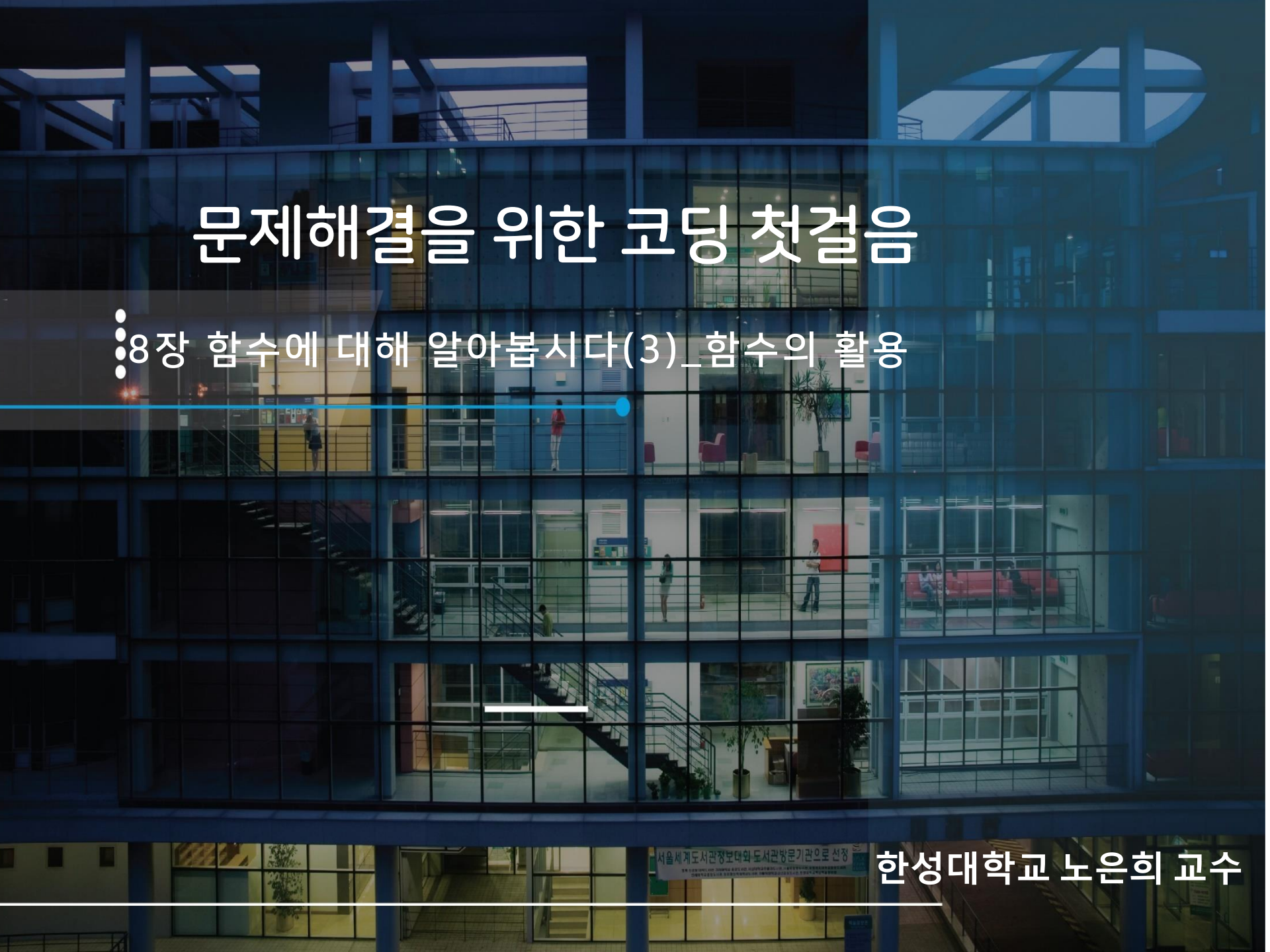


문제해결을 위한 코딩 첫걸음

8장 함수에 대해 알아봅시다(3)_함수의 활용

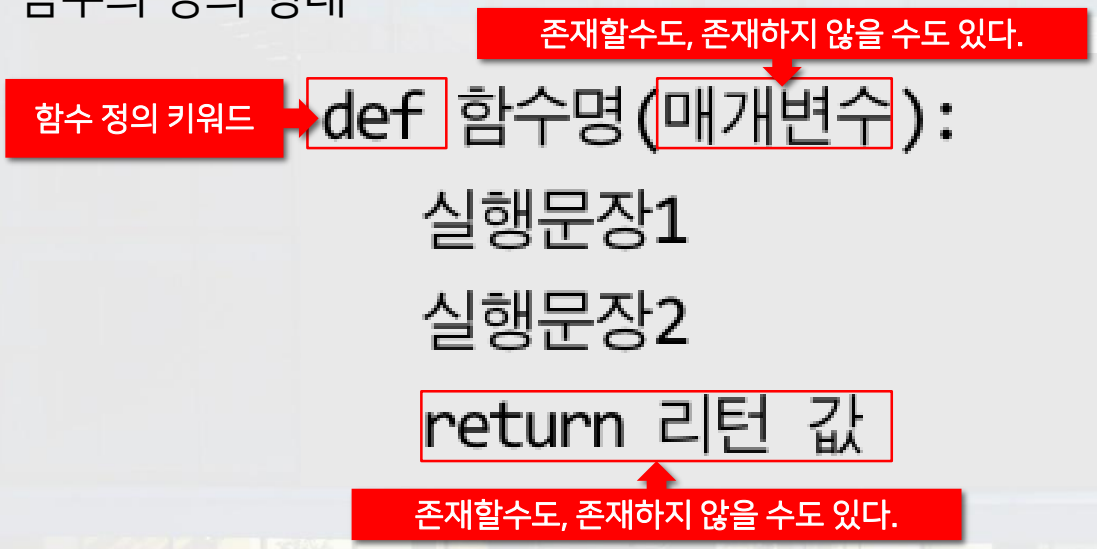
한성대학교 노은희 교수

학습 목표

- 함수를 활용하여 다양한 프로그램을 만들 수 있습니다.

함수(function)란?

- 함수(function)
  - 일정한 기능을 수행하도록 만든 프로그램의 코드 블록
  - 함수의 사용
    - 함수의 정의 : def 키워드 사용
    - 함수의 호출 : 정의한 함수의 이름을 호출
  - 함수의 정의 형태



함수의 값 반환

- 함숫값 반환
  - 정의된 함수가 실행된 결과로 얻어진 값을 호출 함수에 반환
  - 그 값을 함수를 호출한 쪽에서 사용
  - 정의된 함수의 결괏값을 반환하기 위해서는 반드시 return문을 사용

[소스코드] 8-6.py

```
def convInch2Cm(inch) :
    cmV = inch * 2.54
    return cmV

cmVal = convInch2Cm(10)
print(str(10)+'inch', "=>", str(cmVal)+'cm')
```

