



부록 B에서는 이 책의 예제 프로그램에서 사용했거나 사용하지 않았더라도 프로그램 작성할 때 참고하면 좋을 만한 파이썬 표준 라이브러리 함수를 정리했습니다.

표 B-1

turtle 모듈
(import turtle as t)

함수	설명	사용 예
forward(거리) / fd(거리)	거북이가 앞으로 이동합니다.	t.forward(100) # 거북이가 100만큼 앞으로 이동합니다.
backward(거리) / back(거리)	거북이가 뒤로 이동합니다.	t.back(50) # 거북이가 50만큼 뒤로 이동합니다.
left(각도) / lt(각도)	거북이가 왼쪽으로 회전합니다.	t.left(45) # 거북이가 45도 왼쪽으로 회전합니다.
right(각도) / rt(각도)	거북이가 오른쪽으로 회전합니다.	t.right(45) # 거북이가 45도 오른쪽으로 회전합니다.
circle(반지름)	현재 위치에서 원을 그립니다.	t.circle(50) # 반지름이 50인 원을 그립니다.
down() / pendown()	펜(잉크 묻힌 꼬리)을 내립니다.	t.down() # 이제 움직이면 그림이 그려집니다.
up() / penup()	펜(잉크 묻힌 꼬리)을 올립니다.	t.up() # 거북이가 움직여도 선이 그려지지 않습니다.
shape("모양")	거북이 모양을 바꿉니다.	t.shape("turtle") # 진짜 거북이 모양으로 지정합니다. t.shape("arrow") # 화살표 모양의 거북이로 지정합니다. ※ 거북이 모양으로 "circle", "square", "triangle"도 사용할 수 있습니다.
speed(속도)	거북이 속도를 바꿉니다.	t.speed(1) # 가장 느린 속도 t.speed(10) # 빠른 속도 t.speed(0) # 최고 속도
pensize(굵기) / width	펜 굵기를 바꿉니다.	t.pensize(3) # 굵은 선으로 선을 그립니다.
color("색 이름")	펜 색을 바꿉니다.	t.color("red") # 빨간색으로 선을 그립니다.
bgcolor("색 이름")	화면의 배경색을 바꿉니다.	t.bgcolor("black") # 배경색을 흰색에서 검은색으로 바꿉니다.
fillcolor("색 이름")	도형 내부를 칠하는 색을 바꿉니다.	t.fillcolor("green") # 녹색으로 도형 내부를 칠합니다. ※ 색상을 따로 지정하지 않으면 현재 색으로 칠합니다.

<code>begin_fill()</code>	도형 내부를 색칠할 준비를 합니다.	<code>t.begin_fill()</code> # 거북이 움직임을 색칠할 준비를 합니다.
<code>end_fill()</code>	도형 내부를 색칠합니다.	<code>t.end_fill()</code> # <code>begin_fill()</code> 이후부터 지금까지 그려진 그림에 맞춰 내부를 색칠합니다.
<code>showturtle()</code> / <code>st()</code>	거북이를 화면에 표시합니다.	<code>t.st()</code> # 거북이를 화면에 표시합니다(원래 상태).
<code>hideturtle()</code> / <code>ht()</code>	거북이를 화면에서 가립니다.	<code>t.ht()</code> # 거북이를 숨깁니다.
<code>clear()</code>	거북이를 그대로 둔 채 화면을 지웁니다.	<code>t.clear()</code>
<code>reset()</code>	화면을 지우고 거북이도 원래 자리와 상태로 되돌립니다.	<code>t.reset()</code>
<code>pos()</code> / <code>position()</code>	거북이의 현재 위치(좌표)를 구합니다(x, y 둘 다).	<code>t.pos()</code>
<code>xcor()</code> , <code>ycor()</code>	거북이의 x 좌표나 y 좌표를 구합니다(x, y 중 하나만).	<code>a = t.ycor()</code> # 거북이의 y 좌표를 구해 a에 저장합니다.
<code>goto(x, y)</code> , <code>setpos(x, y)</code>	거북이를 특정 위치(좌표)로 보냅니다(x, y 둘 다).	<code>t.goto(100, 50)</code>
<code>setx(x)</code> , <code>sety(y)</code>	거북이의 x 좌표나 y 좌표를 지정한 위치로 이동합니다(x, y 중 하나만).	<code>t.sety(50)</code> # 거북이의 y 좌표를 50만큼 이동합니다. x 좌표는 그대로 둡니다.
<code>distance(x, y)</code>	현재 거북이가 있는 위치에서 특정 위치까지 거리를 구합니다.	<code>d = t.distance(100, 100)</code> # 현재 위치에서 (100, 100)까지의 거리를 구해서 d에 저장합니다.
<code>heading()</code>	거북이가 현재 바라보는 각도를 구합니다.	<code>ang = t.heading()</code>
<code>towards(x, y)</code>	현재 거북이가 있는 위치에서 특정 위치까지 바라보는 각도를 구합니다.	<code>ang = t.towards(10, 10)</code> # 현재 위치에서 (10, 10)까지 가는 데 필요한 각도를 구해 ang에 저장합니다.
<code>setheading(각도)</code> / <code>seth(각도)</code>	거북이가 바라보는 방향을 바꿉니다.	<code>t.setheading(90)</code> # 거북이가 화면 위쪽을 바라봅니다. ※ 거북이가 오른쪽을 바라볼 때의 각도가 0°이며, 시계 반대 방향으로 돌면서 각도가 커집니다.
<code>home()</code>	거북이의 위치와 방향을 처음 상태로 돌립니다.	<code>t.home()</code> # 거북이가 화면 가운데인 (0, 0)에서 오른쪽(0도)을 바라봅니다.
<code>onkeypress(함수, "키 이름")</code>	키보드를 눌렀을 때 실행할 함수를 정합니다.	<code>def f():</code> <code>t.forward(10)</code> <code>t.onkeypress(f, "Up")</code> # 위쪽 방향키 ↑를 누르면 f 함수를 호출합니다(f 함수는 거북이를 10만큼 앞으로 이동시킵니다).

