



게임을 만들며 배우는 파이썬 2 (with pygame)

- 애니메이션

오류중학교 코딩 동아리
강사: 여 은 선

확인합시다!

- ☆ <https://github.com/chpink0518/oryuCC.git>
소스 코드를 다운로드합니다
- ☆ **infos2.ipynb 파일만** make_game폴더에서 카피해 옵니다
main.py는 1장에서 사용한 것에 더해 사용합니다
- ☆ infos2.ipynb 파일의 각 셀 단위를 main.py로 옮겨오며 테스트
-> **주의:** 순서나 위치가 맞지 않음, **셀을 실행하지 마세요!!**

아이콘을 바꿔봅시다

☆ graphics 폴더

☆ infos2.ipynb의 첫 번째 셀

```
# 01. change icon
```

```
icon = pygame.image.load('graphics/snail_icon.png')
```

```
pygame.display.set_icon(icon)
```

-> 제목 지정(set_caption) 아래에 추가

달팽이를 그려 봅시다

☆ infos.ipynb의 2, 3 번째 셀

02. snail

```
snail_surface = pygame.image.load('graphics/snail/snail1.png')  
win.blit(snail_surface, (600,250))
```

-> 다른 서피스 생성 아래에 추가
-> while 루프 다른 블릿들
'아래'에 추가

☆ Ctrl + F5를 눌러 실행해 봅니다

03. animate snail

```
snail_x_pos = 600
```

-> 서피스 생성 아래에 추가

☆ mission!

snail_x_pos 변수로 달팽이를 '왼쪽'으로 움직여 보세요!
(hint: pygame 좌표계를 생각해 보기!!)

사라진 달팽이 데려오기

☆ infos2.ipynb의 4 번째 셀

```
# 04. animate snail continuously  
if snail_x_pos < -100: snail_x_pos = 800
```

-> 어느 위치에 넣어야 할까?

-> snail1.png의 이미지 크기가 72 x 36 이므로
화면에서 완전히 사라진 -100 정도의 위치일 때
화면 끝으로 데려옴

☆ mission!
프레임레이트도 조정해 보세요!

☆ 배경을 지워봅시다

- 하늘과 땅 블라라인을 선택하고 Ctrl+/ 로 주석처리

이미지 블릿 속도 높이기

☆ infos2.ipynb의 5 번째 셀

```
# 05. convert image
```

```
sky_surface = pygame.image.load('graphics/Sky.png').convert()
```

- > 블릿하기 전 포맷에 맞지 않으면 매번 변환 실행
서피스 생성 시 미리 변환 -> 매 블릿마다 변환 필요 x -> 속도 UP!
- > **.convert()**만 복사
이미지 가져온 하늘, 땅, 달팽이 부분에 넣는다

