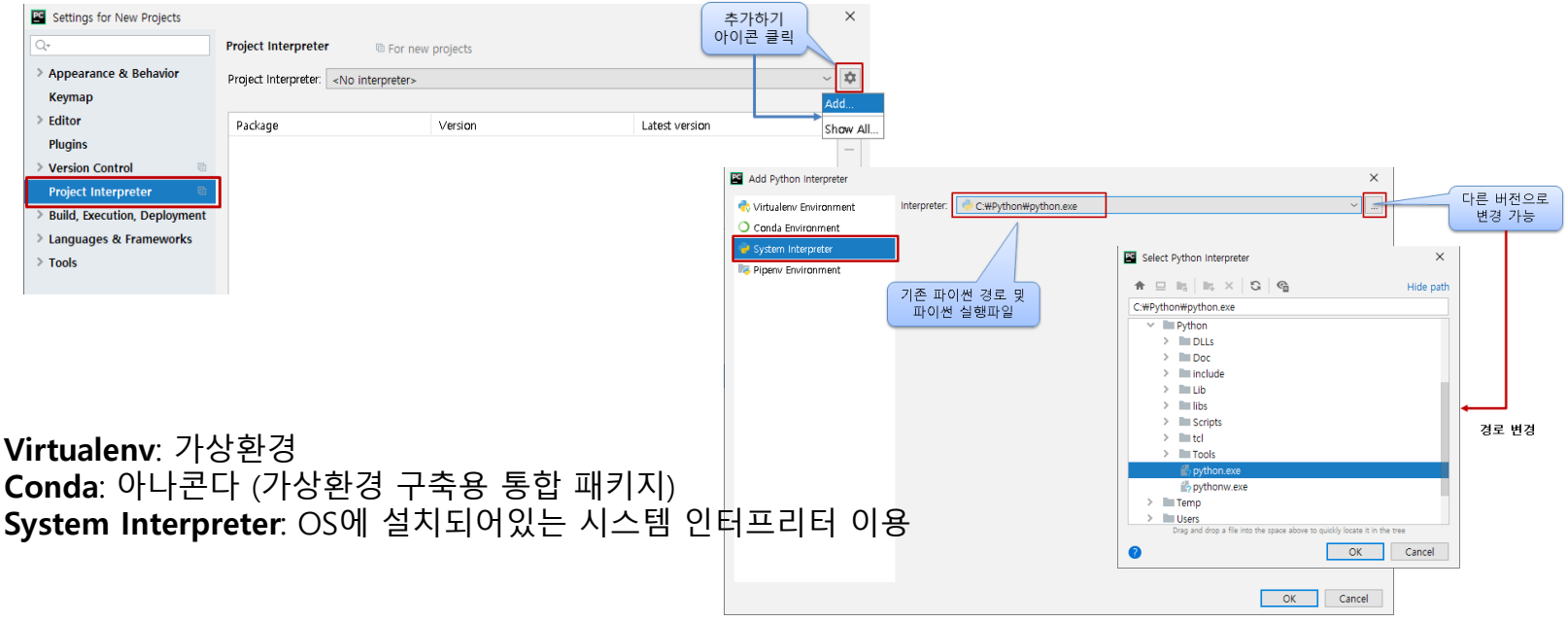
실습 07. 프로젝트 디버깅

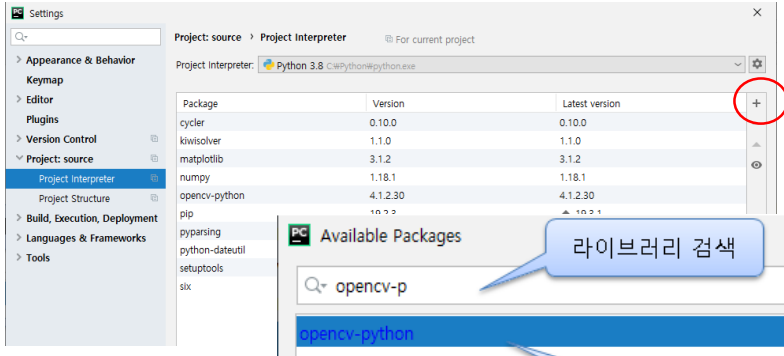
오류 1. Python interpreter 설정

- 파이썬은 오픈소스 프로젝트로 많이 진행되며 라이브러리들이 매우 빈번하게 업데이트 됨.
- 어떤 업데이트에서는 구조가 크게 바뀌거나 인터페이스가 변경되는 경우도 있음.
- 잘 동작하던 코드가 특정 버전에서 갑자기 오류가 발생할 수 있다.
- 특정 버전을 사용하는 프로젝트를 위해, **가상환경**을 구축하여 진행할 수 있음.

