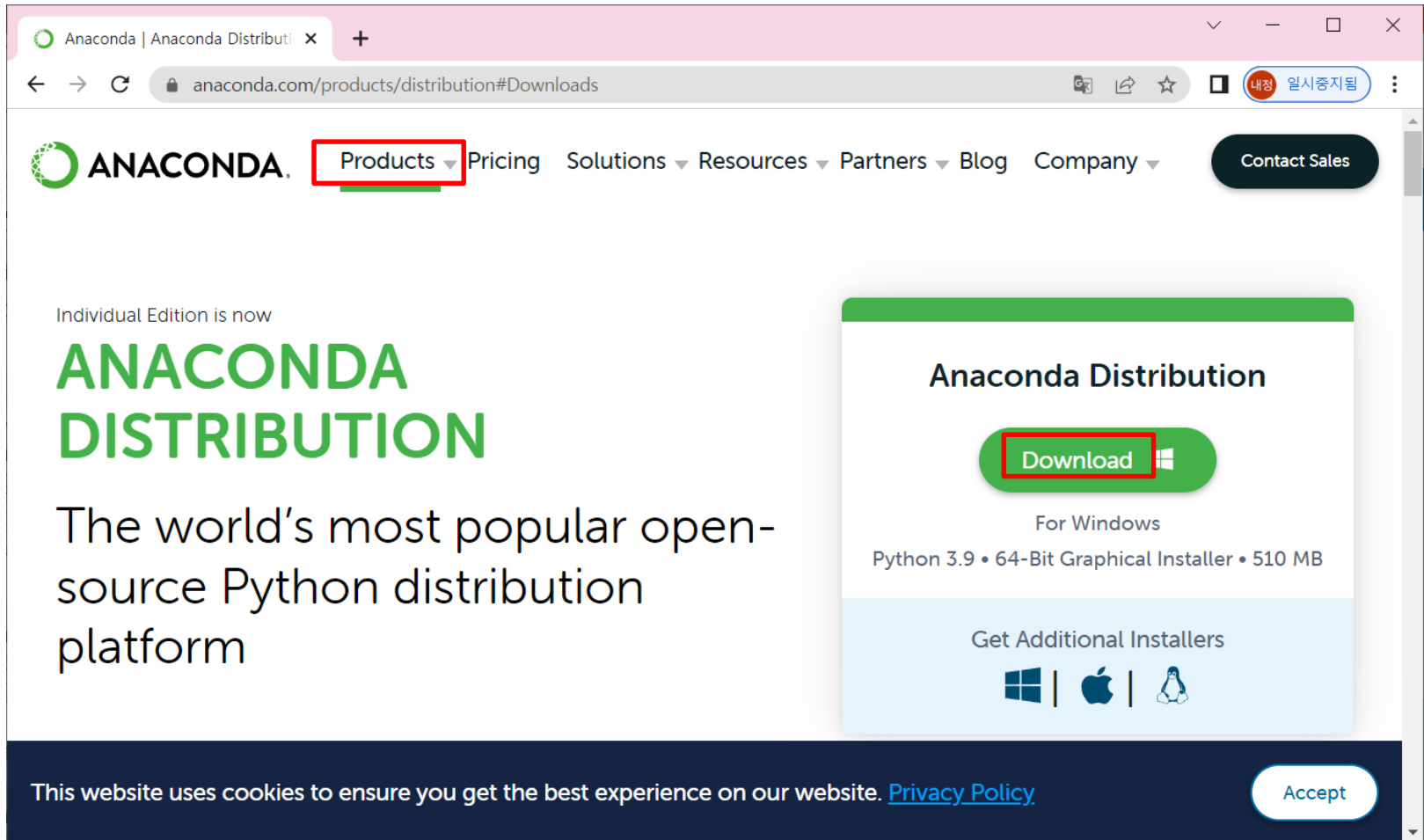


아나콘다 텐서플로 설치

아나콘다 설치

■ 아나콘다 홈페이지

https://anaconda.org/



■ 아나콘다 홈페이지

<https://www.anaconda.com/products/distribution#Downloads>

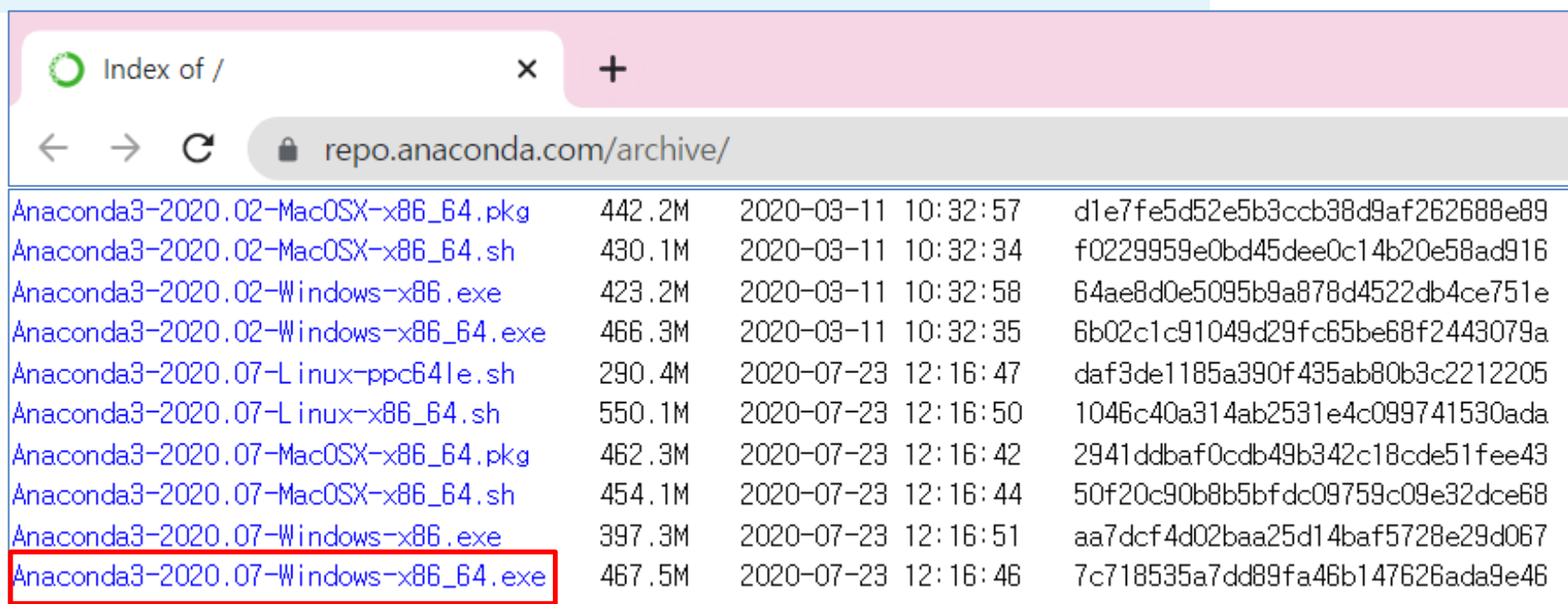
Anaconda Installers

Windows 	MacOS 	Linux 
Python 3.9	Python 3.9	Python 3.9
64-Bit Graphical Installer (510 MB)	64-Bit Graphical Installer (515 MB)	64-Bit (x86) Installer (581 MB)
32-Bit Graphical Installer (404 MB)	64-Bit Command Line Installer (508 MB)	64-Bit (Power8 and Power9) Installer (255 MB)
		64-Bit (AWS Graviton2 / ARM64) Installer (488 MB)
		64-bit (Linux on IBM Z & LinuxONE) Installer (242 MB)

ADDITIONAL INSTALLERS

The archive has older versions of Anaconda Distribution installers. The Miniconda installer homepage can be found here.

