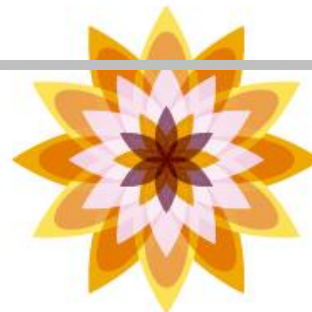
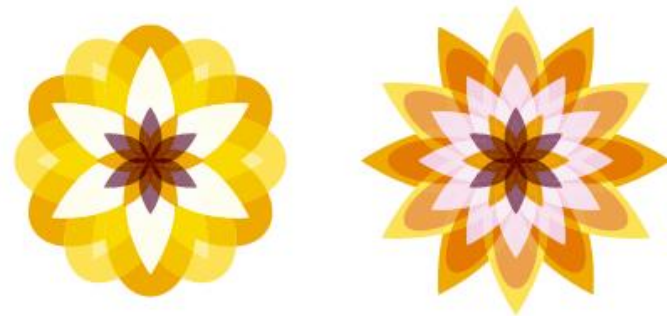


Chapter 03
우주 창조



1. 이번에 만들 코드



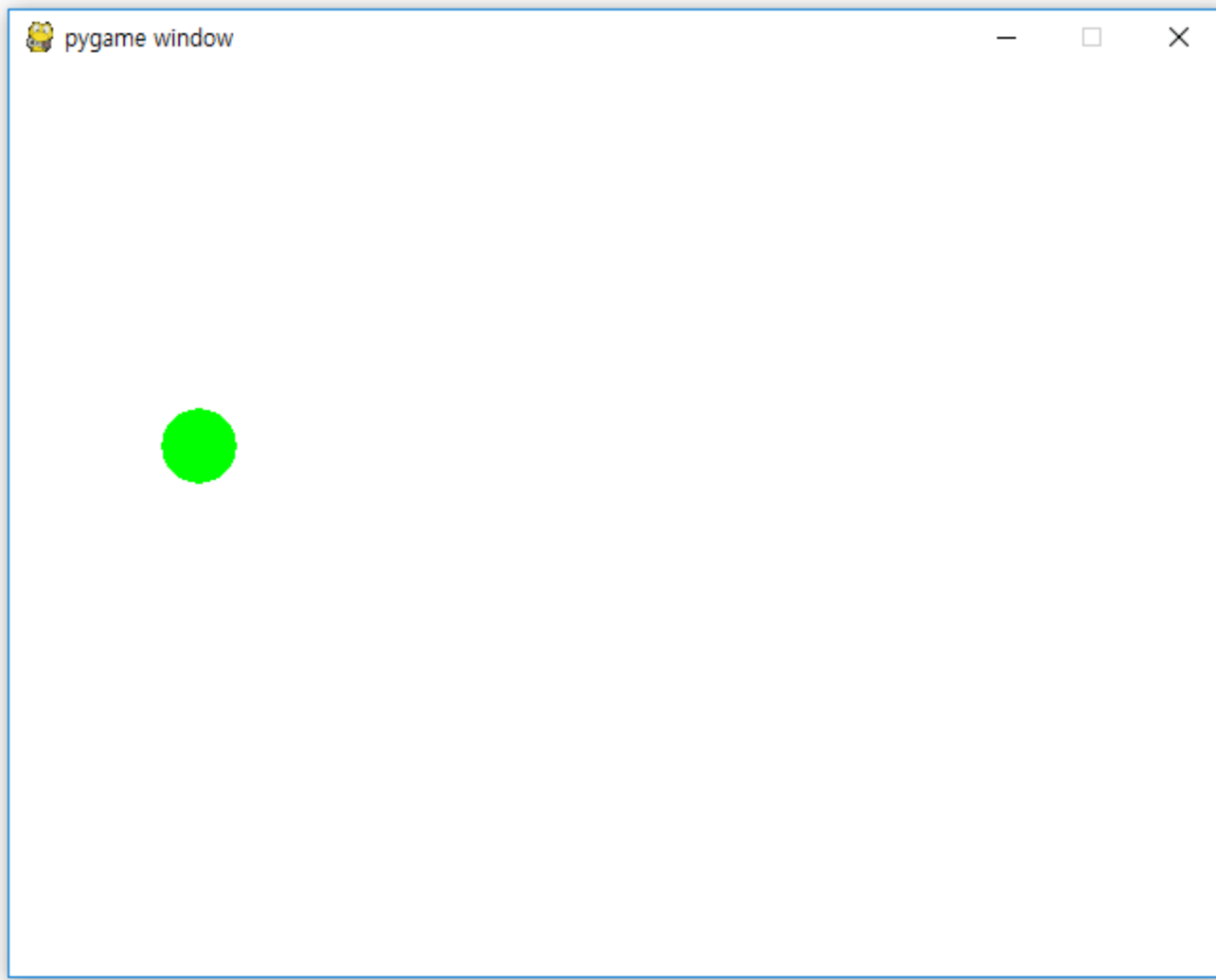
새 파일

firstprogram.py

```
1 import pygame, sys
2 from pygame.locals import *
3 pygame.init()
4 screen = pygame.display.set_mode((640, 480))
5 while 1:
6     for event in pygame.event.get():
7         if event.type == pygame.QUIT:
8             sys.exit()
9     screen.fill((255, 255, 255))
10    pygame.draw.circle(screen, (0, 255, 0), (100, 200), 20)
11    pygame.display.update()
```

실행

1. 이번에 만들 코드



2. 도구상자 가져오기 (import)

```
import pygame, sys
```

- 1번째 줄은 도구를 가져오라는 뜻입니다.
- 자전거를 고칠 때 오븐을, 빵 구울 때 드라이버를 사용하지는 않잖아요?
- 컴퓨터 프로그래밍도 마찬가지입니다.
- 파이썬에 적절한 도구가 있지요.
- 파이게임 pygame이나 시스템 sys 같은 것들이요.
- 이런 것들을 일단 가져오라 Import는 뜻이지요.
 - pygame은 파이썬을 위한 미술 도구입니다. 게임이나 애니메이션을 만들 때 필요한 그래픽 작업에 쓰이는 여러 도구가 담겨 있습니다.
 - sys는 파이썬을 실행시키는 컴퓨터 시스템 부분과 파이썬이 대화할 수 있도록 해 줍니다.

2. 도구상자 가져오기 (import)

```
from pygame.locals import *
```

- 나중에 쓰기 편하게 파이게임의 함수들을 한 번에 모두 준비시키는 코드입니다.

```
pygame.init()
```

- 이 코드는 파이게임을 초기화시킵니다.