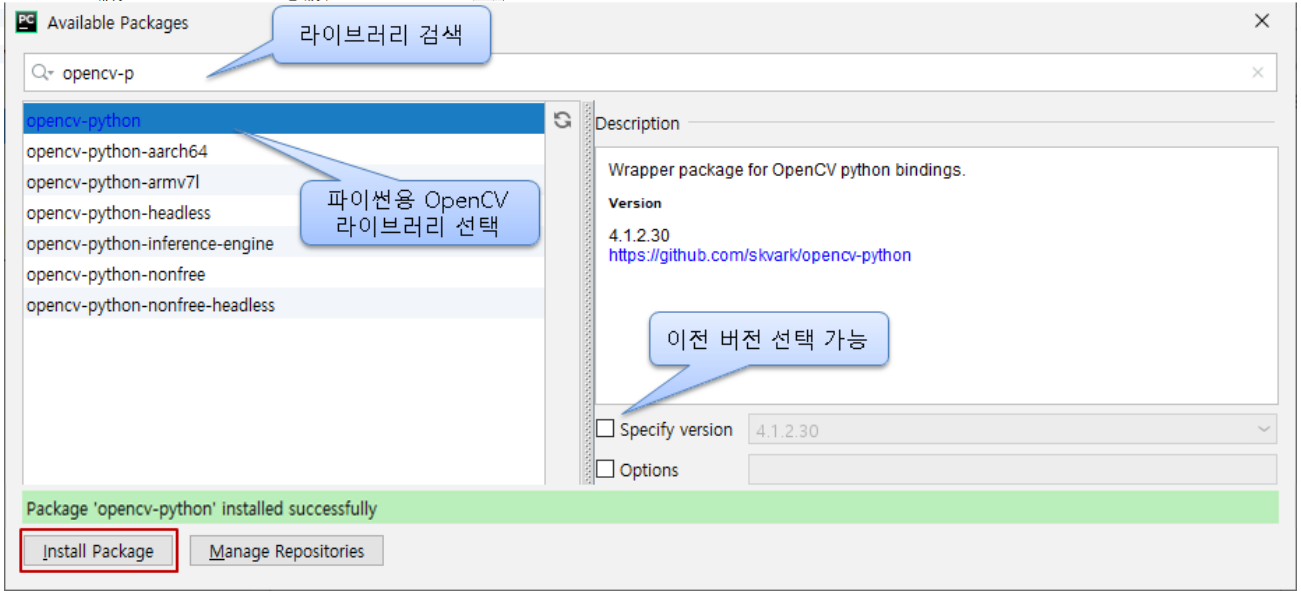


Virtualenv: 가상환경
Conda: 아나콘다 (가상환경 구축용 통합 패키지)
System Interpreter: OS에 설치되어있는 시스템 인터프리터 이용

오류 2. Python 패키지 설치 확인

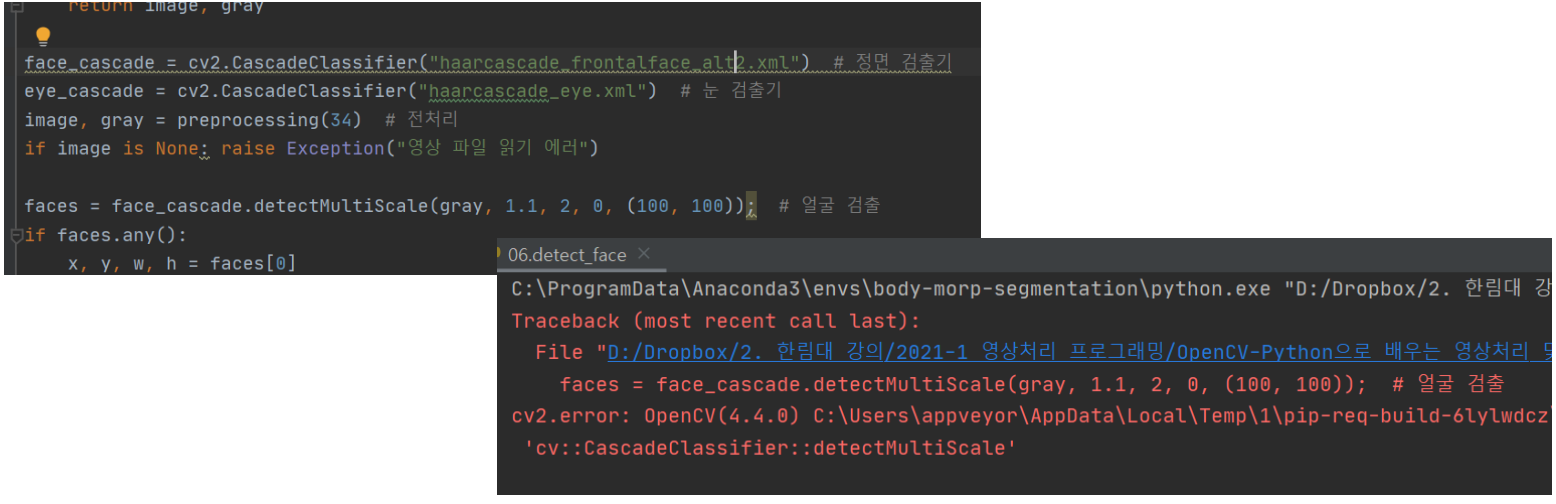


예: opencv가 설치 되어 있지 않다면?