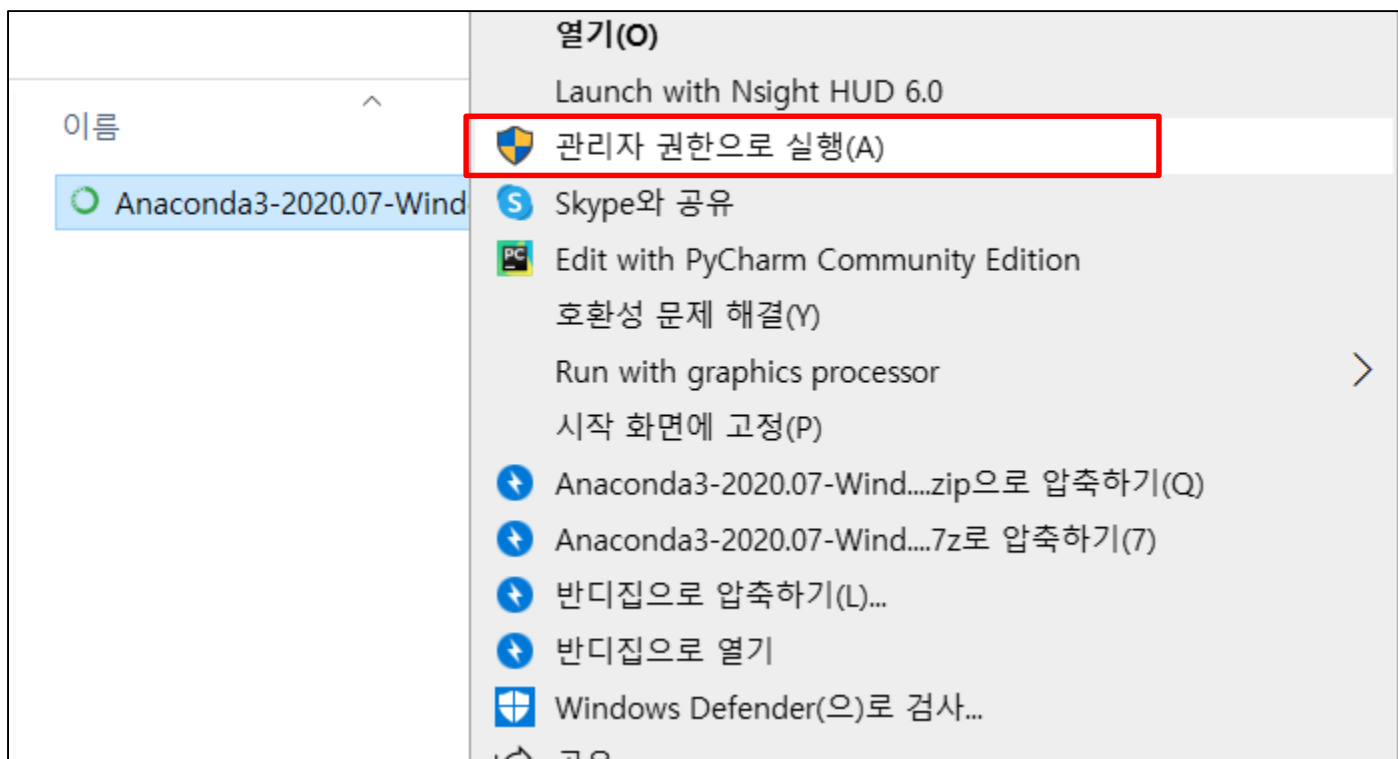
<https://repo.anaconda.com/archive/>



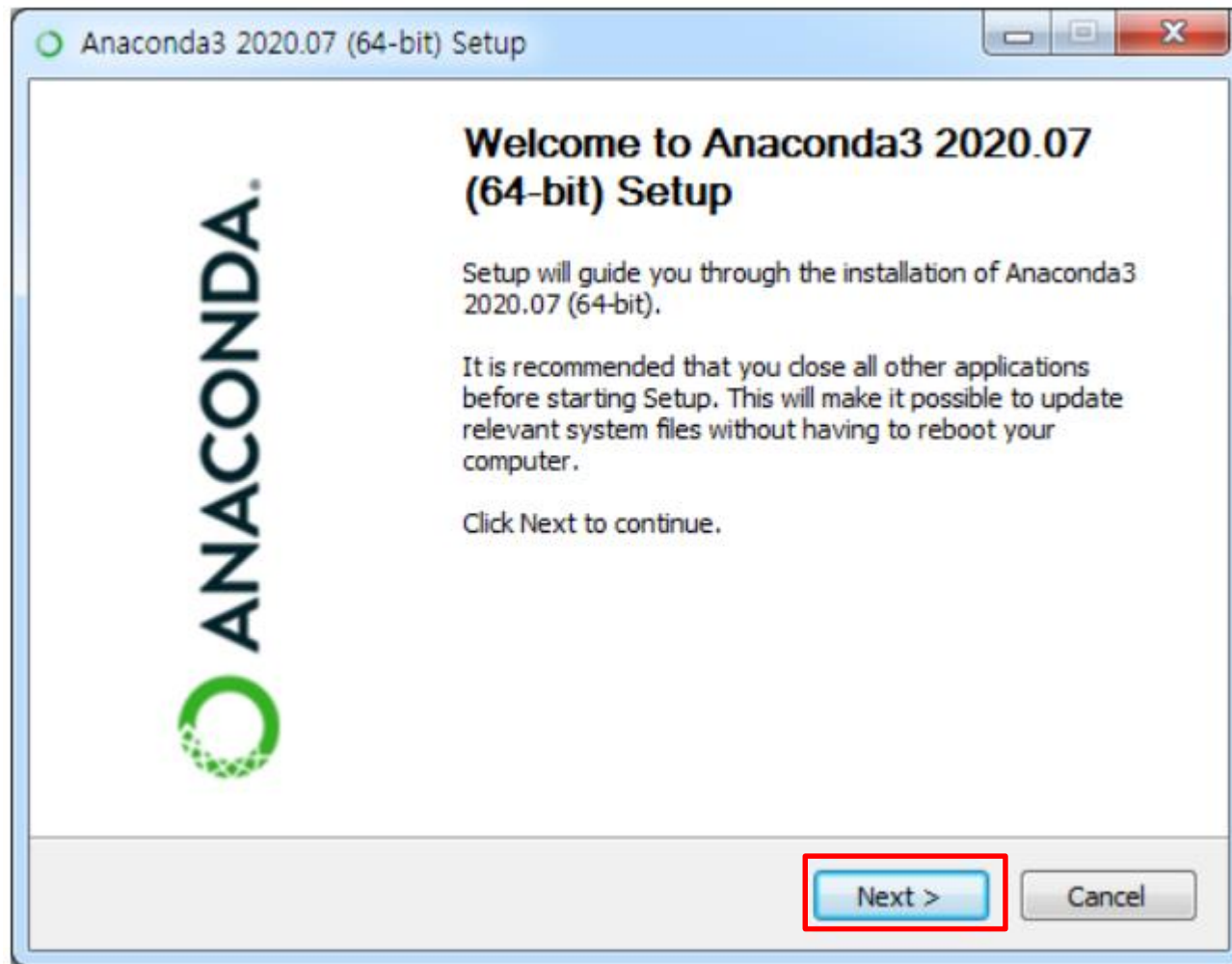
The screenshot shows a web browser window with the address bar displaying `repo.anaconda.com/archive/`. The page content is an index of files, with the file `Anaconda3-2020.07-Windows-x86_64.exe` highlighted by a red box.

Index of /					×	+
←	→	↻	🔒	repo.anaconda.com/archive/		
Anaconda3-2020.02-MacOSX-x86_64.pkg	442.2M	2020-03-11	10:32:57	d1e7fe5d52e5b3ccb38d9af262688e89		
Anaconda3-2020.02-MacOSX-x86_64.sh	430.1M	2020-03-11	10:32:34	f0229959e0bd45dee0c14b20e58ad916		
Anaconda3-2020.02-Windows-x86.exe	423.2M	2020-03-11	10:32:58	64ae8d0e5095b9a878d4522db4ce751e		
Anaconda3-2020.02-Windows-x86_64.exe	466.3M	2020-03-11	10:32:35	6b02c1c91049d29fc65be68f2443079a		
Anaconda3-2020.07-Linux-ppc64le.sh	290.4M	2020-07-23	12:16:47	daf3de1185a390f435ab80b3c2212205		
Anaconda3-2020.07-Linux-x86_64.sh	550.1M	2020-07-23	12:16:50	1046c40a314ab2531e4c099741530ada		
Anaconda3-2020.07-MacOSX-x86_64.pkg	462.3M	2020-07-23	12:16:42	2941ddbaf0cdb49b342c18cde51fee43		
Anaconda3-2020.07-MacOSX-x86_64.sh	454.1M	2020-07-23	12:16:44	50f20c90b8b5bfdc09759c09e32dce68		
Anaconda3-2020.07-Windows-x86.exe	397.3M	2020-07-23	12:16:51	aa7dcf4d02baa25d14baf5728e29d067		
Anaconda3-2020.07-Windows-x86_64.exe	467.5M	2020-07-23	12:16:46	7c718535a7dd89fa46b147626ada9e46		

