

» 학습목표

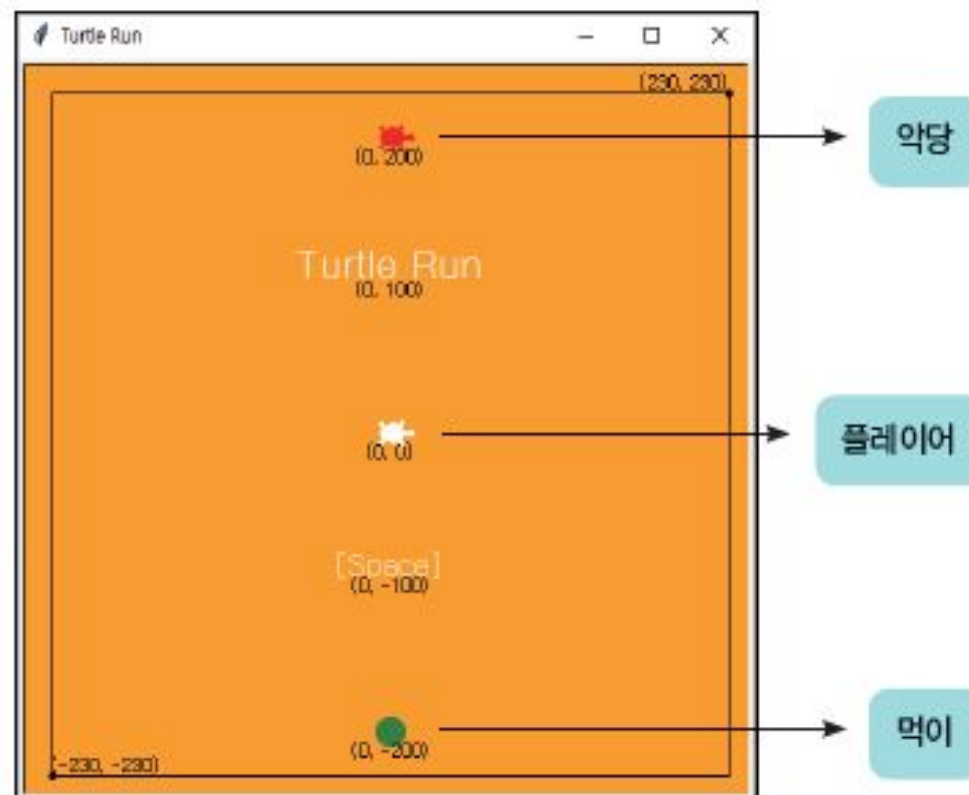
- 파이썬 프로그램의 오브젝트를 응용하여 터틀런 만들기 프로젝트 프로그램을 제작할 수 있다.

Unit 1. 터틀런 만들기

- 01 터틀런이란?
- 02 거북이 여러 마리 키우기
- 03 타이머
- 04 towards 함수
- 05 프로젝트 구조

슬라이드1. 터틀런이란?

- » 게임의 주인공인 거북이가 되어, 뒤에서 쫓아오는 악당 거북이를 피하면서 초록색 동그라미 먹이를 최대한 많이 먹어야 함



슬라이드 2. 거북이 여러 마리 키우기

```
» te = t.Turtle()      # 악당 거북이 te를 새로 만듦
   te.shape("turtle")
   te.color("red")      # 빨간색으로 지정하여 주인공 거북이와 구분

» ts = t.Turtle()      # 먹이로 사용할 거북이 ts를 만듦
   ts.shape("circle")   # 모양을 원으로 지정하여 먹이처럼 보이게 함
   ts.color("green")     # 초록색으로 지정함
```

슬라이드 3. 타이머

- » t.ontimer(실행할 함수, 정해진 시간)
- » '정해진 시간'이 지나면 '지정한 함수'를 실행하는 함수
- » 주의점
 - 정해진 시간의 단위는 초가 아니라 1000분의 1초(1/1000)
 - t.ontimer의 타이머 기능은 정해진 함수를 딱 한 번만 실행

슬라이드 4. towards 함수

- » 특정 위치로 가려면 어떤 방향을 향해서, 즉 어떤 '각도'로 가야 하는지 알려주는 기능
- » 터틀런에서 악당 거북이가 주인공 거북이를 쫓아가는 순서
 - 1 | 악당 거북이를 기준으로 주인공 거북이로 향하는 각도를 구합니다.
 - 2 | 구한 각도에 맞춰 악당 거북이의 머리를 돌립니다.
 - 3 | 악당 거북이가 앞으로 이동합니다.

