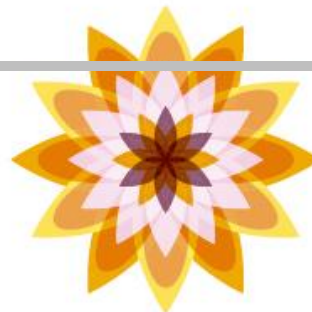
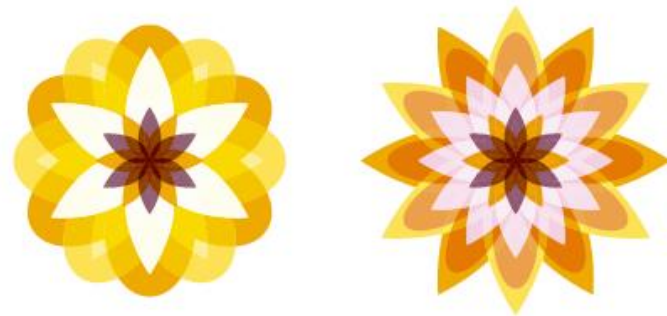


Chapter 04
움직이기

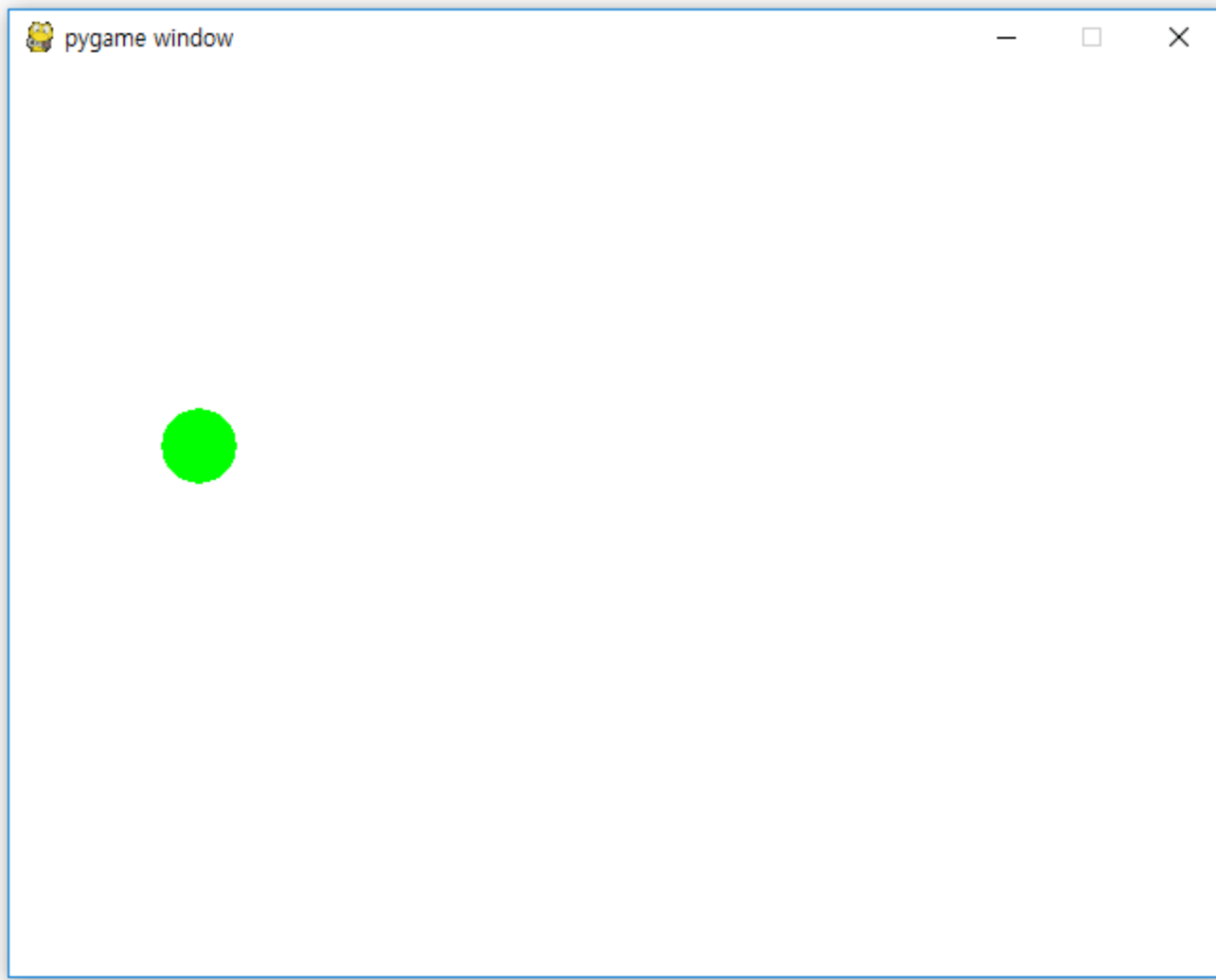


1. 이번에 만들 코드

- 아래 음영 표시된 부분을 프로그램에 써넣고 실행한 뒤 어떤 일이 일어나는지 보세요.

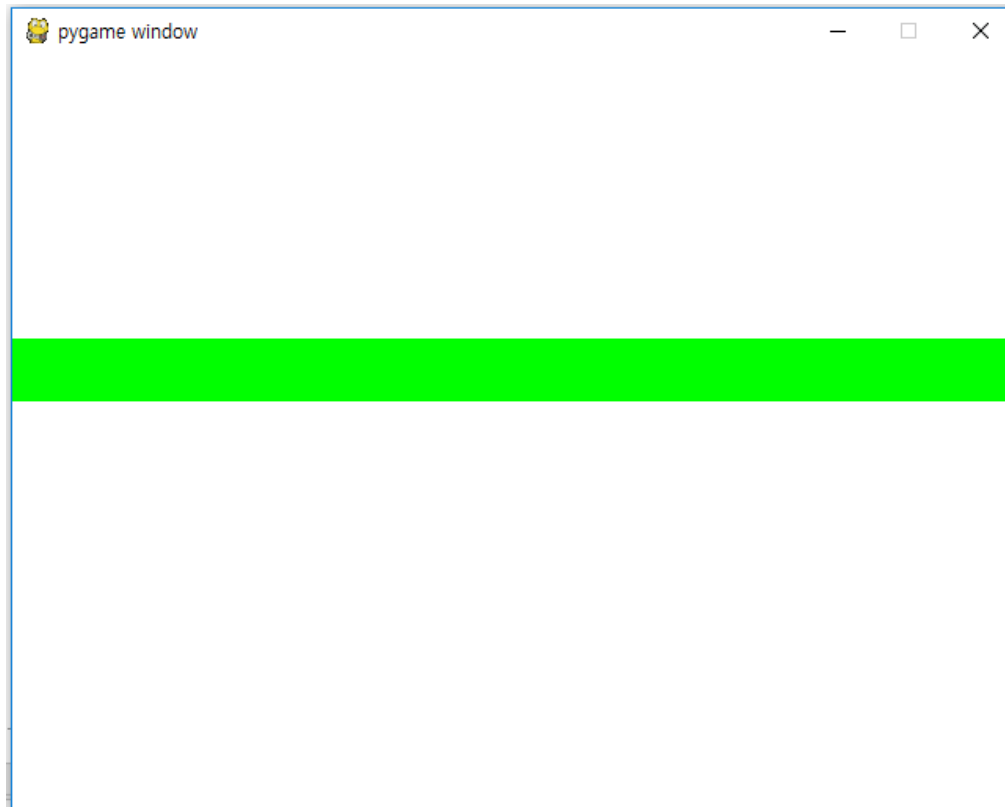
```
import pygame, sys
from pygame.locals import *
pygame.init()
screen = pygame.display.set_mode((640,480))
xpos = 50
while 1:
    for event in pygame.event.get():
        if event.type == QUIT:
            sys.exit()
    xpos += 1
    screen.fill((255,255,255))
    pygame.draw.circle(screen,(0,255,0),(100 xpos,200),20)
    pygame.display.update()
```

1. 이번에 만들 코드



1. 이번에 만들 코드

- 왜 색칠 함수 `screen.fill()`이 게임 루프 안에 있는지 궁금하다고요?
- 스크린은 1번만 색칠해도 되는데, 왜 굳이 루프 안에 써서 또 색칠하냐고요?
- `screen.fill()`을 루프 밖 `while 1:` 코드 위에 써 보세요.
- 들여쓰기는 하지 말고요. 어떤 일이 일어 나나요?



