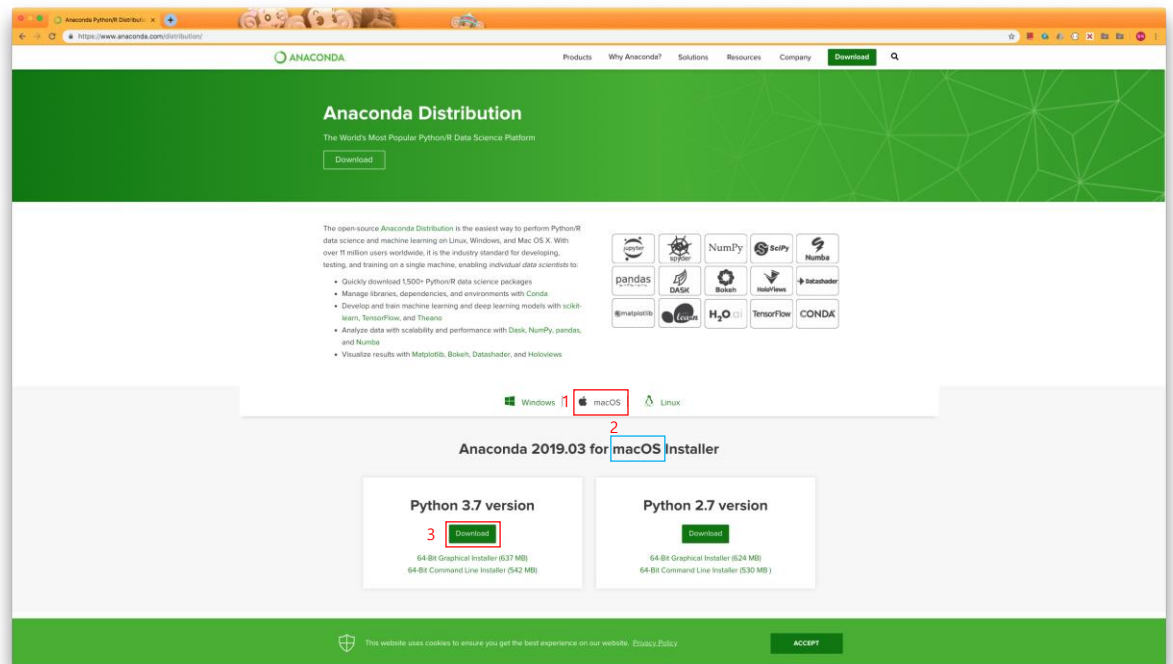
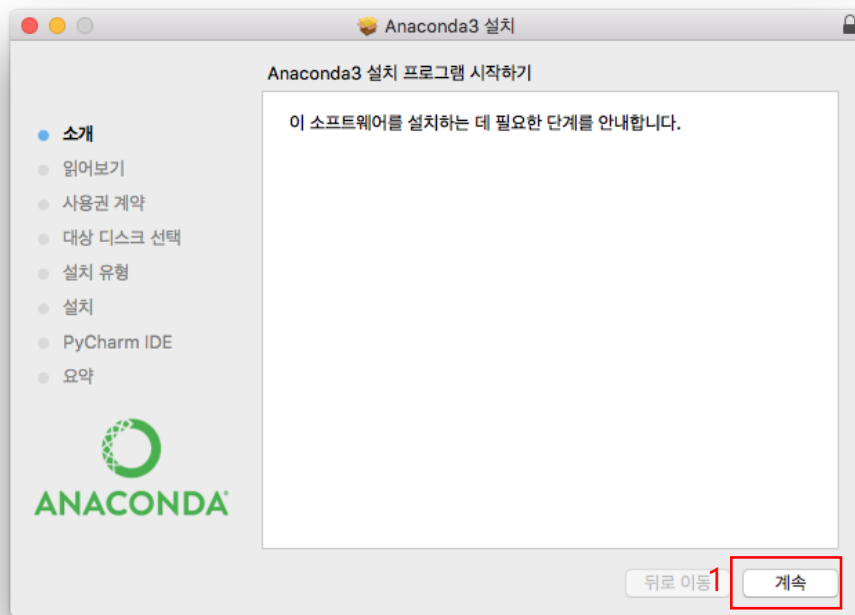


[1] 아나콘다 설치

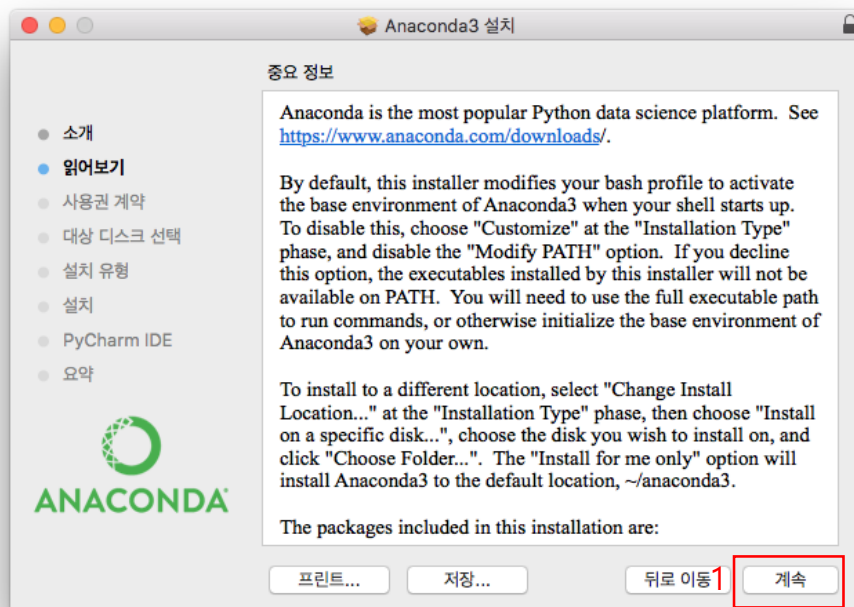
- 1-1. 아나콘다 설치 프로그램을 다운로드받기 위해 <https://www.anaconda.com/download/> 사이트로 이동합니다. 아나콘다는 실습에 필요한 주피터 노트북과 파이썬 인터프리터 등이 모두 포함된 통합 패키지 소프트웨어입니다. 먼저 **2**에 표시된 설치 대상 운영체제를 확인합니다. 이 부분에 'MacOS'가 아닌 내용이 보인다면 **1**을 클릭해 'MacOS'가 나타나게 합니다. 그런 후 **3**을 클릭해 설치 프로그램의 다운로드를 시작합니다.