te: 악당 거북이

t: 주인공 거북이

```
ang = te.towards(t.pos())
```

```
te.setheading(ang)
```

```
te.forward(9)
```

슬라이드 5. 프로젝트 구조

» 프로젝트 구조

- 등장인물 만들기
악당 거북이와 먹이 거북이를 만들고 색과 모양 등 필요한 처리를 함.
- turn_으로 시작하는 함수
turn_으로 시작하는 네 가지 함수는 각각 사용자가 ↑, ↓, ←, → 방향키를 눌렀을 때 주인공 거북이의 방향을 돌리는 역할을 함.
- play 함수
터틀런 프로젝트에서 핵심 역할을 하는 부분입니다. 주인공 거북이가 앞으로 이동하는 처리, 악당 거북이가 주인공 거북이를 쫓아가는 처리, 주인공 거북이가 먹이 또는 악당 거북이에 닿았을 때의 처리를 모두 play 함수에서 함.
- 게임 준비 및 실행
화면 배경을 그리거나 키보드 입력을 처리 함. 즉, 게임에 필요한 준비를 하는 부분임.

5. 프로젝트 구조

» 터틀런 만들기

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

```
te = t.Turtle()                # 악당 거북이(빨간색)  
te.shape("turtle")  
te.color("red")  
te.speed(0)  
te.up()  
te.goto(0, 200)
```

```
ts = t.Turtle()                # 먹이(초록색 동그라미)  
ts.shape("circle")
```


5. 프로젝트 구조

```
ts.color("green")  
ts.speed(0)  
ts.up()  
ts.goto(0, -200)
```

```
def turn_right():           # 오른쪽으로 방향을 바꿈  
    t.setheading(0)
```

```
def turn_up():              # 위로 방향을 바꿈  
    t.setheading(90)
```

```
def turn_left():            # 왼쪽으로 방향을 바꿈  
    t.setheading(180)
```

```
def turn_down():            # 아래로 방향을 바꿈  
    t.setheading(270)
```

5. 프로젝트 구조

```
def play():  
    t.forward(10)          # 게임을 실제로 플레이하는 함수  
                           # 주인공 거북이가 10만큼 앞으로 이동  
    ang = te.towards(t.pos())  
    te.setheading(ang)     # 악당 거북이가 주인공 거북이를 바라보게 함  
    te.forward(9)         # 악당 거북이가 9만큼 앞으로 이동함  
    if t.distance(ts) < 12: # 주인공과 먹이의 거리가 12보다 작으면(가까우면)  
        star_x = random.randint(-230, 230)  
        star_y = random.randint(-230, 230)  
        ts.goto(star_x, star_y) # 먹이를 다른 곳으로 옮김  
    if t.distance(te) >= 12: # 주인공과 악당의 거리가 12 이상이면(멀면)  
        t.ontimer(play, 100)  # 0.1초 후 play 함수를 실행함(게임 계속)
```

5. 프로젝트 구조

```
t.setup(500, 500)
t.bgcolor("orange")
t.shape("turtle")
t.speed(0)
t.up()
t.color("white")
t.onkeypress(turn_right, "Right")
t.onkeypress(turn_up, "Up")
t.onkeypress(turn_left, "Left")
t.onkeypress(turn_down, "Down")
t.listen()
play()
```