☆ Ctrl + F5로 확인해 보기

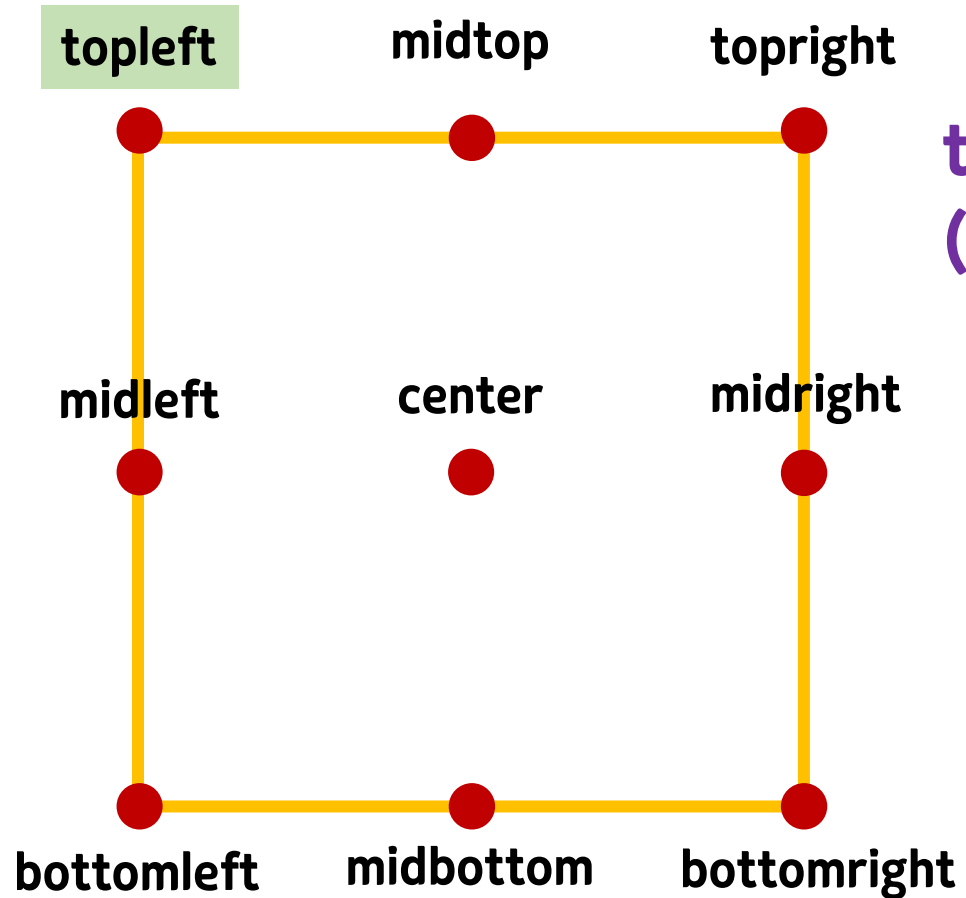
```
# 06. convert image + alpha
```

```
snail_surface = pygame.image.load('graphics/snail/snail1.png').convert_alpha()
```

- > 달팽이는 배경을 투명히 처리하는 알파값을 고려해야 한다
.convert_alpha()만 복사 -> 달팽이 부분에 넣는다

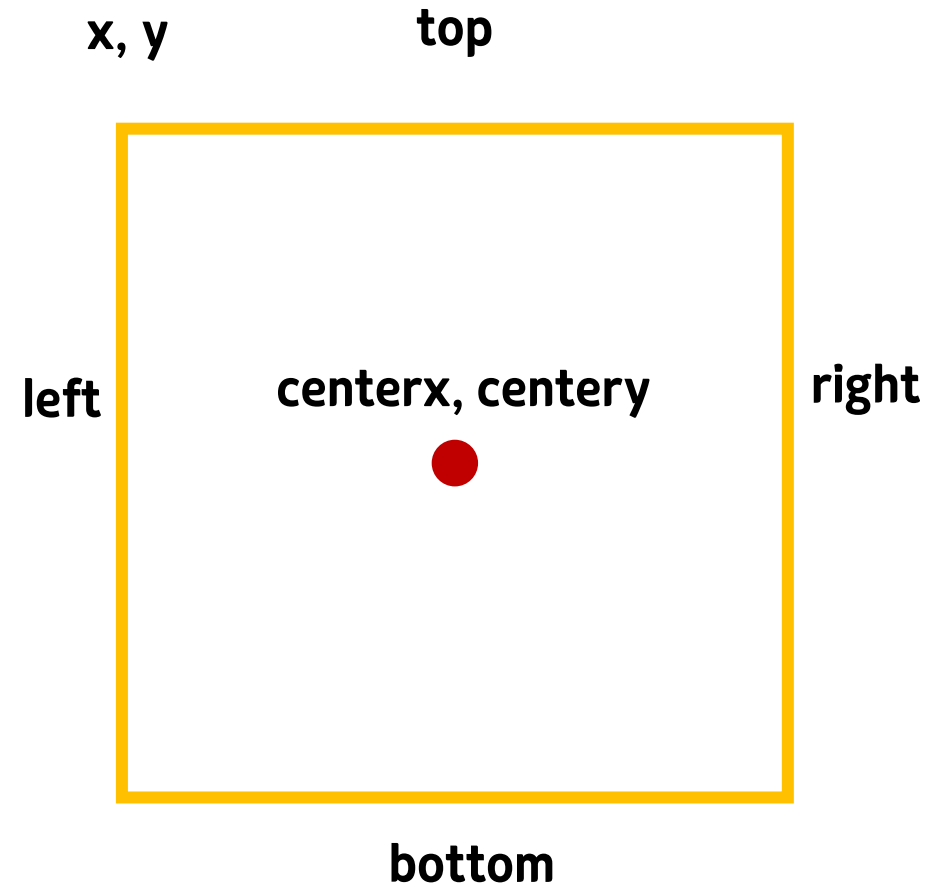
Rectangle - 사각형

☆ 이미지 변환 서피스를 화면에 그리는 기준



tuple
(x, y)

or



Rectangle

☆ infos2.ipynb의 8번째 셀

```
player_rect = player_surf.get_rect(topleft = (80, 200))  
win.blit(player_surf, player_rect)
```