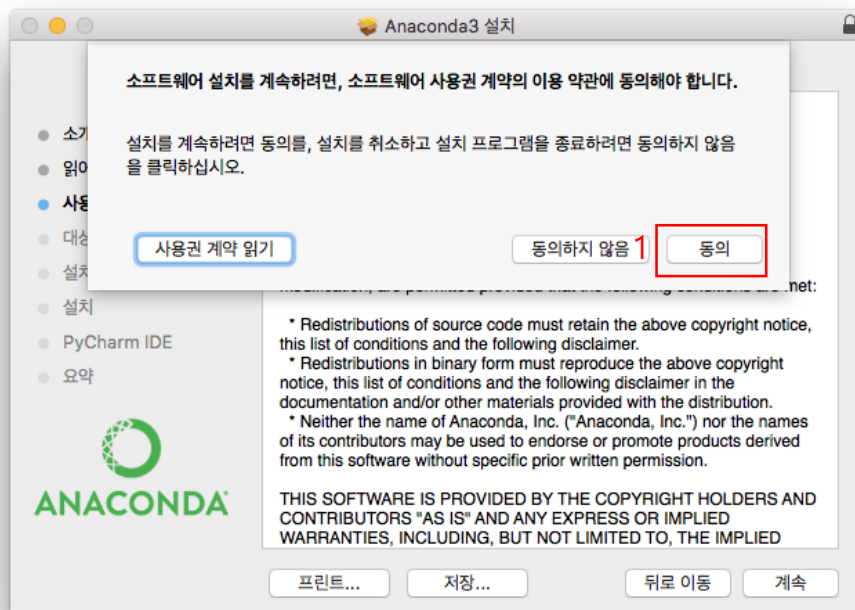
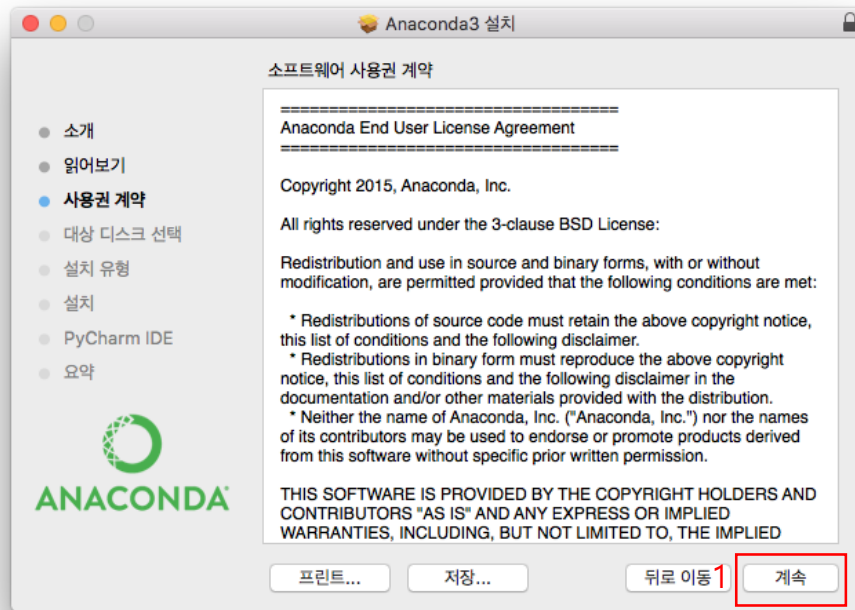
- 1-2. 다운로드 받은 파일을 실행시키면 아래와 같은 설치 프로그램이 시작됩니다. **1**을 클릭해 설치를 시작합니다.



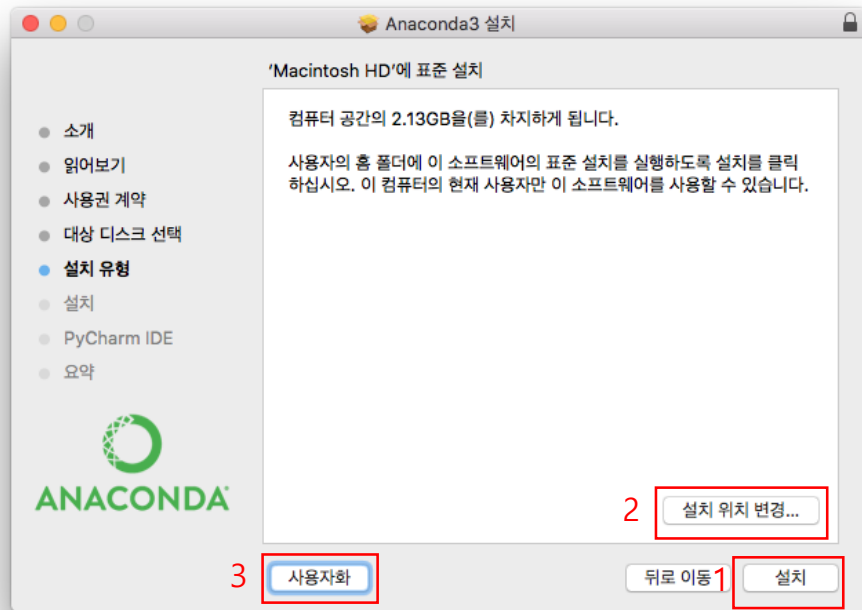
1-3. 중요 정보를 고지하는 화면이 나타나면 1을 클릭해 계속 진행합니다.



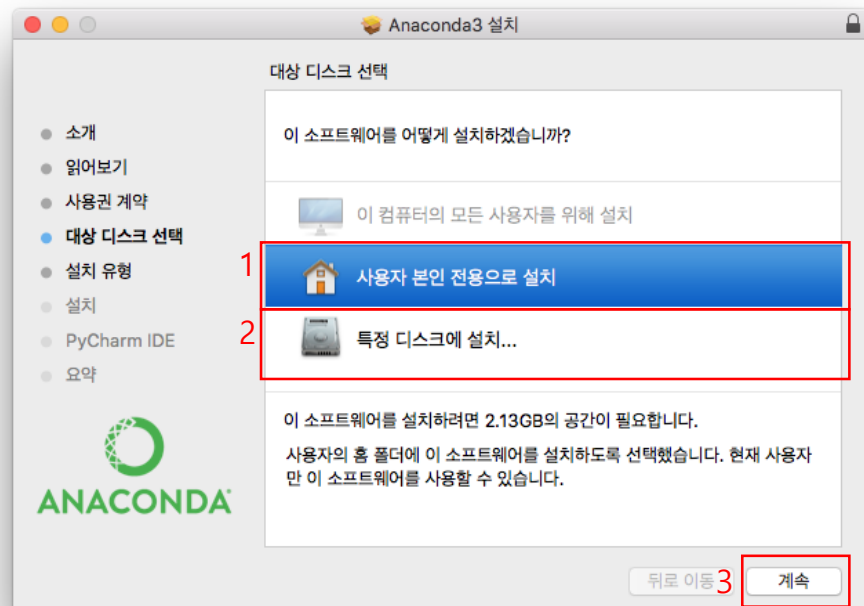
1-4. 라이선스 동의 화면이 나타나면 1을 클릭해 동의합니다.



- 1-5. 설치 유형 설정 화면이 나타나면 1을 클릭해 '현재 계정 사용자'를 위한 설치와 기본 디스크로 설치를 진행합니다. 단 기본 디스크 외의 다른 저장 공간을 지정하거나 컴퓨터에 설정된 여러 계정 사용자와 함께 아나콘다를 이용하고 싶고 관리자 권한이 있는 경우에는 2를 클릭해 1-5-1과 같이 진행하고, 고급 설치를 원한다면 3을 클릭하여 1-5-2의 과정으로 진행합니다.