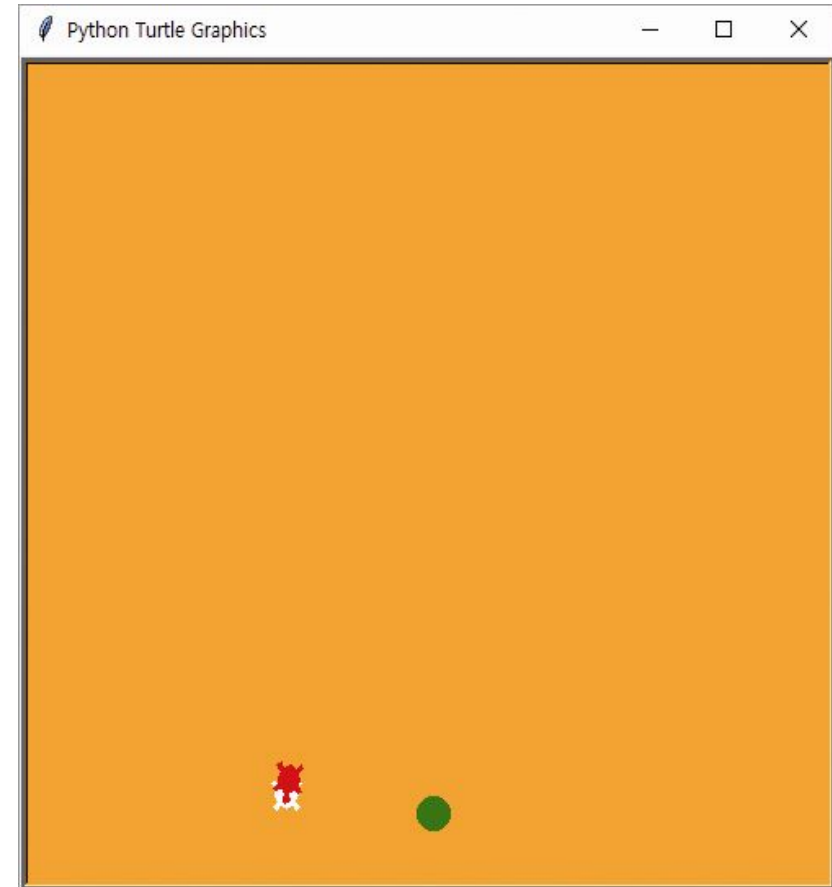
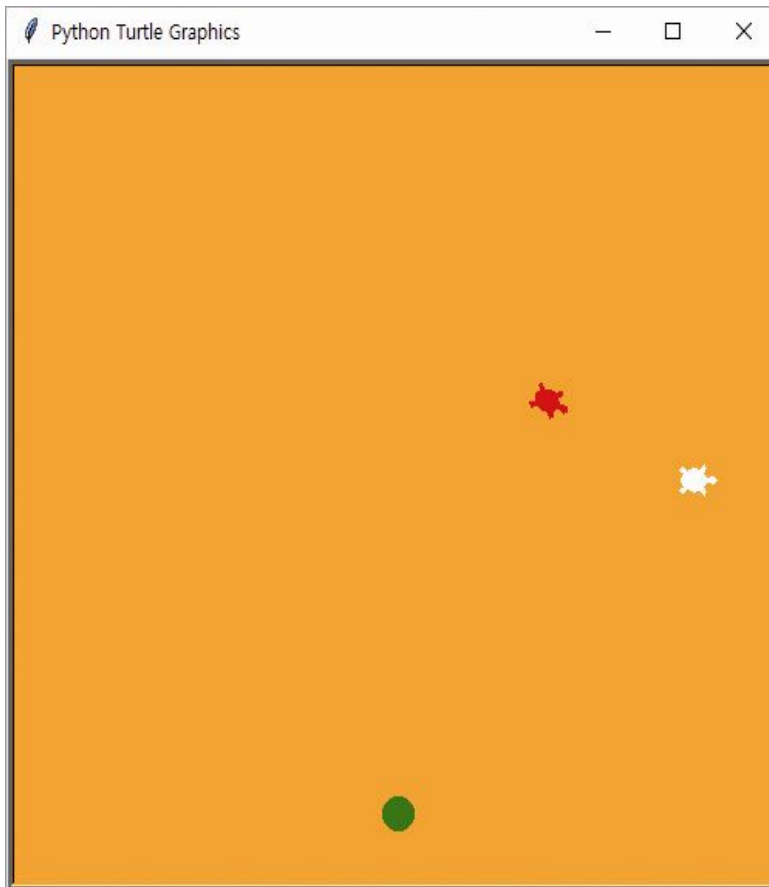
```
# 거북이 모양의 커서를 사용
# 거북이 속도를 가장 빠르게 지정
```

```
# → 를 누르면 turn_right 함수를 실행
```

```
# 거북이 그래픽 창이 키보드 입력을 받도록 함
# play 함수를 호출해서 게임을 시작함
```

5. 프로젝트 구조

» 실행결과



Unit 2.터틀런 2 만들기

- 01** 터틀런 2란?
- 02** global 변수
- 03** clear 함수와 home 함수
- 04** title 함수
- 05** 프로젝트 구조

슬라이드1. 터틀런 2란?

