

Day 23. 나와 놀자 거북이.

여러분 앞에 거북이가 있다고 생각해 봅시다. 거북이는 여러가지 행동을 할 수 있을 것입니다. 앞으로 기어가기, 좌측으로 방향틀기, 우측으로 방향틀기, 밥먹기 등 다양한 행동을 할 수 있습니다.

1966 년 Wally Feurzig와 Seymour Papert는 Logo 라는 프로그래밍 언어를 만들었습니다. 이 Logo라는 프로그래밍 언어에는 프로그래밍 언어로 거북이를 움직여 그림을 그릴 수 있도록 해주는 재미난 기능을 가지고 있었습니다.

로고라는 프로그래밍 언어는 이 재미난 거북이 프로그래밍 기능 때문에 프로그래밍 교육에 널리 사용되었습니다.

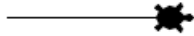
파이썬 2.5에서는 이 파이썬 거북이 그래픽 모듈(Python Turtle Graphics Module)이 추가가 되었습니다. 거북이 그래픽 모듈은 파이썬의 기본 모듈이기 때문에 파이썬만 설치하면 바로 사용가능합니다.

아 모듈(Module)이란 단어가 처음 등장했네요. 음 모듈이란 일단 어떤 작업들을 수행하는 기능을 모아놓은 것 이라고 생각해주세요. 파이썬 거북이 그래픽 모듈이란 거북이를 이용해 그림을 그리기 위한 어떤 것들이 많이 모여져 있는 것이라고 생각하면 될 것 같습니다.

파이참에서 turtle01.py라는 파일을 작성해주세요.

```
turtle01.py
import turtle
turtle.shape("turtle")
turtle.forward(100)
turtle.done()
```

마우스 오른쪽 버튼을 클릭한 후, Run 'turtle01' 을 선택하세요.



프로그램이 실행되면 위와 같은 그림이 그려질겁니다. 어떻게 이런 그림이 그려진 것일까요? 한 줄 한 줄 설명 드리도록 할게요.

**import turtle**

거북이 그래픽 모듈을 사용하려면 반드시 필요한 줄입니다. **import** 다음에는 사용하고 싶은 모듈 이름이 나옵니다. 말그대로 **turtle**(거북이)를 사용하겠다는 의미입니다.

**turtle.shape("turtle")**

**turtle.** 다음에 나오는 것을 메소드라고 합니다. 이 메소드를 이용해 거북이이가 특별한 행동을 하게 할 수 있습니다. **shape()**메소드는 거북이의 모양을 바꿀때 사용합니다. **"turtle"**라고 하면 거북이 모양으로 보여집니다. 모양에는 **classic, circle, square, triangle, arrow** 등이 있습니다. 한번 값을 바꿔가면서 실행해보세요.

**turtle.forward(100)**

**forward()**메소드는 괄호안에 있는 값만큼 앞으로 이동하게 합니다. 숫자가 작아지거나 커질 수록 이동하는 길이가 달라집니다. 숫자를 바꿔가면서 실행해보세요.

**turtle.done()**

위의 문장이 없으면 지금 실행한 창이 바로 닫혀버려서, 어떤 그림이 그려졌는지 확인할 수가 없습니다. 해당 줄을 지우고 실행해 보세요.

turtle01.py

```
import turtle

turtle.shape("turtle")

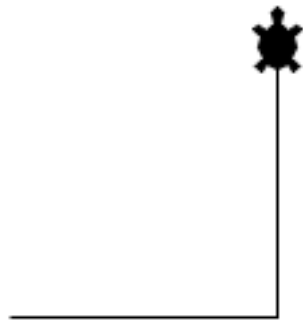
turtle.forward(100)

turtle.left(90)

turtle.forward(100)

turtle.done()
```

이번엔 위와 같이 수정하고 실행해 보도록 하겠습니다.



이번에는 위와같이 실행됩니다.

```
turtle.left(90)
```

위의 문장은 거북이를 왼쪽으로 90도 돌리라는 뜻입니다.

```
turtle.forward(100)
```

거북이를 다시 100만큼 이동합니다.

결국 그림과 같은 결과가 나오는 것을 알 수 있습니다.

숙제.

