

# 문제해결을 위한 코딩 첫걸음

## 9장 윈도우 프로그램을 만들어 봅시다(1)

서울세계도서관정보대학도서관방문기관으로 선정

한성대학교 노은희 교수



# 학습 목표

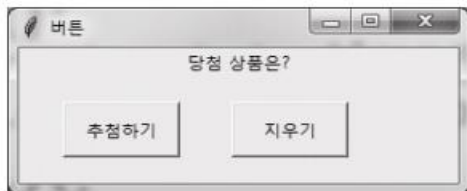
- tkinter 모듈에 대한 이해와 위젯의 배치와 크기를 조절하는 방법을 익힙니다.
- tkinter의 다양한 위젯을 익힙니다.
- tkinter를 이용하여 메뉴 구성하는 방법을 익힙니다.



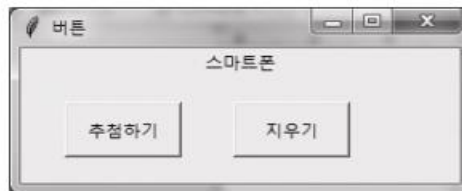
# 이번 장에서 만들 프로그램

## 경품 당첨 프로그램

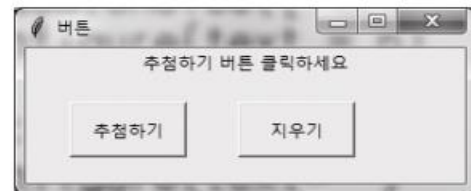
### tkinter 이용



처음 실행 상태



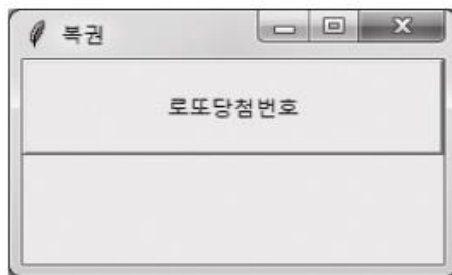
추첨하기 버튼 클릭



지우기 버튼 클릭 상태

## 로또 당첨번호 프로그램

### tkinter 이용

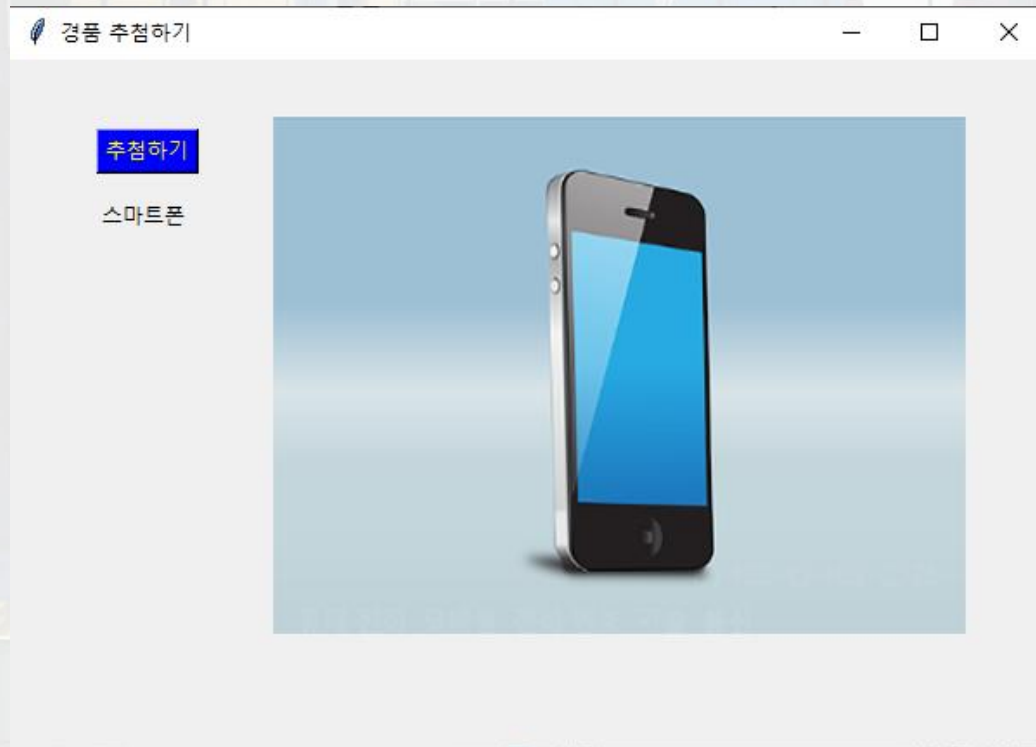


로또당첨번호 버튼 클릭 →



# tkinter

- tkinter(tk interface)
  - 파이썬 설치할 경우 내장되어 제공되는 그래픽 모듈