- » 터틀런 1에 제목, 게임 시작, 게임 종료(Game Over), 난이도 조절, 점수 계산 등의 기능을 추가한 것
- » 터틀런 1은 실행을 하자마자 게임이 시작되고, 악당 거북이에게 잡히면 그 상태로 게임이 끝나 버리지만 터틀런 2는 실행과 동시에 제목이 적힌 화면이 뜨고, 사용자가 SpaceBar 를 누를 때까지 게임이 시작되지 않음.

즉, 사용자는 게임을 시작할 준비를 할 수 있음. 또한, 먹이를 먹을 때마다 점수가 올라가고 점수가 올라갈수록 악당 거북이가 빨라지면서 난이도가 조정됨. 악당에게 잡히면 최종 점수와 Game Over가 표시되면서 게임이 끝남. 게임이 끝나도 SpaceBar 를 누르면 게임을 다시 시작할 수 있음.

슬라이드 2. global 변수

- » 파이썬 프로그래밍에서는 '프로그램 전체에서'라는 뜻으로 사용됨
함수 시작 부분에 다음과 같이 적으면 '이 변수는 함수 안에서는 물론 프로그램 전체에서 사용하는 변수'라는 뜻

global 변수명

- » 다음과 같이 f 함수 안에 global a라고 적으면 변수 a는 '프로그램 전체에서' 사용하는 변수가 됨

```
def f():  
    global a  
    a = 5
```

```
f()  
print(a)
```

슬라이드 3. clear 함수와 home 함수

- » t.clear 함수 : 거북이로 그린 그림을 모두 지우는 함수
거북이가 그린 선, 색칠한 도형, 화면에 쓴 글자 등 모든 흔적을 지움
- » t.home 함수 : 거북이가 그린 그림은 그대로 두고, 거북이를 기본 상태로 돌리는 함수
이 명령을 실행하면 거북이는 처음 거북이 그래픽을 시작했을 때의 위치인 원점(0, 0)으로 돌아가 오른쪽을 바라보게 됨(0°). 하지만 그동안 그린 그림은 화면에 남아 있음.

슬라이드 4. title 함수

» t.title 함수를 사용하여 창에 붙은 이름을 바꿀 수 있음

```
t.title("Turtle Run")
```



슬라이드 5. 프로젝트 구조

» 프로젝트 구조

- playing 변수와 start 함수

터틀런 2에는 '게임 중'인 상태와 '대기 중'인 상태가 있는데, 이를 구분하기 위해 playing 변수를 사용함. playing 변수가 참(True)이면 게임이 실행중이라는 뜻이고, 거짓(False)이면 게임을 시작하기 전이거나 끝났다는 뜻.

start 함수는 게임을 시작하는 함수로 사용자가 게임을 시작하려고 SpaceBar를 누르면 start 함수가 호출됨. 이 함수는 현재 게임이 실행 중인지 먼저 확인한 후 아니라면 play 함수를 호출하여 게임을 실행하는 역할을 함.

- 점수(score) 계산

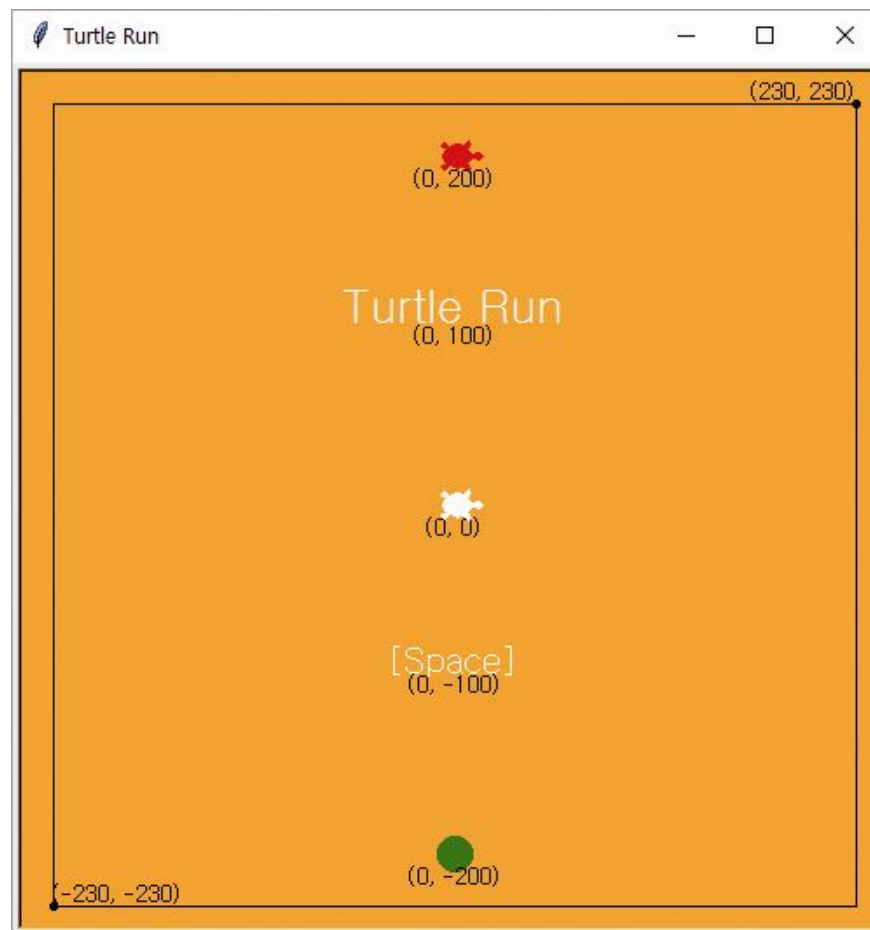
주인공 거북이가 먹이를 먹을 때마다 점수가 올라가도록 하는 부분. 점수에 따라 악당 거북이의 속도를 변화시켜 게임 난이도를 조절.

