CHAPTER 03

그래픽에 대해 알아봅시다.

문제해결을 위한 파이썬 첫걸음

이미향 교수

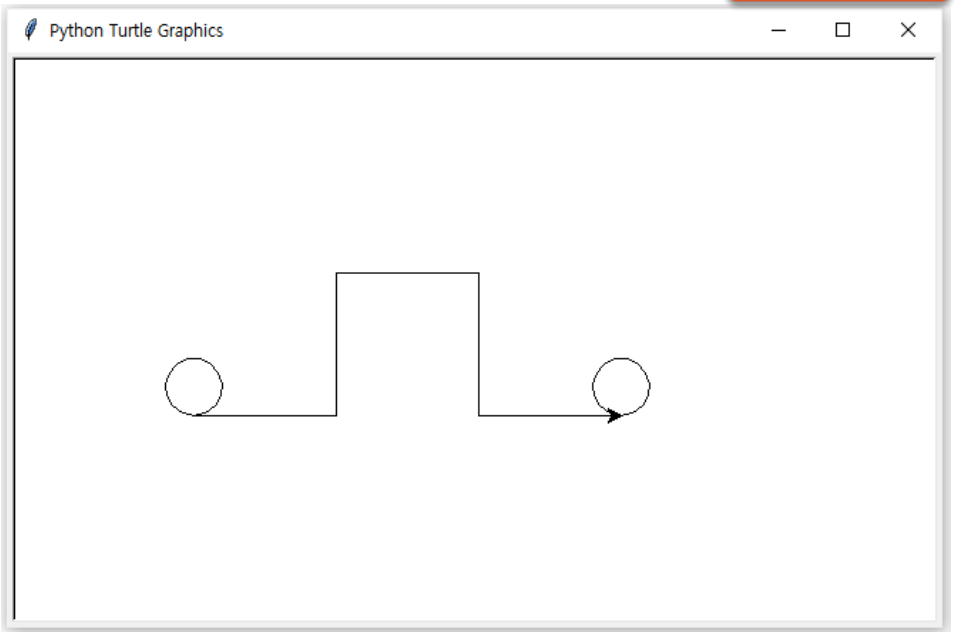
smilequeen@gmail.com

실습_코딩하기2

실습

- 터틀 그래픽을 활용하여 다음 실행결과와 같은 그림이 그려지도록 프로그램을 작성해 봅니다.
- 원의 반지름_20
- 한 선의 길이_100
- 최초의 위치_x축 -200, y축 -50

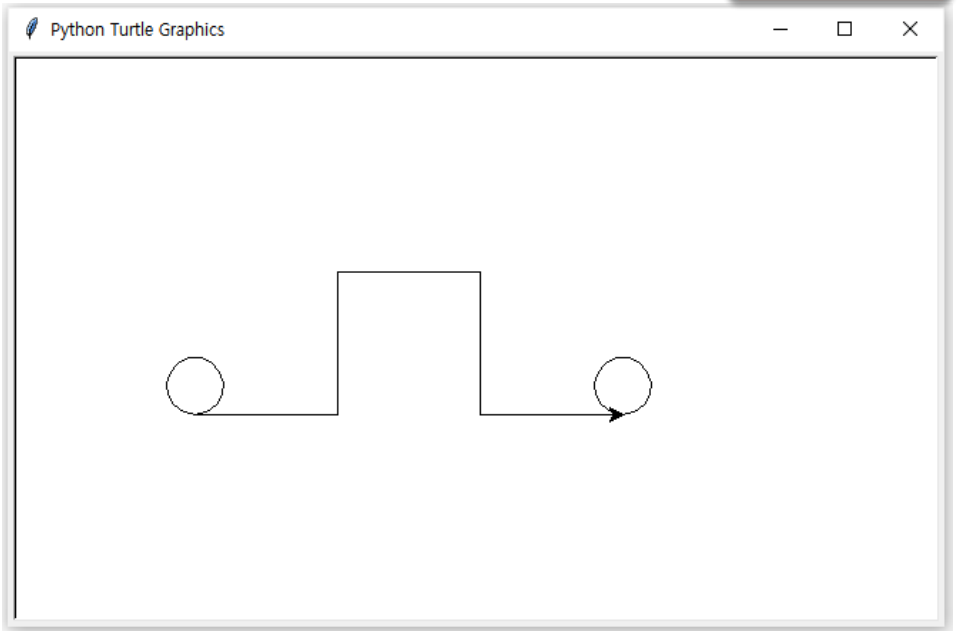
[실행결과]



