



# 게임을 만들며 배우는 파이썬 1 (with pygame)

오류중학교 코딩 동아리  
강사: 여 은 선

## 오늘의 학습 목표

1. 비디오 게임의 동작 원리를 알아본다
2. STEP BY STEP으로 pygame의 코드를 옮겨본다
3. 각 단계마다 그 코드의 원리를 이해한다



Clear Code

# 확인합시다!

- ☆ <https://github.com/chpink0518/oryuCC.git>  
소스 코드를 다운로드합니다. - **make\_game** 폴더에 집중!
- ☆ infos.ipynb 파일과 빈 파일인 main.py 두 개의 파일을 사용
- ☆ infos.ipynb 파일의 각 셀 단위를 main.py로 옮겨오며 테스트  
-> **주의:** 순서나 위치가 맞지 않음, **셀을 실행하지 마세요!!**
- ☆ 터미널에 **pip list** 입력:
  - 현재 **인스톨된 라이브러리 리스트** 보여줌
  - **pygame 라이브러리가 설치된 상태인지 확인**

# 비디오 게임의 동작 원리

## 1. 플레이어의 입력을 체크한다



## 2. 입력된 값과 게임 내부 로직에 따라 요소들을 출력한다 (화면, 소리 등)



# pygame 시작하기

## ☆ pygame의 역할

- 이미지 그리기, 음원 재생 등 미디어 작업을 돕는다
- 플레이어의 입력을 확인하는 방법을 제공한다
- 충돌탐지, 텍스트 그리기, 타이머 등의 기능 제공

## ☆ infos.ipynb의 첫 번째 셀

```
# 1. import pygame
import pygame
```

-> pygame 라이브러리를 가져온다

```
# 2. initialize pygame
pygame.init()
```

-> pygame 라이브러리를 사용하기 위해  
초기화 작업을 한다

# window 만들기

## ☆ infos.ipynb의 두 번째 셀

```
# 3. creating a window
```

```
win = pygame.display.set_mode((800, 400))
```

-> 메소드 **소괄호 안 소괄호**의 정체는? **튜플**  
좌표 표현에 많이 사용됨  
리스트처럼 컨테이너형 자료형, 소괄호 생략 가능  
인덱싱, 슬라이싱 가능하나 튜플 내부 값 수정 불가

-> 가로 800, 세로 400 사이즈의 윈도우(display surface)를 만듭니다

-> **Ctrl + F5로 실행해서 결과를 확인해 봅시다**

# 화면이 사라지지 않게 하려면?

☆ `infos.ipynb`의 세 번째 셀

```
# 4. main loop  
while True:  
    pygame.display.update()
```

- > `while True`로 무한 루프를 만든다
- > 디스플레이를 업데이트한다
- > `Ctrl + F5`로 실행해서 결과를 확인해 보시다

# 이벤트 루프를 추가합시다

☆ 이벤트란? - 사용자의 입력

☆ `infos.ipynb`의 네 번째 셀

```
# 5. event handling loop
for event in pygame.event.get():
    if event.type == pygame.QUIT:
        pygame.quit()
```

- > 사용자에게 입력받아 화면을 업데이트하는 구조이므로 `pygame.display.update()` 전에 이벤트 루프가 있어야 함
- > 이벤트 큐에서 이벤트를 가져와 처리
- > 윈도우 종료버튼을 누르면 `pygame.QUIT`이벤트 발생
- > `Ctrl + F5`로 실행해서 결과를 확인해 보시다



# 정말 프로그램을 끝내봅시다

☆ **sys 모듈**: 표준 라이브러리, 인터프리터 제어

☆ **infos.ipynb**의 다섯 번째 셀

```
# 6. import exit method from sys  
from sys import exit  
exit()
```

-> **import pygame**아래에 **import** 윗줄 붙여넣기

-> **pygame.quit()** 아래줄에 **exit()** 붙여넣기

-> **Ctrl + F5**로 실행해서 결과를 확인해 봅시다

☆ **infos.ipynb**의 여섯 번째 셀

```
# 7. change caption  
pygame.display.set_caption("YES의 게임")
```

-> **window** 생성 아래줄에 넣기

# 화면 갱신 속도를 정합시다

- ☆ 실행속도는 시스템 사양에 따라 변한다
  - 사양과 관계없이 일정한 속도를 유지하는 게 이상적

## ☆ infos.ipynb의 일곱 번째 셀

```
# 8. Use clock for fps  
clock = pygame.time.Clock()  
clock.tick(60)
```

- > 첫째 줄은 제목 아래 줄에  
둘째 줄은 display update 아래줄에 각각 붙여넣기
- > 1초에 이 while loop을 최대 60회 돈다

