



**[8차시 : 랜덤으로 수행하기]**

# 【지난 시간에 배운 내용】


# 조건에 따라 수행하기 : **if**


# 반복해서 수행하기 : **while, for**


# 데이터를 묶어서 표현하기 : **list**

# [오늘 학습할 내용]

# 랜덤으로 수행하기 : **random** 모듈

 Ex1\_복권당첨확인하기

 Ex2\_랜덤빨샘문제맞추기

 Ex3\_랜덤으로움직이는거북이(터틀)

 Ex4\_터틀런(터틀)

# [모듈을 쓴다고 하면 뭐다?!]

#1. 사용하고자 하는 모듈 **import**

#2. 해당 모듈의 메서드를 **dot(.)연산**으로 사용하기

random 모듈의 choice( ) 사용하는 경우,

```
import random
```

```
random.choice(seq)
```

# [random 모듈 살펴보기]

- Numeric and Mathematical Modules
  - `numbers` — Numeric abstract base classes
  - `math` — Mathematical functions
  - `cmath` — Mathematical functions for complex numbers
  - `decimal` — Decimal fixed point and floating point arithmetic
  - `fractions` — Rational numbers
  - `random` — Generate pseudo-random numbers
  - `statistics` — Mathematical statistics functions

# [random 모듈 살펴보기]

## Functions for integers

```
random.randrange(stop)
```

```
random.randrange(start, stop[, step])
```

Return a randomly selected element from `range(start, stop, step)`. This is equivalent to `choice(range(start, stop, step))`, but doesn't actually build a range object.

The positional argument pattern matches that of `range()`. Keyword arguments should not be used because the function may use them in unexpected ways.

*Changed in version 3.2:* `randrange()` is more sophisticated about producing equally distributed values. Formerly it used a style like `int(random()*n)` which could produce slightly uneven distributions.

```
random.randint(a, b)
```

Return a random integer  $N$  such that  $a \leq N \leq b$ . Alias for `randrange(a, b+1)`.

# [random 모듈 살펴보기 : 정수를 위한 함수]

```
import random
for i in range(10):
    random.randrange(100)
```

0~99까지 범위의 수에서 랜덤

```
import random
for i in range(10) :
    random.randrange(1,10,2)
```

1~9까지 범위의 수에서 홀수(2씩 증가) 랜덤

```
import random
for i in range(10) :
    random.randint(1,100)
```

1~100까지 범위의 수에서 랜덤

# [Ex1\_복권당첨확인하기]

2자리 숫자로 이루어진 복권에서  
사용자가 가지고 있는 복권 번호를 입력하여 당첨여부를 확인하는  
프로그램을 작성하세요.

(단, 복권 번호는 10~99사이에서 랜덤으로 발생된다.)

가지고 있는 복권 번호를 입력 :68  
꽂!!!  
당첨 번호는 52 입니다



# [Ex1\_복권 당첨 확인하기]

```
import random
```

```
user=int(input("가지고 있는 복권 번호를 입력 : ")) #사용자 입력 값을 user_num 변수에 저장
```

```
lotto= #10~99사이의 숫자를 랜덤으로 발생시켜 lotto변수에 저장
```

```
if(): #만약 사용자 입력값(user)과 랜덤으로 발생된 수(lotto)가 같으면
```

```
    print("당첨입니다!!")
```

```
else : #그렇지 않으면
```

```
    print("꽂!!!")
```

```
    print("당첨 번호는",lotto,"입니다")
```

```
input()
```

# [Ex2\_랜덤 뱀셈문제 맞추기]

1~100사이의 숫자를 랜덤으로 2개 발생시켜  
뱀셈 문제를 만들고 사용자에게 답을 물어본 후  
사용자 답변이 올바른지 검사하는 프로그램 작성하세요.

==랜덤 뱀셈 문제 맞추기==

7-47=-40

맞았습니다.

==랜덤 뱀셈 문제 맞추기==

41-22=18

틀렸습니다.

# [Ex2\_랜덤 뱀셈문제 맞추기]

**import random** #random 모듈 가져오기-**randint**함수 사용

**print**("==랜덤 뱀셈 문제 맞추기==")

**num1**= #1~100사이의 난수 발생

**num2**= #1~100사이의 난수 발생

**answer**= #정답

**user**=**int**(**input**(**str**(**num1**)+"-"+**str**(**num2**)+"=")) #사용자로부터 입력받기

**if**(): #만약, 사용자 입력값과 정답이 같으면

**print**("맞았습니다.") # "맞았습니다" 출력

**else** : #그렇지 않으면

**print**("틀렸습니다.") # "틀렸습니다" 출력

# [Ex2\_랜덤 뱀셈 문제 맞추기(도전)]

사용자 답변이 맞을 때까지 반복하여 답을 입력받아 보세요

**==랜덤 뱀셈 문제 맞추기==**

**75-35=20**

틀렸습니다.

