

이 장에서 만들 프로그램 및 기본 위젯 활용 방법

- 프로그램의 기본 구조를 이해하고, 버튼과 텍스트 상자와 같은 기본 위젯의 역할과 사용 방법을 설명할 수 있다.
- 기본 위젯을 활용하여 사용자 인터페이스를 설계하고, 프로그램에 적절히 배치할 수 있다.



🕎 이 장에서 만들 프로그램



[프로그램 1] 사진 앨범



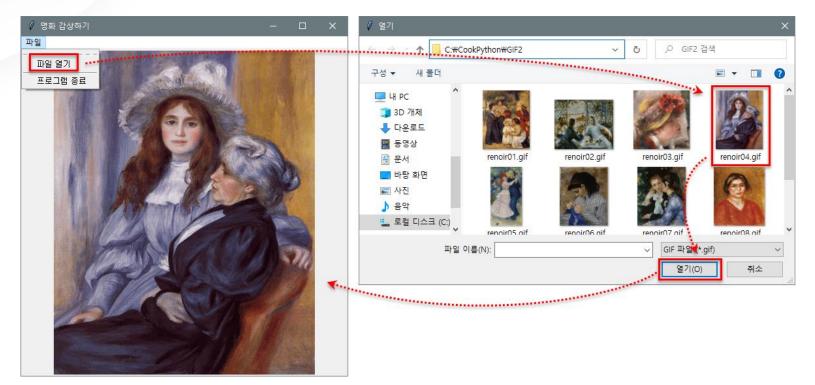




🛒 이 장에서 만들 프로그램



[프로그램 2] 명화 감상











기본 윈도창의 구성

❖ 위젯(Widget) : 윈도창에 나올 수 있는 문자, 버튼, 체크박스, 라디오버튼 등을 의미

Code10-01.py

- from tkinter import * window = Tk() ## 이 부분에서 화면을 구성하고 처리 ## window.mainloop() 7행에서 window.mainloop() 함수 실행
 - 1 tkinter는 파이썬에서 GUI 관련 모듈을 제공해 주는 표준 윈도 라이브러리
 - 3 Tk()는 기본이 되는 윈도를 반환, 이를 루트 윈도 또는 베이스 윈도라고 함. 3행 실행하면 윈도창 화면에 출력 3행에서 베이스 윈도를 window 변수에 넣고



여기서 잠깐!

 tkinter는 TK Interface의 약어. Tk는 Tcl/Tk라는 전통적인 GUI 인터페이스 윈도, 리눅스, 맥 등에서 모두 동일한 코드로 사용 가능







기본 윈도창의 구성

❖ 윈도창 조절

Code10-02.py









❖ 레이블(Label) : 문자를 표현할 수 있는 위젯

Code10-03.py

```
1 from tkinter import *
   window = Tk()
                                      4 Label() 함수가 레이블을 만들어 줌
   label1 = Label(window, text = "COOKBOOK~~ Python을")
   label2 = Label(window, text = "열심히", font = ("궁서체", 30), fg = "blue")
   label3 = Label(window, text = "공부 중입니다.", bg = "magenta", width = 20, height = 5,
           anchor = SE)
                                                              COOKBOOK~~ Python을
                  ③ pack() 함수로 해당 레이블을 화면에 표시
   label1.pack()
   label2.pack()
   label3.pack()
                     Font: 글꼴과 크기 지정
11
                     Fg: 'foreground'의 약어, 글자색 지정
   window.mainloop()
                                                                    공부 중입니다
                     Bg: 'background'의 약어, 배경색 지정
                     width와 height : 위젯의 폭과 높이 지정
                     Anchor: 위젯이 어느 위치에 자리 잡을지 결정
```





❖ 레이블에 글자 대신 이미지 넣기

Code10-04.py

```
1 from tkinter import *
  window = Tk()
  photo = PhotoImage(file = "gif/dog.gif") ② PhotoImage()에서 이미지 파일을 준비
  label1 = Label(window, image = photo)
                                     5 Label()의 옵션 image에서 속성 지정
```

label1.pack()

8

window.mainloop()











SELF STUDY

- Cody10-04.py를 수정해서 이미지를 2개 출력해 보자.
 - [힌트] 위젯을 가로로 나타내려면 pack(side=LEFT)를 사용한다.



```
##Code04.py

from tkinter import *
window = Tk()

photo1 = PhotoImage(file="GIF/dog.gif")
photo2 = PhotoImage(file="GIF/cat.gif")
label1 = Label(master = window, image=photo1)
label2 = Label(master = window, image=photo2)

label1.pack(side=LEFT)
label2.pack(side=LEFT)

window.mainloop()
```