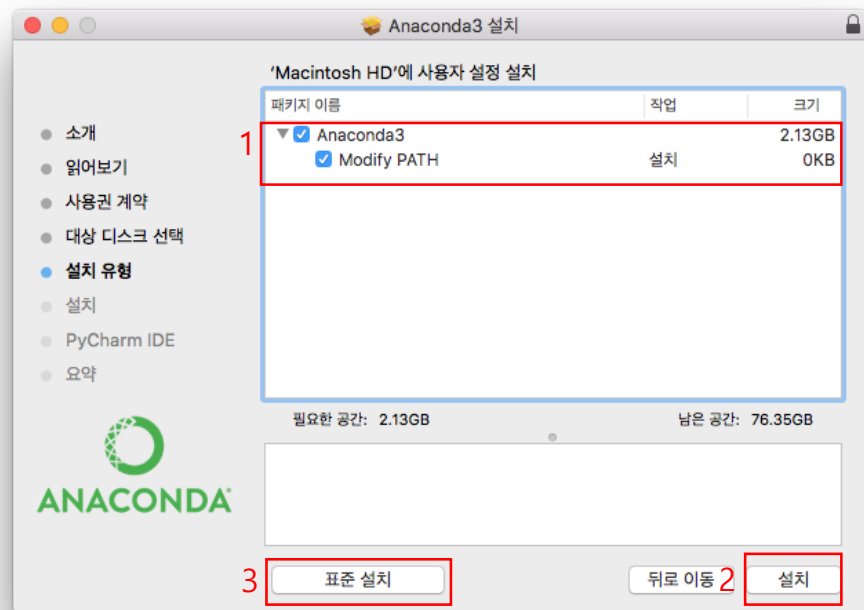


1-5.1. 기본 설정 값인 사용자 본인 전용으로 설치를 하려면 1을 선택 후 3을 선택하여 계속 진행하고, 디스크 위치나 경로를 변경하고자 하는 경우에는 2를 선택하여 설정 후 3을 선택하여 계속 진행합니다.

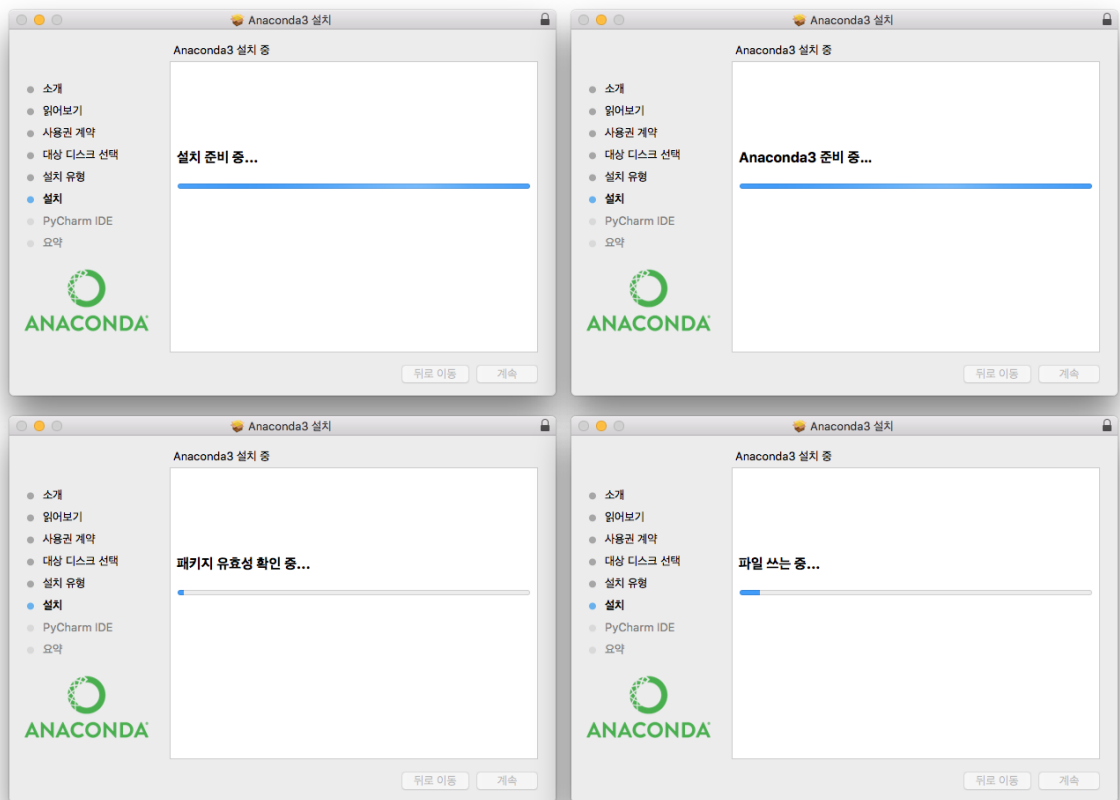


1-5.2. 사용자 설정 설치에서는 설치되는 패키지 및 환경 설정을 변경할 수 있

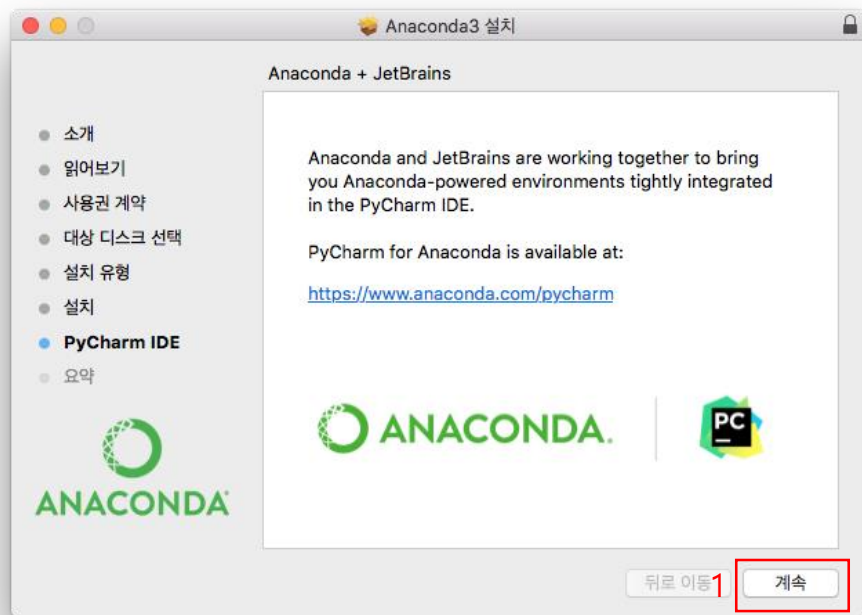
습니다. 1에서 Anaconda3을 체크 해제 하거나 Modify PATH를 체크 해제 하는 경우 Anaconda3이 설치되지 않거나 정상적으로 작동하지 않을 수 있습니다. 따라서 변경하는 것은 권장하지 않습니다. 원하는 값으로 변경 후 2를 클릭해 설치를 계속 진행 하거나 3을 클릭하여 기본 설정인 표준 설치를 진행하도록 합니다.



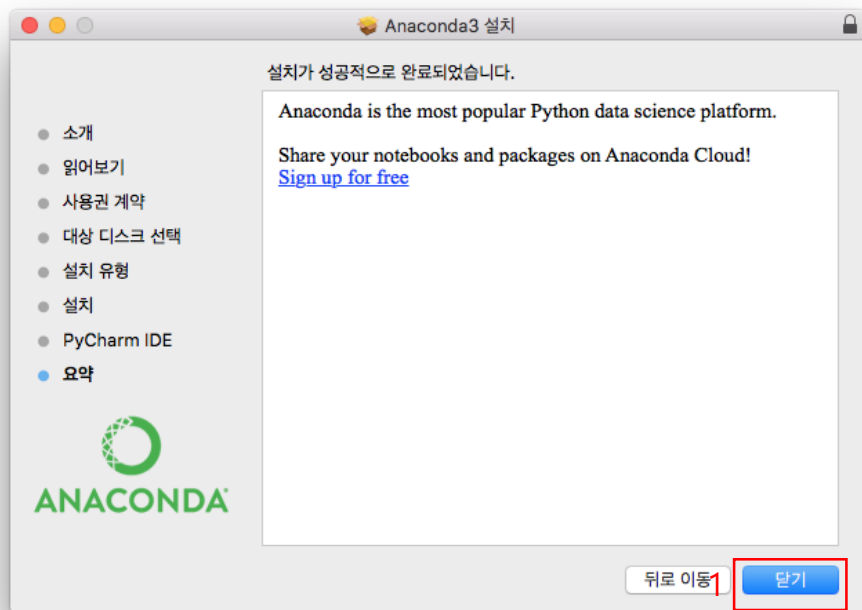
1-6. 설치가 시작되면 시간이 생각보다 오래 걸리므로 인내를 가지고 기다립니다.