실습

```
1  import turtle          #turtle 모듈 불러오기
2  t = turtle.Turtle()    #객체 생성하기
3
4  t.up()                 #펜 들기
5  t.goto(-200,-50)       #처음 위치로 이동하기
6  t.down()               #펜 내기리
7  t.circle(20)           #원 그릭
8  t.forward(100)         #직진
9  t.left(90)             #90도 좌회전
10 t.forward(100)         #직진
11 t.right(90)            #90도 우회전
12 t.forward(100)         #직진
13 t.right(90)            #90도 우회전
14 t.forward(100)         #직진
15 t.left(90)             #90도 좌회전
16 t.forward(100)         #직진
17 t.circle(20)           #원 그리기
```

[실행결과]

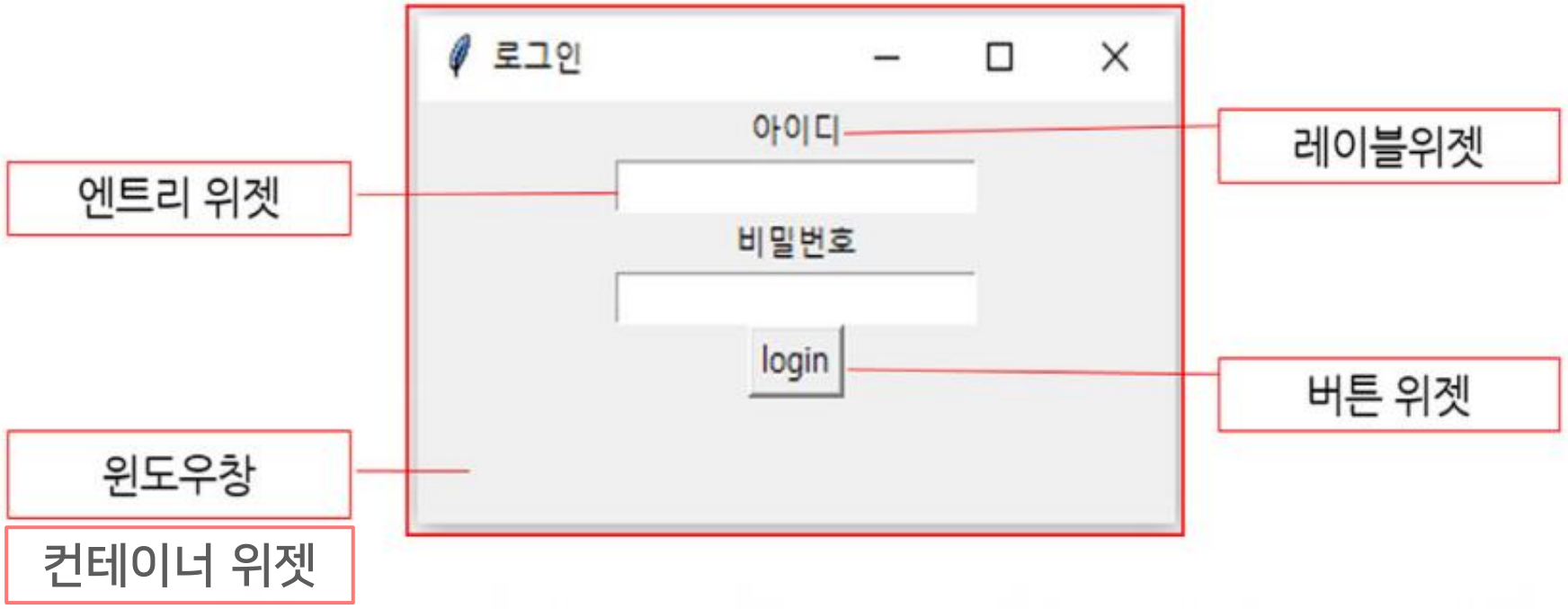


3.3 tkinter 기초

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

3.3 tkinter 기초 - 윈도우창

- 위젯(widget:window gadget)은 레이블, 버튼과 같은 GUI기반 운영체제에서 사용하는 각종 시각적인 요소
- 레이블과 버튼 같은 위젯을 이용하여 상호 작용하는 프로그램 을 작성할 수 있음



3.3 tkinter 기초 - 프로그래밍 순서

3.3 tkinter 기초

tkinter 모듈 명령어

명령어 유형	의미