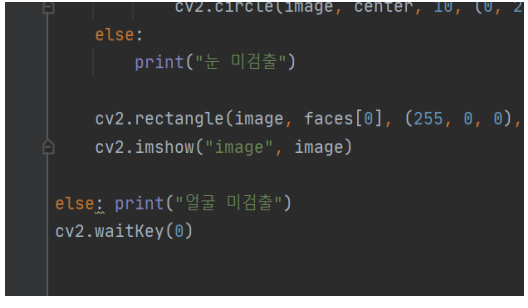
오류 3. 추가 파일 설명을 따르지 않음

- 지정한 파일을 다운로드 안한경우
- 지정한 파일 대신 다른 이름의 파일을 설치한 경우



오류 4. 소스코드 잘못 기입, 오타.

- imshow() 함수 입력 빠뜨림 → 결과 출력X
- waitKey() 함수 이름을 waitkey()로 입력 → 대소문자 오타
- face_cascade 대신 face_casacde로 입력
- 들여쓰기 잘못함 → if/else 구문의 로직이 다르게 동작: 논리적 오류



따라치기 바른 예

따라치기 나쁜 예1

따라치기 나쁜 예2

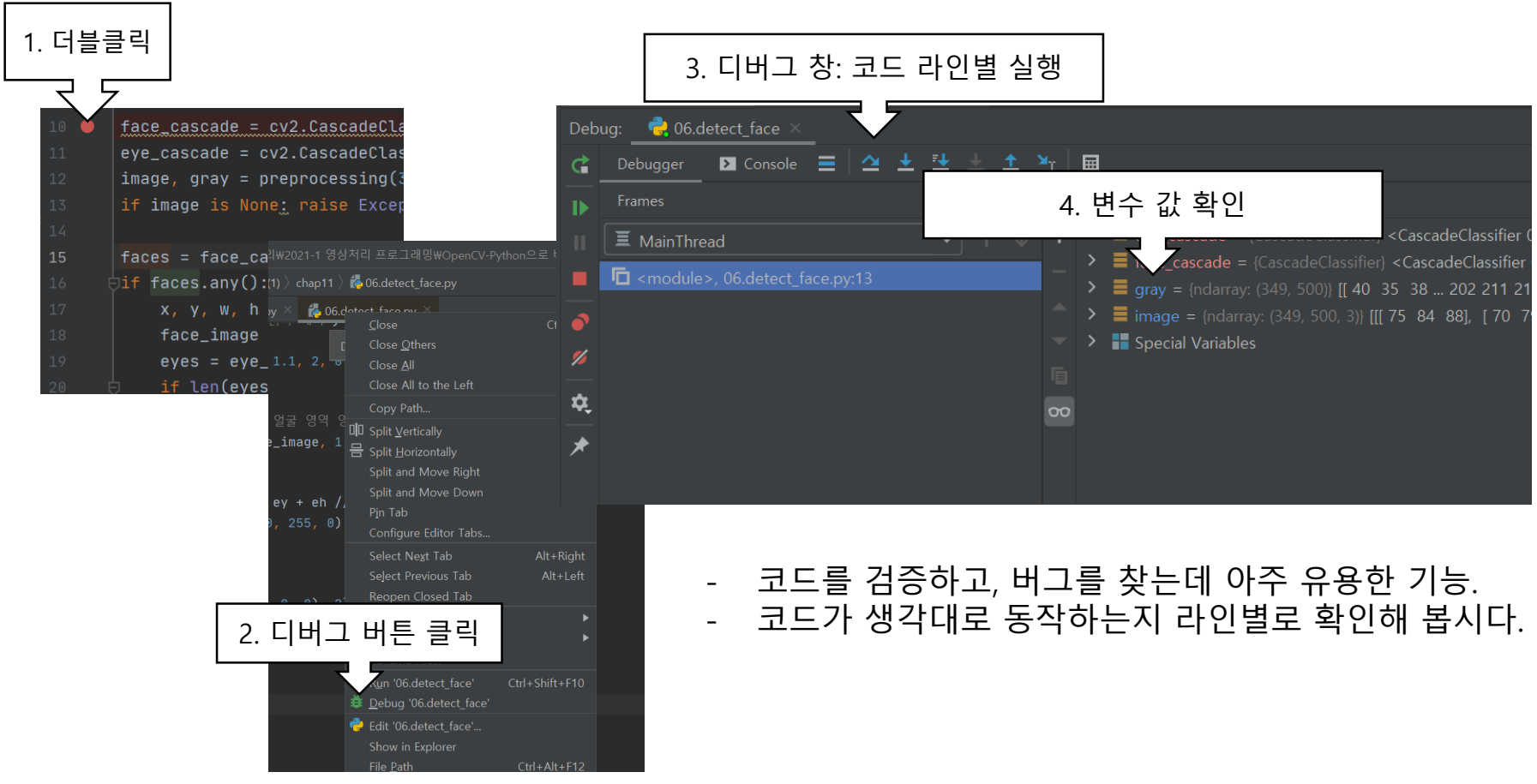
에러 메시지를 읽어보자

- 모든 사람은 오류를 낸다. (코딩의 신 조차)
- 대부분의 오류는 오타에서 발생 → **꼼꼼하게!!**
- 많은 경우 에러메시지를 읽음으로써 파악 가능
  - 에러메시지 해석
  - 구글 검색

```
    import no_pakage
ModuleNotFoundError: No module named 'no_pakage'

Process finished with exit code 1
```

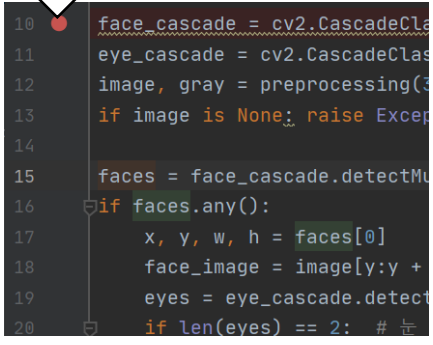
