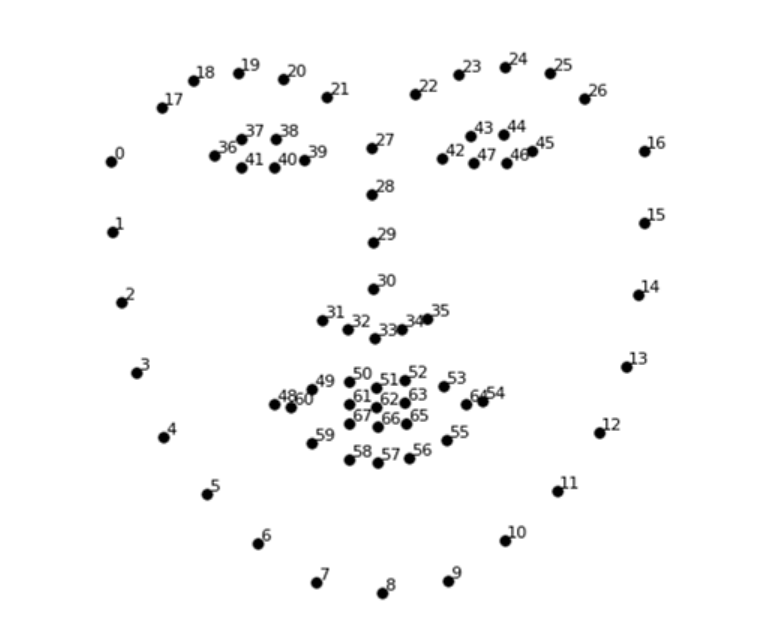
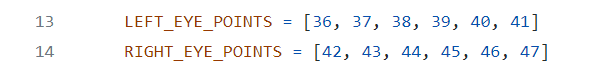
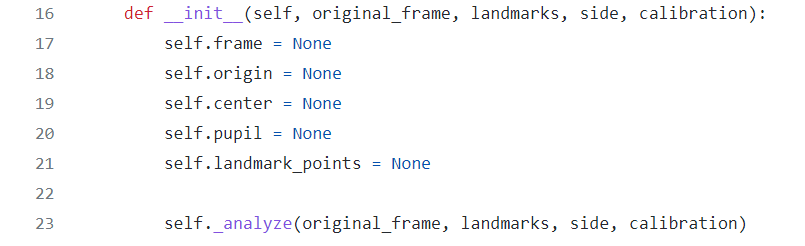
Class Eye : 눈을 분리하고 동공 감지를 시작하는 클래스



dlib을 이용하여 인식한 얼굴



왼쪽 눈을 36부터 41번으로 설정하고, 오른쪽 눈을 42부터 47로 설정해준다.

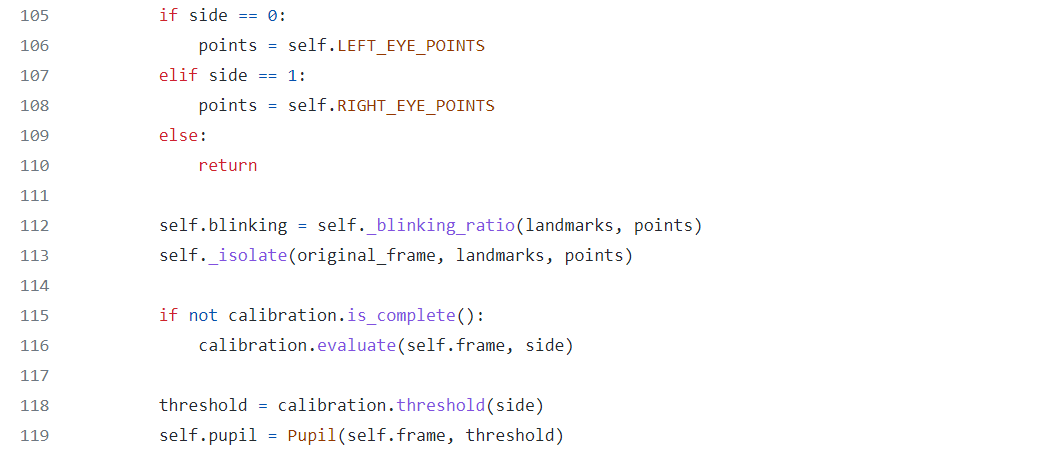


frame. origin, center, pupil, landmark\_points를 none으로 초기설정

analyze함수로 인자 전달

analyze 함수





analyze 함수 : 새 프레임에서 눈을 감지 및 분리하고 데이터를 보정으로 보냄, pupil 객체 초기화

origin\_frame은 사용자가 전달한 프레임,

landmarks는 얼굴 영역의 얼굴 랜드마크,

side는 왼쪽 눈인지 오른쪽 눈인지 나타낸다. 왼쪽 눈일 경우 0, 오른쪽 눈일 경우 1이다.

Calibration은 이진화 임계값을 관리한다.

왼쪽 눈일 경우에는 points를 left\_eye\_points로, 오른쪽 눈일 경우에는 points를 right\_eye\_points로 설정해준다.