1-7. 자매품 광고 화면입니다. 1을 클릭해 다음 단계로 이동합니다.

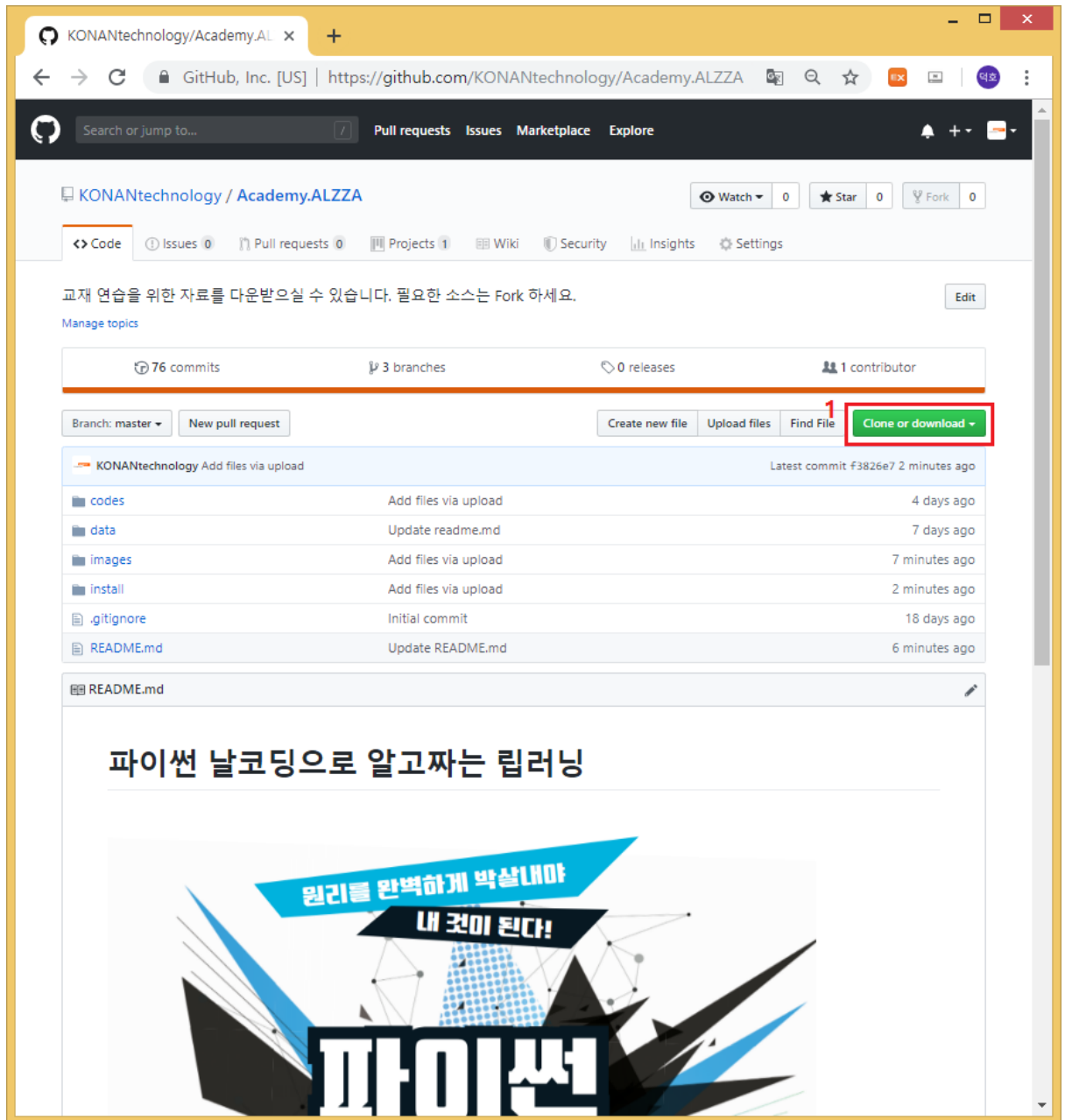


1-8. 아나콘다 설치가 완료되었습니다. 1을 클릭해 작업을 마칩니다.