- 파이게임의 전원을 누른다고 해야 할까요?

3. 스크린

- 우리가 만든 우주의 이름은 스크린screen입니다.
- 우리는 이곳에 '게임'이라는 빅뱅을 일으킬 거예요.
- 모든 일이 스크린 위에서 일어나지요.

```
screen = pygame.display.set_mode((640, 480))
```

- 맨 끝, 괄호 안의 숫자 2개는 이 우주의 크기를 알려 줍니다.
- 640은 스크린의 너비, 480은 높이지요.
- 이 숫자들을 바꾸면 스크린의 크기와 모양이 바뀝니다.
- 이 숫자들은 scree를 만드는 set_mode()라는 함수의 인자입니다.

3-1. 인자

- 인자arguments는 괄호 안에 들어 있는 정보 조각입니다
- 예를 들어 봅시다. `clock.tick(60)`이란 코드에서 `clock.tick()`은 함수고 60은 인자입니다
- `clock.tick()` 함수에는 1개의 인자가 필요합니다. 괄호 안에 써야 합니다.
- 다른 숫자를 써도 됩니다. 하지만 하나만 써야해요
- `clock.tick()` 함수는 괄호 안에 숫자 1 개만 있을 거라고 예상합니다.
- 만약 괄호 안에 단어 1개나 숫자 2개를 쓴다면 이렇게 말할 거예요.
 - “도대체 뭘 어쩌라는 거야?”
- 하지만 숫자 1 개를 쓰면 웃으면서 말하겠지요
 - “난 이걸로 무엇을 할 지 알고 있어 ”
- 무엇을 하냐고요? `clock.tick(60)`은 1초에 60번 똑딱거리는 시계를 만듭니다. 왜 만들까요?
 - 뒤에서 알게 됩니다.

4. 게임루프

- 여기까지는 실제로 뭔가를 한다기보다는 프로그램을 준비시키는 코드라고 할 수 있습니다.
- 준비, 즉 셋업 setup 부분이지요.
- 준비를 다 마쳤으면 프로그램을 실행시켜야 합니다.
- 아래 코드는 게임 루프game loop의 시작을 나타냅니다.

```
while 1:
```

- 보통 메인 루프main loop라고 부르지만, 우리는 게임 루프라고 부르겠습니다.
- 게임을 만들 때 올 거니까요!
- 루프는 코드의 반복되는 부분을 가리킵니다.
- For 루프. While 루프 등이 있습니다.
- 둘 다 그 밑에 나오는 코드를 반복해서 실행하라는 뜻입니다.


