

너는 청년의 때에 너의 창조주를 기억하라 곧 곤고한 날이 이르기 전에, 나는 아무 낙이 없다고 할 해들이 가깝기 전에 (전12:1)



**NOTE:** The following materials have been compiled and adapted from the numerous sources including my own. Please help me to keep this tutorial up-to-date by reporting any issues or questions. Send any comments or criticisms to [idebtor@gmail.com](mailto:idebtor@gmail.com). Your assistances and comments will be appreciated.

## 제 1-2 강: 기계학습 개발환경

### 학습 목표

- 기계학습에 필요한 개발환경에 대해 이해한다
- 각자의 컴퓨터에 기계학습 개발환경을 구축하고, 주피터 노트북 사용법을 익힌다

### 학습 내용

- 왜 파이썬인가?
- 파이썬 라이브러리
- 안나콘다 배포판 *Anaconda distribution*
- 주피터 노트북 *Jupyter notebook* 사용하기
- 마크다운 사용법

## 기계학습 개발환경

기계학습은 자료로부터 지식을 추출하는 작업입니다. 이러한 작업을 위해 여러 지식과 더불어 이를 실현하기 위한 몇가지 도구들이 필요합니다. 이번 시간에는 우리가 앞으로 사용할 기계학습을 개발도구 및 환경에 대해 간단하게 알아보겠습니다.

### 1. 왜 파이썬인가?

파이썬은 간단하고 배우기 쉬운 언어입니다. 불편한 컴파일 과정이 없는 스크립트 언어라서 편합니다. 특히 아이디어를 프로토타입으로 즉시 코딩하여 결과를 볼 수 있습니다. 파이썬은 자료의 저장, 분석, 시각화, 통계, 이미지 처리, 자연어 처리 등에 필요한 라이브러리를 풍부하게 제공합니다. 나아가 딥러닝 프레임워크 즉 카페, 텐서플로, 테아노, 케라스같은 유명 딥러닝 프레임워크에서 파이썬 API를 제공합니다. 그래서, 파이썬은 기계학습 분야에서 독보적인 존재가 되어가고 있습니다. 그래서, 파이썬을 배우면 딥러닝을 공부할 때도 도움이 됩니다. 열심히 파이썬을 지금 배우는 것은 아주 유용할 것입니다.

### 파이썬 버전

파이썬을 컴퓨터에 설치할 때 한 두가지 주의할 점이 있습니다. 현재 파이썬은 2.7 버전과 3.x버전이 동시에 사용되고 있습니다. 버전 3이 나왔지만, 아직도 2.7버전이 사용되는 이유는 두 버전이

서로 호환되지 않기 때문입니다. 우리는 3.x를 사용합니다.

파이썬을 처음 배울 때 가볼만한 곳

- [파이썬 코딩을 연습하면서 배우는 튜토리얼](#)
- [파이썬 공식 튜토리얼\(참조용\)](#)

## 3. 파이썬 라이브러리

강의 제목에서 이미 나타났듯이 우리는 기계학습을 파이썬으로 손코딩으로 구현하는 것이 목표입니다. 우리의 목표를 비행기에 비유한다면, 비행기 운항을 배우려는 것이 아니라 비행기의 원리를 이해하고 비행기를 직접 만들어 보고 날아보자는 것입니다.

### 3.1 파이썬 표준 라이브러리

우리가 비행기를 제작을 해도 기본적인 장비, 재료, 부속품은 공급이 된다고 가정합니다. 마찬가지로, 우리도 기계학습에 필요한 파이썬 라이브러리를 사용할 것입니다. 프로그램 개발자들이 다른 프로그램에 포함할 수 있도록 다양한 프로그램들을 미리 만들어 묶어 둔 것을 **라이브러리**라고 합니다

다만, 이미 개발된 기계학습을 위한 라이브러리는 가능한 사용하지 않지만, 파이썬의 표준 라이브러리(math, random, pickle, csv, os, time, urllib, 등등)와 몇 개의 외부 라이브러리를 사용할 것입니다.

