

거북이 피하기 게임 만들기

1. 모듈 import 하기

```
import turtle as t  
import random
```

2. 변수 초기화 하기

```
score = 0  
playing = False #시작 여부 : True면 시작  
t.bgcolor("blue") #배경색 설정
```

3. 악당 거북이 설정하기

악당 거북이 변수 te에 거북이를 만들고
모양은 거북이로 색깔은 빨강색, 스피드는 0
거북이가 움직일 때 선을 그리지 않도록 설정하고
거북이 위치는 0, 200으로 이동시키는 코드를 작성하세요.

정답코드

```
te = t.Turtle()  
te.shape("turtle")  
te.color("red")  
te.speed(0)  
te.penup()  
te.goto(0, 200)
```

4. 먹이 설정

먹이 변수 ts에 거북이를 만들고
모양은 원으로 색깔은 오렌지색, 스피드는 0
움직일 때 선을 그리지 않도록 설정하고
위치는 0, -200으로 이동시키는 코드를 작성하세요.

정답코드

```
ts.Turtle()  
ts.shape("circle")  
ts.color("orange")  
ts.speed(0)  
ts.penup()  
ts.goto(0, -200)
```

5. 거북이 방향 설정 함수 만들기

거북이의 방향전환을 처리하는 함수를 만드세요.

상하좌우 4개의 함수를 만들면 되고

함수이름은 아래처럼 작성하고

turn_left, turn_right, turn_up, turn_down

함수에서는 거북이의 머리 방향을 설정하도록 합니다.

정답코드

```
def turn_right() :  
    t.setheading(0)
```

```
def turn_up() :  
    t.setheading(90)
```

```
def turn_left() :  
    t.setheading(180)
```

```
def turn_down() :  
    t.setheading(270)
```

6. 게임 시작 함수 만들기

게임을 시작하는 함수를 만들어 보세요.

함수이름은 start로 하고

2번에서 설정한 playing 변수의 값을 True로 바꿔주고
플레이어 거북이 t를 초기화 해줍니다.

마지막으로 play 함수를 호출하도록 합니다.

정답코드

```
def start() :  
    global playing  
    if playing == False :  
        playing = True  
        t.clear()  
        play()
```

7. 게임 플레이 함수 만들기

6번에서 마지막으로 호출한 play 함수를 만들어 보세요.

플레이어 거북이가 앞으로 10전진합니다.

1~5 사이의 랜덤수를 생성하여 3이 나온 경우 악당 거북이의 방향을 플레이어 거북이 위치로 향하게 합니다.

스피드를 현재 점수에 5만큼 더한 값으로 설정합니다.

악당 거북이의 스피드를 바꿔 줍니다.

플레이어거북이와 악당거북이의 거리가 12미만이면 게임종료 처리를 합니다.

플레이어거북이와 먹이의 거리가 12미만이면 점수를 올려주고 먹이를 화면 내 랜덤한 위치에 옮겨줍니다.

게임이 종료되지 않았다면 0.1초 후 play 함수를 다시 호출하도록 합니다.

정답코드

```
def play() :  
    global score  
    global playing  
    t.forward(10)  
    if random.randint(1,5) == 3 :  
        ang = te.towards(t.pos())  
        te.setheading(ang)  
    speed = score + 5  
    if speed > 15 :  
        speed = 15  
    te.forward(speed)
```

뒷장 계속

```
if t.distance(te) < 12 :  
    text = "Score: " + str(score)  
    message("game over", text)  
    playing = False  
    score = 0  
if t.distance(ts) < 12 :  
    score = score + 1  
    t.write(score)  
    start_x = random.randint(-230, 230)  
    start_y = random.randint(-230, 230)  
    ts.goto(start_x, start_y)  
if playing :  
    t.ontimer(play, 100)
```

8. 메시지 출력 함수 만들기

#2개의 문자열을 받아 각각 화면의 정해진 위치에 출력해주는 함수입니다. 그대로 작성하세요.

```
def message(m1, m2) :  
    t.clear()  
    t.goto(0,100)  
    t.write(m1, False, "center",("Consolas", 20, "italic", "bold",  
"underline"))  
    t.goto(0,-100)  
    t.write(m2, False, "center", ("Arial", 15))  
    t.home()
```

9. 거북이 설정하기

플레이어 거북이 설정을 합니다.

`t.title("Turtle Run")` #게임창 제목 설정

플레이어 거북이 변수는 `t`입니다.

플레이어 거북이 모양을 거북이로 하고

스피드는 0, 이동 시 선을 그리지 않도록 설정하세요.

색깔은 하얀색으로 설정합니다.

정답코드

```
t.shape("turtle")
```

```
t.speed(0)
```

```
t.penup()
```

```
t.color("white")
```

10. 키보드 이벤트 처리하기

키보드 이벤트를 등록해보세요.

키보드의 상하좌우 키, 스페이스 키를 등록해 줍니다.

Right, Left, Up, Down 키는 5번에서 작성한 플레이어 거북이의 방향을 설정하는 함수를 호출합니다.

스페이스키는 6번에서 작성한 게임을 시작하는 함수를 호출합니다.

정답코드

```
t.onkeypress(turn_right, "Right")  
t.onkeypress(turn_left, "Left")  
t.onkeypress(turn_up, "Up")  
t.onkeypress(turn_down, "Down")  
t.onkeypress(start, "space")  
t.listen()
```

11. 게임 초기화면 메시지 설정하기

```
message("Turtle Run", "[space]")
```

12. 게임 루프 설정하기

```
t.mainloop()
```