2. 변수


```
pygame.draw.circle(screen,(0,255,0),(340,280),20)
```

- 위 코드에서 원이 스크린 어디에 놓일지는 3번째 인자인 (340 , 280)이 결정합니다.
- 원을 움직이고 싶다면, 위 숫자들을 바꿔야 합니다.
- 하지만 340은 340이고, 바뀌지 않지요.
- 그러니 바꿀 수 있는 숫자 바로 변수가 필요합니다.
- 변수를 만들려면 이름을 붙이고 값을 정하면 됩니다.
- 예를 들어 다음과 같습니다.
 - 생명 = 4
- 이렇게 쓰면 “생명”이라는 변수를 만들고 값을 4로 정한 것입니다.
- 중요한 것은 4가 바뀔 수 있는 값이라는 거예요.
- 만약 죽으면 4에서 1 을 빼라는 코드를 쓸 수도 있지요. 그러면 3이 되지요.
- 원을 움직이기 위해 xpos라는 이름의 변수를 만들고 그 값을 정했습니다.
- 이제 이 값을 바꿔 원을 움직여 봅시다.

3. 변수정하기

- xpos라는 이름의 변수를 만들고 값을 정했습니다.
- x의 위치 X-position의 줄임말이지요.
- 이 값을 바꾸면 원을 움직일 수 있습니다.

```
xpos = 50
```



- 변수는 원하는 대로 이름 붙여도 되지만, 가급적 역할과 관련된 이름을 붙이는 것이 좋습니다.
- 그래야 혹시라도 오류가 발생할 때 발견하기가 쉬우니까요.
- 여러분의 코드를 다른 사람들이 이해하기도 쉽고요.
- 위 코드는 프로그램의 첫 번째 섹션, 게임 루프 위에 씁니다.
- 변수는 1번만 만들면 되므로 루프 안에 있을 필요가 없습니다.

3. 변수정하기

```
pygame.draw.circle(screen, (0,255,0), (xpos,200),20)
```

- 이제 draw.circle() 함수에게 주어지는 x좌표는 xpos입니다.
- 파이썬은 우리가 xpos에게 50이라는 값을 주었다는 사실을 이미 알고 있기 때문에 원의 x좌표를 xpos, 즉 50으로 정했습니다.

```
xpos += 1
```

- += 1은 어떤 값에 1을 더하라는 뜻입니다.
- += 5는 5를 더하라는 뜻이고, -= 2는 2를 빼라는 뜻이지요.
- *= 3은 3을 곱하라는 뜻이고, /= 6은 6으로 나누라는 뜻입니다.
- xpos += 1은 게임 루프 안에 있습니다.
- 루프를 돌 때마다 xpos의 값에 1을 더하라는 뜻입니다.
- 그러면 어떻게 될까요?
- 원의 x좌표가 1씩 커지겠지요.

4. 키를 눌러 움직이기

- 원의 움직임을 제어하기 위해서 `xpos += 1`을 지우고 아래 음영으로 추가된 코드를 넣겠습니다.
- 프로그램 실행 후 오른쪽 또는 왼쪽 화살표 키를 눌러 보세요.