### 3.2 파이썬 외부(제3자) 라이브러리

외부 라이브러리들 중에서 특별히 다음 두 라이브러리는 많이 사용할 것입니다.

- [numpy\(넘파이\)](#)
- [matplotlib\(맷플롯립\)](#)

**넘파이**는 수치 계산용 라이브러리입니다. 넘파이는 다양한 수학 알고리즘과 행렬(배열)을 생성하고 다루는 많은 클래스와 함수들이 있습니다.. 자료들을 다차원 배열로 저장하여 다루는 기능과 선형 대수 연산, 변환 등과 같은 수준 높은 함수들을 포함합니다. 기계학습에서 사용할 자료는 모두 NumPy배열로 변환해서 사용합니다. 넘파이가 없이 기계학습이나 딥러닝을 구현하는 것이 상당히 가질 않을 정도입니다. 넘파이를 위한 특강 시간도 곧 가질 예정입니다.

- [넘파이 튜토리얼\(참조용\)](#)
- [기계학습을 위한 넘파이 튜토리얼\(한글\)](#)

**Matplotlib**은 파이썬에서 사용하고 있는 과학 계산용 그래프 라이브러리입니다. 자료 분석을 할 때 다양한 관점에서 분석 결과를 시각화를 하면 자료에 대한 중요한 통찰력을 얻게 되므로, 기계학습에서 시각화 과정은 필수적입니다. 맷플롯립은 선 그래프, 히스토그램, 산점도scatter plot 등을 고품질로 그려주는 그래프 함수들을 가지고 있습니다.

기계학습을 위해 이미 개발된 외부 라이브러리는 우리가 가능하면 사용하지 않겠지만, 그러나 필요하다면 언제든지 사용할줄 알아야 합니다. 만약 외부 라이브러리도 사용할 필요가 있으면 놀라지 말고, 반갑게 사용하길 바랍니다.

### 3.3 기계학습을 위한 외부(제3자) 라이브러리

기계학습에 관심이 있는 우리가 사용할지도 모르지만, 아무튼 우리가 사용할 줄 알면 도움이 되는 라이브러리입니다.

### 판다스 *Pandas*

*Pandas*는 자료 처리와 분석을 위한 파이썬 라이브러리입니다. *DataFrame*이라는 자료 구조를 기반으로 만들어졌으며, 엑셀의 스프레드시트와 비슷한 테이블 형태라고 생각하면 됩니다. *pandas*는 이 테이블을 수정하고 조작하는 다양한 기능을 제공합니다. 넘파이 배열은 동일한 형식(타입)의 자료만 각 열에 저장할 수 있지만, *pandas DataFrame*에는 다른 형식의 자료도 저장할 수 있습니다.

### 싸이파이 *SciPy*

*SciPy*는 Scientific Python으로 다양한 과학적 계산을 하기 위한 파이썬 모듈입니다. 자료들을 다차원 배열로 저장하여 다루는 기능과 선형 대수 연산, 변환 등과 같은 수준 높은 함수들을 포함합니다. 기계학습에서 사용할 자료는 모두 넘파이 배열로 변환해서 사용합니다. *SciPy*뿐만 아니라 *NumPy*와 Python 언어의 기초에 대한 간략한 소개는 *SciPy* 강의 노트 (<http://www.scipy-lectures.org>)가 탁월합니다.

### 싸이킷런 *Scikit-learn*

*Scikit-learn*은 기계학습에서 가장 많이 사용하는 파이썬 오픈소스 라이브러리 중에 하나입니다. 다양한 기계학습 모델뿐만 아니라 기계학습에 필요한 다양한 알고리즘들도 제공합니다. 이러한 모델이나 알고리즘들은 학계에서 뿐만 아니라 산업 현장에서도 많이 사용되고 있으나 본 강의에서는 다루지 않습니다. *scikit-learn*은 Python의 *NumPy*와 *SciPy*를 사용하여 구현되었습니다.