- ❖ 버튼(Button) : 마우스로 클릭하면 눌리는 효과와 함께 지정한 작업 실행
- ❖ 예 : 버튼을 누르면 파이썬 IDLE이 종료되는 코드

Code10-05.py



/프로그래밍기초

록 기본 위젯 활용



❖ 예 : 이미지 버튼을 누르면 간단한 메시지창이 나오는 코드

```
Code10-06.py
  1 from tkinter import *
                                     2 messagebox 모듈 사용 임포트
  2 from tkinter import messagebox
  4 ## 함수 선언 부분 ##
  5 def myFunc():
       messagebox.showinfo("강아지 버튼", "강아지가 귀엽죠? ^^")
                                    6~6 myFunc() 함수 정의,
이 함수는messagebox.showinfo(제목, 내용)
    ## 메인 코드 부분 ##
                                           형식으로 화면에 메시지 상자 출력
    window = Tk()
    photo = PhotoImage(file = "gif/dog2.gif") ⑩~⑪ 이미지 준비하고 버튼에 글자 대신 이미지 표현
 12 button1 = Button(window, image = photo, command = myFunc) ∅ tk
 13
    button1.pack()
                                                                             ∅ 강아지 버튼
 15
    window.mainloop()
                                                                                 강아지가 귀엽죠? ^^
                                                                                       확인
```





록 기본 위젯 활용



- ❖ 체크버튼(Checkbutton) : 켜고 끄는 데 사용하는 위젯
- ❖ 예 : 체크버튼을 켜거나 끄면 메시지창이 열리게 하는 코드



/ 프로그래밍기초

록 기본 위젯 활용



Code10-07.py

```
1 from tkinter import *
                                                               체크버튼이 켜졌어요.
                                                from tkinter import messagebox
                                             ☑ 클릭하세요
   window = Tk()
                                                                      확인
   ## 함수 선언 부분 ##
                                 ⑥~⑩ myFunc() 함수는 chk.get() 함수,
체크버튼에 설정된 값을 가져와서 메시지 출력
   def myFunc():
       if chk.get() == 0:
          messagebox.showinfo("", "체크버튼이 꺼졌어요.")
       else:
          messagebox.showinfo("", "체크버튼이 켜졌어요.")
10
11
   ## 메인 코드 부분 ##
   chk = IntVar()
                                  ® chk 변수를 준비, IntVar() 함수 사용
   cb1 = Checkbutton(window, text = "클릭하세요", variable = chk , command = myFunc)
                                  15
                                     변수 사용
   cb1.pack()
17
   window.mainloop()
```









❖ 라디오버튼(Radiobutton) : 여러 개 중에서 하나를 선택하는 데 사용하는 위젯

Code10-08.py

```
1 from tkinter import *
   window = Tk()
   ## 함수 선언 부분 ##
                                          ⑤~① myFunc() 함수는 var 변수값으로 맨
   def myFunc():
                                                아래쪽 레이블의 텍스트를 변경
       if var.get() == 1:
           label1.configure(text = "파이썬")
       elif var.get() == 2:
           label1.configure(text = "C++")
10
       else:
            label1.configure(text = "Java")
11
12

♪ 정수형 변수 var 준비
   ## 메인 코드 부분 ##
                                          (6~1) 라디오버튼 3개 준비
   var = IntVar()
   rb1 = Radiobutton(window, text = "파이썬", variable = var, value = 1, command = myFunc)
   rb2 = Radiobutton(window, text = "C++", variable = var, value = 2, command = myFunc)
   rb3 = Radiobutton(window, text = "Java", variable = var, value = 3, command = myFunc)
```







❖ 라디오버튼(Radiobutton) : 여러 개 중에서 하나를 선택하는 데 사용하는 위젯

```
18
19 label1 = Label(window, text = "선택한 언어: ", fg = "red")
20
21 rb1.pack()
22 rb2.pack()
23 rb3.pack()
24 label1.pack()
25
26 window.mainloop()
```











❖ 수평으로 정렬하는 방법 : pack() 함수의 옵션 중 side=LEFT, RIGHT

Code10-09.py

```
1 from tkinter import *
2 window = Tk()
3
4 button1 = Button(window, text = "버튼1")
5 button2 = Button(window, text = "버튼2")
6 button3 = Button(window, text = "버튼3")
7
8 button1.pack(side = LEFT)
9 button2.pack(side = LEFT)
10 button3.pack(side = LEFT)
11 window.mainloop()
```

```
-Ø □ ×

버튼1 버튼2 버튼3
```