4. 게임루프

- 일단 루프 안에 들어가면 빠르게 일이 진행됩니다.
- 루프 안의 코드는 아주 짧은 순간에도 여러번 반복됩니다.
- 루프란 계속해서 반복 실행되는 거니까요.
- 키보드를 눌렀는지, 마우스 클릭이 있었는지 여부도 여기서 감지하지요.
- 스크린 위에서 움직이는 것을 제어하거나 충돌 또는 점수, 생명 등의 변화를 기록하는 부분이기도 하고요.
- 이런 것들에 대해서는 나중에 설명하겠습니다.
- 지금은 while 1: 밑에 나오는 코드 몇 줄은 들여쓰기 해야 한다는 것만 기억하세요.
- 들여쓰기 한 코드들은 게임 루프 안에서 실행될 거예요.
- 아주 짧은 순간에도 여러 번 반복된다는 뜻입니다.
 - 어떤 프로그래머는 while 1: 대신 while True : 라고 쓰기도 해요. 같은 뜻입니다.

5. 빠른 끝내기

- 루프 안에 들어갔다면, 나갈 수도 있어야겠지요?
- 아래 세 줄만 알면 루프에서 빠져나올 수 있습니다.
- 아래 코드를 모른 채 '파이썬'이란 버스에 올라타면 하루 종일 원을 그리며 뱅뱅 돌 뿐 절대 내릴 수 없을 거예요.

```
for event in pygame.event.get():  
    if event.type == pygame.QUIT:  
        sys.exit()
```

- 1번째 줄은 마지막 게임 루프를 도는 동안 일어난 일을 감지합니다.
- 키보드 눌림 또는 마우스 클릭 같은 것들 말이지요.
- 2번째 줄은 버튼을 눌렀는지 확인합니다.
- 파이게임에서는  아이콘의 이름이 끝QUIT 입니다.
- 3번째 줄은 버튼이 클릭되면 프로그램을 끝내라는 뜻입니다.

5-1. 함수

- 함수는 어떤 것을 하는 어떤 것입니다.
- 원을 그리거나 스크린을 채우거나 정보를 전달할 수도 있습니다
- 모든 함수의 끝에는 괄호를 씁니다
- 소괄호() 한 쌍 말이지요
- 괄호 안에는 함수가 필요로 하는 정보를 넣을 때도 있지만, 그런 정보가 없어도 괄호는 항상 써 주는 게 좋습니다
- 다른 사람들도 함수가 함수라는 걸 알아야 하니까요

6. 색칠하기

- 스크린을 색칠합니다.

```
screen.fill((255, 255, 255))
```

- `screen.fill()` 함수는 스크린 색칠에 사용됩니다.
- 이 함수에는 인자가 1개 있습니다.
- 괄호 안에 들어 있는 숫자 3개 즉 (255, 255, 255)가 하나의 인자지요.
- 각 숫자는 하나의 색깔을 나타냅니다.
- `screen.fill()` 함수의 인자는 3개의 숫자로 이루어져 있습니다.
- 파이썬에게 이 인자는 세 조각으로 이루어진 하나의 인자라는 것을 알려 주려면, 숫자를 괄호 안에 묶어야 합니다.
- 그래서 괄호 2개를 쓴 것입니다.
- 하나는 함수 때문에 쓴 거고, 하나는 함수 안의 인자 때문에 쓴 거지요.
- 이렇게 숫자를 묶은 것을 튜플tuple이라고 부릅니다.
- 튜플에 대해서는 뒤에서 자세히 배울게요.