## 3.4 기계학습 개발 프레임워크

기계학습 개발 프레임워크는 기계학습(딥러닝)에서 어려운 부분들을 추상화함으로써 사용자들 혹은 어플리케이션 개발자들이 기계학습을 좀 더 쉽게, 빠르게 사용할 수 있도록 해줍니다. 기계학습의 자세한 구현 부분을 실제로 코딩하지 않아도 기계학습의 기능을 활용하여 새로운 응용할 수 있도록 합니다.

### 텐서플로 *TensorFlow*

텐서플로 *TensorFlow*는 구글이 만든 2세대 기계학습과 심화학습 오픈소스 라이브러리 시스템입니다. 데이터 플로우 그래프 *Data Flow Graph* 방식을 사용하였으며, 핵심 기술은 C++로 작성되었지만, 프론트 엔드 부분은 파이썬으로 작성되었기 때문에, 사용자들은 주로 파이썬을 사용하여 기계학습(딥러닝) 알고리즘을 개발합니다. 이미지, 음성, 비디오 등 다양하고 많은 자료를 처리할 수 있으며, GPU(*Graphics Processing Unit*)와 같은 하드웨어를 사용합니다.

참고로, 수학에서 0차원의 수를 스칼라 *scalar*, 1차원 배열을 벡터 *vector*, 2차원 배열을 행렬 (*matrix*)라고 부릅니다. 이와 같이 스칼라, 벡터와 행렬을 일반화한 것을 텐서 (*tensor*)라고 합니다.

### Keras

*TensorFlow*를 백엔드(*back-end*)로 사용하는 기계학습(딥러닝) 라이브러리입니다. 이것은 현존하는 최고의 파이썬 API라고 전문가들이 말하고 있습니다. 케라스는 Python 기반으로 개발되었으며 *TensorFlow*와 CNTK 위에서 실행됩니다. 비전문가도 쉽게 기계학습 모델을 개발할 수 있도록 직관적인 API 제공하는 것으로 유명합니다. Keras는, 태생적으로, 텐서플로와 CNTK 딥러닝 엔진 위

에서 구동되니까, 텐서플로나 CNTK로 할 수 있는 것을 대부분 할 수 있습니다. Keras는 최소주의, 소위 minimalism을 자랑하는 것이 특징이죠. 무엇이든지 간결하게 하려고 합니다. 이 멋진 케라스를 개발하고 유지보수하고 있는 사람은 프랑소와 쥘레(François Chollet)라는 구글의 엔지니어입니다.

## PyTorch

파이토치는 텐서플로와 케라스에 비해 후발주자이지만, 사용하기 쉽고 구현이 간단하여 상당히 인기를 얻고 있습니다. 다른 것과 마찬가지로 파이토치 역시 기존의 파이썬 라이브러리와 호환성이 아주 좋습니다. 또한, 반가운 소식은 파이토치에는 파이썬 넘파이와 아주 유사한 Tensor가 있어서, 이번 강의를 통해 파이썬 넘파이 사용에 익숙하게 될 여러분이 아주 쉽게 사용할 수 있을 것입니다. 파이토치는 페이스북에서 개발되었습니다.

## 3.5 개인 모듈 - joy.py

본 강의 위해 강의자가 개발한 joy 모듈을 사용합니다. 물론, 소스코드는 모두 여러분에게 공개되어 제공될 것입니다.

## 3.6 주피터 노트북 (IDE 도구)

IPython(Interactive Python, 대화형 파이썬)의 장점을 모두 물려받은 주피터 노트북(Jupyter Notebook)은 웹 브라우저에서 파이썬 코딩과 문서 작성을 같이 할 수 있는 최고의 파이썬 개발환경(IDE, Interactive Development Environment)입니다.

결론적으로 우리가 기계학습 프로그램을 개발하면서 항상 사용할 도구들은 다음과 같습니다.

- 파이썬 3.x
- 넘파이
- 맷플롯립
- joy.py 모듈
- 주피터 노트북

## 4. 아나콘다 배포판

우리가 사용할 기계학습 개발환경을 구축을 하기 위한 다양한 방법이 있겠지만, 아나콘다 배포판을 설치하는 것을 권장합니다.