from tkinter import *	tkinter 모듈을 작업환경에 포함하기
Tk()	루트 윈도우 생성(다른 위젯보다 먼저 생성해야 함)
geometry("가로크기*세로크기")	윈도우 창의 크기
title()	윈도우 창의 제목
mainloop()	이벤트 메시지 루프(loop)

3.3 tkinter 기초 - 윈도우 창 만들기

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

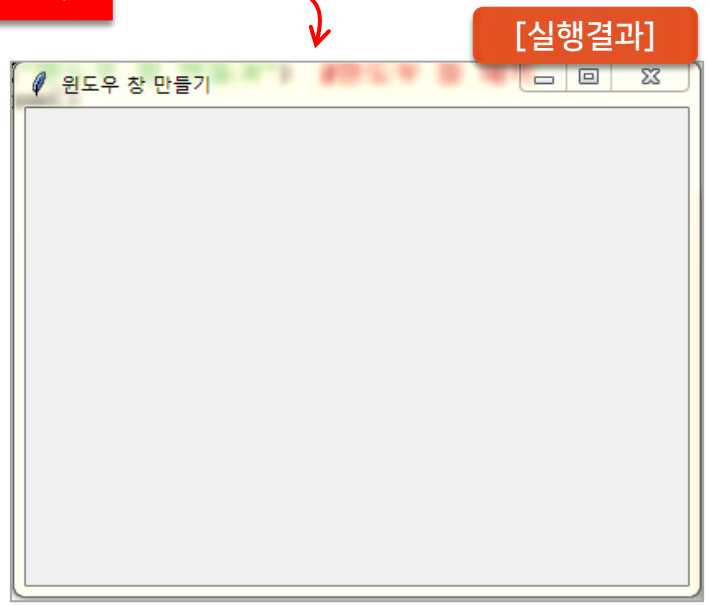
[소스코드] 3-3.py

```
from tkinter import *
root= Tk()
root.geometry("230x120")
root.title("윈도우 창 만들기")
root.mainloop()
```

객체 생성,
root 이름을 갖는 하나의 윈도우 생성

윈도우 크기 (가로 230 X 세로 120)

[실행결과]



3.3 tkinter 기초 - 배치관리자

- **tkinter 배치관리자**
  - 윈도우 창에 위젯(widget)을 배치하는 방법
  - 배치관리에 대한 명령이 없으면 위젯이 화면에 표시되지 않음.
    - 위젯(widget) :
      - 라벨(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

3.3 tkinter 기초 - 배치관리자

- **화면에 위젯의 배치를 담당하는 객체**

① 압축(pack) 배치 관리자

② 격자(grid) 배치 관리자_테이블 형태로 배치

③ 절대(place) 배치 관리자_절대 위치(x,y 매개변수)를 사용하여 배치

배치 관리자_pack

① pack

- pack 메소드는 가장 처음 선언한 pack부터 배치됩니다.
  pack()은 grid()와 같이 사용될 수 없고 place()와 같이 사용할 수 있습니다.