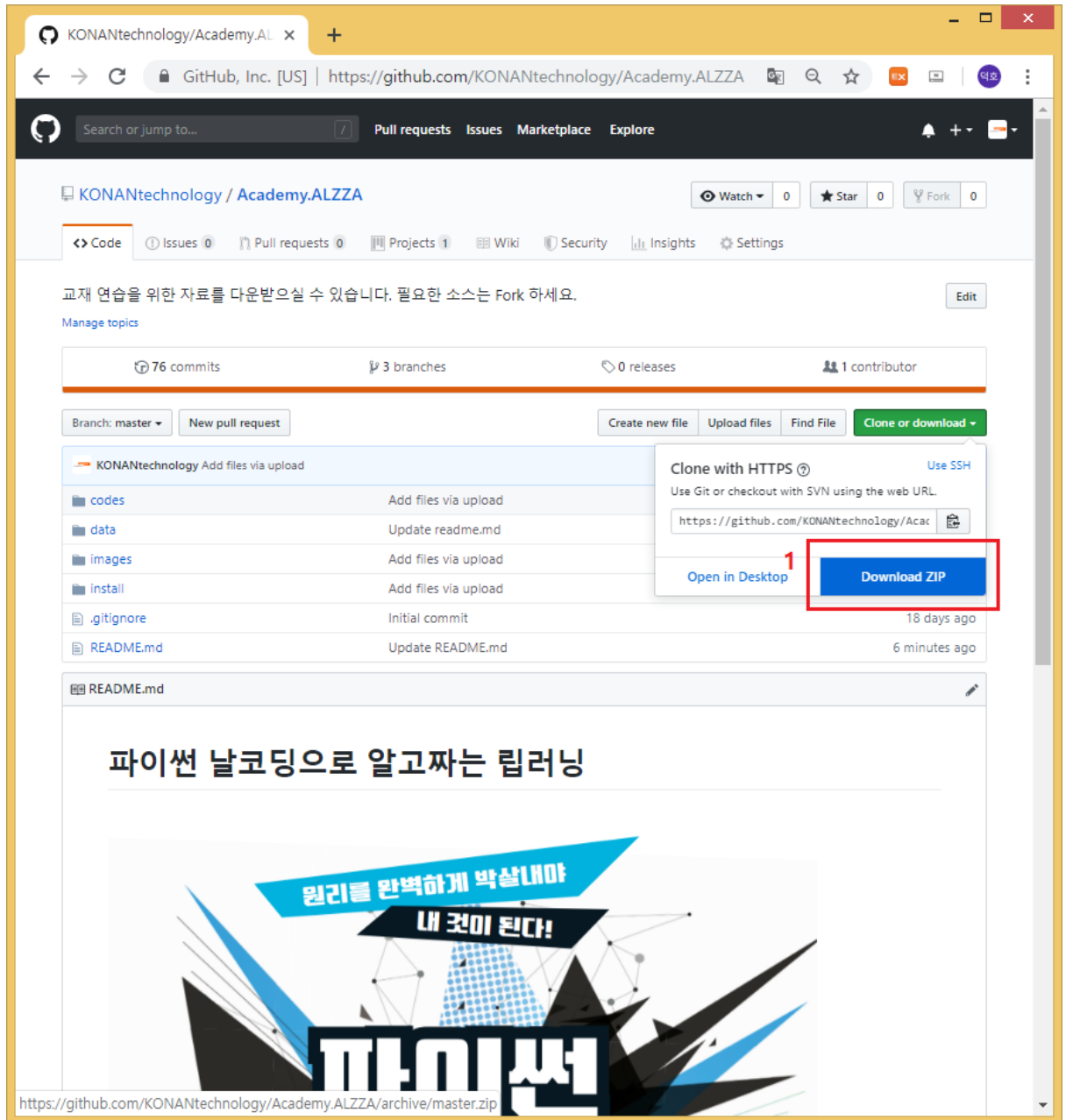


[2] 예제 프로그램 다운로드 받기

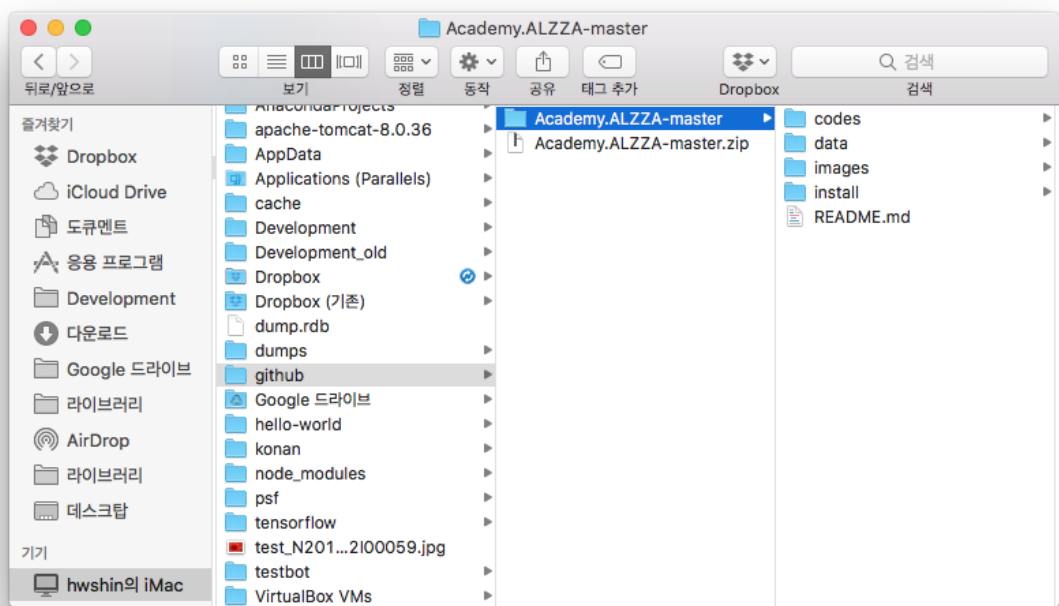
- 2-1. 예제 프로그램들을 다운로드 받기 위해 '파이썬 날코딩으로 알고짜는 딥러닝' 책의 깃허브 저장소인 <https://github.com/konantechnology/academy.alzza> 사이트로 이동합니다. 1을 클릭해 예제 프로그램 다운로드를 시작합니다.



- 2-2. 예제 프로그램 사본을 다운받아 내 컴퓨터에서 실험을 하고 싶다면 1을 클릭해 압축 파일 형태로 다운로드를 받습니다.

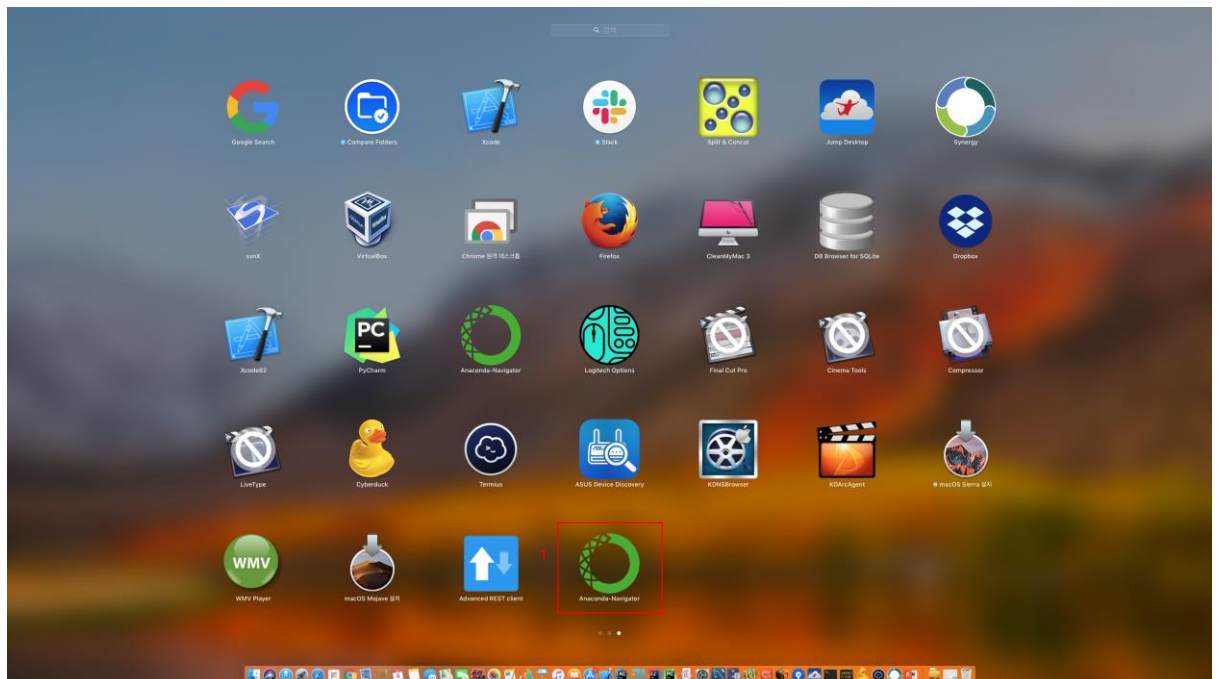


- 2-3. 다운로드가 끝나면 다운로드 받은 Academy.ALZZA-master.zip 파일의 압축을 풀어줍니다. 이때 만들어질 폴더의 위치가 /Users/(사용자명) 아래 있어야 아나콘다에서의 접근이 가능합니다. 여기에서는 1에서처럼 /Users/(사용자명)/github로 압축 풀기 위치를 지정해 이 폴더 아래 다시 Academy.ALZZA-master 폴더가 생겼고 그 아래 다운로드 받은 파일들이 설치되었다고 가정하겠습니다.