배포판이란 사용자가 소프트웨어를 한 묶음으로, 한꺼번에 설치할 수 있도록 필요한 모든 것을 하나로 정리해둔 것입니다. 특히 아나콘다는 데이터 과학에서 가장 많이 사용되는 Python, R 및 Scala 패키지 중 100개 이상을 포함하고 있습니다. 우리가 항상 사용할 Jupyter notebook도 포함하고 있습니다. 우리가 사용하려고 하는 파이썬, 넘파이, matplotlib 모듈도 포함하고 있습니다. 한번 설치로 기본적인 기계학습을 시작할 수 있습니다.

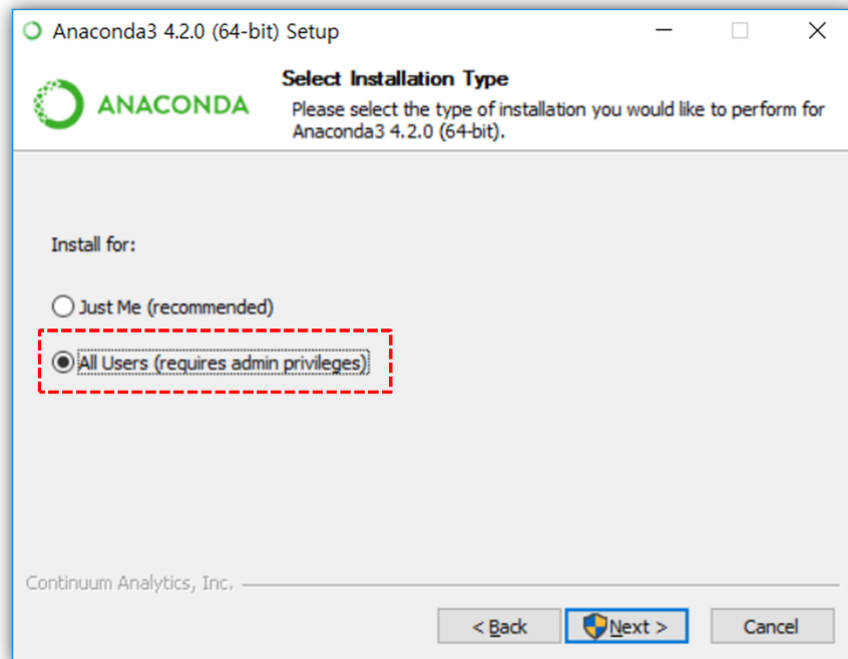
이를 위해 먼저 **Anaconda** 배포판을 설치하는 방법에 대해 알아보도록 하겠습니다. 다음 주소에서 자신의 OS에 맞는 배포판을 내려받아 설치하십시오. 설치하기 전에 다음 주의 사항을 살펴보세요.

### 아나콘다 내려받기

먼저 [아나콘다 내려받기 사이트](#)로 접속합니다.