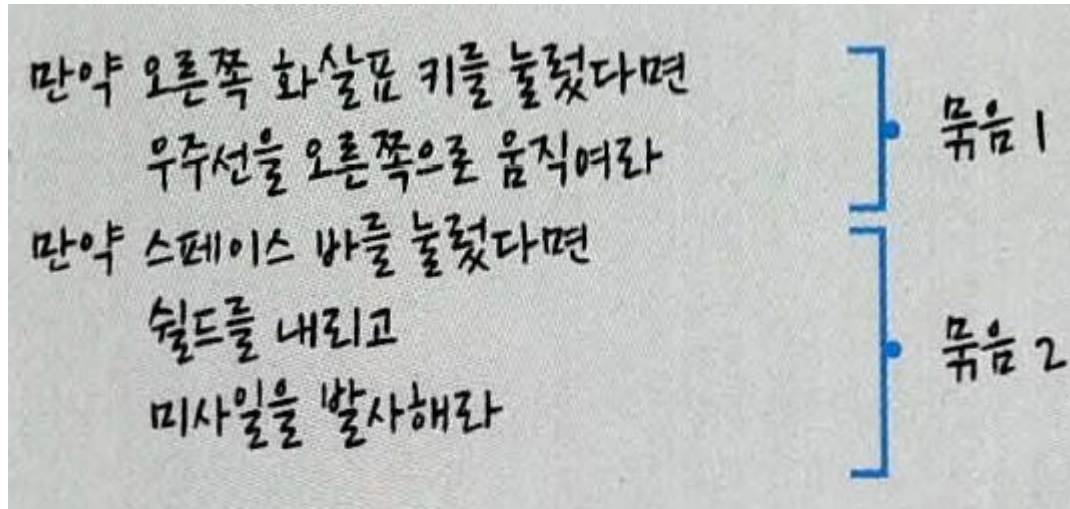
```
import pygame, sys
from pygame.locals import *
pygame.init()
screen = pygame.display.set_mode((640,480))
xpos = 50
while 1:
    for event in pygame.event.get():
        if event.type == QUIT:
            sys.exit()
    xpos += 1
    pressed_keys = pygame.key.get_pressed()
    if pressed_keys[K_RIGHT]:
        xpos += 1
    if pressed_keys[K_LEFT]:
        xpos -= 1
    screen.fill((255,255,255))
    pygame.draw.circle(screen,(0,255,0),(xpos,200),20)
    pygame.display.update()
```


4-1. 들여쓰기

- 파이썬에서 들여쓰기는 아주 중요합니다.
- 들여쓰기는 코드 문치를 구별하는 방법 중 하나니까요.
- 어떤 코드가 들여쓰기 돼 있으면 바로 위 코드와 한 문치라는 뜻입니다.
- 다시 한 번 말하지만, 코드를 들여쓰기 하려면 'space'를 4번 누르면 됩니다.
- 'Tab'을 사용해서 들여쓰기를 할 수도 있지요.

4-1. 들여쓰기

- 다음은 들여쓰기를 사용한 의사 코드의 예시입니다



- 오직 1 번째 줄이 참일 때만 2번째 줄이 실행됩니다.
 - 마찬가지로, 3번째 줄이 참일 때만 4번째, 5번째 줄이 실행됩니다.
 - 만약 1 번째 줄이 참이 아니라면, 즉 오른쪽 화살표 키를 누르지 않았다면, 프로그램은 그 밑에 있는 모든 들여쓰기 줄을 무시하고 그 다음 줄 즉, 3번째 줄로 이동합니다.
- 어떤 줄을 들여쓰기 하려면, 그 줄의 바로 윗줄 끝에는 반드시 콜론(:)이 있어야 합니다.

5. 눌린 키들 리스트

```
pressed_keys = pygame.key.get_pressed()
```

- `pygame.key.get_pressed()`는 키보드의 모든 커와 키 눌림 여부를 리스트로 만듭니다.
- 이 리스트를 눌린 키들 리스트(`pressed_keys`)라고 부릅니다.
- 눌린 키들 리스트 역시 변수입니다.
- 리스트 이름은 마음대로 붙여도 되지만, 보통 지금 사용한 이름이나 이와 비슷한 이름을 사용합니다.
- 어떤 프로그래머들은 `keys`라는 이름을 쓰기도 합니다.
- 리스트에서는 일반적으로 대괄호를 씁니다.

5. 눌린 키들 리스트

```
if pressed_keys[K_RIGHT]:  
    xpos += 1
```

- 위 코드의 1번째 줄에서는 오른쪽 화살표 키("K_RIGHT")의 눌림 여부를 확인합니다.
- 눌렀으면 2번째 줄이 실행됩니다.
- xpos에 1을 더하는 것이지요.
- xpos는 원의 중심의 x좌표이므로 1을 더하면 원이 그려지는 위치가 오른쪽으로 이동합니다.
- 그럼 오른쪽으로 원이 굴러가는 것처럼 보일 거예요.