**75-35=10**

틀렸습니다.

**75-35=40**

맞았습니다.

아래의 링크를 따라가서  
break 구문을 참고해 보세요!

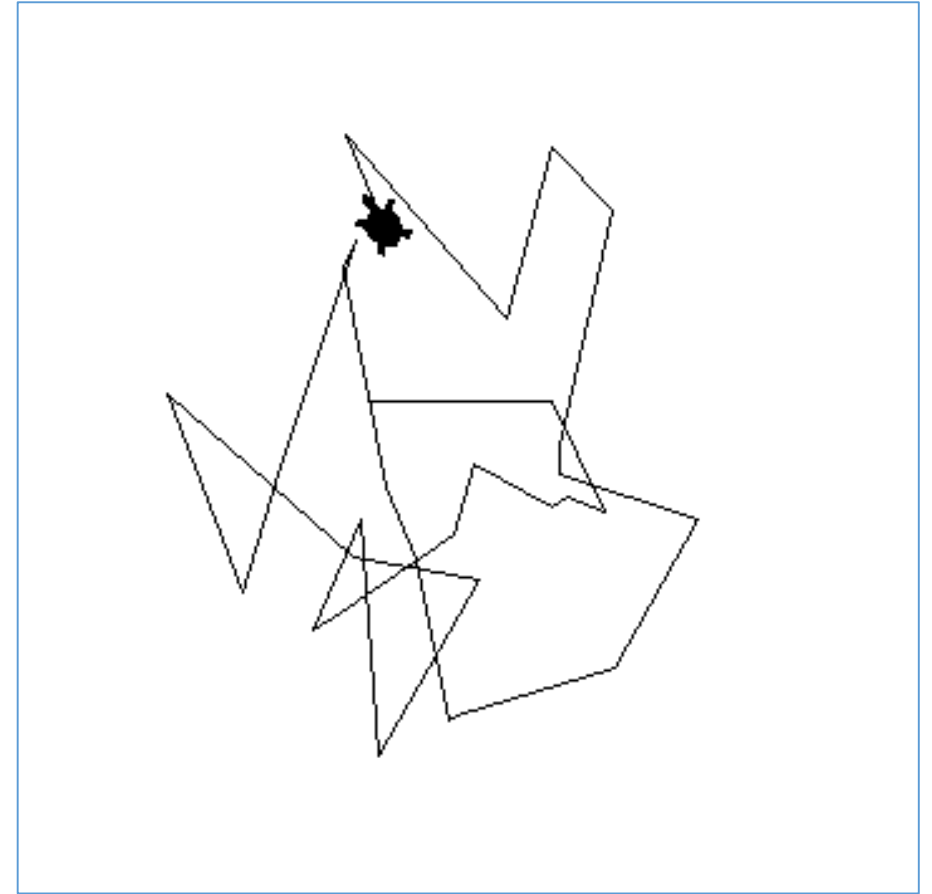
<https://docs.python.org/3/tutorial/controlflow.html#break-and-continue-statements-and-else-clauses-on-loops>

# [Ex3\_랜덤으로움직이는거북이(터틀)]

아래 [조건]에 맞추어서 랜덤으로 움직이는 거북이를 만들어 보세요.

## [조건]

1. 거북이의 움직임을 30회 반복한다.
2. 거북이의 움직임
  - 1) 1~100사이의 난수를 생성하여  
그 길이만큼 이동
  - 2) -180~180사이의 난수를 생성하여  
그 각도만큼 오른쪽 회전