Blinking은 눈을 감고있는지 여부를 나타낼 수 있는 비율을 계산하는 blinking\_ration함수로 전달.

isolate함수에 프레임과 랜드마크, 포인트를 전달해서 얼굴의 다른 부분이 없는 프레임을 만들기 위해 눈을 분리한다.

임계값이 최종 상태가아니라면 프레임과 side를 인자로 evaluate함수를 실행한다.

Pupil을 Pupil파일에 프레임과 한계점을 인자로 전달해준다.

Blinking\_ratio 함수





Blinking\_ratio 함수 : 눈을 감고 있는지 여부를 나타낼 수 있는 비율을 계산하며 눈의 너비를 높이로 나눈 것이다.

Points는 68개의 다중 PIE랜드마크 중 눈의 포인트를 나타낸다

left, right, top, bottom을 각각 points에서 알맞은 값을 가져와 설정해준다.

Points[0]은 눈의 맨왼쪽, points[3]은 눈의 맨오른쪽 으로 맨처음 그림의 눈 부분에서부터 리스트로 되어있다.

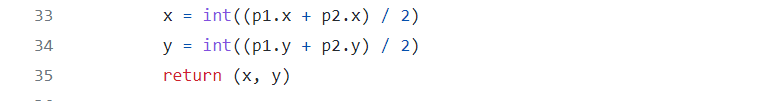
눈의 width는 hypot함수를 이용해서 빗변을 구한다

Height도 마찬가지로 hypot함수를 이용한다.

Ratio는 width/height로 설정하고 최종적으로 계산된 비율을 반환한다.

middle\_point 함수





middle\_point 함수 : 두 포인트 사이의 중간 점을 (x, y)로 리턴하는 함수

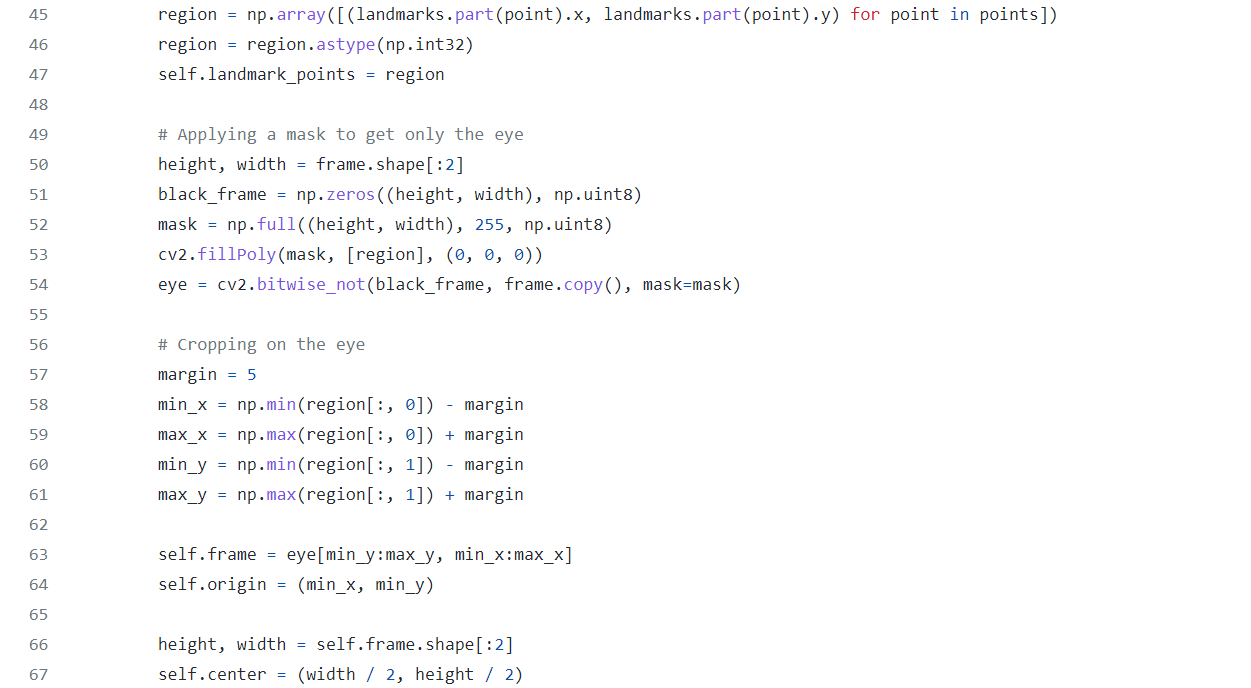
p1은 첫번째 포인트, p2는 두번째 포인트

x는 첫번째 포인트의 x와 두번째 포인트 x의 중간값, y도 마찬가지로 설정해주고,

x와 y를 반환한다.

isolate 함수





isolate 함수 : 얼굴의 다른 부분이 없는 프레임을 얻기위해 눈을 격리한다.

frame 은 얼굴을 포함하는 프레임이고, points는 눈의 포인트다.

Region을 array로 넣어주고 랜드마크 포인트로 설정해준다.

눈만 얻기위해 mask를 적용해준다. (numpy 이용)

눈에서 원하지않은 것들을 제거하기 위해 margin을 5로 넣고, min, max값을 설정한다.

프레임은 eye에서 아까 설정한 min, max값을 이용한다.

height, width도 받아와서 반으로 나누고 center로 설정해준다.