- message 함수

화면에 게임 제목이나 점수, Game Over 등을 표시하는 함수.

5. 프로젝트 구조

» 터틀런 2에 나오는 캐릭터의 시작 좌퓯값, 먹이의 생성 위치, 제목과 점수를 출력하는 위치



5. 프로젝트 구조

» 터틀런 2 만들기

터틀런 2 만들기

import turtle as t

import random

score = 0

playing = False

te = t.Turtle()

te.shape("turtle")

te.color("red")

te.speed(0)

te.up()

te.goto(0, 200)

점수를 저장하는 변수

현재 게임이 플레이 중인지 확인하는 변수

악당 거북이(빨간색)

5. 프로젝트 구조

```
ts = t.Turtle() # 먹이(초록색 동그라미) # 먹이(초록색 동그라미)
ts.shape("circle")
ts.color("green")
ts.speed(0)
ts.up()
ts.goto(0, -200)
```

```
def turn_right():                # 오른쪽으로 방향을 바꿈
    t.setheading(0)
```

```
def turn_up():                  # 위로 방향을 바꿈
    t.setheading(90)
```

```
def turn_left():                # 왼쪽으로 방향을 바꿈
    t.setheading(180)
```

```
def turn_down():                # 아래로 방향을 바꿈
    t.setheading(270)
```

5. 프로젝트 구조

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

게임을 시작하는 함수

메시지를 지움

```
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
```

게임을 실제로 플레이 하는 함수

주인공 거북이 10만큼 앞으로 이동
1~5 사이에서 뽑은 수가 3이면(20% 확률)

악당 거북이가 주인공 거북이를 바라봄
점수에 5를 더해서 속도를 올림
점수가 올라가면 빨라짐

5. 프로젝트 구조

```
if speed > 15:
    speed = 15
te.forward(speed)
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)
    star_x = random.randint(-230, 230)
    star_y = random.randint(-230, 230)
    ts.goto(star_x, star_y)
if playing:
    t.ontimer(play, 100)
```

속도가 15를 넘지는 않도록 함

주인공과 악당의 거리가 12보다 작으면
게임을 종료

주인공과 먹이의 거리가 12보다 작으면(가까우면)
점수를 올림
점수를 화면에 표시

먹이를 다른 곳으로 옮김

게임 플레이 중이면 0.1초 후
play 함수를 실행

5. 프로젝트 구조

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

메시지를 화면에 표시하는 함수

```
t.title("Turtle Run")  
t.setup(500, 500)  
t.bgcolor("orange")  
t.shape("turtle")  
t.speed(0)  
t.up()  
t.color("white")  
t.onkeypress(turn_right, "Right")  
t.onkeypress(turn_up, "Up")  
t.onkeypress(turn_left, "Left")  
t.onkeypress(turn_down, "Down")  
t.onkeypress(start, "space")  
t.listen()  
message("Turtle Run", "[Space]")
```

거북이 모양의 커서를 사용

거북이 속도를 가장 빠르게 지정

→ 를 누르면 turn_right 함수를 실행

거북이 그래픽 창이 키보드 입력을 받도록 함

5. 프로젝트 구조

» 실행결과