  - 사용자와 상호작용할 수 있는 그래픽 사용자 인터페이스 (GUI: Graphical User Interface) 개발에 사용
  - 윈도우 생성 및 버튼, 레이블 같은 위젯(widget) 제공
  - import 명령을 통해 tkinter 모듈을 작업 환경으로 가지고 옴

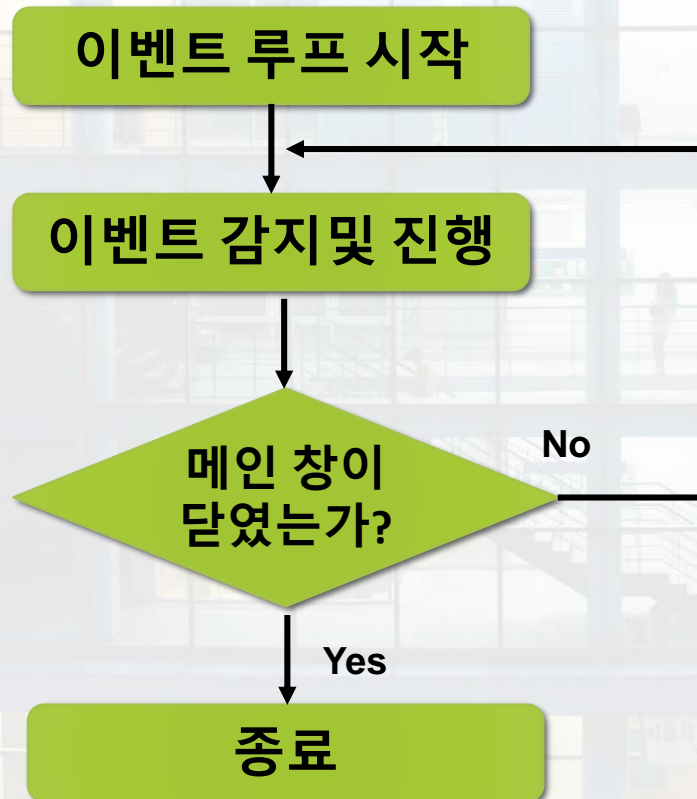




명령어 유형	의미
<code>from tkinter import *</code>	tkinter 모듈을 작업환경에 포함하기
<code>Tk()</code>	루트 윈도우 생성(다른 위젯보다 먼저 생성해야 함)
<code>geometry("가로크기*세로크기")</code>	윈도우 창의 크기
<code>title()</code>	윈도우 창의 제목
<code>mainloop()</code>	윈도우 내부에서 수행되는 마우스 클릭같은 이벤트들이 발생하게끔 유지해주는 함수

# tkinter GUI프그래밍 루프 과정

- ① 유저 인터페이스(User Interface)가 출력
- ② 프로그램은 마우스나 키보드 입력 같은 상호작용을 기다림
- ③ `window.mainloop()`는 이벤트 루프(event loop)를 기다림  
이벤트 루프는 이용자가 메인창을 닫을 때까지 계속해서 이벤트를 진행