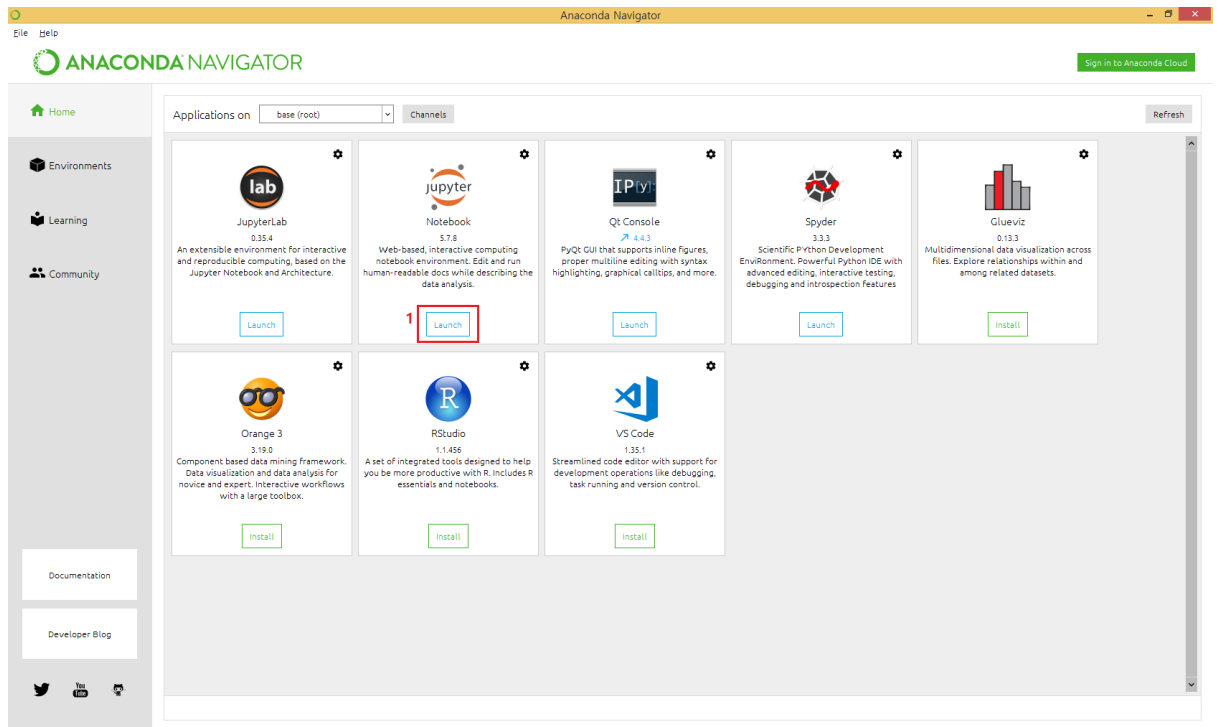


[3] 주피터 노트북 가동 및 실습 개시

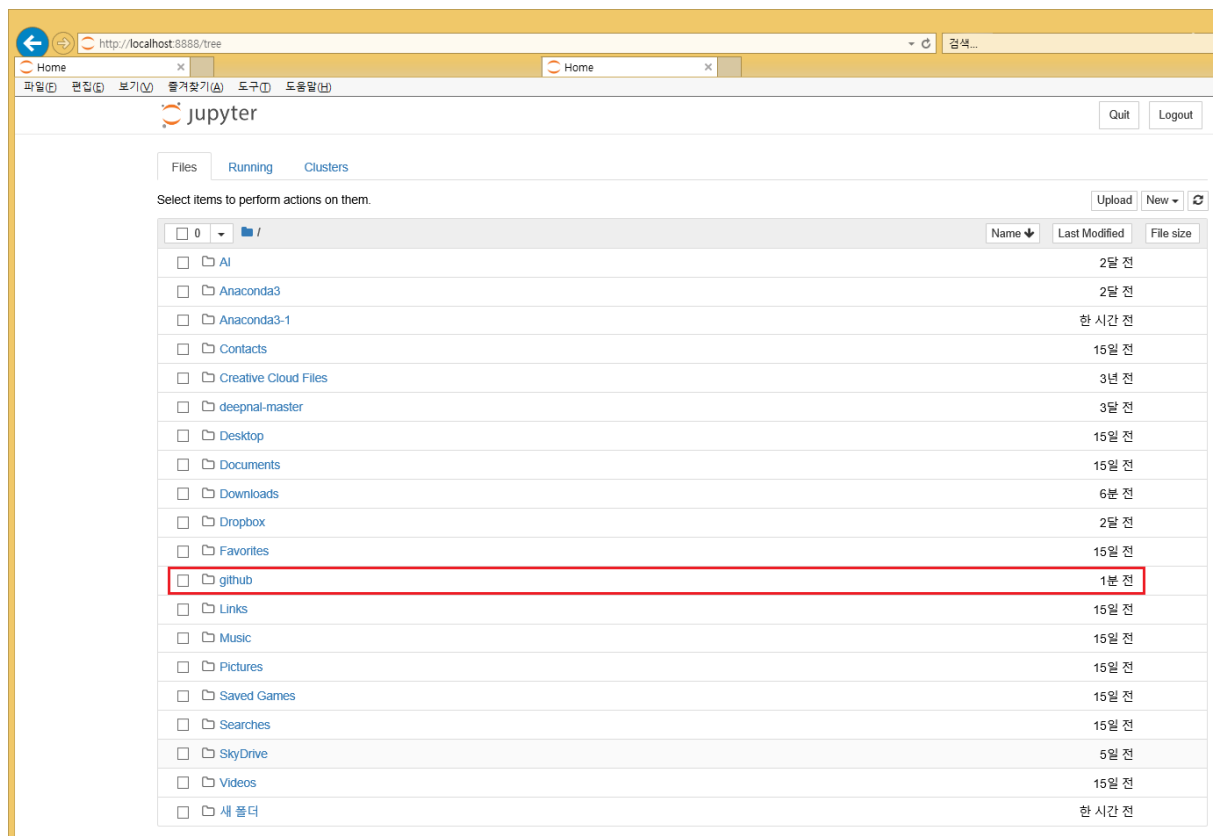
3-1. [LaunchPad]에서 1을 클릭해 아나콘다 네비게이터 프로그램을 가동시킵니다.



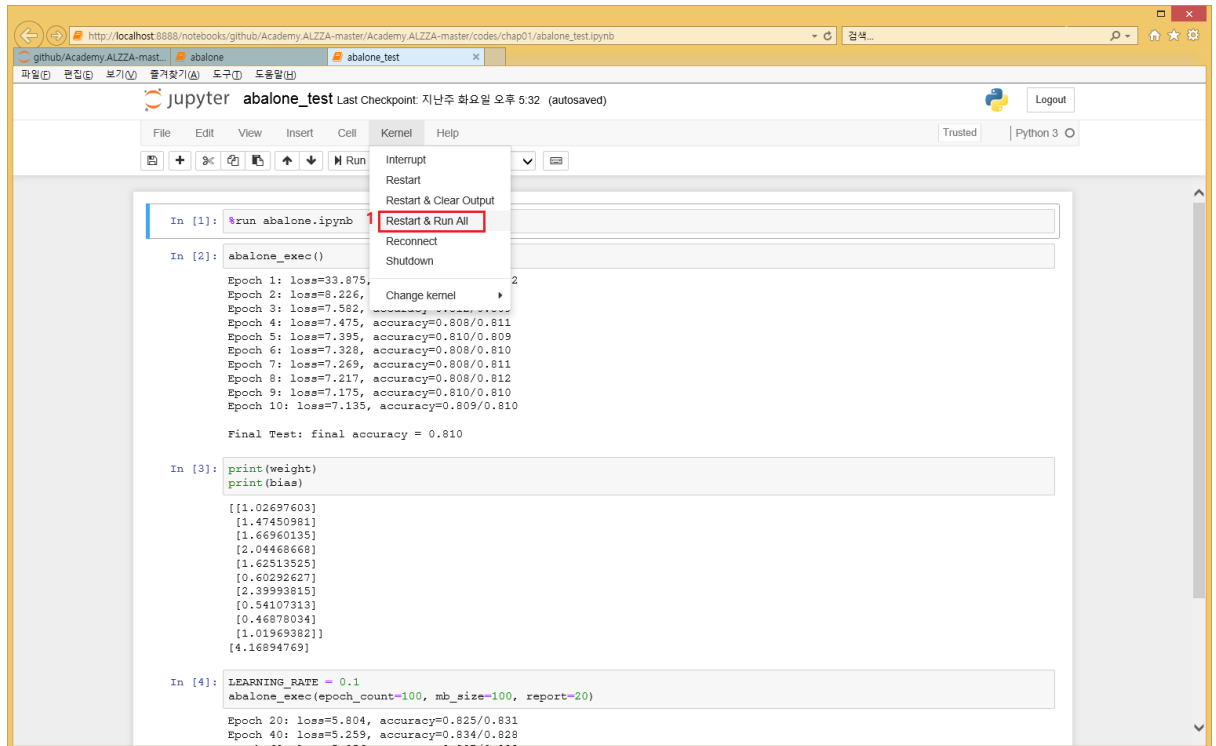
3-2. 아나콘다 네비게이터 화면에서 1을 클릭해 주피터 노트북을 가동시킵니다.