# [Ex3\_랜덤으로 움직이는 거북이(터틀)]

**import turtle** #turtle모듈 가져오기

**import random** #random모듈 가져오기

**t=turtle.Turtle()**

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

**turtle.title("랜덤으로 움직이는 거북이")**

**for i in**  **: # 30회 반복**

**length=random.randint(1,100)** #1~100사이의 난수를 생성하여 **length**에 저장

**t.forward(****)** #거북이가 **length**만큼 앞으로 이동

**angle=** **# -180~180사이의 난수를 생성하여 angle에 저장**

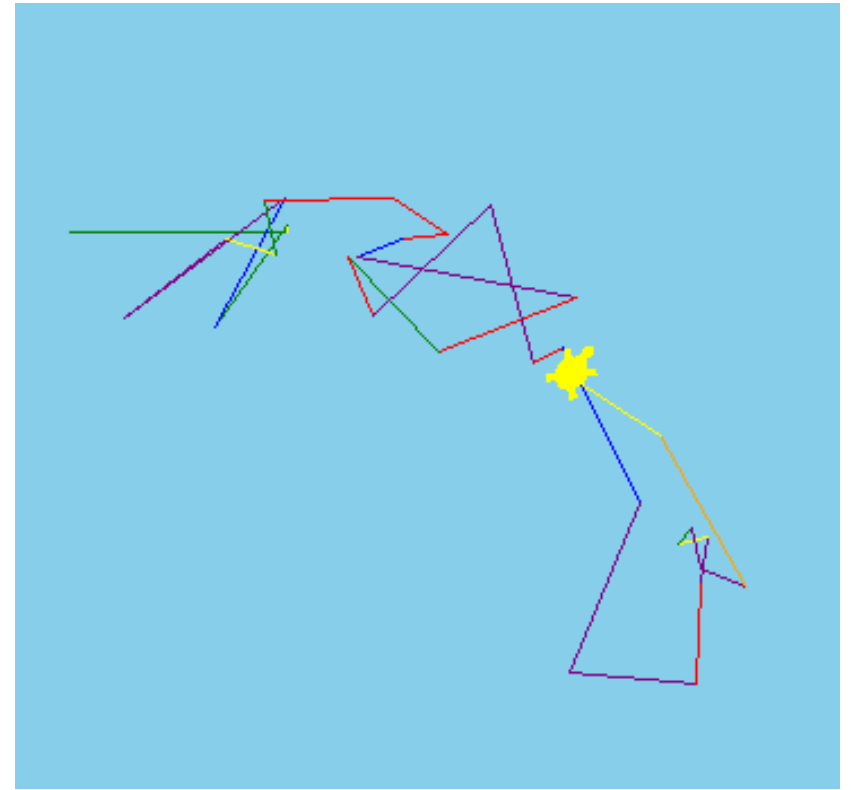
**t.right(****)** #거북이가 **angle**만큼 오른쪽으로 회전

**turtle.done()**

# [Ex3\_랜덤으로움직이는거북이(터틀)]

[추가 조건]

거북이가 움직일 때마다  
7가지 색 중 하나의 색을 랜덤으로 선택하여  
그려지도록 한다.

