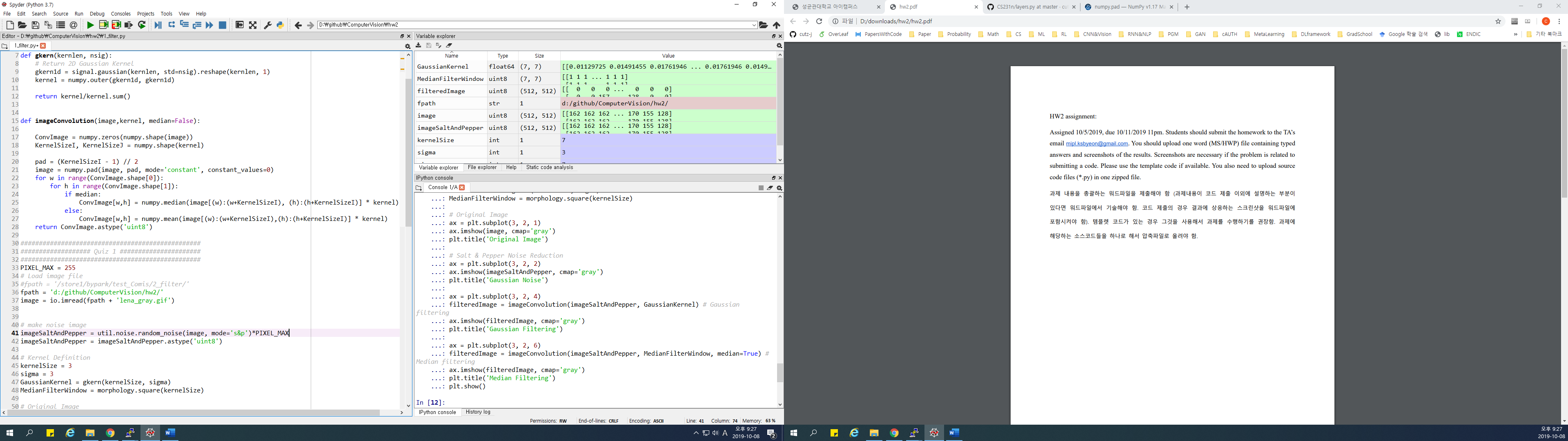
Quiz 1.

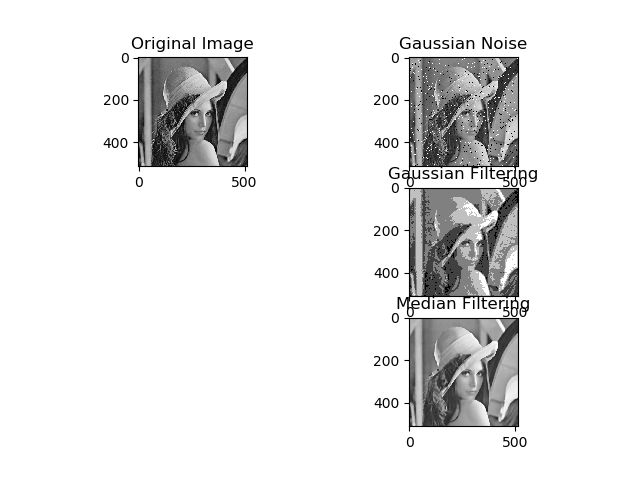
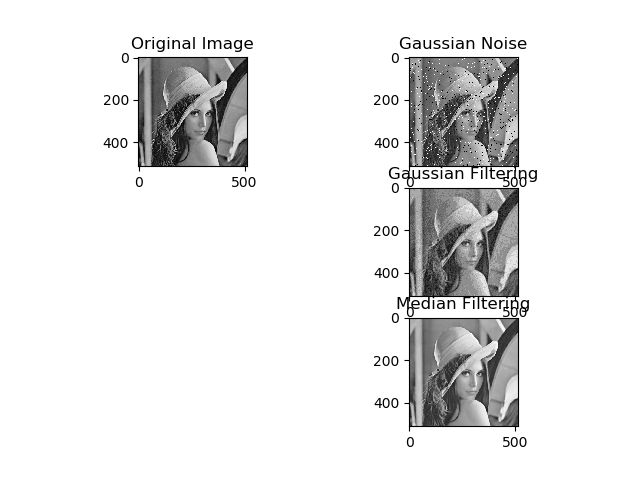