디버깅을 하자 (Debugging: 버그찾기)



- 코드를 검증하고, 버그를 찾는데 아주 유용한 기능.
- 코드가 생각대로 동작하는지 라인별로 확인해 봅시다.

디버깅을 하자 (Debugging: 버그찾기)

코드 옆 더블클릭
(브레이크 포인트)

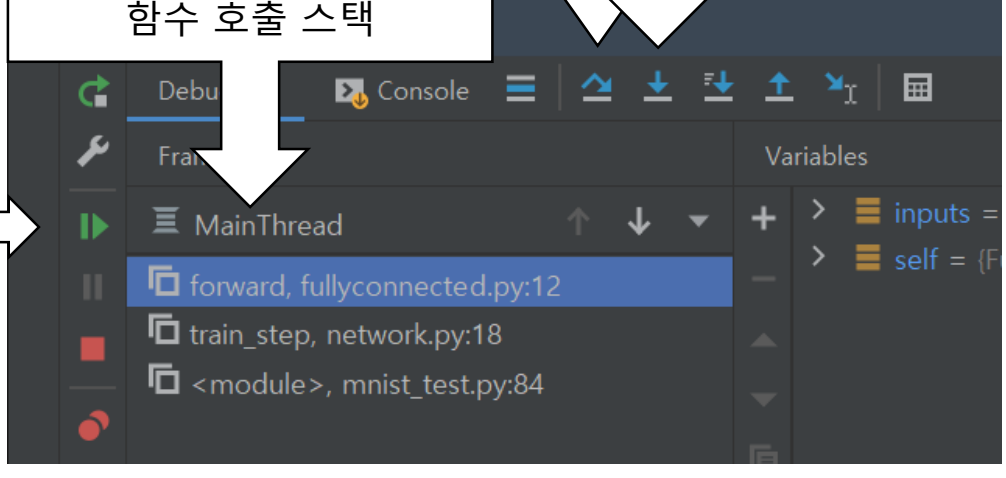


다음 줄 실행

함수 내부 실행

함수 호출 스택

계속 진행 버튼
(브레이크 포인트까지)



디버깅을 하자 (Debugging: 버그찾기)

코드 분석 및 디버깅 연습 실습

- 프로젝트 설치 후 코드를 실행하시오. (학습에 2~3분 시간 소요)
  - mnist_test.py 실행
  - Line 31의 layers 리스트의 역할은 무엇인가?

- Fully connected layer를 한줄 추가해 보시오.

- 첫번째 layer의 입력과 출력 (z)의 shape를 살펴보고 그 의미를 서술하시오.
  1. mnist_test.py의 31번째 줄의 layer 리스트 살펴보기: 첫번째와 두번째 layer의 역할은?
  2. Nn/network.py의 18번째줄에 break point (빨간점)을 찍고, 디버깅을 실행
  3. Step over(단축키 F8)를 눌러보고 입력과 출력 z에 대해 살펴보기
  4. Step into(단축키 F7)를 눌러보고 함수 호출 스택 살펴보기