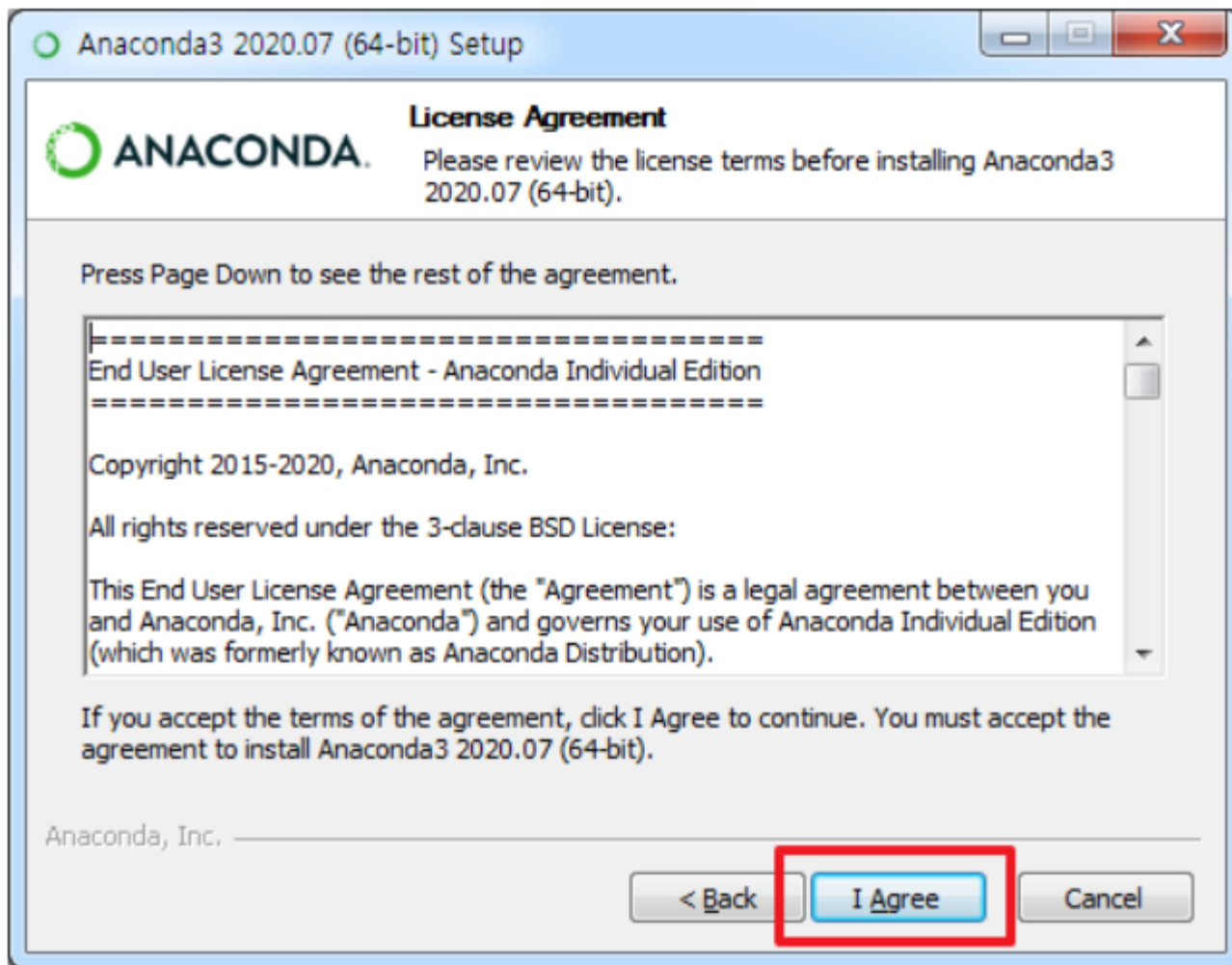
아나콘다 설치



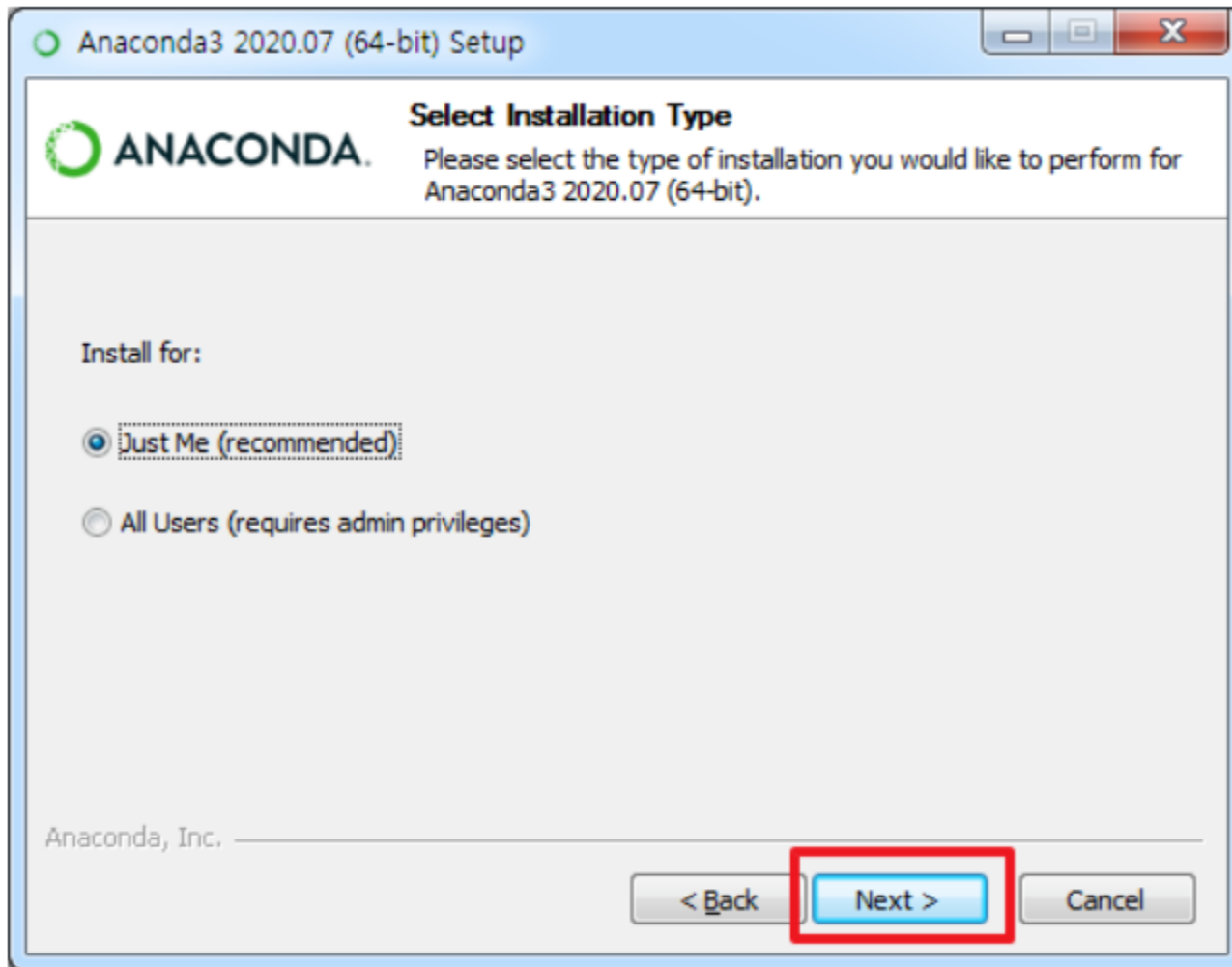
아나콘다 설치



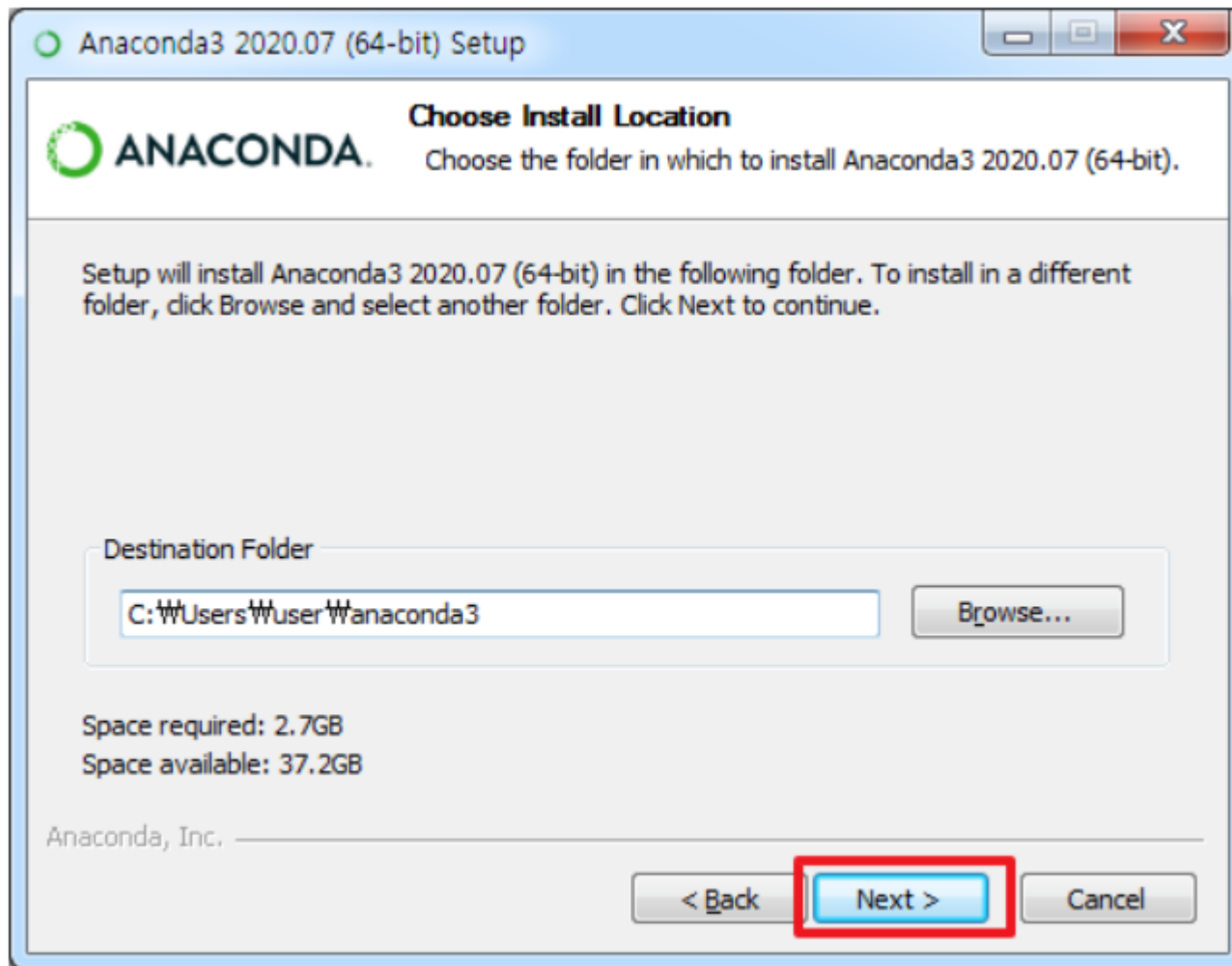
아나콘다 설치

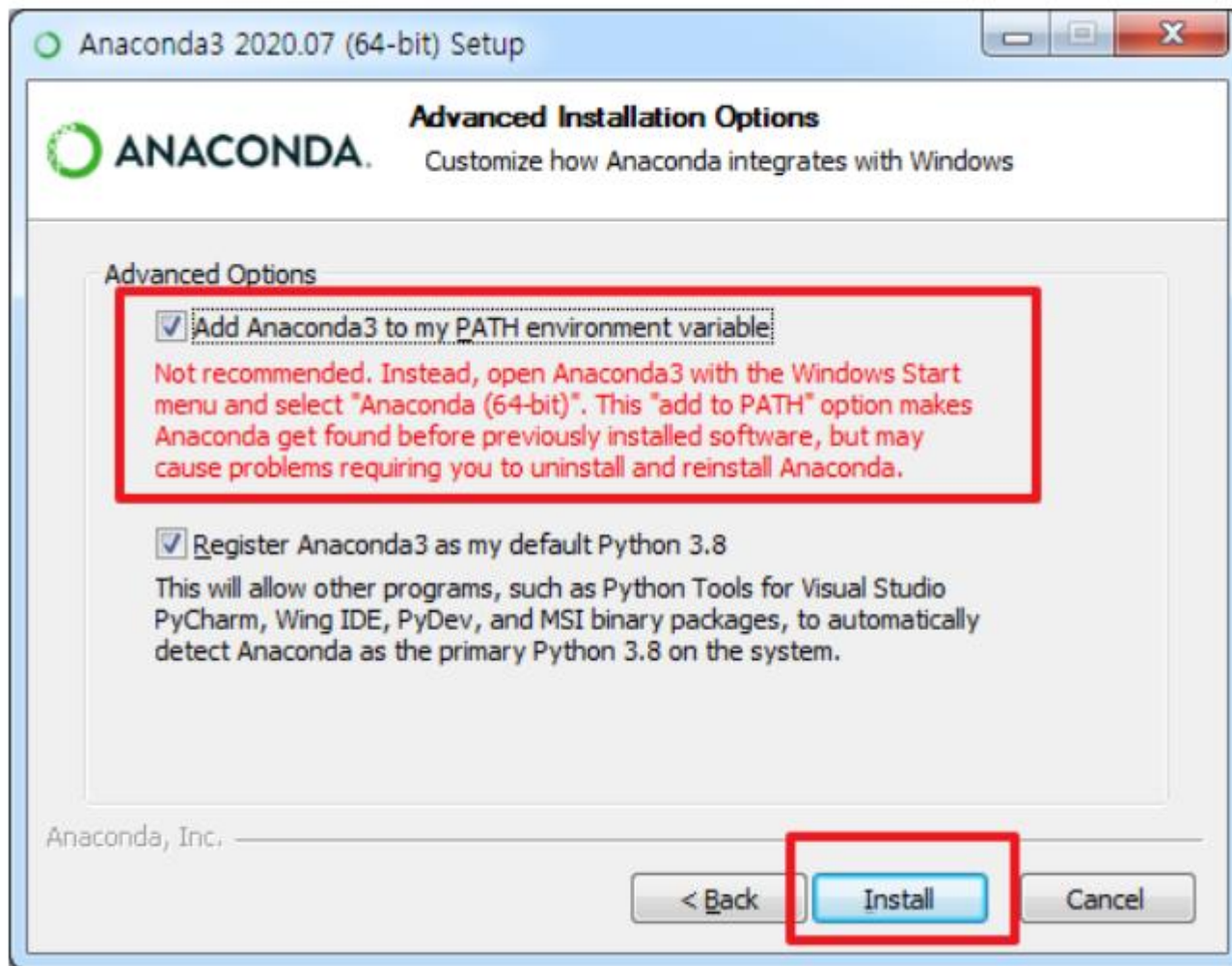


아나콘다 설치

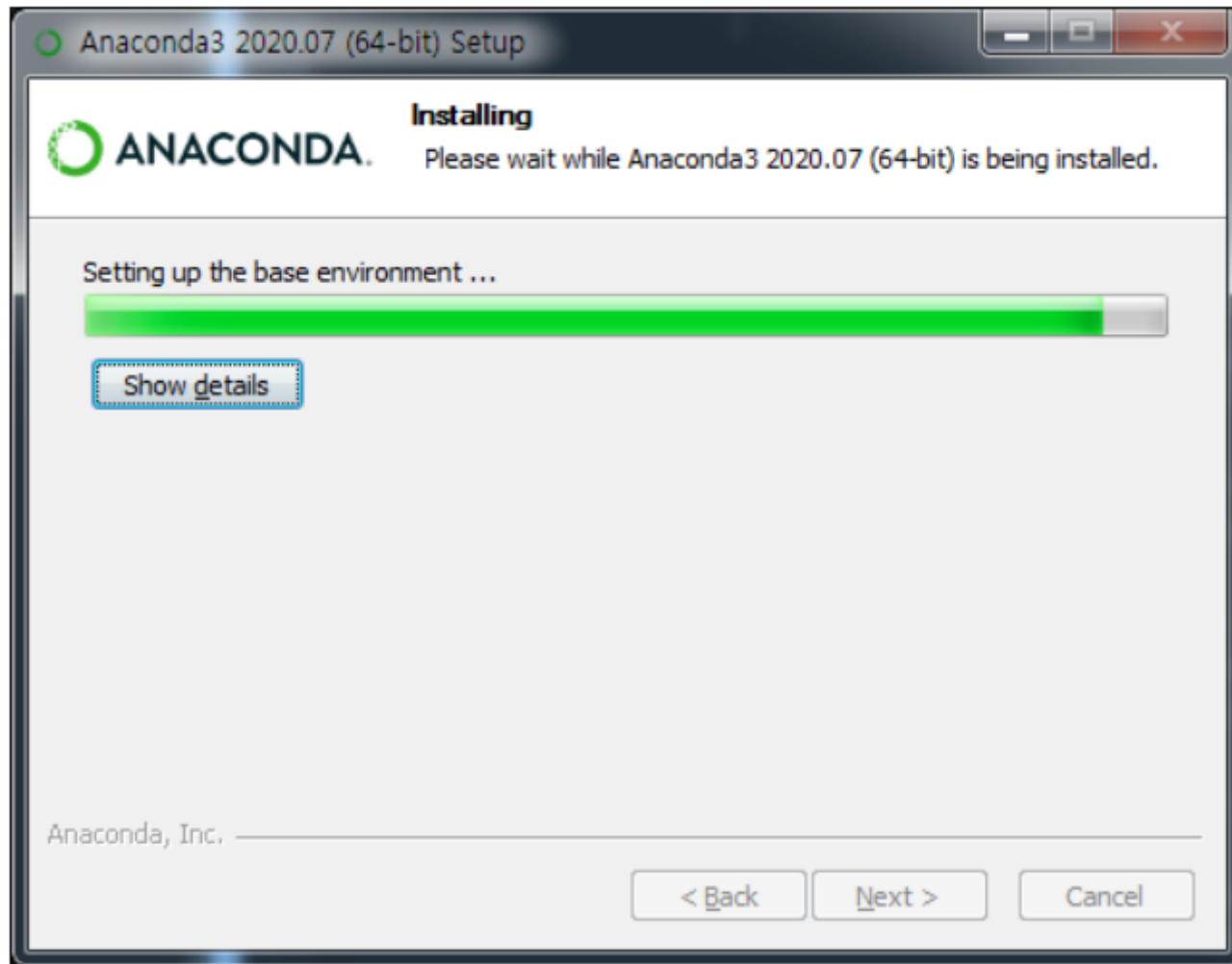


아나콘다 설치

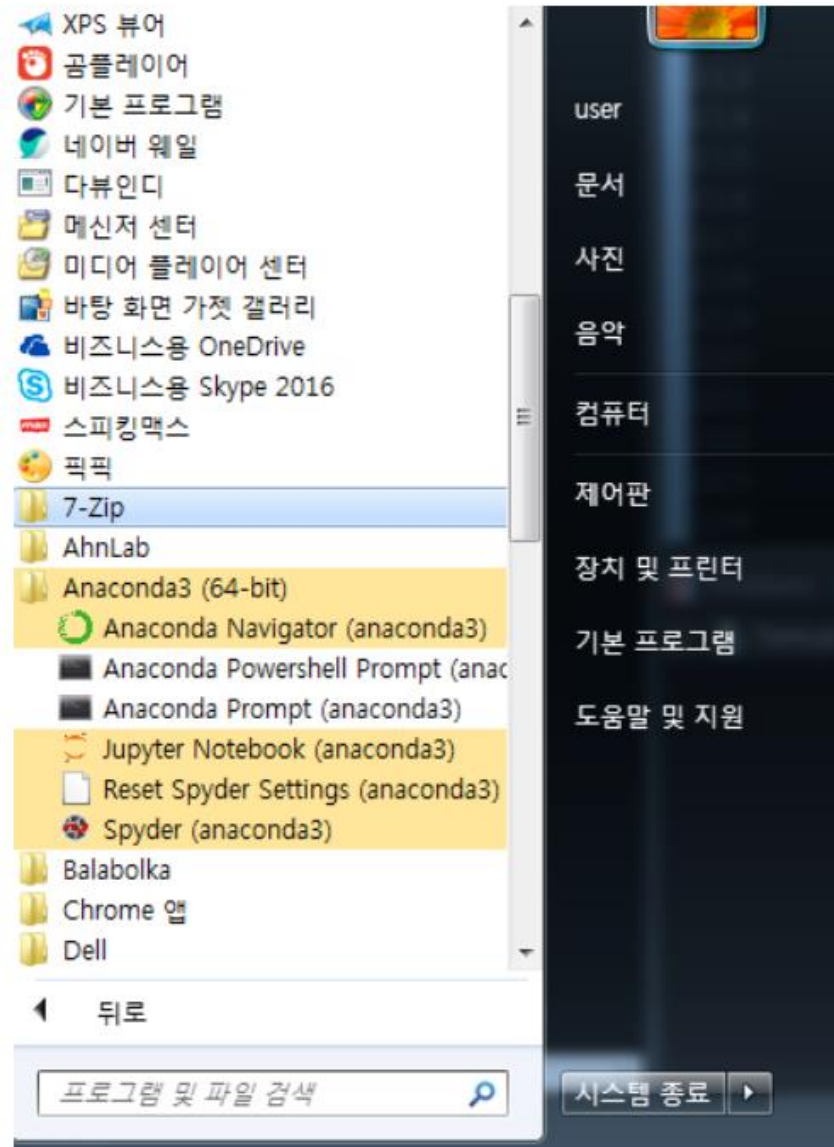