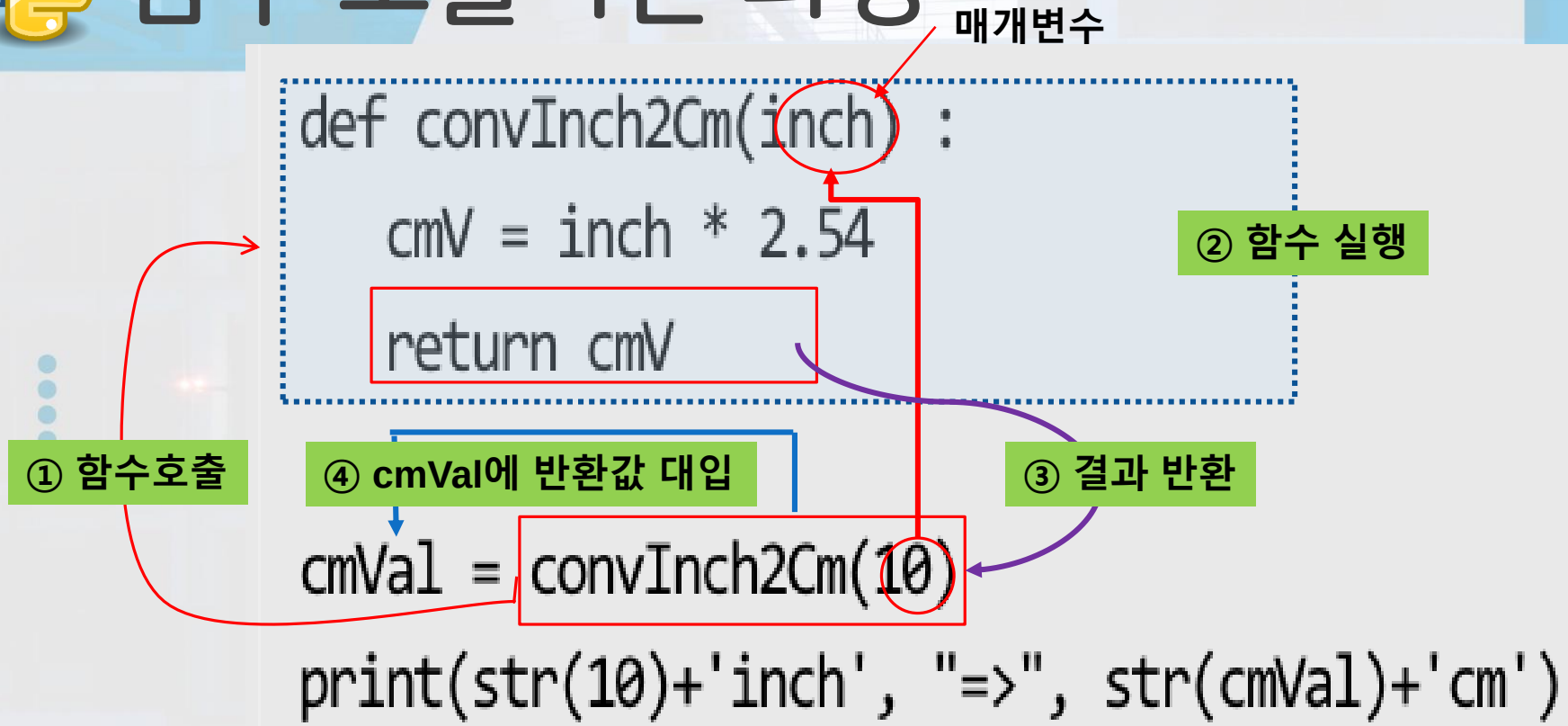
함수 호출

계산 결과를 함수에 되돌림

[실행결과]

```
10inch => 25.4cm
```

함수 호출하는 과정



① 함수호출 : convInch2Cm(10)으로 함수를 호출하면 10이라는 숫자를 가지고 convInch2Cm( ) 함수를 호출합니다. 이때 매개변수인 inch에 10이 할당됩니다.
② 함수실행 : inch * 2.54를 계산하고 cmV에 대입한 후 return cmV문장에 의해 이 함수를 호출했던 곳으로 돌아갑니다.
③ 결과 반환 : 함수를 실행하여 얻은 cmV값(25.4)를 함수를 호출했던곳으로 돌려줍니다.
④ 반환된 값 25.4를 변수 cmVal에 대입합니다.

8.4 함수의 활용 – 난수의 종류

- 난수의 적용과 종류
  - 난수의 적용 : import random 선언 필요
  - random.random() : 0.0에서 1.0 미만의 실수를 반환

```
>>> import random
>>> random.random()
>>> 0.938482478210824
```

  - random.randint(a,b) : a에서 b까지의 정수를 반환

```
>>> import random
>>> random.randint(1,10)
6
```

8.4 함수의 활용 – 난수의 종류

- 난수의 종류
  - random.choice(seq)
    - 문자열 또는 리스트 안에서 하나의 항목을 랜덤하게 선택 반환

```
>>> import random
>>> items = random.choice(["냉장고","세탁기","노트북","스마트폰"])
>>> print("경품은",items)
```