# 서피스(surface)란?

- ☆ - display surface: 메인 윈도우
- regular surface: 이미지, 컬러 채우기, 텍스트  
화면에 표시하기 위해 display surface에 연결돼야 한다

## ☆ infos.ipynb의 여덟 번째 셀

### # 9. test surface

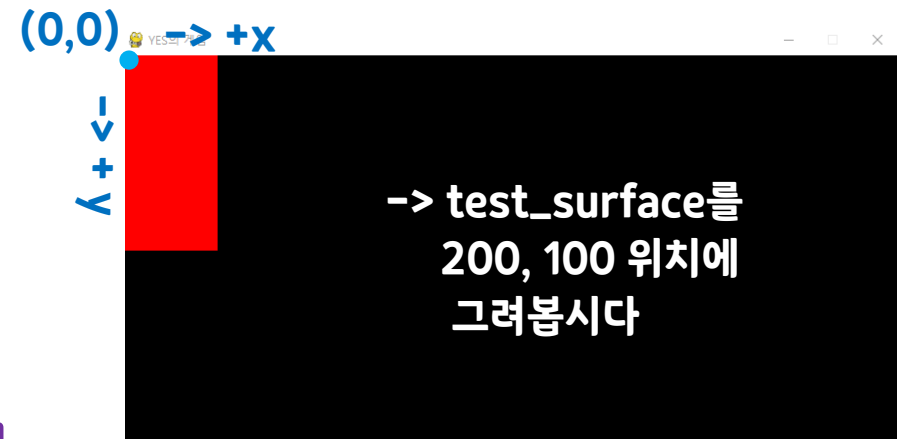
```
test_surface = pygame.Surface((100, 200))  
test_surface.fill('Red')
```

```
win.blit(test_surface, (0,0))
```

### ☆ blit?

2D에서 메모리 블록을 옮겨 넣는 연산을 뜻함

- > 윈도우의 (0,0)에 이 서피스를 블릿(blit)
- > 윗 두줄은 클릭 아래에
- > 맨 아래 줄은 업데이트 전에
- > Ctrl + F5로 실행해서 결과를 확인해 보시다



## 배경 이미지를 그려 봅시다

☆ graphics 폴더에 있는 Sky.png와 ground.png

☆ infos.ipynb의 아홉 번째 셀

```
# 10. load background image
sky_surface = pygame.image.load('graphics/Sky.png')
win.blit(sky_surface, (0,0))
```

-> test\_surface는 삭제하고 그 위치에 각각 넣기

-> Ctrl + F5로 실행해서 결과를 확인해 봅시다

☆ mission!

ground.png를 ground\_surface라는 이름으로 로드해서  
서피스로 만들고 윈도우와 연결해 (0, 300)위치에 그려 보세요!

# 텍스트를 그려 봅시다

1. 폰트를 만든다 (size, style)

2. 텍스트를 서피스에 쓴다

3. 윈도우에 블릿한다

☆ 폰트 만들기

pygame.font.SysFont( font name, size, bold=False, italic=False)  
pygame.font.Font(font filename, size) <- 우리는 이 방법으로

☆ 시스템 폰트 가져오기

print( pygame.font.get\_fonts() )

☆ infos.ipynb의 열 번째 셀

```
# 11. draw text
```

```
my_font = pygame.font.Font('font/Pixeltype.ttf', 50)  
text_surface = my_font.render('My game', False, 'Black')  
win.blit(text_surface, (300,50))
```

☆ 안티 에일리어싱(Anti-aliasing)

그래픽의 계단현상 방지

-> 우리는 픽셀을 도드라지게 표현하므로 False

-> 폰트 생성은 클락 생성 아래, 서피스는 서피스들 생성 아래

-> Ctrl + F5로 실행해서 결과를 확인해 봅시다

# 달팽이를 그려 봅시다

☆ graphics/snail폴더

☆ infos.ipynb의 11,12 번째 셀

# 12. snail

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

# 13. animate snail

```
snail_x_pos = 600
```

-> 서피스 생성 아래에 추가

☆ mission!

snail\_x\_pos 변수로 달팽이를 왼쪽으로 움직여 보세요!  
(tip: pygame 좌표계를 생각해 보기!!)

# 사라진 달팽이 데려오기

☆ infos.ipynb의 13 번째 셀

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

☆ mission!  
프레임레이트도 조정해 보세요!

☆ 보너스: infos.ipynb의 14 번째 셀

```
# 15. change icon  
icon = pygame.image.load('graphics/snail_icon.png')  
pygame.display.set_icon(icon)
```