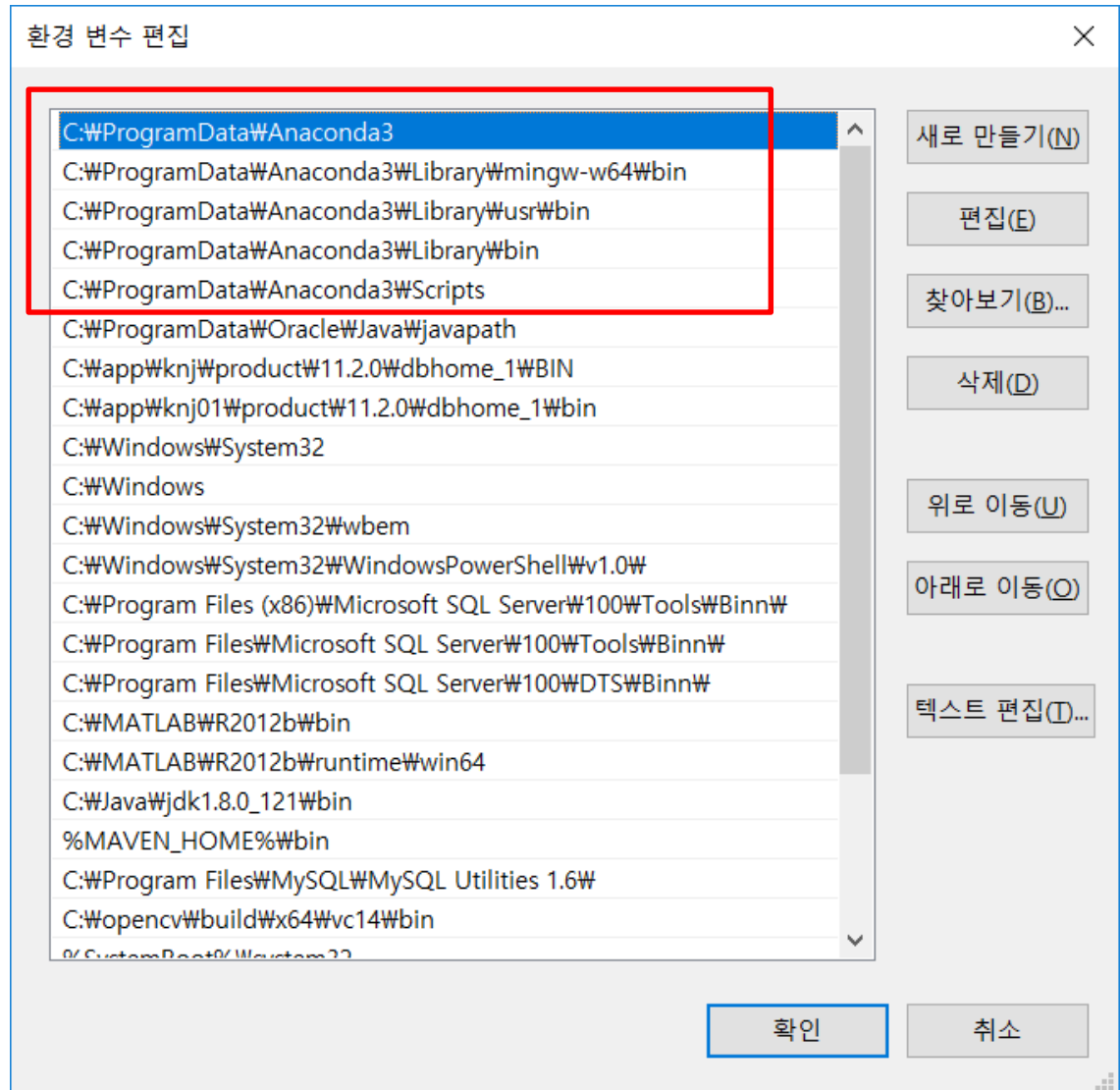
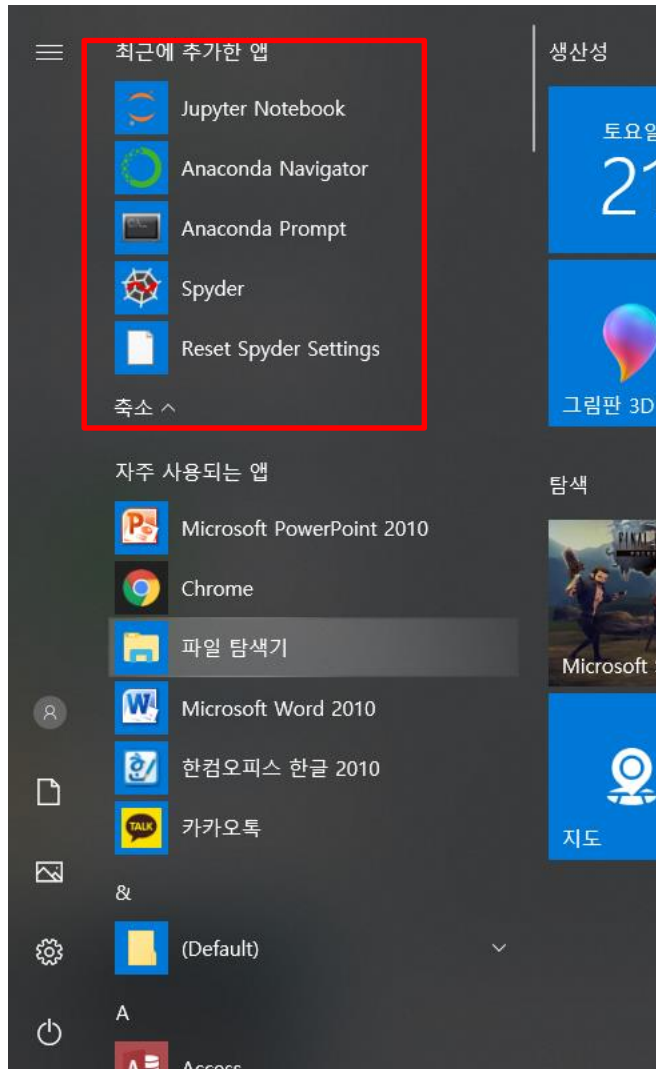
아나콘다 설치



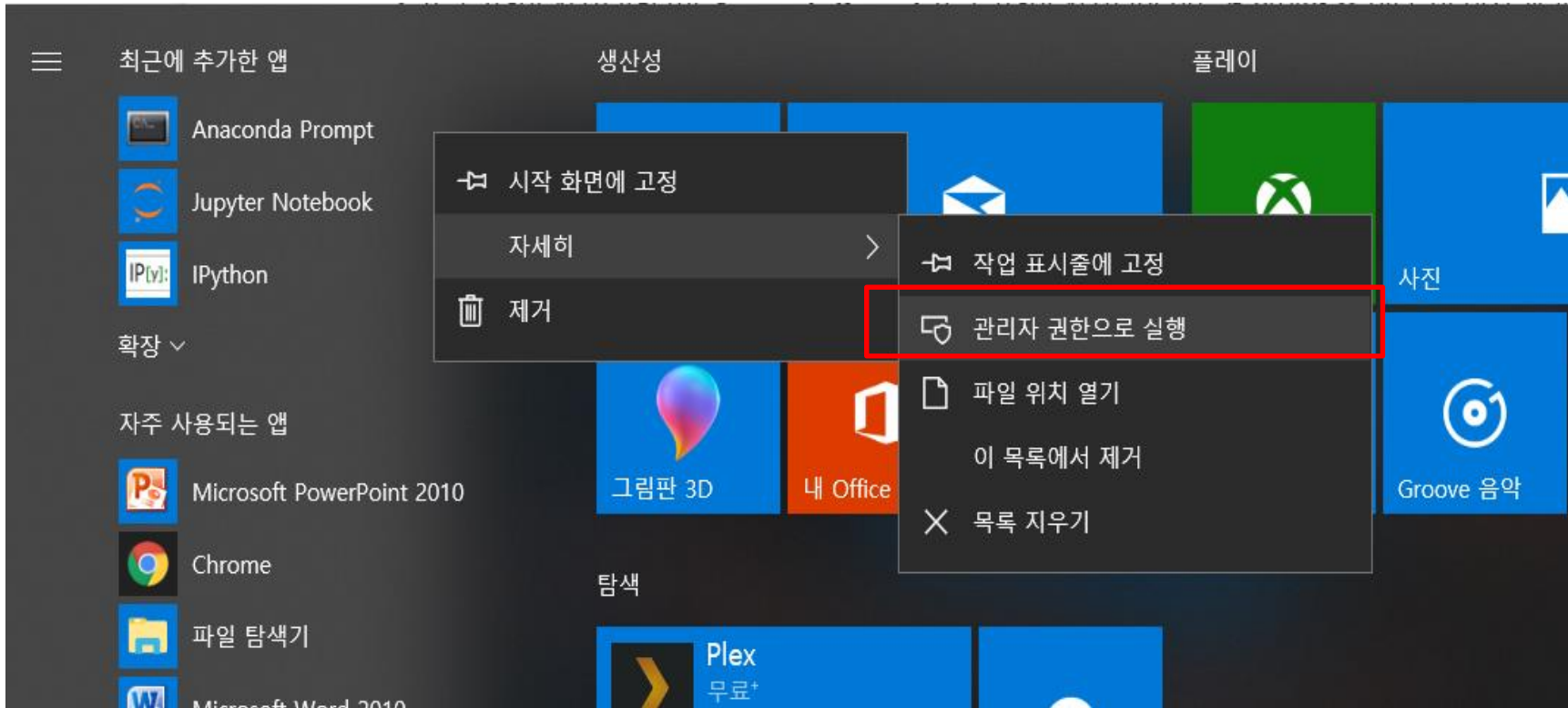
아나콘다 설치



아나콘다 및 파이썬 설치



아나콘다 및 파이썬 설치



아나콘다 및 파이썬 설치

■ 아나콘다 버전 확인 및 업데이트

```
1 #아나콘다 버전 확인
2 conda --version
3
4 #아나콘다 업데이트
5 conda update conda
```

