



이번 장에서 만드 프로그램

(1) 거북이 경주 게임을 작성해본다. 난수를 이용하여야 한다.





난수 발생하기

- 난수(random number)는 게임과 시뮬레이션에 필수적이다.
- 파이썬은 random 모듈을 통하여 난수 발생을 지원한다.

```
>>> import random
```

```
>>> random.random() # 0.0부터 1.0보다 작은 실수 난수  
0.8345121533431609
```

```
>>> random.randint(1, 100) # 1부터 100 사이의 정수 난수  
49
```



리스트에서 랜덤하게 선택

```
>>> import random  
>>> random.choice('abcdefghij') # 하나의 랜덤한 항목  
'c'
```

```
>>> items = [1, 2, 3, 4, 5, 6, 7]  
>>> random.shuffle(items)  
>>> items  
[7, 3, 2, 5, 6, 4, 1]
```

항목을 섞는 기능



터틀 그래픽 함수들

함수	인수	설명
forward()	픽셀값	거북이를 지정된 거리만큼 앞으로 이동한다.
backward()	픽셀값	거북이를 지정된 거리만큼 뒤로 이동한다.
right()	각도	거북이를 시계방향으로 회전시킨다.
left()	각도	거북이를 반시계방향으로 회전시킨다.
penup()	None	거북이의 펜을 올린다. 그림이 그려지지 않는다.
pendown()	None	거북이의 펜을 내린다. 그림이 그려진다.
up()	None	거북이의 펜을 올린다. 그림이 그려지지 않는다.
down()	None	거북이의 펜을 내린다. 그림이 그려진다.
color()	색상 이름	거북이 펜의 색상을 변경한다.
fillcolor()	색상 이름	다각형을 채우는 색상을 변경한다.
heading()	None	현재의 방향을 반환한다.
position()	None	현재 위치를 반환한다.
goto()	x, y	거북이를 (x, y) 위치로 이동시킨다.
begin_fill()	None	채워진 다각형을 시작한다.
end_fill()	None	채워진 다각형을 닫는다.
dot()	None	현재 위치에 점을 남긴다.
stamp()	None	현재 위치에 거북이 모양을 남긴다.
shape()	모양 이름	거북이의 모양을 'arrow', 'classic', 'turtle', 'circle' 중의 하나로 변경한다.



거북이 경주 게임 #1

- 거북이 2마리를 만들려면 다음과 같이 한다.

```
import turtle
```

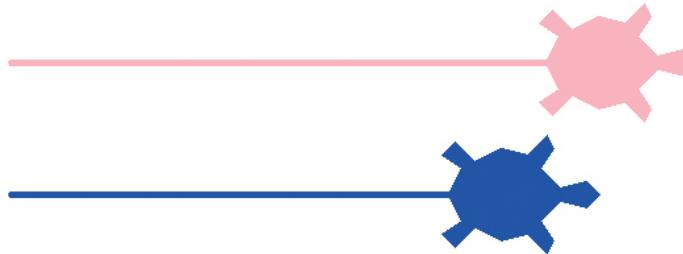
```
t1 = turtle.Turtle() # 첫 번째 거북이  
t2 = turtle.Turtle() # 두 번째 거북이
```





거북이 경주 게임 #2

- 거북이들을 구별하기 위하여 색상을 다르게 하고 모양도 다르게 하자.



```
t1.color("pink")
t1.shape("turtle")
t1.shapesize(5)
t1.pensize(5)
```

```
t2.color("blue")
t2.shape("turtle")
t2.shapesize(5)
t2.pensize(5)
```

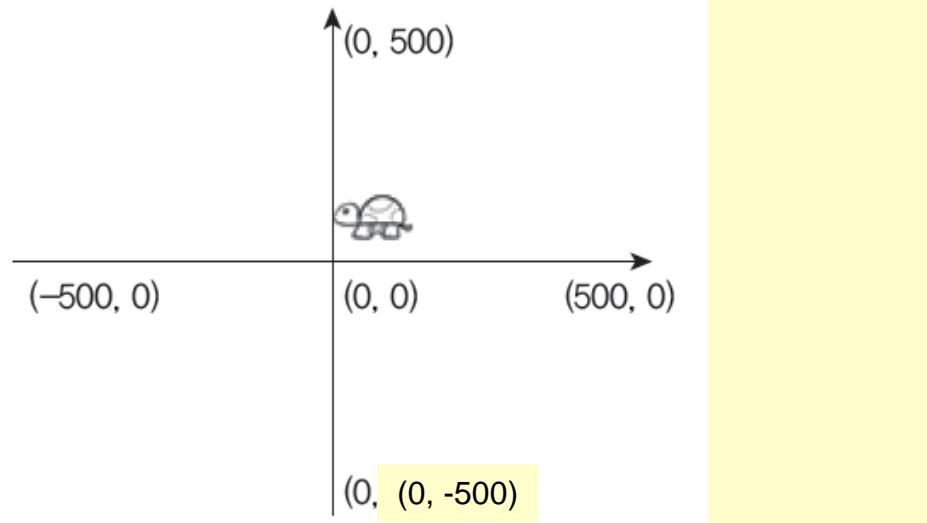


거북이 경주 게임 #3

- 출발점에 세우기

```
t1.penup()  
t1.goto(-300, 0)
```

```
t2.penup()  
t2.goto(-300, -100)
```