1. 위의 코드에 몇줄 추가하여 사각형을 그려보세요.
2. turtle02.py 를 작성하고 거북이를 이용해 정삼각형을 그려보세요. 정삼각형의 한변의 길이는 150으로 해주세요.

참고 : 자주 사용하는 turtle 모듈 메소드

Turtle motion	Move and draw	forward(거리)   fd(거리)	거북이가 앞으로 이동합니다.
		backward(거리)   bk(거리)   back(거리)	거북이가 뒤로 이동합니다.
		left(각도)   lt(각도)	거북이가 왼쪽으로 회전합니다.
		right(각도)   rt(각도)	거북이가 오른쪽으로 회전합니다.
		circle(반지름)	현재 위치에서 원을 그립니다.
		goto(x, y)   setpos(x, y)   setposition(x, y)	거북이를 특정 위치(좌표)로 보냅니다.
		setx(x)	거북이의 x 좌표를 지정한 위치로 이동합니다.
		sety(y)	거북이의 y 좌표를 지정한 위치로 이동합니다.
		setheading(각도)   seth(각도)	거북이가 바라보는 방향을 바꿉니다.

		home()	거북이의 위치와 방향을 처음 상태로 돌립니다.
		speed(속도)	거북이 속도를 바꿉니다. 0 : 최고 속도 1 : 가장 느린 속도 10 : 빠른 속도
	Tell Turtle's state	pos()   position()	거북이의 현재 위치(좌표)를 구합니다.
		towards(x, y)	현재 거북이가 있는 위치에서 특정 위치까지 바라보는 각도를 구합니다.
		xcor()	거북이의 x 좌표를 구합니다.
		ycor()	거북이의 y 좌표를 구합니다.
		heading()	거북이가 현재 바라보는 각도를 구합니다.
		distance(x, y)	현재 거북이가 있는 위치에서 특정 위치까지 거리를 구합니다.
Pen control	Drawing state	pendown()   pd()   down()	펜(잉크 묻힌 꼬리)를 내립니다. (그릴 수 있는 상태)
		penup()   pu()   up()	펜 (잉크 묻힌 꼬리)를 올립니다.
		pensize(굵기)   width(굵기)	펜 굵기를 바꿉니다.
		isdown()	펜이 내려져 있는지 여부 반환
	Color control	color("색상명")	펜 색을 바꿉니다.
		fillcolor("색상명")	도형 내부를 칠하는 색을 바꿉니다.
	Filling	begin_fill()	도형 내부를 색칠할 준비를 합니다.
		end_fill()	도형 내부를 색칠합니다.
	More drawing control	reset()	화면을 지우고 거북이도 원래 자리와 상태로 되돌립니다.
		clear()	거북이를 그대로 둔 채 화면을 지웁니다.
		write() 선언부 : write(arg, move=False, align="left", font=("Arial", 8, "normal"))	현재 거북이 위치에 문자를 출력합니다. ex) t.write("Hello") #현재 위치에 Hello 를 출력 t.write("Hello", False, "center", ("",20)) # 현재 거북이 위치에 가운데 정렬로 크기가 20 인 Hello 를 출력
Turtle state	Visibility	showturtle()   st()	거북이를 화면에 표시합니다.
		hideturtle()   ht()	거북이를 화면에서 가립니다.
		isvisible()	거북이가 화면에 보여지는지 여부
	Appearance	shape("모양명")	거북이 모양을 바꿉니다. 모양명 : classic(기본값), circle, square, triangle, arrow, turtle
Methods of	Using Screen events	listen()	사용자 입력이 잘 처리되도록 거북이 그래픽 창에 포커스를 줍니다.

TurtleScreen			
		onkeypress(함수, "키이름")	키보드를 눌렀을 때 실행할 함수를 정합니다.
		onclick(함수)   onscreenclick(함수)	마우스 버튼을 눌렀을 때 실행할 함수를 정합니다.
		ontimer(함수, 시간)	일정한 시간(ms)이 지난 뒤 실행할 함수를 정합니다.
	Methods specific to Screen	title("창 이름")	<p>거북이 그래픽 창의 이름을 지정합니다..</p> <p><b>[출처]</b> <a href="#">[Python3] 파이썬 거북이 그래픽 (Python Turtle Graphics) 모듈 사용하기 - 그림 그리기</a>   작성자 <a href="#">자바 킴</a></p>