# tkinter – 윈도우 창 만들기

- tkinter(tk interface)로 윈도우 창 만들기
  - 윈도우 창 만들기라는 제목의 빈 다이얼로그 화면 생성하기

[소스코드] 3-3.py

```
from tkinter import *
```

```
root= Tk()
```

객체 생성,  
root 이름을 갖는 하나의 윈도우 생성

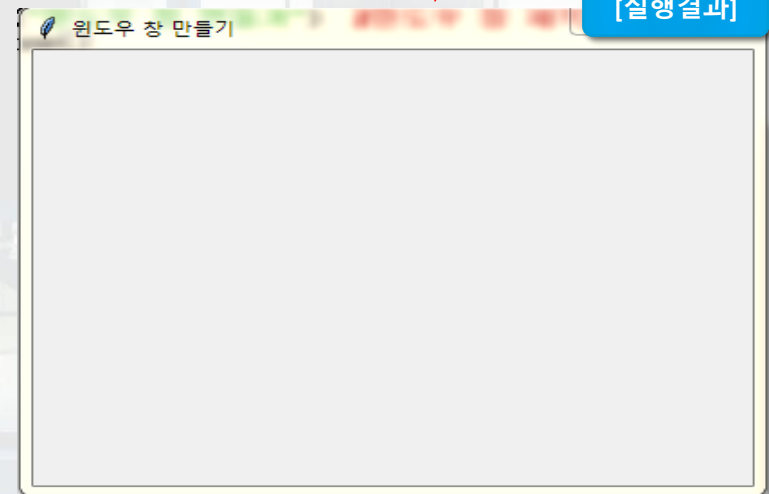
```
root.geometry("230x120")
```

윈도우 크기 (가로 230 X 세로 120)

```
root.title("윈도우 창 만들기")
```

```
root.mainloop()
```

[실행결과]



mainloop()는 이벤트 메시지 루프로써  
키보드나 마우스 같은 다양한 이벤트로부터  
오는 메시지를 받고 전달하는 역할을 한다.

해설 : tkinter 모듈을 import한 다음에는 Tk 클래스 객체(root)를 생성하고, 이 객체의 mainloop() 메서드를 호출



# tkinter 기초 - 배치관리자

- tkinter 배치관리자
  - 윈도우 창에 위젯(widget)을 배치하는 방법
  - 배치관리에 대한 명령이 없으면 위젯이 화면에 표시되지 않음
    - 위젯(widget) :
      - 레이블(Label), 버튼(Button), 엔트리(Entry) 등
      - 윈도우 창에 배치할 수 있는 컴포넌트(Component)
  - 배치관리자 종류
    - pack()
      - 명령을 부여한 순서대로 위젯을 부모 윈도우에 배치 하는 형식
    - grid(row=행번호, column=열번호)
      - 행과 열을 갖는 배열의 형태로 위젯을 배치하는 방식
      - 초기 번호는 0부터 인식함
    - place(x=좌푯값, y=좌푯값)
      - 위젯이 배치될 시작 위치를 절대 좌표 x, y에 값으로 지정
      - 윈도우 크기에 따라 영향을 받음





## 3.3 tkinter 기초 - 컴포넌트 활용

- tkinter 컴포넌트(Component)
  - 라벨(Label), 버튼(Button), 엔트리(Entry) 등과 같이 윈도우 창에 배치할 수 있는 컴포넌트(Component)
- tkinter 모듈 명령어

위젯 명령어	의미
Label(루트윈도우, 옵션)	문자열 표시
Entry(루트윈도우, 옵션...)	문자열 데이터를 입력받을 수 있는 박스형 위젯
Button(루트윈도우, 옵션...)	버튼 지정

### 옵션

- text : 표시 문자열
- height : 줄 수 / 위젯 높이 지정
- width : 글자 수 / 위젯 너비 지정
- bg : 배경색 지정, fg : 전경색 지정
- show : Entry 위젯의 옵션, 패스워드 형식으로 표시할 문자 지정
- command : 버튼이 클릭 된 경우 호출하여 이벤트를 처리할 함수명 지정

# tkinter 기초 - 버튼 위젯과 이벤트 처리

- Button 위젯과 pack() 배치관리자 이용 프로그램
- 버튼(Button) 위젯을 통해 이벤트 처리하여 상호작용 가능
- 이벤트(Event)
  - 마우스의 버튼 또는 키보드의 키(Key)가 눌러진 상태와 같은 사건을 의미
  - 버튼의 이벤트는 정의된 함수에 대해 처리함
- Button() 위젯에 이벤트를 등록하여 처리하는 방법
  - ① 함수 정의 : 이벤트로 처리할 명령문 정의
  - ② 이벤트를 등록할 버튼 위젯 생성  
: 버튼 위젯 옵션 'command = 함수명' 형식으로 등록