5. 눌린 키들 리스트

```
if pressed_keys[K_LEFT]:  
    xpos -= 1
```

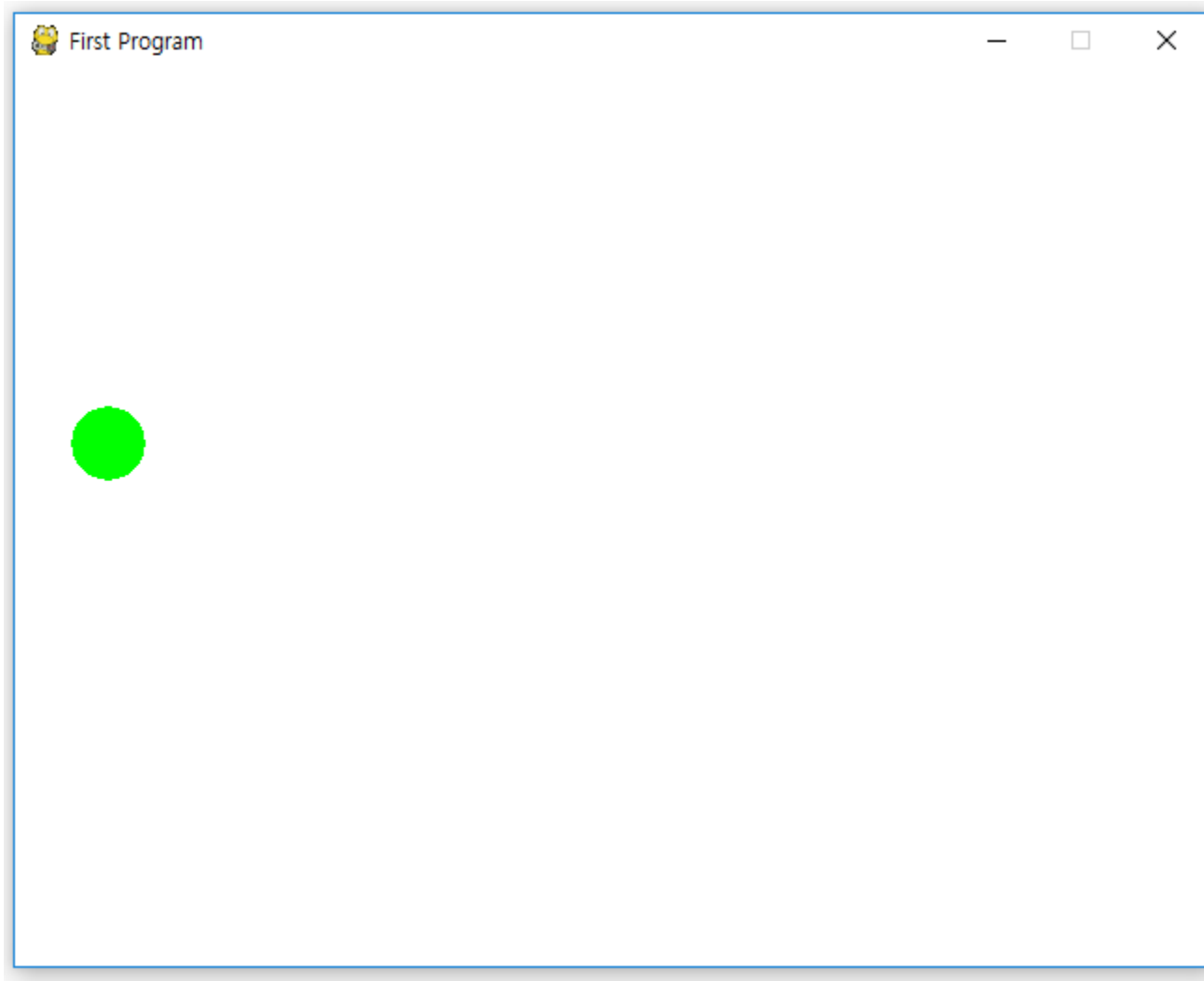
- 오른쪽 화살표를 누를 때와 같은 방식으로 작동합니다.
- 위 코드의 1번째 줄에서는 왼쪽 화살표("K_LEFT")의 눌림 여부를 확인합니다.
- 눌렀으면 2번째 줄이 실행됩니다.
- xpos에서 1을 빼는 것이지요.
- xpos에서 1을 빼면 원이 그려지는 위치가 왼쪽으로 이동합니다.
- 왼쪽으로 원이 굴러가는 것처럼 보이겠지요.