  - random.sample(범위,개수)
    - 지정된 범위에서 지정한 개수만큼 랜덤하게 추출 반환

```
>>> import random
>>> lotto = random.sample(range(1,46),6)
>>> sorted(lotto)
>>> print(lotto)
[1, 4, 11, 12, 20, 32]
```

8.4 함수의 활용

- 난수의 종류
  - random.shuffle(a)
    - 리스트를 인자로 받아 무작위로 순서를 변경
    - 원본의 리스트 객체 순서 변경

```
>>> import random
>>> a = [1,2,3,4,5,6,7,8,9,10]
>>> random.shuffle(a)
>>> print(a)
[7, 6, 10, 5, 2, 4, 8, 9, 1, 3]
```

모듈에 대해 알아보기

- 모듈(Module) : 함수의 집합, 별도의 파일에 함수들을 모아놓은것
- 모듈을 사용하기 위해서는 import 모듈명
- 함수를 호출할때는 '모듈명.함수이름()'형식으로 모듈이름을 앞에 붙인다.
- import는 외부 파일에서 작성한 함수를 현재 코드에서 사용할 수 있도록 해주는 예약어

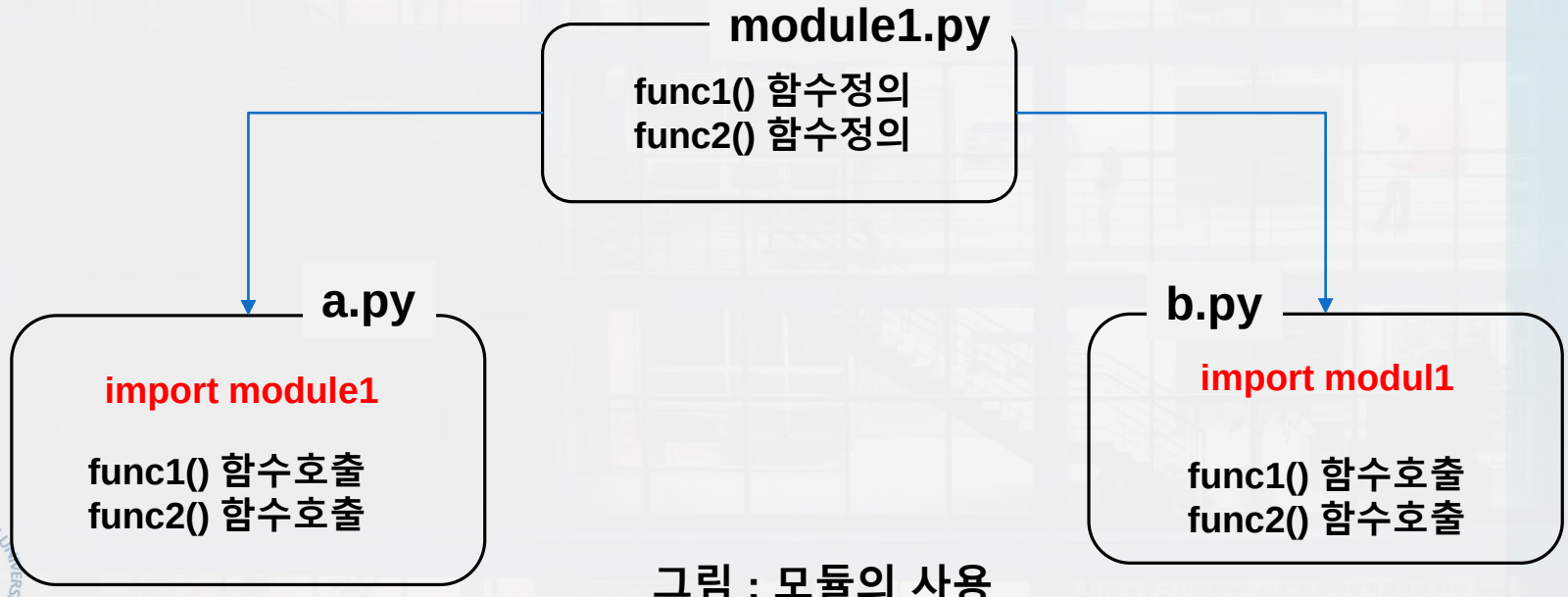


그림 : 모듈의 사용

8.4 함수의 활용 – 이벤트 처리 함수

- 이벤트(Event) : 키보드나 마우스를 누르는 것
- 윈도프로그램에서 mainloop()함수 : 이벤트가 발생하는 것을 기다리는 함수

터틀 객체에 대한 마우스/키보드 이벤트처리 메서드/함수

함수/메서드	설명
listen()	키보드 이벤트를 받기 위해 대기
onkey(fun,key)	키를 fun() 함수에 바인딩 key : “Up”, “Down”, “Left”, “Right” 등
onclick(fun,btn=1,add=None) onscreenclick(fun,btn=1,add=None)	마우스 클릭을 fun(x,y) 함수에 바인딩 btn =1(마우스왼쪽 버튼) btn =3(마우스 오른쪽 버튼) add = True면 새로운 바인딩 추가(add) add=False면 이전 바인딩을 대체(replace)
mainloop()	이벤트 루프 시작, tkinter의 메인 루프 함수를 호출