# 9.1 tkinter 위젯의 배치와 크기 조절

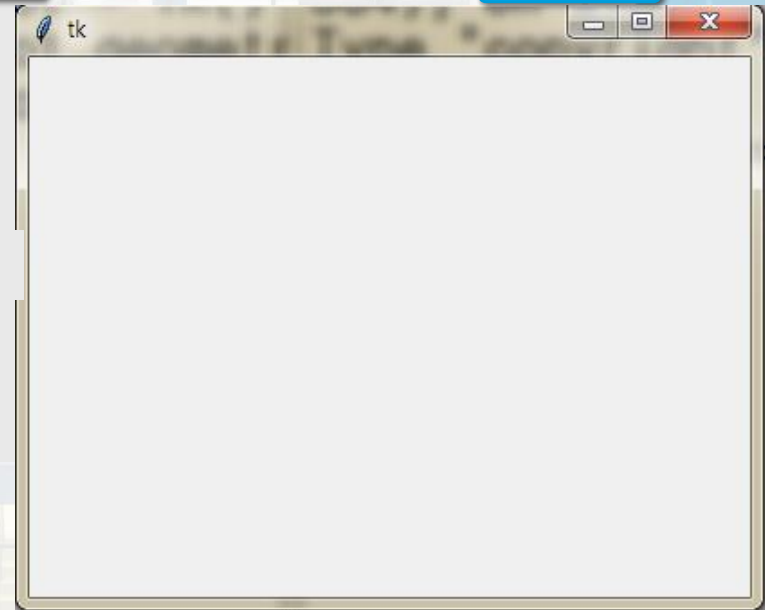
- tkinter(Tk interface) 모듈
  - 윈도우 창, 버튼, 이미지, 체크박스 등
  - 파이썬 기본 패키지에 내장된 모듈
  - 다양한 tkinter 위젯을 사용하여 다양한 GUI 응용프로그램 작성 가능
- 빈 다이얼로그 창을 화면에 표시하는 프로그램

[소스코드] 9-1.py

```
from tkinter import *  
root = Tk()  
root.geometry("400x300")  
root.mainloop()
```

너비 400, 높이 300  
윈도우 창 지정

[실행결과]



# 9.1 tkinter 위젯의 배치와 크기 조절

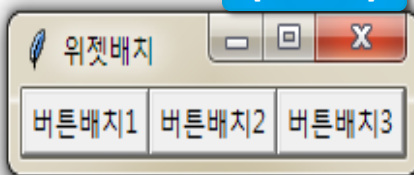
- 위젯 배치와 크기 조정
  - pack 배치 관리자 : 위젯.pack()
    - 위젯을 부모 위젯에 모두 패킹하여 불필요한 공간 제거
    - 정렬 옵션 : side=LEFT / RIGHT / TOP / BOTTOM

[소스코드] 9-2.py

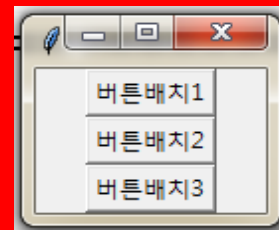
```
from tkinter import *  
root = Tk()  
  
root.title("위젯배치")  
  
btn1 = Button(root, text = "버튼배치1")  
btn2 = Button(root, text = "버튼배치2")  
btn3 = Button(root, text = "버튼배치3")  
  
btn1.pack(side=LEFT)  
btn2.pack(side=LEFT)  
btn3.pack(side=LEFT)  
  
root.mainloop()
```

버튼이 왼쪽부터 차례로 정렬됨

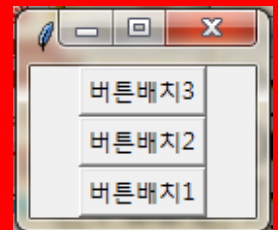
[실행결과]



side=TOP



side=BOTTOM



# 9.1 tkinter 위젯의 배치와 크기 조절

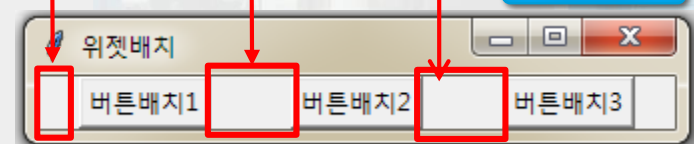
- pack 배치 관리자
  - 폭 조정하기와 위젯 사이의 여백 조절
    - 위젯 사이 여백 지정 : padx = 픽셀값, pady = 픽셀값
    - 위젯 내부 여백 지정 : ipadx = 픽셀값, ipady=픽셀값

[소스코드] 9-4.py

```
from tkinter import *  
root = Tk()  
  
root.title("위젯배치")  
  
btn1 = Button(root, text = "버튼배치1")  
btn2 = Button(root, text = "버튼배치2")  
btn3 = Button(root, text = "버튼배치3")  
  
btn1.pack(side = LEFT, padx = 20)  
btn2.pack(side = LEFT, padx = 20)  
btn3.pack(side = LEFT, padx = 20)  
  
root.mainloop()
```

위젯 사이 여백 지정

[실행결과]