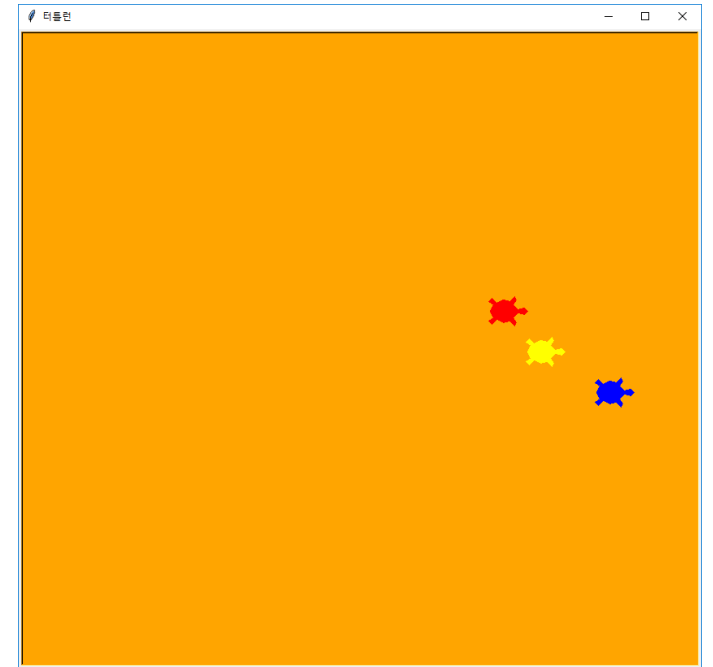
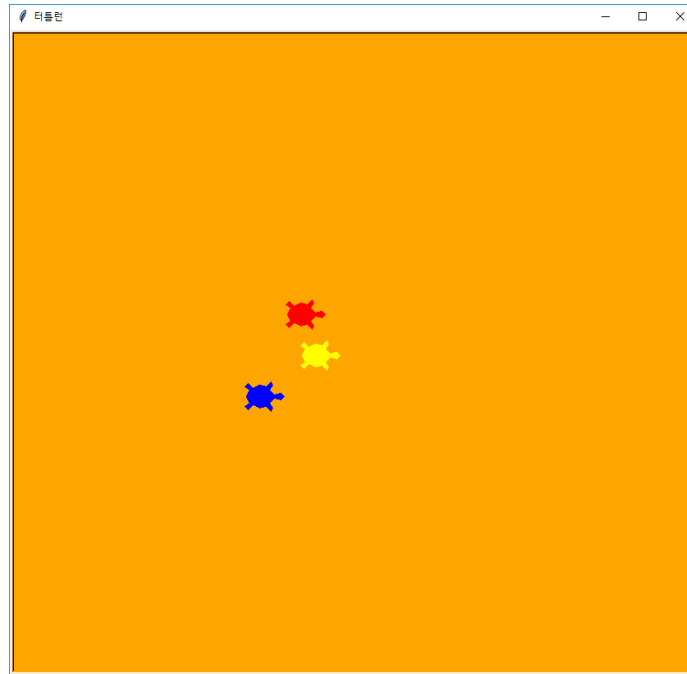
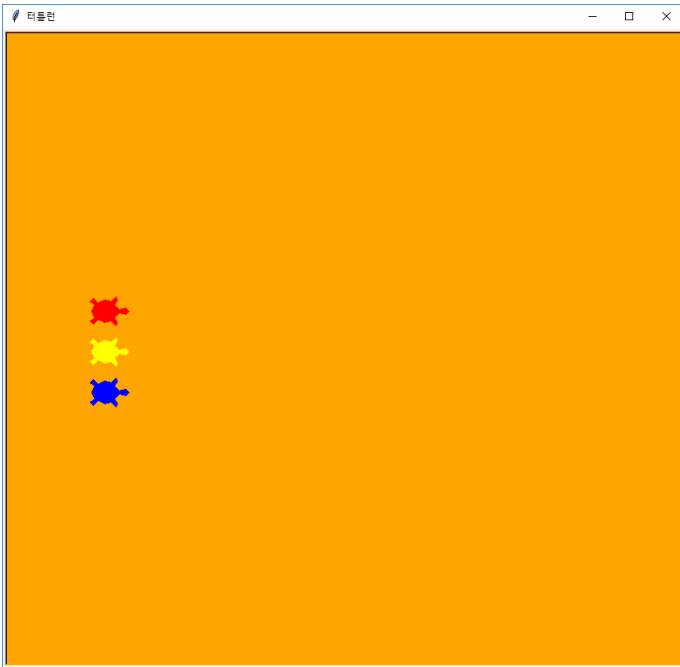


# [Ex4\_터틀런(터틀)]

거북이 3마리(red, yellow, blue)가 경주를 합니다. [조건]을 참고하여 프로그램을 완성해 보세요.

[조건]

1. 거북이의 초기 위치 : red(-300,50), yellow(-300,0), blue(-300,-50)
2. 거북이들은 앞으로 가는 동작을 35번 반복한다.  
단, 앞으로 갈 때 거리 값은 1~30사이의 값을 랜덤으로 발생시켜 이동함