/ 프로그래밍기초

🖳 위젯의 배치와 크기 조절



수평 정렬

❖ 리스트와 for 문 활용

Code10-10.py



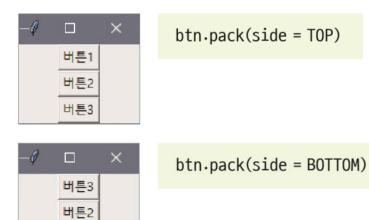








- ❖ pack() 함수의 옵션 중 수직으로 정렬하는 방법 : side=TOP, BOTTOM
 - ✔ Code10-10.py의 10행을 다음과 같이 변경하고 실행



버튼1







- ❖ pack() 함수의 옵션 중에서 윈도창 폭에 맞추는 방법 : fill=X
 - ✔ Code10-10.py의 10행을 다음과 같이 변경하고 실행

- ∅		×
	버튼1	
	버튼2	
	버튼3	

btn.pack(side = TOP, fill = X)







- ❖ pack() 함수의 옵션 중에서 위젯 사이에 여백 주는 방법 : padx=픽셀값 또는 pady=픽셀값
 - ✔ Code10-10.py의 10행을 다음과 같이 변경하고 실행



btn.pack(side = TOP, fill = X, padx = 10, pady = 10)







- ❖ pack() 함수의 옵션 중에서 위젯 내부에 여백 주는 방법 : ipadx=픽셀값 또는 ipady=픽셀값
 - ✔ Code10-10.py의 10행을 다음과 같이 변경하고 실행



btn.pack(side = TOP, fill = X, ipadx = 10, ipady = 10)









- ❖ pack() 함수의 옵션 중에서 위젯 내부에 여백 주는 방법 : ipadx=픽셀값 또는 ipady=픽셀값
 - ✔ 위젯 내부와 외부에 모두 여백



btn.pack(side = TOP, fill = X, ipadx = 10, ipady = 10, padx = 10, pady = 10)







고정 위치에 배치

- ❖ 위젯을 고정 위치에 배치하려면 pack() 대신 place() 함수 사용
- ❖ 그림 9개를 2차원으로 배치하는 코드

Code10-11.py

```
1 from tkinter import *
  ## 전역 변수 선언 부분 ##
                                       4 버튼을 저장할 9개짜리 리스트 준비
4 btnList = [None] * 9
5 fnameList = ["froyo.gif", "gingerbread.gif", "honeycomb.gif", "icecream.gif",
       "jellybean.gif", "kitkat.gif", "lollipop.gif", "marshmallow.gif", "nougat.gif"]
6 photoList = [None] * 9
                                       6 이미지 파일명 9개를 리스트로 준비
7 i, k = 0, 0
                                       6 PhotoImage()로 생성할 9개짜리 리스트
8 xPos, yPos = 0, 0
                                       ❸ xPos, yPos는 그림을 출력할 X, Y 좌표로 사용
9 num = 0
                                       9 num은 그림의 순차 번호로 사용
11 ## 메인 코드 부분 ##
   window = Tk()
   window.geometry("210x210")
14
                                       № 0 9번 반복하면서 버튼을 생성
   for i in range(0, 9):
      photoList[i] = PhotoImage(file = "gif/" + fnameList[i])
16
17
      btnList[i] = Button(window, image = photoList[i])
```



COSS 혁신용합대학 MATERIALS-NANO & MANUFACTURING



- ❖ 위젯을 고정 위치에 배치하려면 pack() 대신 place() 함수 사용
- ❖ 그림 9개를 2차원으로 배치하는 코드

```
18
   for i in range(0, 3):
                                           ⑩~☎ 3×3=9번 반복
20
        for k in range(0, 3):
21
            btnList[num].place(x = xPos, y = yPos)
22
            num += 1
23
           xPos += 70
24
      xPos = 0
        yPos += 70
25
26
   window.mainloop()
```









SELF STUDY

- Cody10-11.py를 실행할 때마다 그림을 임의로 뒤섞어서 나타내게 하자.
 - で random 모듈을 임포트하고 shuffle(리스트) 함수를 사용하면 리스트를 임의로 뒤섞어 준다.