# 9.1 tkinter 위젯의 배치와 크기 조절

- pack 배치 관리자
  - 폭 조정하기와 위젯 사이의 여백 조절
    - 위젯 사이 여백 지정 : padx = 픽셀값, pady = 픽셀값
    - 위젯 내부 여백 지정 : ipadx = 픽셀값, ipady=픽셀값

[소스코드] 9-5.py

```
from tkinter import *
root = Tk()

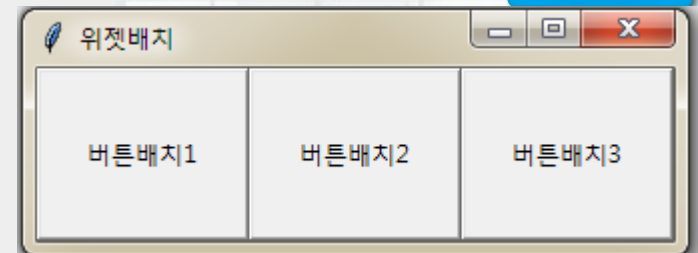
root.title("위젯배치")

btn1 = Button(root, text = "버튼배치1")
btn2 = Button(root, text = "버튼배치2")
btn3 = Button(root, text = "버튼배치3")

btn1.pack(side = LEFT, ipadx = 20, ipady = 30)
btn2.pack(side = LEFT, ipadx = 20, ipady = 30)
btn3.pack(side = LEFT, ipadx = 20, ipady = 30)

root.mainloop()
```

[실행결과]



위젯 내부 여백 지정

# 9.1 tkinter 위젯의 배치와 크기 조절

- pack 배치 관리자
  - 폭 조정하기와 위젯 사이의 여백 조절
    - 위젯 사이 여백 지정 : padx = 픽셀값, pady = 픽셀값
    - 위젯 내부 여백 지정 : ipadx = 픽셀값, ipady=픽셀값

[소스코드] 9-5.py

```
from tkinter import *
root = Tk()

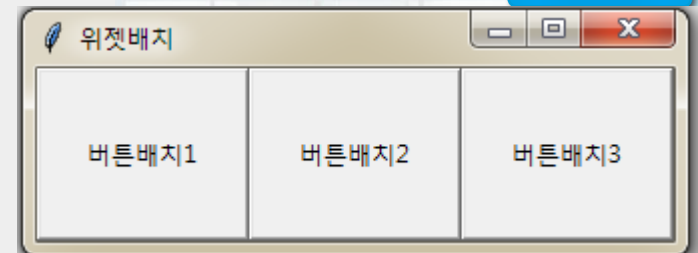
root.title("위젯배치")

btn1 = Button(root, text = "버튼배치1")
btn2 = Button(root, text = "버튼배치2")
btn3 = Button(root, text = "버튼배치3")

btn1.pack(side = LEFT, ipadx = 20, ipady = 30)
btn2.pack(side = LEFT, ipadx = 20, ipady = 30)
btn3.pack(side = LEFT, ipadx = 20, ipady = 30)

root.mainloop()
```