8.4 함수의 활용 – 클릭될 때 별 그리기

- 마우스 왼쪽 버튼 클릭할 때마다 별 그리기
  - 윈도우 화면에서 마우스의 왼쪽 버튼이 클릭되는 곳마다 별을 그리는 프로그램

[소스코드] 8-10.py

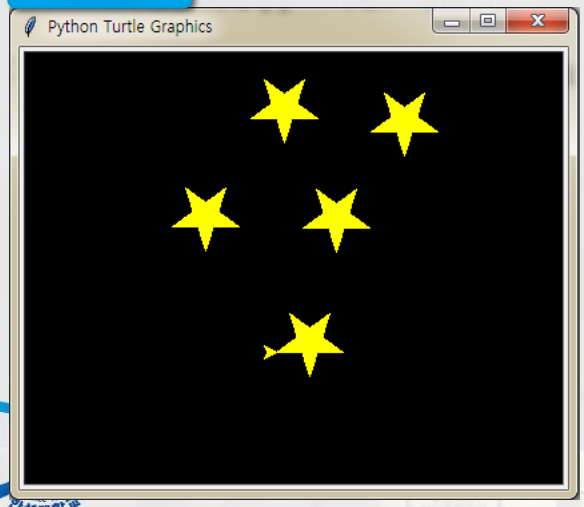
```
import turtle
t = turtle.Turtle()

def star(length):
    for n in range(5):
        t.forward(length)
        t.left(144)

def s_draw(x, y):
    turtle.bgcolor("black")
    t.up()
    t.goto(x, y)
    t.down()
    t.color("yellow")
    t.begin_fill()
    star(50)
    t.end_fill()

scr = turtle.Screen()
scr.onscreenclick(s_draw)
```

[실행결과]



8.4 함수의 활용 – 마우스로 그림 그리기

- 마우스와 키보드를 활용하여 그림 그리기
  - 화살표 키와 'Delete' 키를 사용하여 사용자가 직접 그림을 그릴 수 있는 프로그램

[소스코드] 8-11.py

```
import turtle
t = turtle.Turtle()
scr = turtle.Screen()
t.shape("turtle")
t.pensize(3)

def s_draw(x, y):
    t.goto(x, y)
def turnleft():
    t.left(15)
def turnright():
    t.right(15)
scr.onscreenclick(s_draw)
scr.onkey(t.penup,"Up")
scr.onkey(t.pendown,"Down")
scr.onkey(turnleft,"Left")
scr.onkey(turnright,"Right")
scr.onkey(t.clear,"Delete")
scr.listen()
scr.mainloop()
```

[실행결과]

마무리 및 다음 차시 예고

- 이번 수업에서는 함수를 활용하여 다양한 프로그래을 만들어보았습니다.