6-1. 색깔

- 파이썬에서는 항상 RGB 시스템으로 색을 표현합니다.
- 빨강Red 초록Green 파랑Blue을 섞어서 다양한 색을 나타내지요
- 이렇게 색들을 섞을 때는 빛이 섞일 때의 규칙을 따르지, 물감이 섞일 때의 규칙을 따르지 않습니다.
- 빛을 섞을 때는 빛의 3원색, 물감을 섞을 때는 색의 3원색을 따릅니다.
- 각 색의 양은 0부터 255 사이에서 정해집니다.
- 따라서 (255, 0, 0)이라고 하면 빨강은 255, 녹색과 파랑은 0이라는 뜻입니다
- (0, 0, 0)은 색이 하나도 없으므로 검정색입니다.
- (255, 255, 255)는 흰색이겠지요
- (255, 140, 0)이라고 하면 다크오렌지색이 됩니다.
- 인터넷에서 찾아보면 어떤 색은 어떤 번호로 나타내는지 알려 주는 사이트가 많이 있습니다.
 - 만약 (75 , 75, 75)처럼 세 숫자가 모두 같으면 회색이 됩니다
 - 은색(192, 192, 192)도 회색의 일종입니다

7. 원그리기

- 스크린 위에 원을 그렸습니다.

```
pygame.draw.circle(screen, (0, 255, 0), (100, 200), 20)
```

- `pygame.draw.circle()` 함수는 4개의 인자를 갖습니다.
- 순서대로 4개의 인자를 주지 않으면 이 함수는 꿈쩍하지 않을 거예요.
- 자, 지금부터 차례로 살펴봅시다.
- 실제 함수는 `circle()` 입니다.
- 이 함수는 `draw`라는 서브모듈 안에 있는데, `draw`는 `pygame`파이게임이라는 모듈 안에 있지요.

7. 원그리기

- 아래는 각 인자 설명입니다.

```
pygame.draw.circle(screen, (0, 255, 0), (100, 200), 20)
```

■ Screen

- 첫 번째 인자는 원이 그려지는 곳을 지정합니다.
- 스크린 위에 원이 그려지는게 당연한 것처럼 보이지만, 'screen'이란 우리가 게임 화면에 붙인 이름일 뿐입니다.
- 원은 스크린 위에 있는 다른 객체 위에 있을 수도 있어요.
- 따라서 함수는 원을 어디에 그릴지 알아야만 합니다.

■ (0, 255, 0)

- 이것은 원의 색을 나타냅니다. 이 경우에는 녹색입니다.

■ (100, 200)

- 스크린 위에서 원의 위치를 나타냅니다.
- 위치는 2개의 숫자로 이루어진 튜플tuple로 주어집니다.
- 처음 숫자는 x좌표입니다. 이것은 스크린의 가장 왼쪽 끝에서부터 원의 중심까지 이르는 거리를 픽셀로 나타낸 것입니다.
- 두 번째 숫자는 y좌표입니다. 이것은 스크린의 가장 위에서부터 원의 중심까지의 거리입니다.

7. 원그리기

- 아래는 각 인자 설명입니다.

```
pygame.draw.circle(screen, (0, 255, 0), (100, 200), 20)
```

- 20
 - 원의 반지름입니다.
- 5번째 인자를 추가할 수도 있습니다.
- 5번째 인자로 테두리 두께를 추가하여, 색이 채워지지 않은 원을 그릴 수 있습니다.
- 이 인자를 쓰지 않으면, 파이게임은 그냥 원을 단색으로 채웁니다.
- 일단 넣어 보세요. 20 다음에 쉼표를 쓰고 5나 1을 써보세요.