Convolution 연산을 numpy로 직접 구현, im2col의 방법으로 한번에 연산을 할 수 있지만, 직관적으로 Width, Height로 2개의 반복문을 실행하고, W, H의 index마다 kernel과 element-by-product를 실시 후 mean 🡪 numpy 연산을 이용하여 간단하게 구현하였습니다.

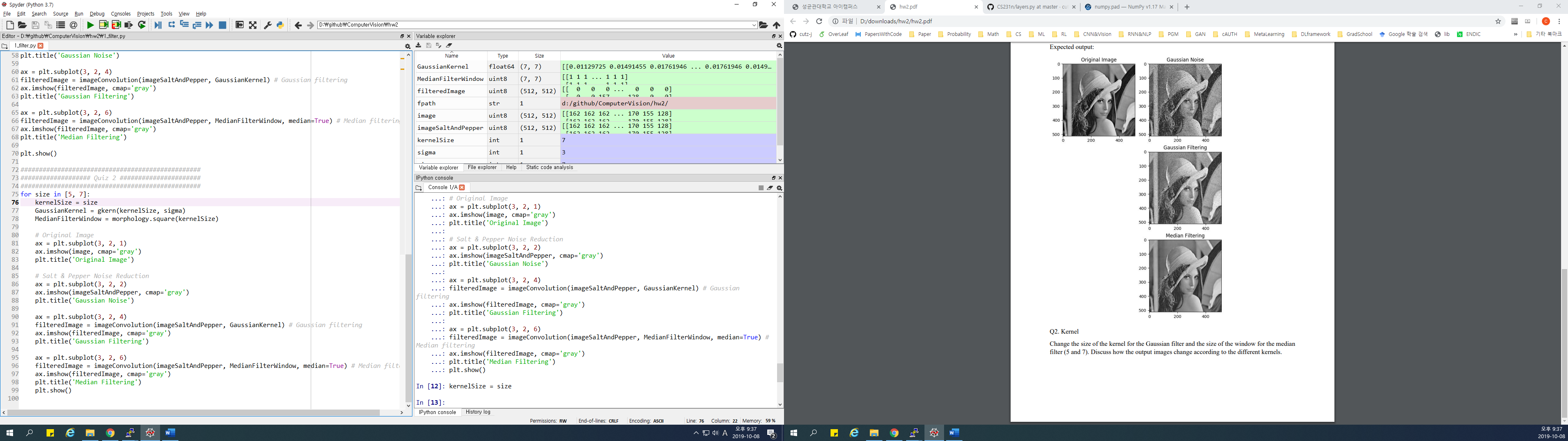
Kernel\_size에 따라서, image resolution이 감소하기 때문에, 미리 원본 이미지를 Zero-padding하였습니다.

< Quiz 2는 다음 장에 >

Quiz 2



왼쪽이 Kernel 5, 오른쪽이 Kernel 7 입니다. Kernel이 커질수록 parameter 연산이 증가하기 때문에, 연산이 많이 느려졌습니다. 결과물은 kernel이 커질수록 큰 범위의 image patch를 가져오기 때문에, kernel 효과가 증가하였으며, 7의 경우는 과도하게 블러링, 중간화가 되어 이미지의 원본을 잃었습니다. 그리고 kernel size를 키울수록, Resolution 유지를 위한 Paddding이 더 필요하기 때문에 이 점도 원본 이미지보다 과도하게 변하게 되는 효과가 크다고 생각합니다.



코드 구현의 경우, 기존 코드에 kernelSize만 변경하여 구현하였습니다.