- 3-3. 시스템의 기본 브라우저로 지정된 브라우저 프로그램이 가동되면서 주피터 노트북 화면이 나타납니다. 주피터 노트북 시작 상태는 /Users/(사용자명) 폴더의 서브 폴더를 보여주면서 시작됩니다. 방금 설치한 github 항목을 찾아 1을 클릭해 폴더 안으로 이동합니다.



- 3-4. 작업 위치를 github 폴더 아래 설치된 codes/chap01 폴더로 이동한 후 abalone.ipynb 파일 abalone_test.ipynb 파일을 클릭해 내용을 살펴보고 프로그램 실행도 시켜봅니다. 특히 abalone_test.ipynb 파일에서 1을 클릭해 '전체 프로그램 재실행' 명령을 가동시키면 교재에서와 같은 실험 결과를 얻을 수 있습니다.



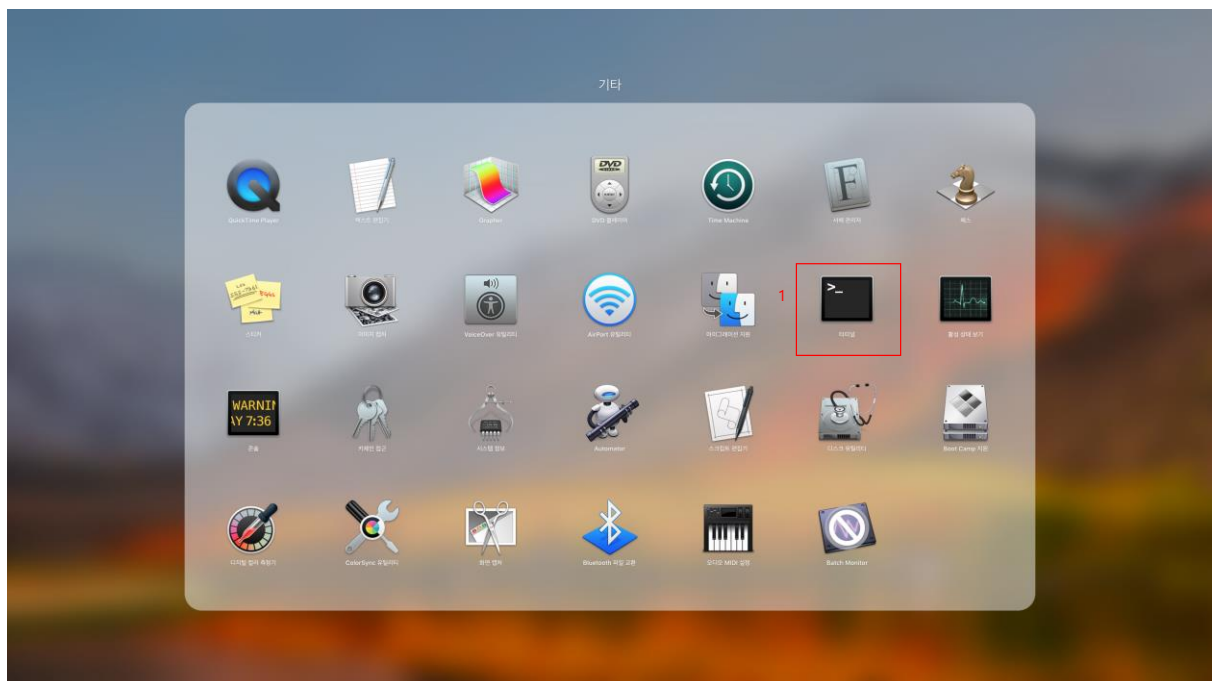
- 3-5. 같은 방법으로 4장까지의 실습은 어려움 없이 진행할 수 있을 것입니다. 하지만 5장 실습을 위하여 codes/chap05/mlp_model_test.ipynb 파일을 실행시키면 아래와 같은 오류 메시지가 나타나게 됩니다.

The screenshot shows a Jupyter Notebook running on a local host. The code cell contains the following commands:

```
In [1]: %run ../chap05/mlp_model.ipynb
        %run ../chap05/dataset_chap123.ipynb
        %run ../chap05/dataset_flowers.ipynb
```

The output shows three consecutive `ModuleNotFoundError` messages, each indicating that the module `'cv2'` is not found. The traceback for each error points to the `import cv2` line in the notebook code.

3-6. 이런 오류가 발생하는 이유는 5장부터 이미지 처리를 위하여 이용되는 파이썬의 cv2 패키지가 아나콘다의 기본 구성요소로 포함되어 있지 않기 때문입니다. 이 패키지의 추가 설치를 위하여 [LaunchPad]에서 mac OS의 기본 설치 프로그램인 터미널을 실행합니다.



3-7. 터미널 프로그램이 가동되면 1에서 'conda install opencv' 명령을 내려 추가 설치를 지시합니다. 프로그램이 설치 내용을 조사해 보고한 후 설치 실행 여부를 물어오면 2에서 'y'