구문

```
위젯.pack(side)
```

side = TOP, BOTTOM, LEFT, RIGHT 위젯을 추가할 위치를 지정합니다.

배치 관리자_grid

② grid

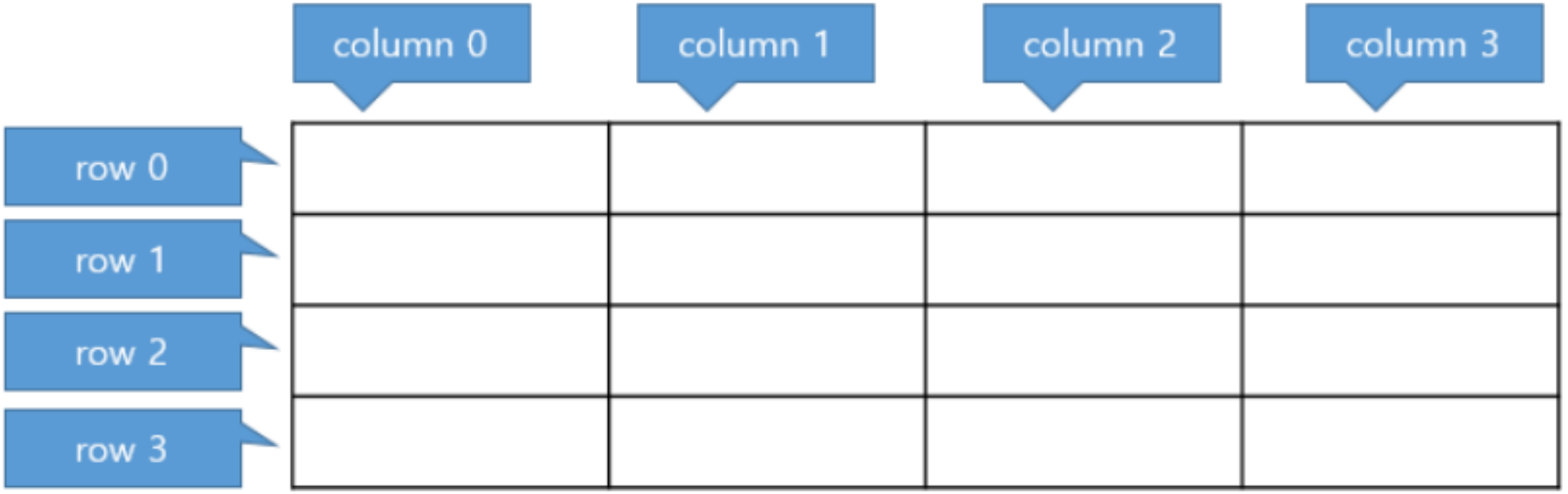
- grid는 위젯들을 테이블 레이아웃에 배치하는 것으로 지정된 row, column에 위젯을 놓습니다.
  위젯.grid() 메소드를 사용합니다.

구문

```
위젯.grid( grid_options )
```

- widget.grid(파라미터1, 파라미터2, 파라미터3, ...)을 사용하여 해당 윈도우 창에 표시할 위젯의 배치 속성을 설정할 수 있습니다.
- grid_options
  column – 위젯을 넣을 열
  row -위젯을 넣을 행

배치 관리자_grid



배치 관리자_place

③ place

- Place는 절대 위치로 배치되며, 윈도우의 크기가 변경되면 위젯 위치들이 변경되지 않습니다. place()은 pack(), grid()와 같이 사용할 수 있습니다.