onscreenclick(함수)	마우스 버튼을 눌렀을 때 실행할 함수를 정합니다.	t.onscreenclick(t.goto) # 마우스 버튼을 누르면 앞에서 정의한 goto 함수를 호출합니다(goto 함수는 거북이를 마우스 버튼을 누른 위치로 이동시킵니다).
ontimer(함수, 시간)	일정한 시간이 지난 뒤 실행할 함수를 정합니다.	def f(): t.forward(10) t.ontimer(f, 1000) # 1000밀리초(1초) 후에 f 함수를 호출합니다(f 함수는 거북이를 10만큼 앞으로 이동시킵니다.)
listen()	사용자 입력이 잘 처리되도록 거북이 그래픽 창에 포커스를 줍니다.	t.listen()
title("창 이름")	거북이 그래픽 창의 이름을 지정합니다.	t.title("welcome") # 거북이 그래픽 창의 이름이 Untitled에서 welcome으로 바뀝니다.
write("문자열")	현재 거북이 위치에 문자를 출력합니다.	t.write("Hello") # 현재 거북이 위치에 Hello를 출력합니다. t.write("Hello", False, "center", ("", 20)) # 현재 거북이 위치에 가운데 정렬로 크기가 20인 Hello를 출력합니다.

표 B-2

random 모듈(import random)

함수	설명	사용 예
random()	0.0 이상 1.0 미만의 소수 한 개를 임의로 뽑아 줍니다.	>>> random.random() 0.9013164080966248
randint(a, b)	a 이상 b 이하의 정수 한 개를 임의로 뽑아 줍니다.	>>> random.randint(1, 10) 2 # 1과 10을 모두 포함하여 선택합니다.
choice(seq)	주어진 seq 안에서 한 개의 값을 임의로 뽑습니다.	>>> a = [1, 2, 3, 4, 5] >>> random.choice(a) 3
randrange(n) / randrange(a, b)	0부터 n-1까지 중 한 개 / a부터 b-1까지 중 한 개의 정수를 임의로 뽑습니다. range() 함수와 선택 범위가 같습니다.	>>> random.randrange(10) 6 # 0, 1, 2, ..., 9 중 하나를 뽑습니다.

표 B-3

time 모듈(import time)

함수	설명	사용 예
time()	1970년 1월 1일 자정 12시부터 현재 시각까지 몇 초 지났는지 알려 줍니다.	<pre>>>> time.time() 1442101084.201666</pre> <p># 1970년 1월 1일 자정부터 현재 시각까지 약 14억 4천만초 지났습니다.</p> <p>※ 이 자체로는 큰 의미가 없는 숫자지만, 두 시각의 time.time() 값을 구해서 빼면 두 시각의 차이를 구할 수 있습니다.</p>
sleep(secs)	주어진 시간(초)만큼 프로그램의 실행을 멈춥니다.	<pre>>>> time.sleep(5) >>></pre> <p># 5초 동안 멈춘 다음 >>>이 다시 표시됩니다.</p>
localtime()	time.time()을 사용해서 구한 값 (초 단위)을 연월일/시분초로 표시합니다.	<pre>>>> time.localtime(time.time()) time.struct_time(tm_year=2015, tm_mon=9, tm_mday=13, tm_hour=8, tm_min=42, tm_sec=28, tm_wday=6, tm_yday=256, tm_isdst=0)</pre> <p># 1970년부터 약 14억 4천만 초가 흐른 지금은 '2015년 9월 13일 오전 8시 42분 28초'입니다.</p>

표 B-4

calendar 모듈(import calendar)

함수	설명	사용 예
prmonth(y, m)	지정한 연과 월에 해당하는 달력을 출력합니다.	<pre>>>> calendar.prmonth(2015, 9) September 2015 Mo Tu We Th Fr Sa Su 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30</pre>
prcal(y)	지정한 해의 전체 달력을 출력합니다.	<pre>>>> calendar.prcal(2015) # 2015년 달력을 출력합니다.</pre>