Day 26. 거북이 up & down

거북이 그래픽 모듈을 이용하여 거북이를 이동시키면, 이동되면서 선이 그려지는 것을 알 수 있습니다. 그렇다면, 거북이를 이동시킬때 선을 그리지 않고 이동시킬 수 있는 방법은 없을까요?

이때는 거북이가 가지고 있는 up(), down() 메소드(method)를 사용하시면 됩니다. 우리는 앞으로 객체(Object)라는 것을 배우게 될텐데요. 오브젝트가 가지는 기능은 메소드라고 말합니다. 우리가 앞에서 배운 함수에선 오브젝트라는 단어가 등장하지 않았죠? 우리가 배우는 거북이는 수많은 객체 중에 하나입니다. 아직은 너무 깊이 알 필요는 없고, 거북이는 객체고 거북이가 가지고 있는 여러가지 기능. 예를 들어 앞으로 가는 forward(), 왼쪽으로 회전하는 left(), 오른쪽으로 회전하는 right() 등이 메소드라고 생각하면 됩니다.

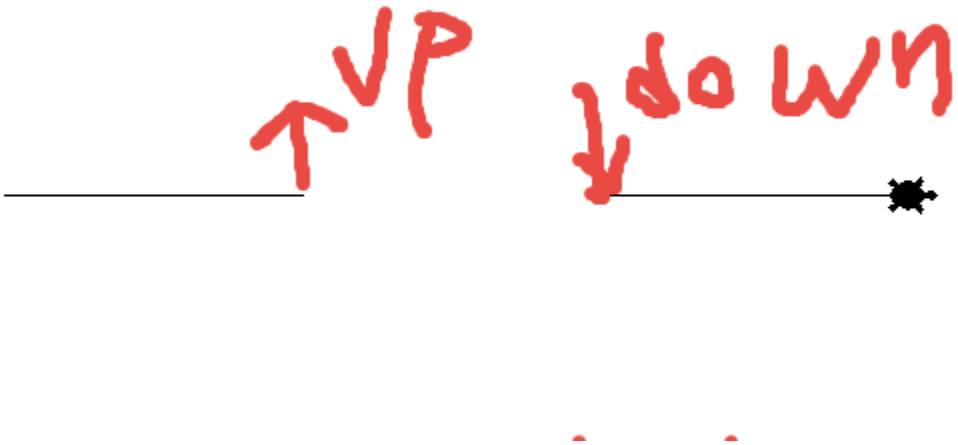
거북이가 가지고 있는 메소드중에서 up()과 down()이 있습니다. up()을 사용하면 거북이를 들었다고 생각하면 될것 같습니다. 그리고 나서 forward()를 하게 되면 거북이를 든 상태로 이동하기 때문에 거북이가 선을 그리질 못합니다. 거북이가 이동하면서 그리게 하려면 down()을 한 이후에야 그릴 수 있습니다. 우리가 앞에서 거북이를 이용해 그릴 수 있었던 것은 거북이는 기본적으로 들리지 않은 상태인 것을 알 수 있습니다.

day26 디렉토리를 생성하고 updown01.py를 아래와 같이 작성합니다.

updown01.py

```
import turtle

turtle.shape("turtle")
turtle.speed(1)


turtle.forward(150)
turtle.up()
turtle.forward(150)
turtle.down()
turtle.forward(150)

turtle.done()
```

위의 프로그램을 실행하면 결과는 다음과 같이 나옵니다.



거북이가 이동하는데, 어떤 부분은 선을 그리지 않고 이동한 것을 알 수 있습니다.

```
turtle.forward(150)
```

거북이를 150만큼 앞으로 이동시킵니다.

```
turtle.up()
```

거북이를 위로 듭니다.

```
turtle.forward(150)
```

거북이를 150만큼 앞으로 이동시킵니다. 들려있는 상태이기 때문에 그리지 않습니다.

```
turtle.down()
```

거북이를 내려놓습니다.

```
turtle.forward(150)
```

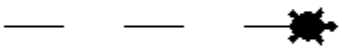
거북이를 150만큼 앞으로 이동시킵니다. 이때는 내려놓아진 상태기 때문에 그리는 것을 알 수 있습니다.

앞에서 배운 함수를 이용하여, 위의 예제를 수정해 보도록 하겠습니다.

updown02.py

```
import turtle

turtle.shape("turtle")
turtle.speed(1)

def go(distance, up = False):
    if(up == True):
        turtle.up()
    else:
        turtle.down()
    turtle.forward(distance)

go(30)
go(30, True)
go(30)
go(30, True)
go(30)

turtle.done()
```

위의 코드를 실행하면 다음과 같은 결과가 나옵니다.



어떻게 이런결과가 나왔는지 코드를 살펴보도록 하겠습니다.

```
def go(distance, up = False):
    if(up == True):
        turtle.up()
    else:
        turtle.down()
```

```
    turtle.forward(distance)
```

위의 코드는 go()라는 함수를 정의하고 있습니다. go라는 함수는 2개의 매개변수를 가지고 있습니다. 각각의 매개변수의 이름은 거리를 의미하는 distance와 up이라는 변수입니다. up은 기본값으로 False를 가지게 하였습니다. 이렇게 기본값을 가지게 하면 함수를 사용할 때 2번째 매개변수에는 값을 넣지 않아도 됩니다.

up이 True일 경우에는 turtle.up()을 실행하고 False일 때는 turtle.down()을 하고 있습니다. 그리고 distance만큼 앞으로 이동하도록 되어 있습니다.

함수가 정의되어 있다고 해서 무조건 실행되는 것은 아니라고 했습니다. 함수를 정의했으면, 함수를 사용해야 합니다.

```
go(30)
```
30 만큼 이동합니다. 전달인자를 1 개만 사용하였기 때문에 go()함수의 2 번째 매개변수는 기본적으로 설정한 False 값을 가집니다. 즉 turtle.down() 후에 turtle.forward(30)이 실행되게 됩니다.

```
go(30, True)
```
30 만큼 이동합니다. 전달인자를 2 개 사용하였습니다. 이 경우 go() 함수의 2 번째 매개변수는 True 값을 가지게 됩니다. 즉 turtle.up() 후에 turtle.forward(30)이 실행되게 됩니다.

```
go(30, False)
```
전달변수를 1 개만 전달하게 되면 매개변수 up 은 기본적으로 False 를 가진다고 하였습니다. 그렇다 하더라도 사용자는 False 값을 다시 전달 할 수 있습니다.

이렇게 하여 코드가 실행되면 30씩 이동하지만 한번을 그리면서 이동, 한번은 그리지 않고 이동을 하게 되는 것을 알 수 있습니다.

숙제

오늘은 함수를 정의하고, 그 함수를 이용하는 코드를 다시한번 살펴봤습니다. 아마 조금 어렵게 느껴

질 수도 있을 텐데요. 오늘 작성한 코드를 살짝 가려놓고 다시한번 작성해보세요. 본인이 이해를 하였는지, 스스로 코드를 작성해 보고 확인해보세요.