■ 아나콘다 가상환경 생성

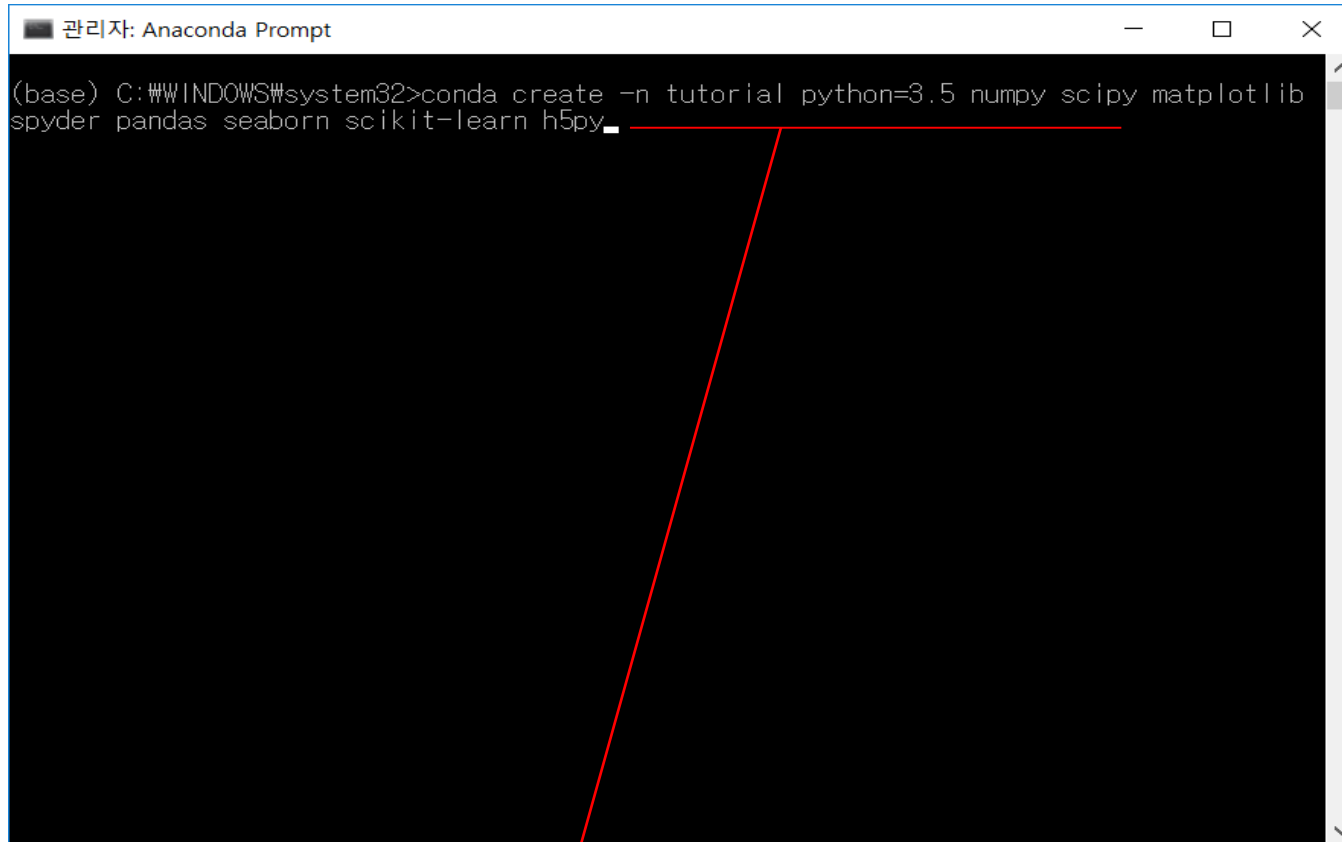
```
1 #아나콘다 가상환경 생성
2 conda create --name(-n) 가상환경명 설치할패키지
3
4 #예) 파이썬 3.5 버전 설치 & test 이름으로 가상환경 생성
5 conda create --name test python=3.5
6
7 #또는
8 conda create --n test python=3.5
```

■ pip가 인스톨이 안되었을 경우

```
conda install pip
```

아나콘다 및 파이썬 설치

■ 아나콘다 가상환경 생성



A screenshot of the Anaconda Prompt window. The title bar reads '관리자: Anaconda Prompt'. The command prompt shows the command: `(base) C:\WINDOWS\system32>conda create -n tutorial python=3.5 numpy scipy matplotlib spyder pandas seaborn scikit-learn h5py`. A red arrow points from the end of this command to a text box below.

`conda create -n tutorial python=3.8 numpy scipy matplotlib
spyder pandas seaborn scikit-learn h5py`

아나콘다 및 파이썬 설치

- **주피터 노트북**
 - 프로그램 코드를 브라우저에서 실행해주는 대화식 환경.
 - 탐색적 데이터 분석에 적합.
 -
- **Numpy 넘파이**
 - 파이썬으로 과학 계산을 하려면 꼭 필요한 패키지.
 - 다차원 배열을 위한 기능과 선형 대수 연산과 푸리에 변환 같은 고수준 수학 함수와 유사 난수 생성기를 포함
 -
- **Scipy 싸이파이**
 - 과학 계산을 함수를 모아놓은 파이썬 패키지.
 - 고성능 선형 대수, 함수 최적화, 신호 처리, 특수한 수학 함수와 통계 분포등을 포함한 많은 기능 제공
 -
- **matplotlib 맷플랏립**
 - 대표적인 과학 계산을 그래프 라이브러리.
 - 선 그래프, 히스토그램, 산점도 등을 지원하며 고품질 그래프를 그려준다.
 -
- **pandas 판다스**
 - 데이터 처리와 분석을 위한 파이썬 라이브러리.
 - R의 data.frame을 본떠서 설계한 DataFrame이라는 데이터 구조를 기반으로 만들어졌다.
 - 엑셀의 스프레드시트와 비슷한 테이블 형태.
 - SQL처럼 테이블에 쿼리나 조인을 수행할 수 있다.
 - SQL, 엑셀 파일, CSV 파일 같은 다양한 파일과 DB에서 읽어들이 수 있다.

아나콘다 및 파이썬 설치

```
관리자: Anaconda Prompt - conda create -n tutorial python=3.5 numpy scipy matplotlib spyder ...  
snowballstemmer: 1.2.1-py35h4c55bfa_0  
sphinx: 1.7.5-py35_0  
sphinxcontrib: 1.0-py35h45f5ca3_1  
sphinxcontrib-websupport: 1.1.0-py35_1  
spyder: 3.3.0-py35_0  
spyder-kernels: 0.2.4-py35_0  
sqlite: 3.24.0-h7602738_0  
statsmodels: 0.9.0-py35h452e1ab_0  
testpath: 0.3.1-py35h06cf69e_0  
tornado: 5.0.2-py35_0  
traitlets: 4.3.2-py35h09b975b_0  
urllib3: 1.23-py35_0  
vc: 14.1-h0510ff6_3  
vs2015_runtime: 15.5.2-3  
wcwidth: 0.1.7-py35h6e80d8a_0  
webencodings: 0.5.1-py35h5d527fb_1  
wheel: 0.31.1-py35_0  
win_inet_pton: 1.0.1-py35hbef1270_1  
win_unicode_console: 0.5-py35h56988b5_0  
wincertstore: 0.2-py35hfebbdb8_0  
wrapt: 1.10.11-py35h54666f7_0  
zeromq: 4.2.5-he025d50_1  
zlib: 1.2.11-h8395fce_2  
  
Proceed ([y]/n)? y  
  
Downloading and Extracting Packages  
certifi-2018.4.16 | 143 KB | ##### | 100%  
qt-5.9.6 | 92.5 MB | #3 | 3%
```

■ 가상환경 정보 확인 및 활성화 비활성화

```
1 #설치 된 가상환경 리스트 확인
2 conda info --envs
3
4 #가상환경 활성화
5 #예) activate test
6 activate 가상환경명
7
8 #가상환경 비활성화
9 #예) deactivate test
10 deactivate 가상환경명
```

아나콘다 및 파이썬 설치

■ 가상환경 정보 확인 및 활성화 비활성화

```
1 #설치 된 가상환경 리스트 확인
2 conda info --envs
3
4 #가상환경 활성화
5 #예) activate test
6 activate 가상환경명
7
8 #가상환경 비활성화
9 #예) deactivate test
10 deactivate 가상환경명
```

아나콘다 및 파이썬 설치

■ 설치 후 화면과 가상환경 활성화

```
관리자: Anaconda Prompt
setuptools-39.2.0 | 593 KB | ##### | 100%
isort-4.3.4 | 77 KB | ##### | 100%
win_inet_pton-1.0.1 | 5 KB | ##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: # DEBUG menuinst_win32:__init__(196): Menu: name: 'Anaconda${PY
_VER} ${PLATFORM}', prefix: 'C:\ProgramData\Anaconda3\envs\tutorial', env_name: 'tutor
ial', mode: 'system', used_mode: 'system'
DEBUG menuinst_win32:create(320): Shortcut cmd is C:\ProgramData\Anaconda3\pythonw.exe
, args are ['C:\ProgramData\Anaconda3\cwp.py', 'C:\ProgramData\Anaconda3\envs\t
utorial', 'C:\ProgramData\Anaconda3\envs\tutorial\pythonw.exe', 'C:\ProgramData\
Anaconda3\envs\tutorial\Scripts\spyder-script.py']
| DEBUG menuinst_win32:create(320): Shortcut cmd is C:\ProgramData\Anaconda3\python.ex
e, args are ['C:\ProgramData\Anaconda3\cwp.py', 'C:\ProgramData\Anaconda3\envs\t
utorial', 'C:\ProgramData\Anaconda3\envs\tutorial\python.exe', 'C:\ProgramData\
Anaconda3\envs\tutorial\Scripts\spyder-script.py', '--reset']
done
#
# To activate this environment, use
#
#     $ conda activate tutorial
#
# To deactivate an active environment, use
#
#     $ conda deactivate
#
(base) C:\WINDOWS\system32>activate tutorial
(tutorial) C:\WINDOWS\system32>
```

04.Tensorflow 설치

- **Tensorflow 설치**

pip install tensorflow==2.4(or 2.5)

■ 케라스(Keras):위키백과 참조

- 파이썬으로 작성된 오픈 소스 신경망 라이브러리이다. MXNet, Deeplearning4j, 텐서플로, Microsoft Cognitive Toolkit 또는 Theano 위에서 수행할 수 있다. 딥 신경망과의 빠른 실험을 가능케 하도록 설계되었으며 최소한의 모듈 방식의 확장 가능성에 초점을 둔다

05.Keras 설치

■ Keras 설치

```
관리자: Anaconda Prompt - pip install keras
Python 3.5.5 [Anaconda, Inc.] (default, Apr 7 2018, 04:52:34) [MSC v.1900 64 bit (AMD 64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import tensorflow as tf
>>> print(tf.__version__)
1.9.0
>>> hello =tf.constant('Hello, TensorFlow!')
>>> sess=tf.Session()
2018-07-21 10:25:27.181581: I T:\src\github\tensorflow\tensorflow\core\platform\cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not
compiled to use: AVX2
>>> print(sess.run(hello))
b'Hello, TensorFlow!'
>>> a=tf.constant(32)
>>> b=tf.constant(10)
>>> print(sess.run(a+b))
42
>>> exit
Use exit() or Ctrl-Z plus Return to exit
>>> exit()

(tutorial) C:\WINDOWS\system32>deactivate

(base) C:\WINDOWS\system32>activate tutorial
(tutorial) C:\WINDOWS\system32>pip install keras
Collecting keras
  Downloading https://files.pythonhosted.org/packages/68/12/4cab5c01451eb3b413d19ea151f36e33026fc0efb932bf51bcaf54acbf5/Keras-2.2.0-py2.py3-none-any.whl (300kB)
    20% |#####| 61kB 240kB/s eta 0:00:01
```

(base) C:\WINDOWS\system32>activate tutorial
(tutorial) C:\WINDOWS\system32>pip install keras

05.Keras 설치

관리자: Anaconda Prompt - python

```
8c236b37cb3ee6b37e8d57152b1a5e0b47/Keras_Applications-1.0.2-py2.py3-none-any.whl (43kB)
100% |#####| 51kB 227kB/s
Collecting keras-preprocessing==1.0.1 (from keras)
  Downloading https://files.pythonhosted.org/packages/f8/33/275506afe1d96b221f66f95adb
a94d1b73f6b6087cfb6132a5655b6fe338/Keras_Preprocessing-1.0.1-py2.py3-none-any.whl
Requirement already satisfied: six>=1.9.0 in c:\programdata\anaconda3\envs\tutorial\li
b\site-packages (from keras) (1.11.0)
mkl-fft 1.0.0 requires cython, which is not installed.
mkl-random 1.0.1 requires cython, which is not installed.
Installing collected packages: pyyaml, keras-applications, keras-preprocessing, keras
Successfully installed keras-2.2.0 keras-applications-1.0.2 keras-preprocessing-1.0.1
pyyaml-3.13

(tutorial) C:\WINDOWS\system32>import keras
'import'은(는) 내부 또는 외부 명령, 실행할 수 있는 프로그램, 또는
배치 파일이 아닙니다.

(tutorial) C:\WINDOWS\system32>python
Python 3.5.5 [Anaconda, Inc.] (default, Apr 7 2018, 04:52:34) [MSC v.1900 64 bit (AMD
64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import keras
Using TensorFlow backend.
>>>
```

