정
9	<code>root.geometry('400x200')</code>	윈도우 크기 고정

4. 메시지 박스 표시하기

■ 메시지 박스 사용 방법

- 메시지 박스를 사용하기 위해서는 `tkinter.messagebox` 모듈을 임포트합니다.
- 예제 `ex0604_1.py`

```
10 btn = tkinter.Button(text="테스트",  
11   command=click_btn)  
12 btn.pack()  
   root.mainloop()
```

버튼 생성, 클릭 시 실행할 함수 지정

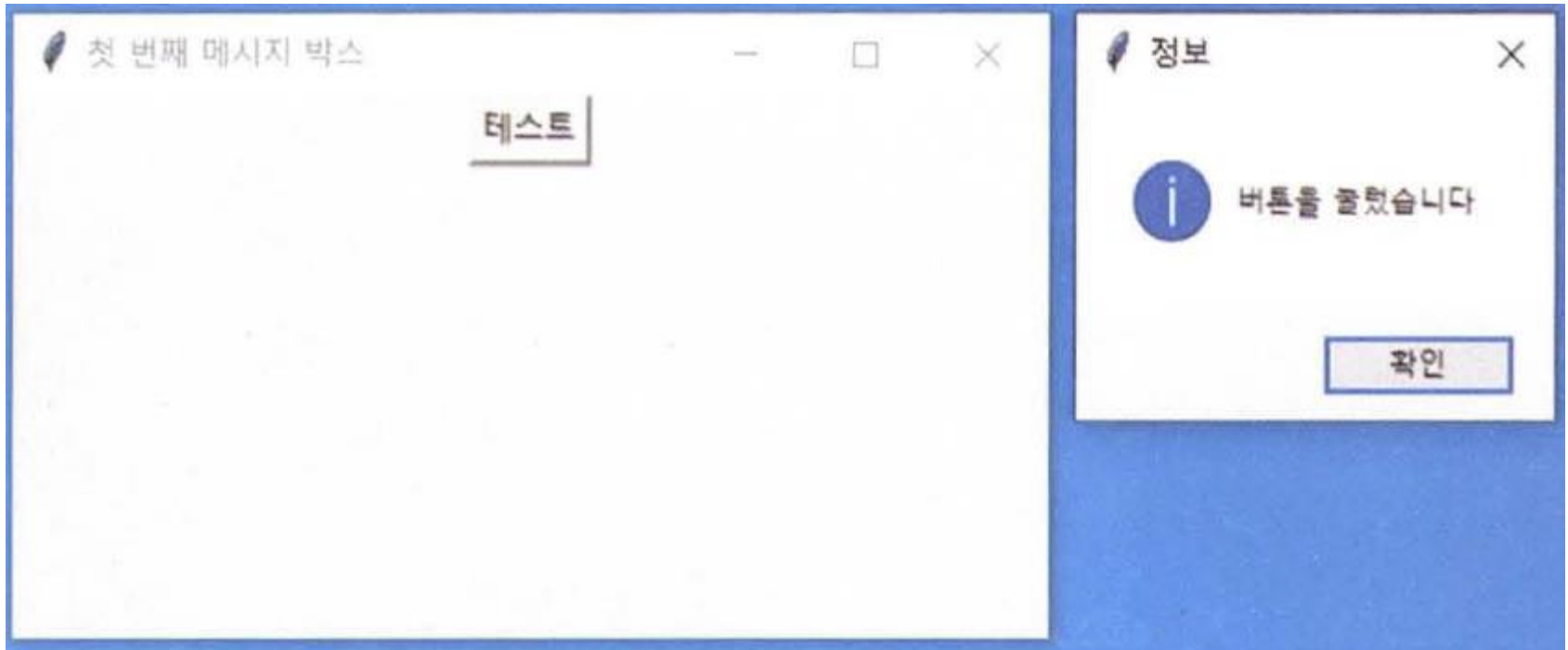
버튼 배치

윈도우 표시

4. 메시지 박스 표시하기

■ 메시지 박스 사용 방법

- 이 프로그램을 실행하고 윈도우 상의 버튼을 클릭하면 메시지 박스가 표시됩니다.



- 이 프로그램에서는 `showinfo()` 명령으로 메시지 박스를 표시합니다.

4. 메시지 박스 표시하기

■ 메시지 박스 사용 방법

- 메시지 박스에는 여러 종류가 있습니다.
- 메시지 박스를 표시하는 주요 명령어는 다음과 같습니다.

<code>showinfo()</code>	정보를 표시하는 메시지 박스
<code>showwarning()</code>	경고를 표시하는 메시지 박스
<code>showerror()</code>	에러를 표시하는 메시지 박스
<code>askyesno()</code>	'네', '아니오' 버튼이 있는 메시지 박스
<code>askokcancel()</code>	'OK', '취소' 버튼이 있는 메시지 박스

5. RGB 값을 사용한 색 지정

- 파이썬에서 색 지정은 앞서 학습한 것처럼 red 혹은 white 등 영단어로 지정하는 방법과 16진수 RGB 값으로 지정하는 방법이 있습니다.
- 여기에서는 16진수로 지정하는 방법을 설명합니다.
- 우선 빛의 삼원색에 관해 알아보니다.
- 빨강, 초록, 파랑을 빛의 삼원색이라고 부릅니다.
- 빛의 빨강과 초록을 섞으면 노랑, 빨강과 파랑을 섞으면 분홍(magenta), 초록과 파랑을 섞으면 하늘색(cyan)이 됩니다.
- 빨강, 초록, 파랑을 섞으면 흰색이 됩니다.
- 빛의 강하기가 약한(=어두운 색) 경우, 섞은 색도 그만큼 어두운 색이 됩니다

5. RGB 값을 사용한 색 지정

- 컴퓨터에서는 빨강(Red) 빛의 세기, 초록(Green) 빛의 세기, 파랑(Blue) 빛의 세기를 각각 0 ~ 255 단계의 숫자로 표시합니다.
- 예를 들어, 밝은 빨강은 R=255, 어두운 빨강은 R=128입니다.
- 어두운 하늘색을 표현하고 싶다면, 'R=0, G=128, B=128'을 사용합니다.
- 0 ~ 255는 우리들이 일상에서 사용하는 10진수 값입니다.
- 16진수 a ~ f는 대문자로 써도 관계없습니다.
- RGB 값은 이 16진수를 사용해 #RRGGBB로 표기합니다.
- 예를 들면, 밝은 빨강은 #ff0000, 어두운 초록은 #00ff00, 회색은 #808080으로 표기합니다.

5. RGB 값을 사용한 색 지정

- 또는 #RGB로 빨강, 초록, 파랑을 한 문자씩 사용해서 표현하는 방법도 있습니다.
- 이 경우 빨강, 초록, 파랑의 값은 256단계가 아닌 16단계가 됩니다.
- 검정은 #000, 밝은 빨강은 #f00, 회색은 #888, 흰색은 #fff 등으로 지정할 수 있습니다.



Thank You