6. 시계와 제목 추가

- 시계와 제목을 추가합니다.
- 둘 다 필수는 아니지만, 적어도 시계는 추가하는 습관을 들이는 것이 좋습니다.

```
import pygame, sys
from pygame.locals import *
pygame.init()
pygame.display.set_caption("First Program")
screen = pygame.display.set_mode((640,480))
xpos = 50
clock = pygame.time.Clock()
while 1:
    clock.tick(60)
    for event in pygame.event.get():
        if event.type == QUIT:
            sys.exit()
    pressed_keys = pygame.key.get_pressed()
    if pressed_keys[K_RIGHT]:
        xpos += 1
    if pressed_keys[K_LEFT]:
        xpos -= 1
    screen.fill((255,255,255))
    pygame.draw.circle(screen, (0,255,0), (xpos,200), 20)
    pygame.display.update()= pygame
```

6. 시계와 제목 추가



6. 시계와 제목 추가

- 아래 코드를 사용하여 게임 윈도우에 제목을 넣었습니다.

```
pygame.display.set_caption("First Program")
```

- 제목은 반드시 따옴표 안에 써야 합니다.
- 프로그램을 실행시키고 게임 윈도우 윗부분을 확인해보세요.

6. 시계와 제목 추가

```
clock = pygame.time.Clock()
while 1:
    clock.tick(60)
```

- 게임 루프는 1초에도 여러 번, 가능한 한 최고 속도로 돌립니다.
- 그런데 루프 안에 코드를 쓰면 아주 약간이지만, 그 코드를 실행시키는 시간만큼 프로그램이 느려집니다.
- 스크린의 속도가 바뀌는 셈이지요.
- 따라서 모든 루프가 같은 속도로 돌기 위해서는 시계 함수를 써야 합니다.
- 1번 루프를 도는 데 걸리는 시간을 일정하게 고정시키는 것입니다.
- `clock.tick()` 함수에 있는 "60"은 while 루프가 도는 시간을 1/60초로 고정시킵니다.
- 10이라고 쓰면 1/10초로 고정됩니다.
- 우리가 지금 만드는 프로그램에는 60이 적당합니다.

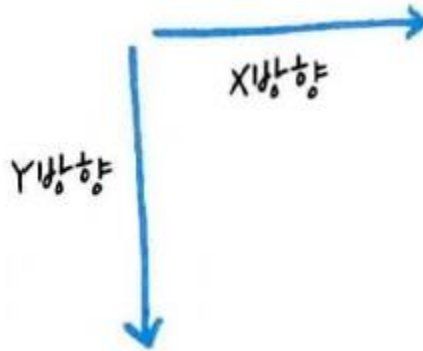
6. 시계와 제목 추가

```
clock = pygame.time.Clock()
while 1:
    clock.tick(60)
```

- 코드를 추가하니 원이 아까보다 느리게 움직이지요?
- 이유가 뭘까요?
- 이전에는 게임 루프가 1/60초보다 훨씬 빠르게 돌았거든요.
- 루프마다 원이 움직이는 정도를 증가시켜 더 빠르게 움직이도록 할 수도 있습니다.
- `xpos += 5`로도 바꿔 보세요.

7. 위아래

- 원을 위아래로 움직이려면 어떻게 할까요?
- "ypos"라는 이름의 변수를 추가합니다. (변수 이름은 마음대로 정해도 됩니다.)
- 위쪽, 아래쪽 화살표 키는 각각 "K_UP"과 "K_DOWN"입니다.



- ypos에 5를 더하면 원이 아래로 내려갑니다.
- y축에서는 아래 방향이 양수기 때문입니다.
- 만약 다른 키를 써 보고 싶으면, "K_a" 형식으로 쓰면 됩니다.
- "z"키는 "K_z"가 되겠지요.
- 우리의 첫 번째 프로그램, 움직이는 원 그리기가 끝났습니다



Thank You