통합 개발 환경에서 코딩하기

■ 통합 개발 환경

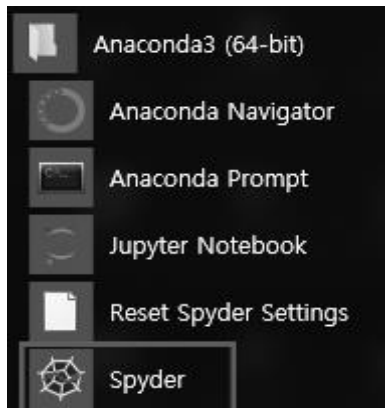
- 텍스트 편집기와 파이썬 개발 환경이 하나의 프로그램에서 동작

■ Spyder 통합 개발 환경: 아나콘다 배포판에 포함돼 있음

- IPython 콘솔(Console)과 내장 편집기(Editor)가 통합

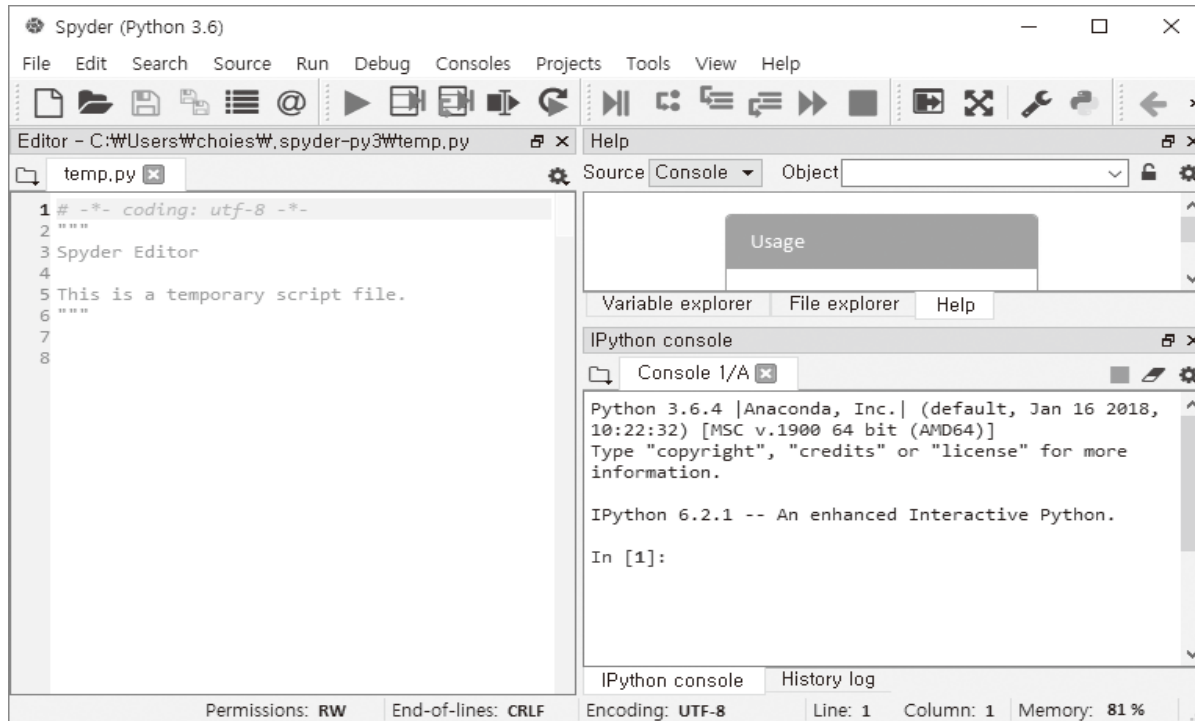
■ Spyder 실행 및 설정

- 아나콘다 메뉴에서 [Spyder]를 클릭



통합 개발 환경에서 코딩하기

■ Spyder 통합 개발 환경

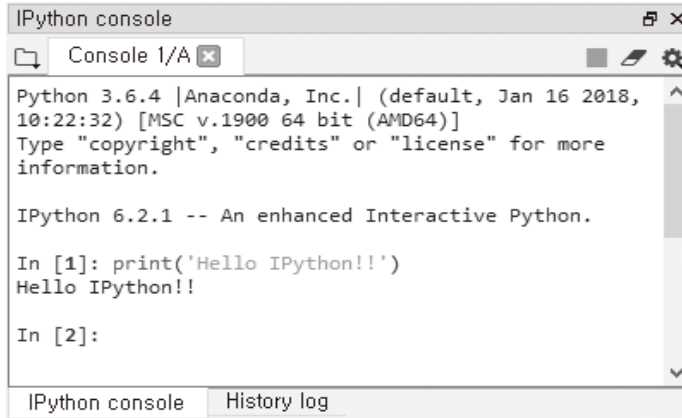


■ IPython 콘솔의 프롬프트는 'In [1]:'

- 대괄호 안의 숫자는 코드를 입력할 때마다 1씩 증가

통합 개발 환경에서 코딩하기

- 'In [1]:' 다음에 `print('Hello IPython!!')`를 입력해서 실행한 결과



```
IPython console
Console 1/A
Python 3.6.4 |Anaconda, Inc.| (default, Jan 16 2018, 10:22:32) [MSC v.1900 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 6.2.1 -- An enhanced Interactive Python.

In [1]: print('Hello IPython!!')
Hello IPython!!

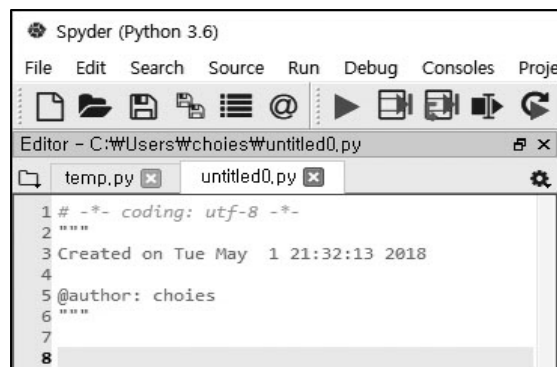
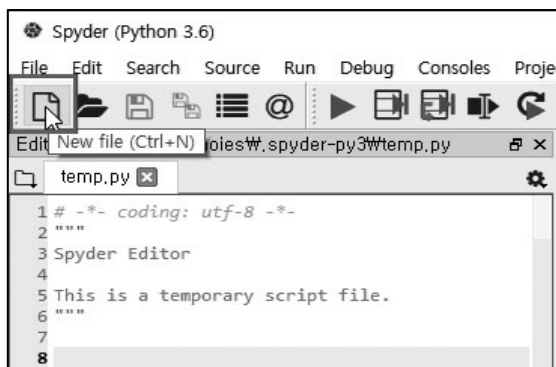
In [2]:

IPython console History log
```

- IPython에서 `print`를 입력하면 글자색이 자동으로 바뀜
 - IPython이 파이썬 내장 명령어를 인식해서 글자색을 변경

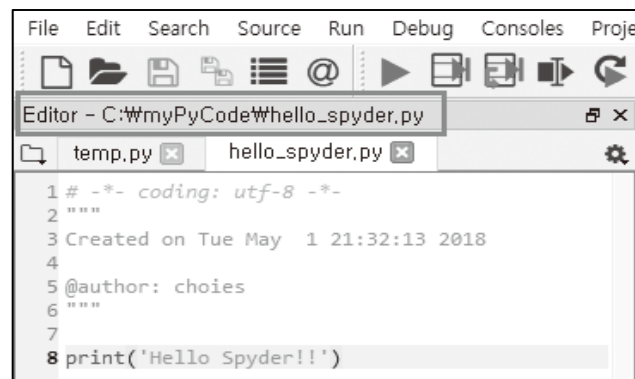
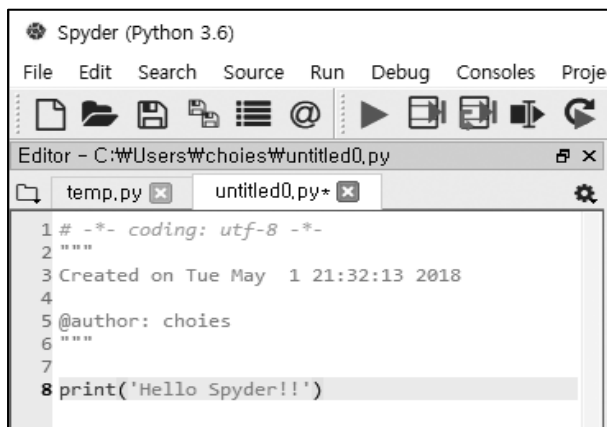
Spyder 에디터에서 코드 작성하기

- Spyder에서 키보드로 Ctrl + N을 누르거나 마우스로 상단의 New file 아이콘을 클릭
- 새 파일이 열리고 문자 인코딩('utf-8') 형식, 파일 생성 날짜, 파일을 생성한 현재 사용자 정보가 자동으로 앞에 삽입됨



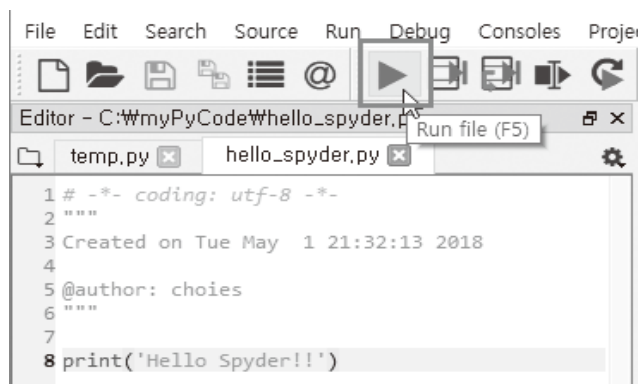
Spyder 에디터에서 코드 작성하기

- 새로 생성한 파이썬 파일에 `print('Hello Spyder!!')` 입력
- 키보드로 `Ctrl + S`를 입력하거나 마우스로 [Save file] 아이콘을 클릭한 후 작업 폴더인 'C:\myPyCode' 폴더에 파일 이름을 `hello_spyder.py`로 저장