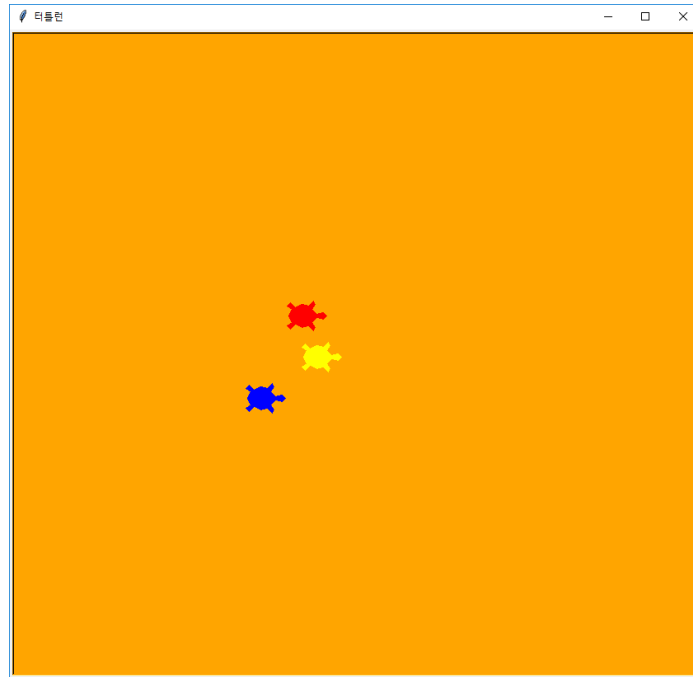
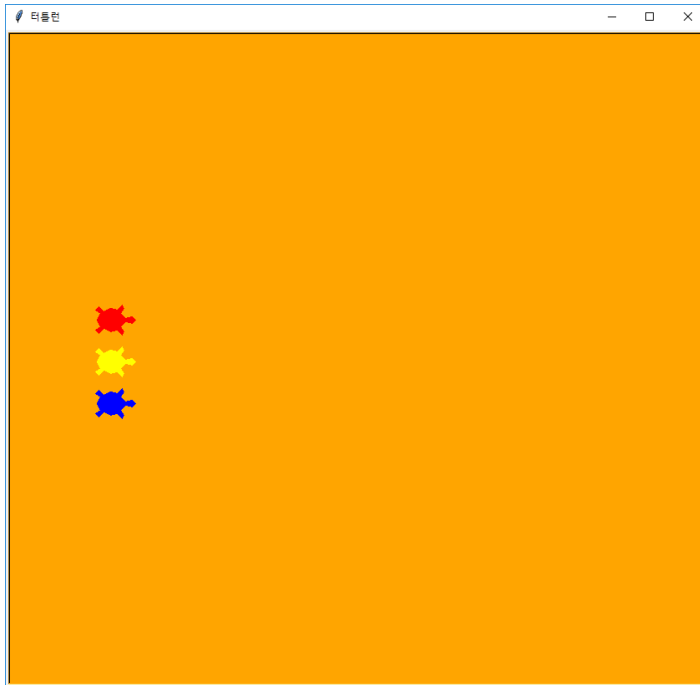
```
import turtle
import random
turtle.bgcolor("orange")
turtle.title("터틀런")
red=turtle.Turtle() #거북이 red생성
yellow=turtle.Turtle() #거북이 yellow생성
blue=turtle.Turtle() #거북이 blue생성
#거북이 red의 설정
red.shape("turtle")
red.color("red")
red.shapesize(2)
red.penup()
red.goto()
#거북이 yellow의 설정
yellow.shape("turtle")
yellow.color("yellow")
yellow.shapesize(2)
yellow.penup()
yellow.goto()
#거북이 blue의 설정
blue.shape("turtle")
blue.color("blue")
blue.shapesize(2)
blue.penup()
blue.goto()
```

#거북이 경주

```
for i in  :
    d1=
    red.forward(d1)
    d2=
    yellow.forward(d2)
    d3=
    blue.forward(d3)
turtle.done()
```

# [Ex4\_터틀런(터틀) : 도전]

거북이 3마리(red, yellow, blue)가 경주를 합니다.  
경기가 종료 된 후, 1등인 거북이를 출력하고  
각 거북이의 x축 위치를 출력합니다.  
[hint] write( )이용하기



## 터틀런 프로그램에 이어서 작성하세요

#거북이들의 x좌표

x1= #red의 x좌표를 x1에 저장

x2= #yellow의 x좌표를 x2에 저장

x3= #blue의 x좌표를 x3에 저장

#가장 멀리 간 거북이 찾기

if(): #red가 1등인 경우

win=red

elif (): #yellow가 1등인 경우

win=yellow

else : #blue가 1등인 경우

win=blue

#1등 출력하기

t= #출력을 위한 새로운 거북이 생성

t.hideturtle()

t.penup()

t.goto(-20,180)

t.write(win.fillcolor()+"가 1등입니다", align="center",font=("Arial",20,"bold")) #1등 출력

t.goto(-20, 160)

t.write("red의 위치 : "+str(x1), align="center",font=("Arial",15,"bold")) #red의 위치 출력

t.goto(-20, 140)


t.write("yellow의 위치 : "+str(x2), align="center",font=("Arial",15,"bold")) #yellow의 위치 출력

t.goto(-20, 120)

t.write("blue의 위치 : "+str(x3), align="center",font=("Arial",15,"bold")) #blue의 위치 출력

# [수업 정리]

1. 구글 드라이브 <2019\_Python\_학번\_이름> 폴더에  
<오늘날짜\_랜덤으로수행하기>폴더를 생성
2. 오늘 실습한 파일을 모두 업데이트(총 4개 파일)

 Ex1\_복권당첨확인하기

 Ex2\_랜덤빨샘문제맞추기

 Ex3\_랜덤으로움직이는거북이(터틀)

 Ex4\_터틀런(터틀)