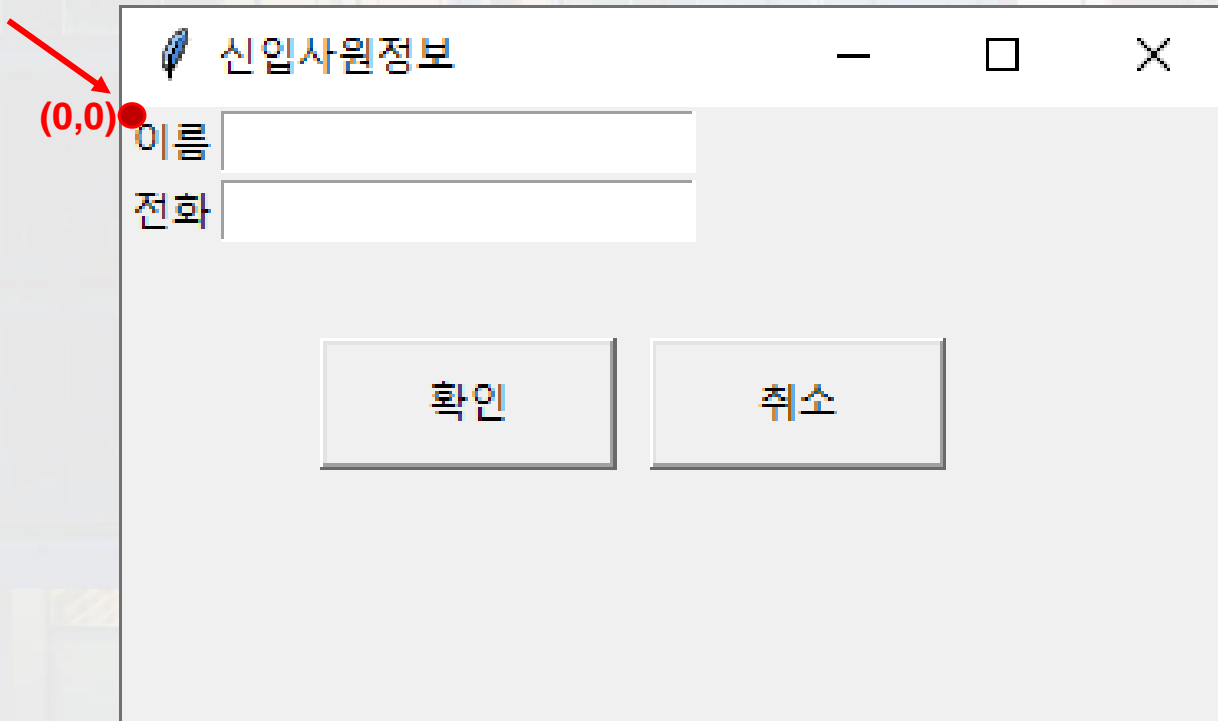
[실행결과]



위젯 내부 여백 지정

# 9.1 tkinter 위젯의 배치와 크기 조절

- place(혹은 absolute) 배치 관리자
  - 절대 배치 관리자, 위젯을 위치를 절대 좌표로 정함.
  - 형식
    - 위젯.place(x=X좌표, y=Y좌표, width=너비, height =높이)
    - width, height가 생략되면 위젯의 기본 크기로 설정



# 9.1 tkinter 위젯의 배치와 크기 조절

- grid
  - 격자 배치 관리자, 위젯을 테이블 레이아웃에 배치하는 것
  - 지정된 row, column에 위젯을 배치
  - 형식
    - 위젯.grid()

	column(0)	column(1)	column(2)
row(0)	(0,0)	(0,1)	(0,2)
row(1)	(1,0)	(1,1)	(1,2)
row(2)	(2,0)	(2,1)	(2,2)

# 9.1 tkinter 위젯의 배치와 크기 조절

- grid와 place를 이용하여 만든 프로그램

시작

[소스코드] 9-6.py

```
from tkinter import *
root = Tk()
root.title("신입사원정보")
root.geometry("400x300")

lbl1 = Label(root, text = "이름")
lbl2 = Label(root, text = "전화")
lbl1.grid(row=0, column=0)
lbl2.grid(row=1, column=0)

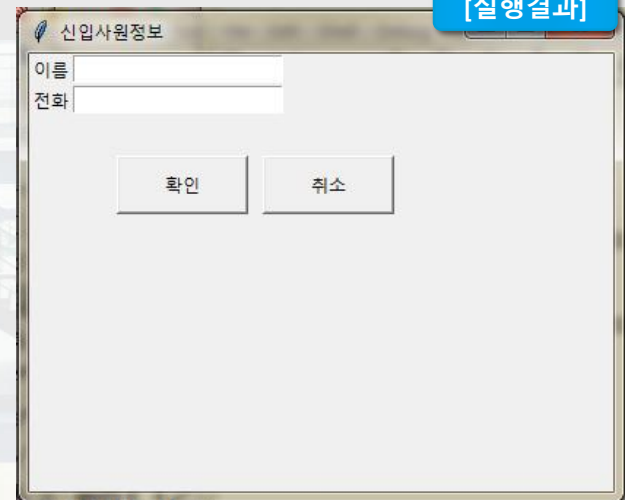
ent11 = Entry(root)
ent12 = Entry(root)
ent11.grid(row=0, column=1)
ent12.grid(row=1, column=1)
```

이어서

```
btn1 = Button(root, text ="확인")
btn2 = Button(root, text = "취소")
btn1.place(x=60,y=70, width =90, height=40)
btn2.place(x=160,y=70, width =90, height=40)

root.mainloop()
```

[실행결과]







## 마무리

- tkinter 모듈에 대한 이해와 tkinter 위젯의 배치와 크기를 조절하는 방법을 학습하였습니다.