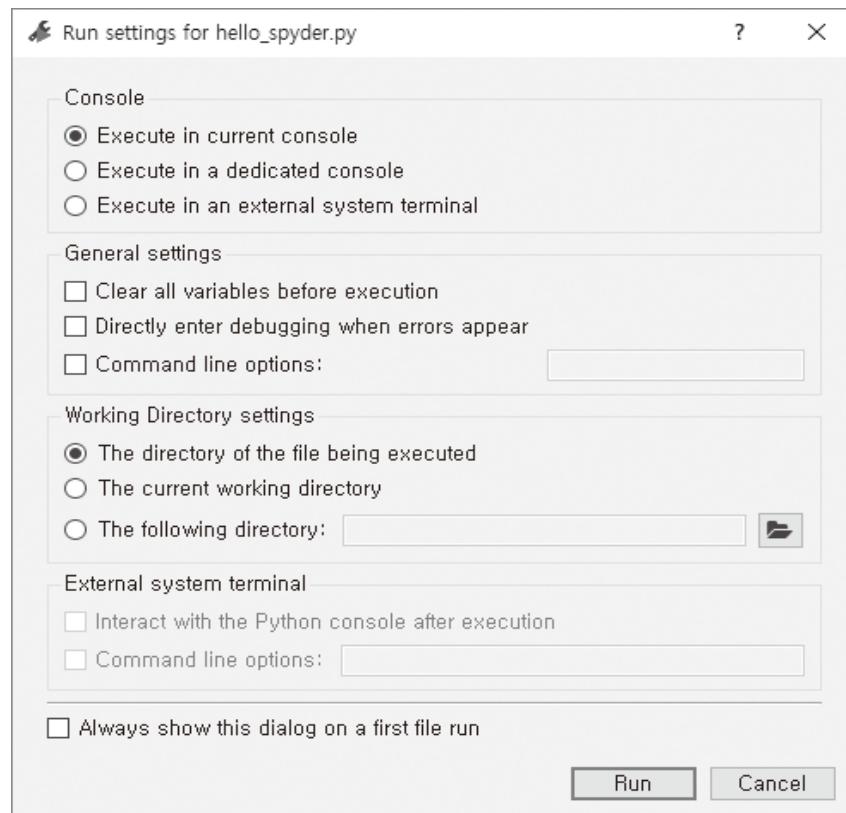


Spyder 에디터에서 코드 작성하기

- F5 키를 누르거나 [Run file(F5)] 아이콘을 클릭

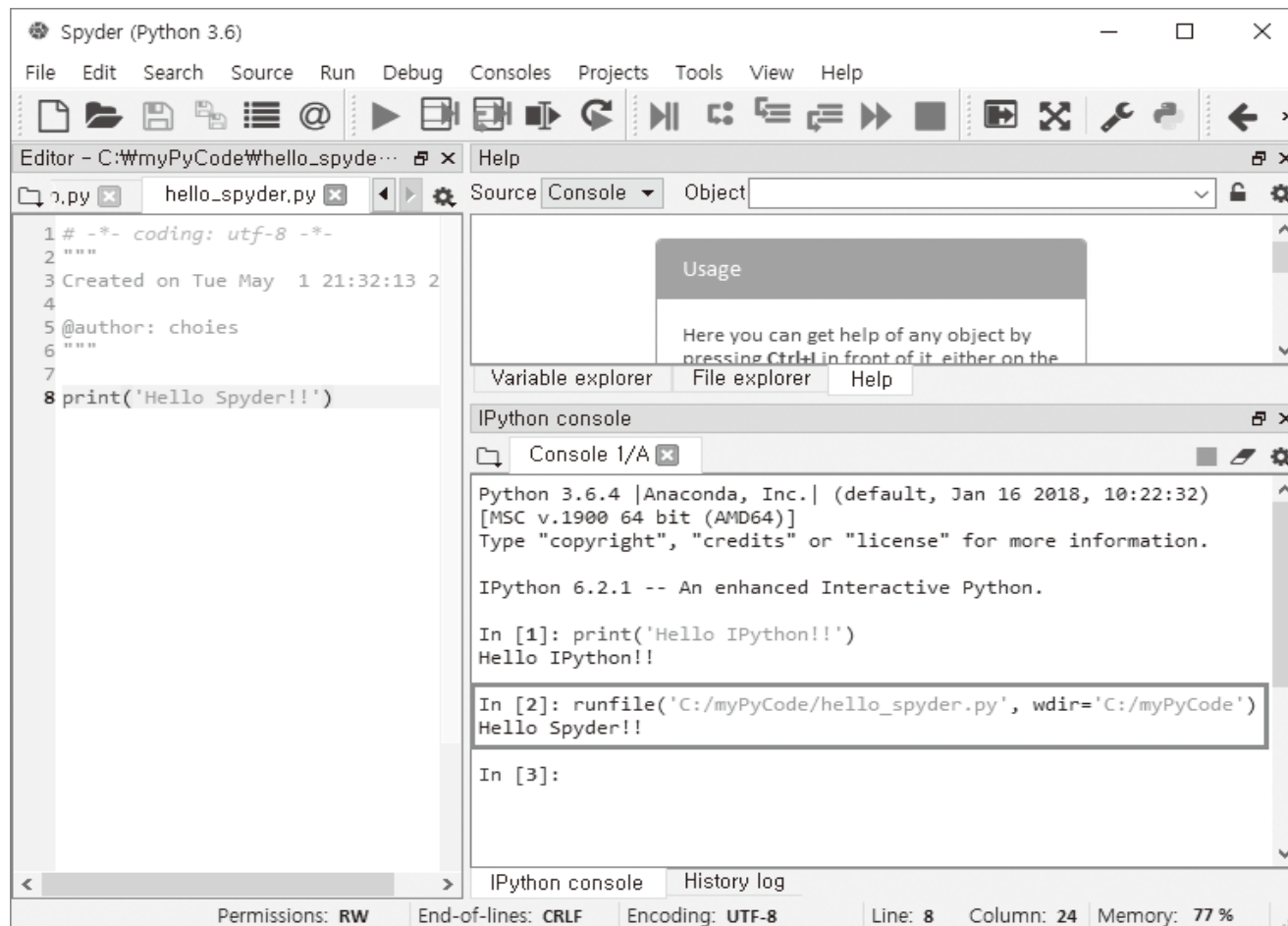


- Spyder에서 저장된 코드를 처음 실행하는 경우 나타나는 설정 창에서 하단의 [Run] 버튼을 클릭



Spyder 에디터에서 코드 작성하기

- 모든 것이 정상적으로 수행되면 IPython 콘솔에서 'hello_spyder.py' 파일을 실행하고 실행 결과가 표시됨



주피터 노트북에서 코딩하기

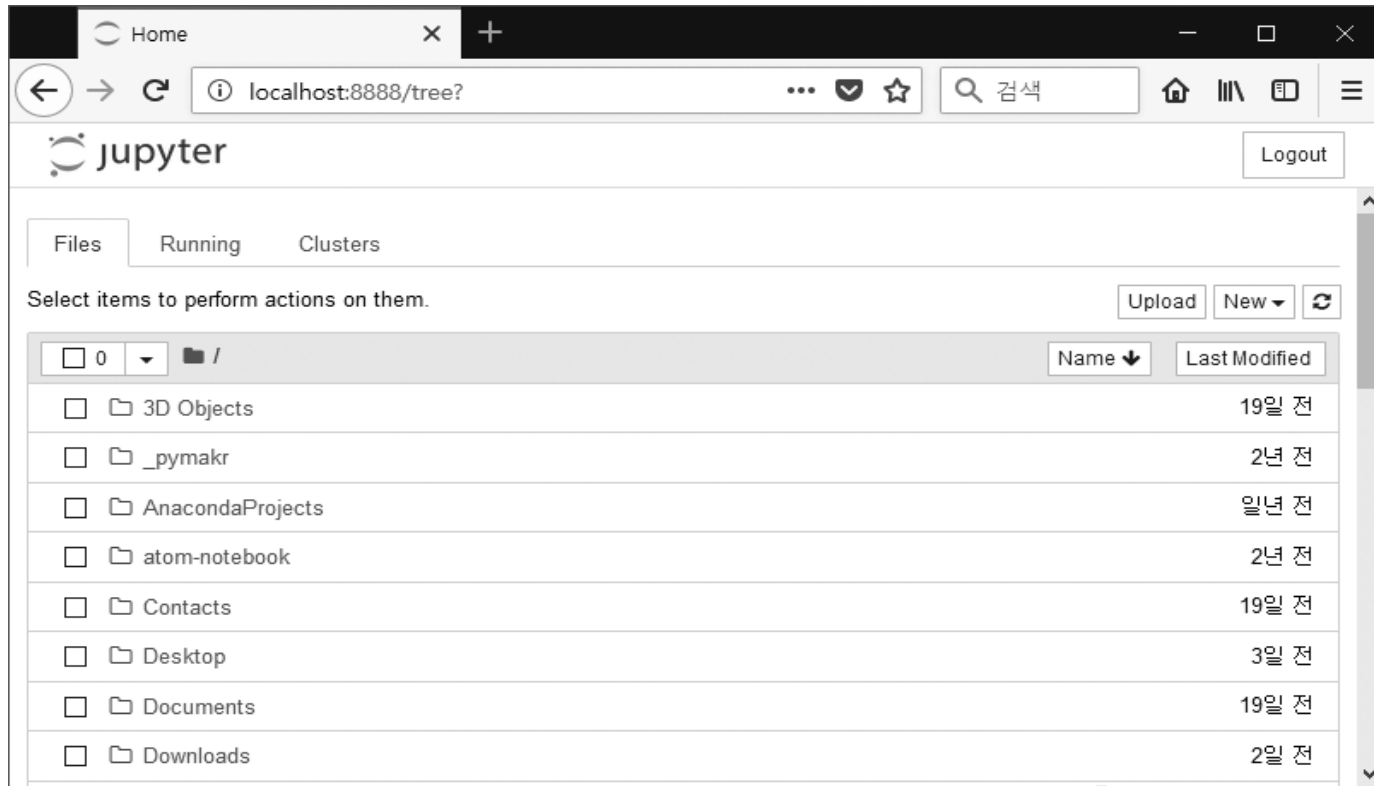
■ 주피터 노트북(Jupyter Notebook)

- 코드 작성 및 실행뿐만 아니라 코드 설명을 위한 문서 작성을 편리하게 할 수 있는 웹 응용 프로그램

■ 주피터 노트북 실행

- 아나콘다 메뉴에서 [Jupyter Notebook]을 클릭해서 실행
- 주피터 서버가 시작되고 기본 브라우저의 새 창에서 주피터 노트북이 열림
- Home이 주피터 노트북의 시작점

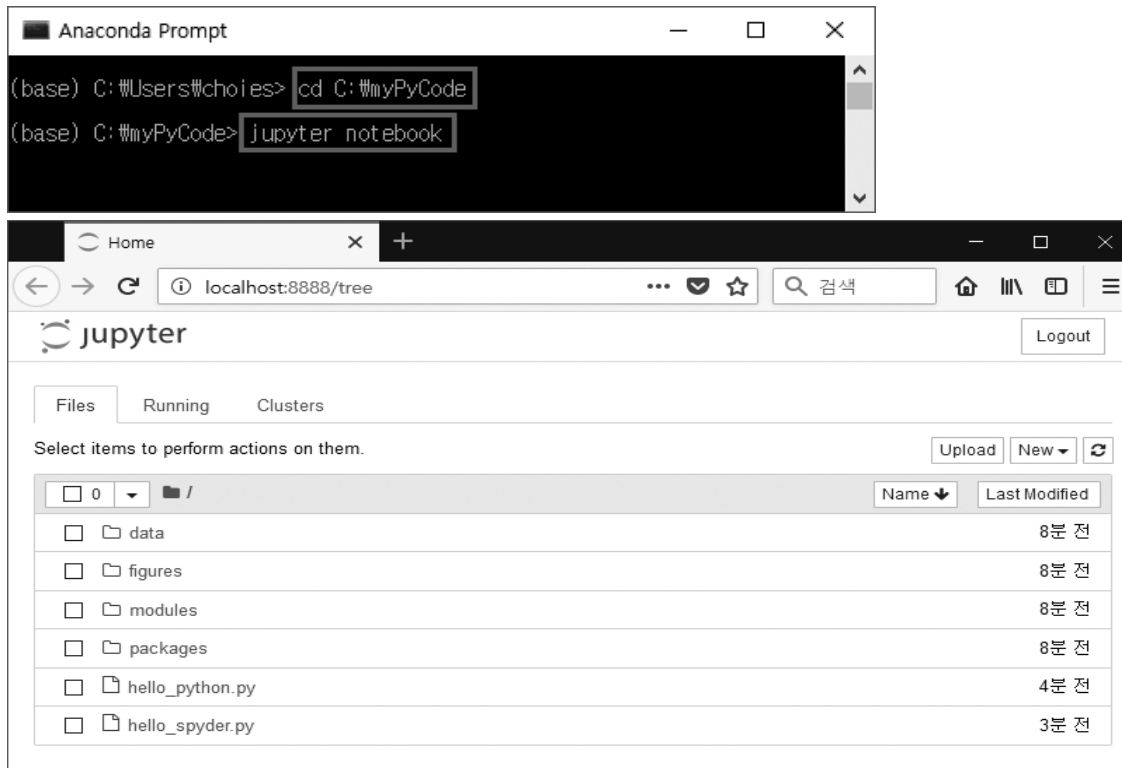
주피터 노트북에서 코딩하기



- 주피터 노트북의 시작 폴더는 'C:\Users\사용자명'
- 주피터 노트북에서 새로운 노트북을 생성하면 이 폴더나 그 아래의 폴더에만 저장 가능