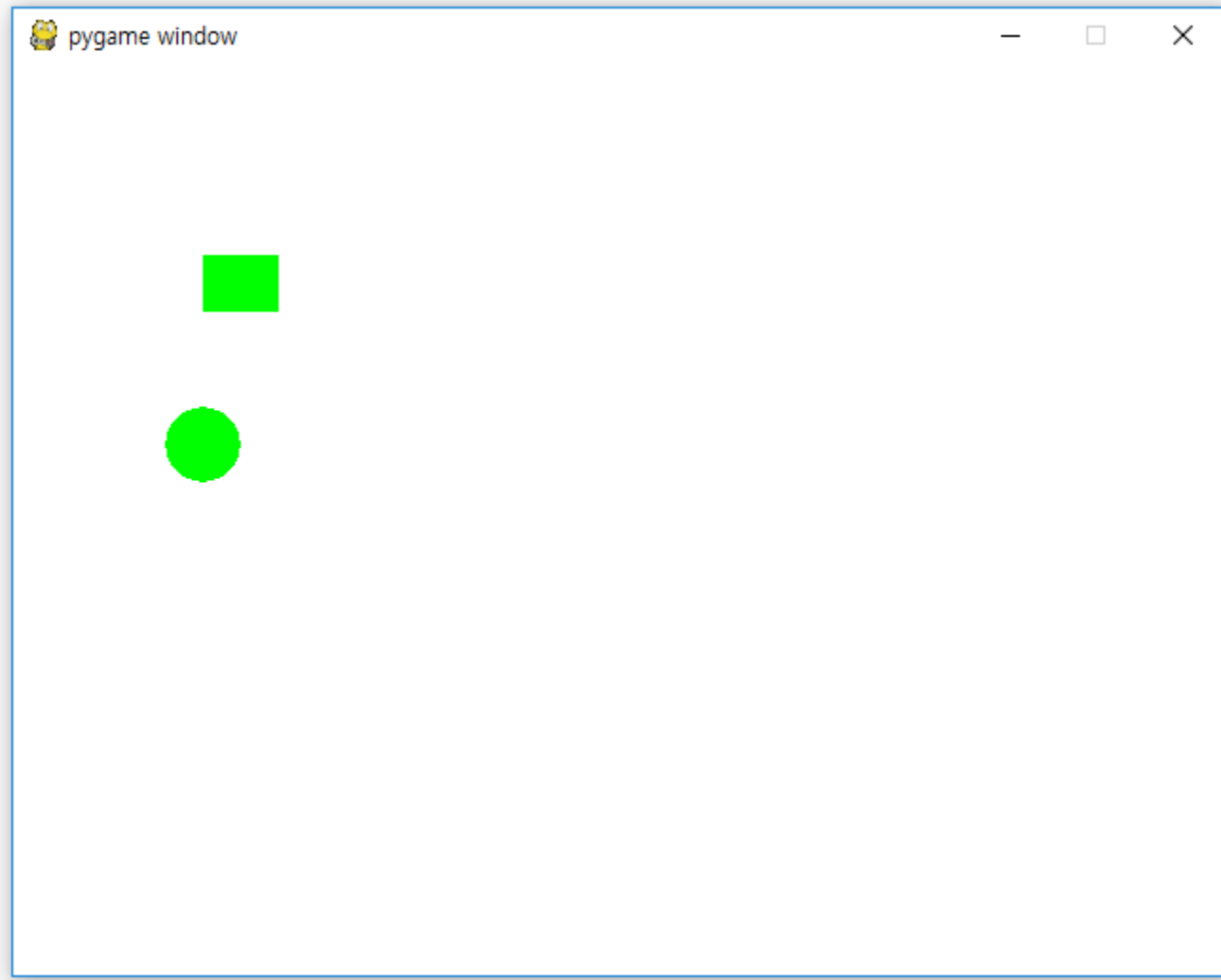
7. 원그리기

- 원의 경우 (100 , 200)은 위치 인자로 원의 중심의 위치를 나타낸다고 위에서 말했지요.
- 하지만 직사각형을 그릴 때는 위치 인자는 직사각형의 왼쪽 위 꼭짓점의 위치를 나타냅니다.
- 직사각형의 크기는 별도의 인자로 따로 추가하지 않고, 가로와 세로의 길이를 좌표 인자 끝에 추가하여 나타냅니다.
- 따라서 직사각형의 경우에는 세 번째 인자는 서로 다른 4개의 값을 갖습니다.
- 2개는 위치를 나타내고 2개는 크기를 나타냅니다.
- 한 번 확인해 볼까요?
- 아래 함수를 코드 안에 넣어보세요.

```
pygame.draw.rect(screen, (0, 255, 0), (100, 100, 40, 30))
```

- 직사각형을 그리려면 이 코드를 `pygame.draw.circle()` 밑에 쓰세요.
- 물론 들여쓰기도 같게 합니다.

7. 원그리기



7. 원그리기

```
pygame.draw.rect(screen, (0, 255, 0), (100, 100, 40, 30))
```

- 세 번째 인자의 숫자도 바꿔 보세요.
- 원과 직사각형이 같은 위치에 나타나게 해 보세요.
- 두 코드의 위치를 바꿔서 직사각형 코드가 원 코드 앞에 오게도 해 보세요.
- `pygame.draw.rect()` 함수 또한 테두리 두께를 추가 인자로 가질 수 있습니다.
- 맨끝에 있는 `)`의 앞에 `5`나 `1`을 쓰고 실행시켜 보세요.

8. 화면에 표시하기

```
pygame.display.update()
```

- 마침내 마지막 코드네요.
- 이 코드는 게임 루프를 도는 동안 스크린에 어떤 변화가 생겼는지 확인하여 화면에 보여 줍니다.
- 이 코드가 없으면, 뭘 해도 화면에 안 나타날 거예요.



Thank You