구문

```
위젯.place(place_options)
```

x, y - 픽셀 단위의 수평 및 수직 위치 지정

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

3.3 tkinter 기초 - 따라 해보기

- 라벨(Label) 위젯과 pack() 배치관리자 이용 프로그램
  - 라벨(Label) 위젯에 텍스트 표시하기

[소스코드] 3-4.py

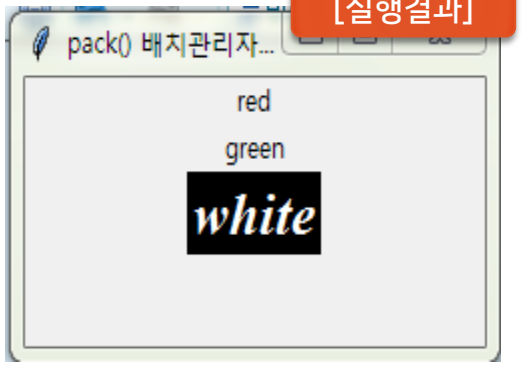
```
from tkinter import *
root = Tk()
root.title("pack() 배치 관리자와 Label() 예제")
root.geometry("230x120")
lbl1 = Label(root, text = "red")
lbl1.pack()
lbl2 = Label(root, text = "green")
lbl2.pack()
lbl3 = Label(root, text = "white", font="Times 20 bold italic",
             fg="white", bg="black")
lbl3.pack()
root.mainloop()
```

lbl1 이름을 갖는 Label객체 생성, Label 표시 글자 : red

lbl1이름을 갖는 Label 위젯을 윈도우에 배치

lbl3 이름을 갖는 Label객체 생성, Label 표시 글자 : white
글꼴 Times, 크기 20, 굵게, 이탤릭 형태,
글자색 white, 배경색 black 으로 표시글자 형식 지정

[실행결과]



3.3 tkinter 기초 - 따라 해보기

- **엔트리(Entry), 라벨(Label) 위젯과 pack() 배치관리자 이용 프로그램**
  - 라벨(Label) 위젯과 엔트리(Entry) 위젯에 텍스트 표시하기

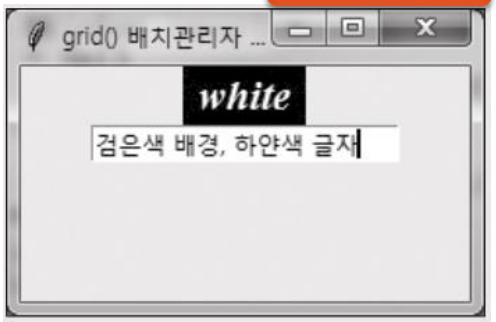
[소스코드] 3-5.py

```
from tkinter import *
root = Tk()
root.title("pack() 배치 관리자 예제")
root.geometry("230x120")
lbl1 = Label(root, text = "white", font="Times 16 bold italic",
             fg="white", bg="black")
ent1 = Entry(root, width=22)
lbl1.pack()
ent1.pack()
root.mainloop()
```

ent1이름을 갖는 Entry객체 생성, 엔트리 상자의 크기 22

생성된 lbl1 Label 위젯과 ent1 Entry 위젯을 윈도우에 배치

[실행결과]



3.3 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

실습_코딩하기3

프로그램2_이벤트 처리 프로그램

- **Button 위젯과 pack() 배치관리자 이용, 이벤트 처리 프로그램**
  - '인사하기' 버튼을 클릭하면 "Hello~~~" 문자열 출력하기

[소스코드] 3-6.py

```
from tkinter import *
def proc():
    print("Hello~~~")
root = Tk()
root.title("버튼과 이벤트")
root.geometry("230x120");
btn = Button(root, text = "인사하기", command=proc)
btn.pack(side=BOTTOM)
root.mainloop()
```

① 이벤트 처리 위해 proc() 함수 정의

② 정의한 함수를 이벤트 처리 가능하도록 등록

버튼 위젯 btn이 윈도우창 아래에 위치하도록 배치

[실행결과]