-> 서피스 생성 아래에 추가

-> 블릿 시리즈 아래에 추가

-> get_rect의 매개변수 **topleft**를 **bottomleft**으로 바꾸면?

-> rect(사각형)를 활용하면 **기준점을 9개로** 늘릴 수 있음

☆ 플레이어가 땅을 디디고 서게 해 봅시다

```
player_rect = player_surf.get_rect(bottomleft = (80, 300))
```

-> 땅 서피스의 top좌표인 300 사용

☆ mission!

달팽이에게도 사각형을 만들어 주세요!

사각형 기준점 중심으로 움직이게 해 봅시다

(**hint**: 땅을 그린 위치 참고하기!!)

달팽이도 사각형이 필요해요

☆ infos2.ipynb의 9번째 셀

```
# 09. snail_surface with rectangle
snail_rect = snail_surface.get_rect(bottomright = (600, 300))

snail_rect.x -= 4
if snail_rect.right <= 0: snail_rect.left = 800

win.blit(snail_surface, snail_rect)
```

-> 보다 정교한 처리 가능

- **bottomright** 를 기준으로 잡은 이유는?
- 플레이어 뒤로 달팽이가 지나가는 이유는?

텍스트 서피스를 점수판으로

☆ infos2.ipynb의 10번째 셀

```
# 10. score_surf with rectangle
score_surf = my_font.render('SCORE', False, (64, 64, 64))

score_rect = score_surf.get_rect(center = (400, 50))

pygame.draw.rect(win, '#c0e8ec', score_rect)
pygame.draw.rect(win, '#c0e8ec', score_rect, 10)

win.blit(score_surf, score_rect)
```

-> text_surface 이름을
score_surf로 바꾸고
센터 기준 사각형을 추가

`pygame.draw.rect(서피스, 색상, 그릴 사각형)` -> 여백 없이 꽉 찬 사각형 그림

+ 두께:

기본설정 (0):	채운 사각형
양수 (>0):	윤곽선 두께, 내부 채움 없이 윤곽선만 그림
음수 (<0):	그리지 않는다

필 준비를 합시다

☆ infos2.ipynb의 11번째 셀

#11. make event loop code for jump

```
if event.type == pygame.KEYDOWN:  
    if event.key == pygame.K_SPACE:  
        print('점프')
```

```
if event.type == pygame.MOUSEBUTTONDOWN:  
    if player_rect.collidepoint(event.pos):  
        print('점프')
```

-> 어떤 키든 키가 눌렸을 때

-> 그 키가 스페이스 키면

-> 어떤 마우스 버튼이 눌렸을 때

-> 플레이어의 사각형이
마우스 위치와 충돌?

가짜지만 좀 더 그럴 듯하게

☆ infos2.ipynb의 12번째 셀

```
#12. add gravity  
player_gravity = 0  
  
player_gravity += 1  
player_rect.y += player_gravity
```

-> 매 프레임마다 가속도 증가
증가한 가속도 만큼 플레이어 y값 증가
: 오래 떨어질수록 빠른 속도

☆ infos2.ipynb의 13번째 셀

```
# 13. event loop re-visit  
player_gravity = -20
```

-> 이전에 추가한 loop 부분에서
print('점프')를 이것으로 대체

☆ infos2.ipynb의 14번째 셀

```
#14. set y boundary for the player  
if player_rect.bottom > 300:  
    player_rect.bottom = 300
```

-> 플레이어가 땅을 뚫고 내려가지 않게

2단 점프 막기

☆ infos2.ipynb의 15번째 셀

```
#15. player only jump once at a time  
player_rect.bottom == 300
```

☆ mission!

이 비교문을 사용해서
한번에 한 번만 점프할 수 있도록 만드세요