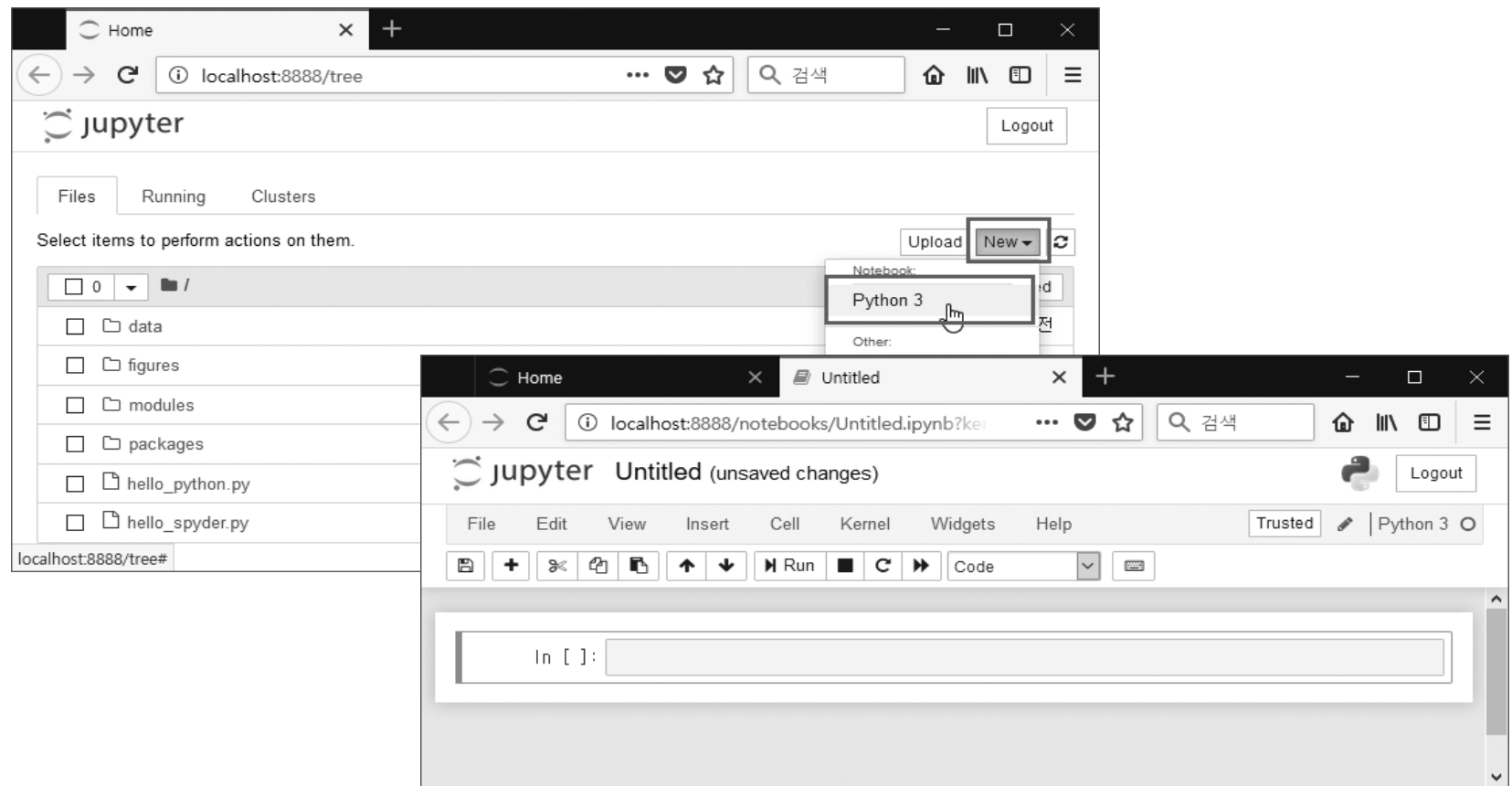
주피터 노트북에서 코딩하기

- 아나콘다 메뉴에서 [Anaconda Prompt]를 클릭
- 다음과 같이 주피터 노트북을 실행
 - 프롬프트에 'cd C:\myPyCode'를 입력해 작업 폴더로 이동
 - 프롬프트에 'jupyter notebook'을 입력해 주피터 노트북 실행



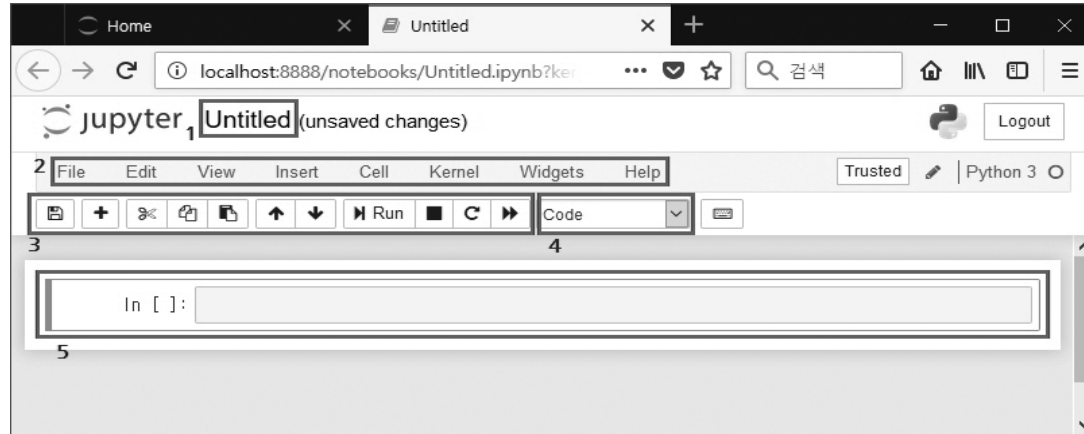
주피터 노트북에서 코딩하기

- 노트북을 생성하면 'C:\WmyPyCode' 폴더에 저장
- 실행된 주피터 노트북의 오른쪽 위에 [New] → [Python 3]를 차례대로 클릭

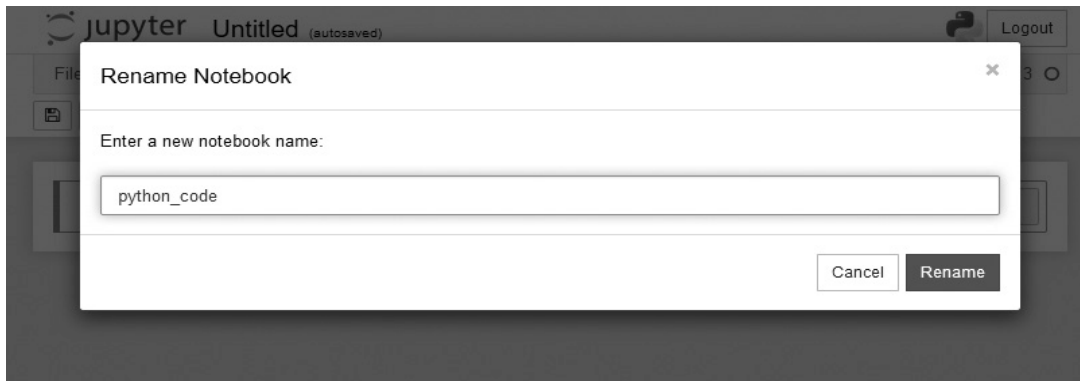


주피터 노트북 사용법

■ 주피터 노트북의 사용자 메뉴



- 노트북 제목(파일 이름): 노트북을 새로 생성하면 임의로 제목이 지정. 노트북 제목을 클릭해 노트북 제목을 변경



주피터 노트북 사용법

- **메뉴 바:** 노트북의 모든 메뉴를 선택
- **툴 바:** 메뉴 바에서 많이 사용하는 기능을 아이콘으로 제공
- **셀 타입**
 - Code: 코드를 작성하기 위한 타입
 - Markdown: 마크다운 형식의 문서를 작성하기 위한 타입
- **셀: 파이썬 코드와 문자를 입력하는 셀**
 - 코드 셀: 셀 앞에 'In []:'이 표시됨. 코드를 입력하고 실행하면 [] 안에는 숫자가 나타나며, 입력한 순서대로 숫자가 1씩 증가
 - 코드 셀의 안쪽에 편집할 수 있는 영역을 편집 영역, 그 밖의 영역을 편집 외 영역이라고 함

■ 편집 모드와 명령 모드

- 편집 모드(Edit mode): 코드나 문서를 작성하기 위한 모드

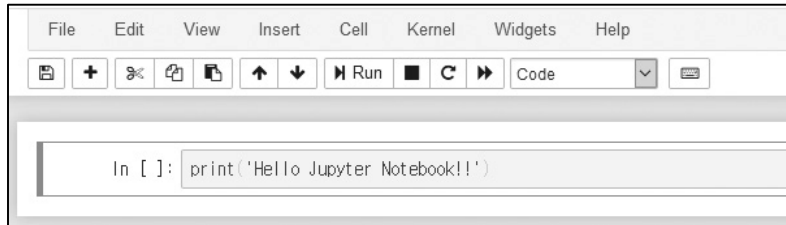
```
In [ ]: print("Jupyter Notebook: Edit mode")
```

- 명령 모드(Command mode): 셀을 다루기 위한 모드

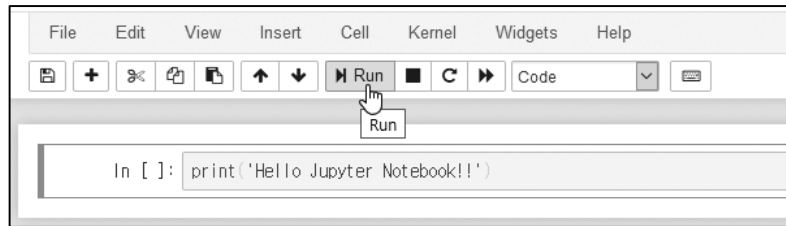
```
In [ ]: print("Jupyter Notebook: Command mode")
```

주피터 노트북에서 코드 작성

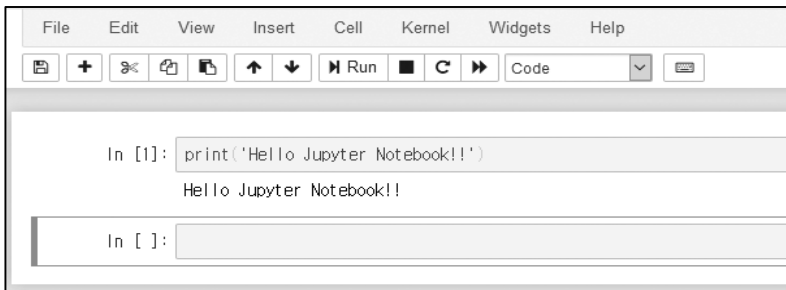
- 생성한 노트북의 첫 번째 셀 편집 영역을 마우스로 클릭해 선택하고 `print('Hello Jupyter Notebook!!')`을 입력



- 셀틀 바에서 [셀 실행] 아이콘을 클릭하거나 Shift + Enter를 누름

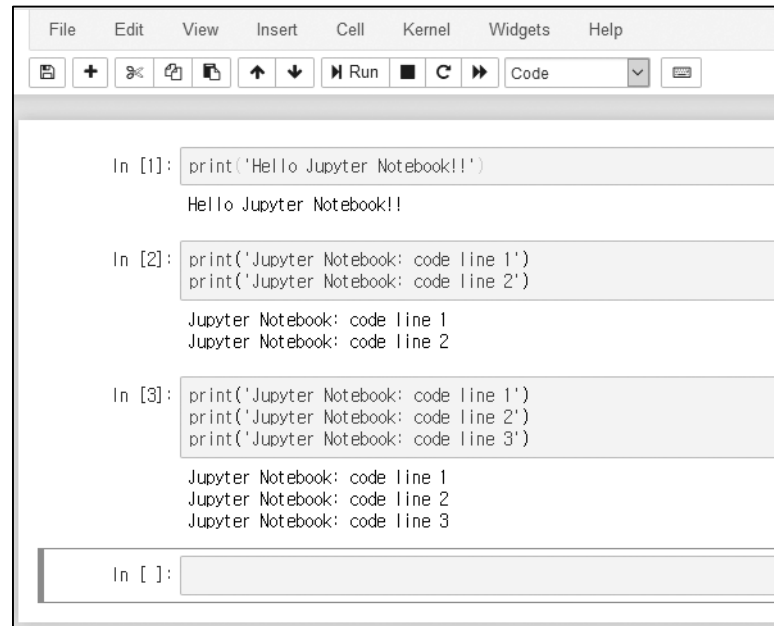


- 코드가 실행되고 그 아래 새로운 코드 셀이 추가됨



주피터 노트북에서 코드 작성

- 코드 셀에는 코드를 여러 줄 입력할 수 있음. 코드 셀에 코드를 여러 줄 입력해 실행하면 위에서부터 순차적으로 코드가 모두 실행됨



The screenshot displays the Jupyter Notebook interface with a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar containing icons for file operations, running, and cell management. The notebook contains three code cells, each with its input and output shown.

```
In [1]: print('Hello Jupyter Notebook!!')
Hello Jupyter Notebook!!
```

```
In [2]: print('Jupyter Notebook: code line 1')
        print('Jupyter Notebook: code line 2')
Jupyter Notebook: code line 1
Jupyter Notebook: code line 2
```

```
In [3]: print('Jupyter Notebook: code line 1')
        print('Jupyter Notebook: code line 2')
        print('Jupyter Notebook: code line 3')
Jupyter Notebook: code line 1
Jupyter Notebook: code line 2
Jupyter Notebook: code line 3
```

At the bottom, there is an empty code cell labeled 'In []:'.