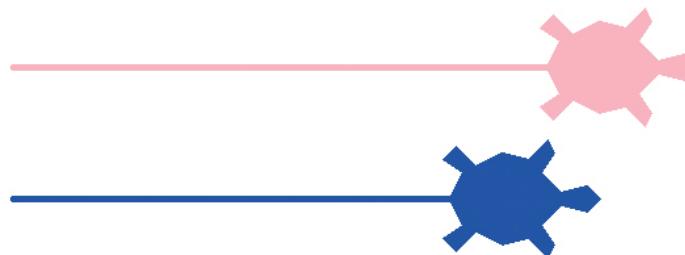




거북이 경주 게임 #4

- 100번 정도 반복하면서 한 번 반복할 때마다 난수만큼 이동하도록 하자.

```
for i in range(100):      # 100번 반복한다.  
    d1 = random.randint(1, 60)  # 1부터 60 사이의 난수를 발생한다.  
    t1.forward(d1)           # 난수만큼 이동한다.  
    d2 = random.randint(1, 60)  # 1부터 60 사이의 난수를 발생한다.  
    t2.forward(d2)           # 난수만큼 이동한다.
```





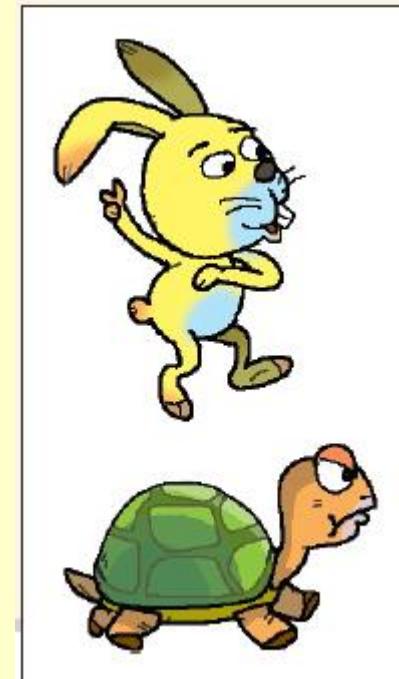
외부 이미지를 사용하자

```
import turtle # 터틀 그래픽 모듈을 불러온다.  
import random # 난수 모듈을 불러온다.
```

```
screen = turtle.Screen()  
image1 = "rabbit.gif"  
image2 = "turtle.gif"  
screen.addshape(image1)  
screen.addshape(image2)
```

```
t1 = turtle.Turtle() # 첫 번째 거북이를 생성한다.  
t1.shape(image1)
```

```
t2 = turtle.Turtle() # 두 번째 거북이를 생성한다.  
t2.shape(image2)
```





전체 소스

```
import turtle      # 터틀 그래픽 모듈을 불러온다.  
import random    # 난수 모듈을 불러온다.  
  
screen = turtle.Screen()  
image1 = "rabbit.gif"  
image2 = "turtle.gif"  
screen.addshape(image1)  
screen.addshape(image2)  
  
t1 = turtle.Turtle()      # 첫 번째 거북이를 생성한다.  
t1.shape(image1)  
t1.pensize(5)            # 팬의 두께를 5로 한다.  
t1.penup()               # 펜을 든다.  
t1.goto(-300, 0)         # (-300, 0) 위치로 간다.
```



전체 소스

```
t2 = turtle.Turtle()      # 두 번째 거북이를 생성한다.  
t2.shape(image2)  
t2.pensize(5)            # 팬의 두께를 5로 한다.  
t2.penup()               # 펜을 든다.  
t2.goto(-300, -200)      # (-300, -200) 위치로 간다.  
  
t1.pendown()             # 첫 번째 거북이의 펜을 내린다.  
t2.pendown()             # 첫 번째 거북이의 펜을 내린다.  
t1.speed(1)  
t2.speed(1)  
  
for i in range(100):      # 100번 반복한다.  
    d1 = random.randint(1, 60)  # 1부터 60 사이의 난수를 발생한다.  
    t1.forward(d1)           # 난수만큼 이동한다.  
    d2 = random.randint(1, 60)  # 1부터 60 사이의 난수를 발생한다.  
    t2.forward(d2)
```



도전문제

- (1) 3마리의 거북이가 경주하는 것으로 코드를 변경하여 보자.