를 입력해 설치를 지시합니다. 설치 작업이 진행되어 요청된 패키지 설치가 끝났다는 메시지를 확인한 후 터미널 프로그램을 종료시킵니다. 만약 1.5.2에서 환경 변수를 설치 안했다거나, 1.5.1에서 경로를 변경 시 conda 명령어가 작동하지 않을 수 있습니다. 이런 경우에는 처음부터 권장 사양으로 재설치 하도록 합니다.

```
hwshin -- -bash -- 112x78
skeleton
verify
1
((base) hwshinui-iMac:~ hwshin$ conda install opencv
WARNING: The conda.compat module is deprecated and will be removed in a future release.
Collecting package metadata: done
Solving environment: done

## Package Plan ##

  environment location: /Users/hwshin/anaconda3

  added / updated specs:
    - opencv

The following packages will be downloaded:

package | build | size
-----|-----|-----
ca-certificates-2019.5.15 | 0 | 133 KB
cairo-1.14.12 | hc4e6be7_4 | 1.2 MB
certifi-2019.6.16 | py37_0 | 154 KB
conda-4.7.5 | py37_0 | 3.0 MB
conda-package-handling-1.3.10 | py37_0 | 259 KB
ffmpeg-4.0 | h01ea3c9_0 | 41.2 MB
fontconfig-2.13.0 | h5d5b041_1 | 245 KB
graphite2-1.3.13 | h2098e52_0 | 83 KB
h5py-2.8.0 | py37h878f3e3_3 | 880 KB
harfbuzz-1.8.8 | hb8d4a28_0 | 703 KB
hdf5-1.10.2 | hf1e0ec_1 | 4.5 MB
jasper-2.0.14 | h636a363_1 | 1.1 MB
libopencv-3.4.2 | h7c891bd_1 | 27.5 MB
libopus-1.3 | h1de35cc_0 | 571 KB
libvpx-1.7.0 | h378b8a2_0 | 2.3 MB
opencv-3.4.2 | py37h6fd60c2_1 | 11 KB
openssl-1.1.1c | h1de35cc_1 | 3.4 MB
pixman-0.38.0 | h1de35cc_0 | 611 KB
py-opencv-3.4.2 | py37h7c891bd_1 | 1.3 MB
pytables-3.4.4 | py37h13c8a08_0 | 1.4 MB
-----|-----|-----
Total: | 90.5 MB

The following NEW packages will be INSTALLED:

cairo | pkgs/main/osx-64::cairo-1.14.12-hc4e6be7_4
conda-package-handling | pkgs/main/osx-64::conda-package-handling-1.3.10-py37_0
ffmpeg | pkgs/main/osx-64::ffmpeg-4.0-h01ea3c9_0
fontconfig | pkgs/main/osx-64::fontconfig-2.13.0-h5d5b041_1
graphite2 | pkgs/main/osx-64::graphite2-1.3.13-h2098e52_0
harfbuzz | pkgs/main/osx-64::harfbuzz-1.8.8-hb8d4a28_0
jasper | pkgs/main/osx-64::jasper-2.0.14-h636a363_1
libopencv | pkgs/main/osx-64::libopencv-3.4.2-h7c891bd_1
libopus | pkgs/main/osx-64::libopus-1.3-h1de35cc_0
libvpx | pkgs/main/osx-64::libvpx-1.7.0-h378b8a2_0
opencv | pkgs/main/osx-64::opencv-3.4.2-py37h6fd60c2_1
pixman | pkgs/main/osx-64::pixman-0.38.0-h1de35cc_0
py-opencv | pkgs/main/osx-64::py-opencv-3.4.2-py37h7c891bd_1

The following packages will be REMOVED:

anaconda-2019.03-py37_0

The following packages will be UPDATED:

ca-certificates | 2019.1.23-0 --> 2019.5.15-0
certifi | 2019.3.9-py37_0 --> 2019.6.16-py37_0
conda | 4.6.11-py37_0 --> 4.7.5-py37_0
openssl | 1.1.1b-h1de35cc_1 --> 1.1.1c-h1de35cc_1

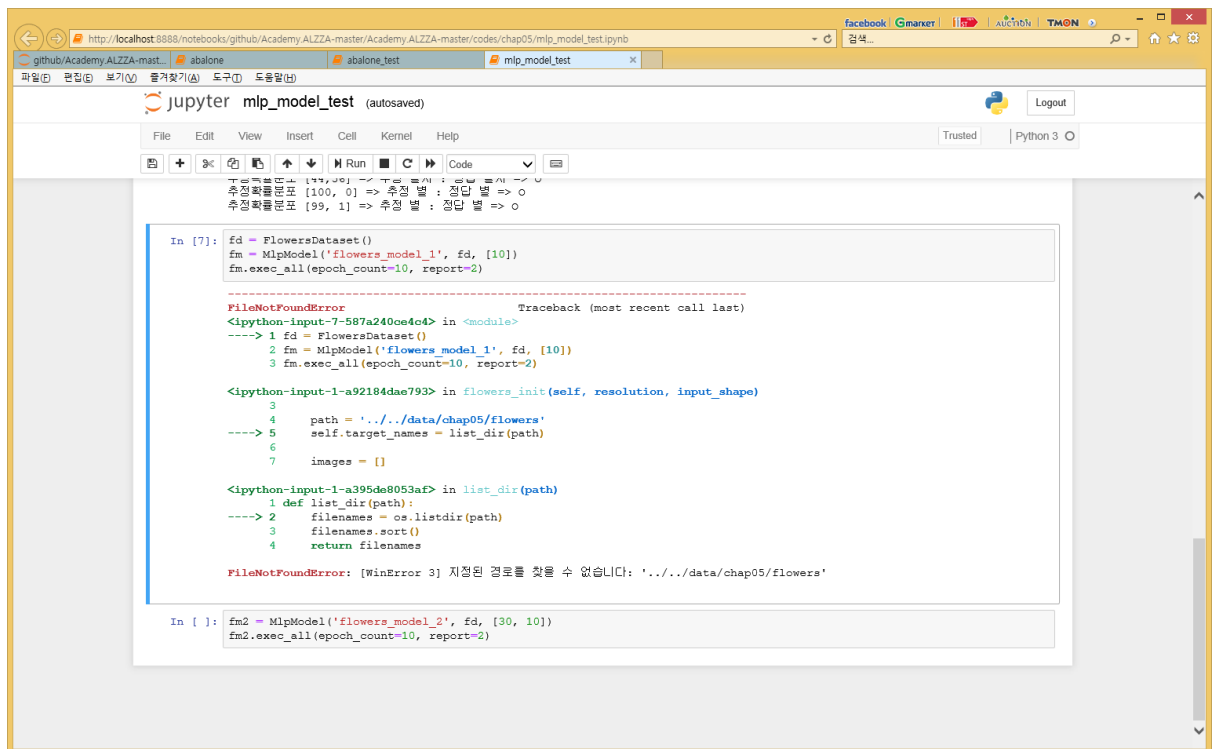
The following packages will be DOWNGRADED:

h5py | 2.9.0-py37h3134771_0 --> 2.8.0-py37h878f3e3_3
hdf5 | 1.10.4-hf1e0ec_0 --> 1.10.2-hf1e0ec_1
pytables | 3.5.1-py37h5bccee9_0 --> 3.4.4-py37h13c8a08_0

2
Proceed ([y]/n)? y
```

3-8. 이제 주피터 노트북에서 codes/chap05/mlp_model_test.ipynb 파일을 다시 실행시키면 이번에는 아래와 같이 데이터 폴더를 찾을 수 없다는 오류 메시지가 나타나게 됩니다. 깃허

브에는 용량 제한이 있어서 딥러닝 실습에 필요한 데이터 파일들을 직접 올릴 수 없습니다. 이에 따라 앞으로 여러 단원에서 이와 유사한 오류가 발생할 수 있습니다.



```
In [7]: fd = FlowersDataset()
fm = MlpModel('flowers_model_1', fd, [10])
fm.exec_all(epoch_count=10, report=2)

FileNotFoundError: Traceback (most recent call last)
<ipython-input-7-587a240ce4c4> in <module>
----> 1 fd = FlowersDataset()
      2 fm = MlpModel('flowers_model_1', fd, [10])
      3 fm.exec_all(epoch_count=10, report=2)

<ipython-input-1-a92184dae793> in flowers_init(self, resolution, input_shape)
      3
      4     path = '../data/chap05/flowers'
----> 5     self.target_names = list_dir(path)
      6
      7     images = []

<ipython-input-1-a395de8053af> in list_dir(path)
----> 1 def list_dir(path):
      2     filenames = os.listdir(path)
      3     filenames.sort()
      4     return filenames

FileNotFoundError: [WinError 3] 지정된 경로를 찾을 수 없습니다: '../data/chap05/flowers'

In [ ]: fm2 = MlpModel('flowers_model_2', fd, [30, 10])
fm2.exec_all(epoch_count=10, report=2)
```

3-9. 이에 대한 처리 방법은 data/chap05/readme.md 파일에 제시되어 있습니다. 깃허브 사이트의 /data/chap05 폴더를 방문하면 보여주는 화면이기도 합니다. 이 화면의 안내 내용에 따라 실습 데이터를 다운로드 받아 실습 폴더 안에 설치하면 탈 없이 실습을 진행할 수 있습니다. 이후 다른 단원의 실습에 필요한 데이터도 같은 방법으로 data 폴더 아래에서 해당 단원의 readme.md 파일이 안내하는 방법에 따라 구하면 됩니다.