• Cody10-11.py를 실행할 때마다 그림을 임의로 뒤섞어서 나타내게 하자.

```
from tkinter import *
2 import random
4 \text{ btnList} = [None] * 9
fnameList = ["honeycomb.gif", "icecream.gif", "jellybean.gif", "kitkat.gif", "lollipop.gif", "marshmallow.gif", "nougat.gif", "oreo.gif", "pie.gif"]
 7 random.shuffle(fnameList)
 9 photoList = [None] * 9
2 \times Pos = 0
3 \text{ vPos} = 0
|4| num = 0
6 \text{ window} = \text{Tk()}
 window.geometry("210x210")
19 for i in range(0,9):
    photoList[i] = PhotoImage(file = "gif/" + fnameList[i])
btnList[i] = Button(window, image = photoList[i])
3 for i in range(0,3):
        for k in range(0,3):
          btnList[num].place(x = xPos, y = yPos)
          num += 1
         xPos += 70
       xPos = 0
       vPos += 70
31window.mainloop()
```







[프로그램 1]의 완성

❖ (이전) 버튼이나 (다음) 버튼을 누르면 사진들을 표시하는 사진 앨범 프로그램

Code10-12.py

```
from tkinter import *
   from time import *
  ## 전역 변수 선언 부분 ##
  fnameList = ["jeju1.gif", "jeju2.gif", "jeju3.gif", "jeju4.gif", "jeju5.gif",
             "jeju6.gif", "jeju7.gif", "jeju8.gif", "jeju9.gif"]
  photoList = [None] * 9 6 fnameList 변수에 사진 9장의 파일명을 저장
  num = 0
                     🕝 photoList에는 PhotoImage() 함수로 생성할 변수 9개 준비
                     7 num은 현재 사진의 번호
   ## 함수 선언 부분 ##
                     ⑩~❶ <다음> 버튼을 누르면 실행되는 함수
   def clickNext():
                     🛈 global num은 num 전역 변수를 함수 안에서 사용 의미
      global num
11
12
      num += 1
                     ♠ 사진 번호를 하나 증가
      if num > 8:
                     №~ 1 사진 번호가 최대 8, 8 넘으면 0번 사진 표시
```







[프로그램 1]의 완성

❖ (이전) 버튼이나 (다음) 버튼을 누르면 사진들을 표시하는 사진 앨범 프로그램

```
14
            num = 0
        photo = PhotoImage(file = "gif/" + fnameList[num])
15
        pLabel.configure(image = photo)
16
        pLabel.image = photo
17
                             ☞ 변경된 사진 번호에 해당하는 이미지 파일로
18
                                   pLabel 변경
19
   def clickPrev():
                             №~② <이전> 버튼을 누르면 처리되는 함수
20
        global num
21
        num -= 1
       if num < 0:
22
23
           num = 8
24
        photo = PhotoImage(file = "gif/" + fnameList[num])
        pLabel.configure(image = photo)
25
        pLabel.image = photo
26
27
   ## 메인 코드 부분 ##
   window = Tk()
   window.geometry("700x500")
   window.title("사진 앨범 보기")
```







[프로그램 1]의 완성

❖ (이전) 버튼이나 (다음) 버튼을 누르면 사진들을 표시하는 사진 앨범 프로그램

```
32
    btnPrev = Button(window, text = "<< 이전", command = clickPrev)
33
    btnNext = Button(window, text = "다음 >>", command = clickNext)
35
    photo = PhotoImage(file = "gif/" + fnameList[0])
    pLabel = Label(window, image = photo)
                                     사진 앨범 보기
38
    btnPrev.place(x = 250, y = 10)
    btnNext.place(x = 400, y = 10)
    pLabel.place(x = 15, y = 50)
42
    window.mainloop()
❸~❷ 버튼 누르면 이 함수와 연결
❸~❸ 프로그램 실행 후 첫 번째 사진 표시
❸~️ 버튼 및 이미지 위치
     place(x=좌표, y=좌표)로 지정
```







■ Cody10-12.py를 수정해서 버튼 사이에 파일명을 표시해 보자.



```
31# 배인 코드 부분
32 window = Tk()
33 window.geometry("700x500")
34 window.title("사진 앨범 보기")
35
36 btnPrev = Button(window, text="<< 이전", command=clickPrev)
37 btnNext = Button(window, text="다음 >>", command=clickNext)
38
39 photo = Photolmage(file="gif/" + fnameList[0])
40 pLabel= Label(window, image=photo)
41
42 nameLabel = Label(window, text=fnameList[0])
43 btnPrev.place(x=250, y=10)
44 nameLabel.place(x=330, y=10)
45 tnNext.place(x=400, y=10)
46 btnNext.place(x=400, y=10)
47 pLabel.place(x=15, y=50)
48
49 window.mainloop()
```