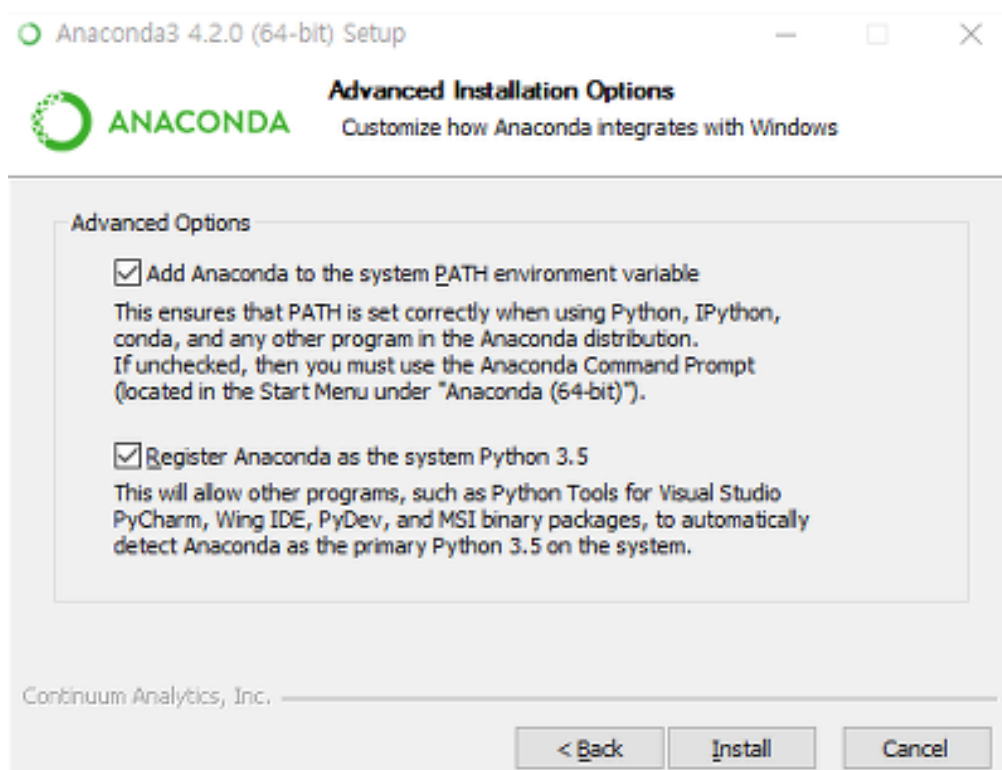
## 4.1 Window에서 설치하기

우리는 3.X버전 64비트를 사용할 것입니다. 다운로드를 누르고 실행파일을 실행 시킵니다.



### 4.1 설치 옵션 - 권장

설치파일을 실행하면 개인용으로 사용할 것인지 공용으로 사용할 것인지 고르는 화면이 나옵니다. 위와 같이 공용으로 체크하고 다음으로 넘어갑니다. 설치 프로그램에서는 개인용으로 설치할 것을 권장하고 있지만 개인용으로 할 경우 설치 경로가 USER 폴더로 되는데 이 경우 USER폴더가 한글인 경우 오류가 나며 해당 경로에 설치가 되지 않습니다. 그러므로 개인용으로 설치를 하고 싶은 경우 설치 위치를 변경해주면 됩니다.



그 다음으로 설치가 되고나면 마지막으로 옵션으로 환경변수와 가상환경 설정을 선택하게 되어 있습니다. 2개의 체크박스 모두 체크하는 것을 권장합니다.

#### 설치 옵션 노트

체크박스를 체크하지 않는다면 설치자가 직접 환경변수를 등록해주어야 하며 커맨드창 혹은 터미널에서 가상환경을 생성해주어야 합니다. 혹시 가상환경에 체크를 하지 않았다면 아래의 방법을 통하여 가상환경을 생성하면 됩니다.

## 4.2 설치 옵션 - 가상환경 생성하기

가상환경은 다른 말로 독립적 환경이라고 합니다. 다른 프로젝트 혹은 다른 버전과 충돌하는 것을 피하기 위해 독립적 환경이 필요할 경우 이런 개발환경이 필요합니다. 독립된 개발 환경을 선호한다면 다음 pip 명령으로 virtualenv를 설치하십시오.

콘솔 창에서 다음과 같은 명령들을 실행함으로 가상환경 즉 독립적 환경을 만드십시오.

**pip3 install --user --upgrade virtualenv**

**conda create -n <환경명> python=<버전(ex:3.5이나 2.7 등)>**

가상환경을 생성하는 명령어입니다. 여기서 환경명은 마음대로 지어도 됩니다. 하지만 버전에서는 3.5로 생성하기를 바랍니다.

**conda env list**

만들어져 있는 가상환경의 목록을 보여주는 명령어입니다. 위의 명령어를 입력하여 가상환경이 제대로 만들어졌는지 확인해보아야 합니다.

**activate <환경명>**

원하는 개발환경을 실행시키는 명령어입니다. 위의 명령어를 입력하여 만든 가상환경을 실행시킵니다. 혹시 위의 명령어가 작동하지 않는다면 아래의 명령어를 입력하기 바랍니다.

**source activate <환경명>**

**conda env**

이 명령어를 입력하여 잘 작동하는지 확인해보면 됩니다. 아래의 사진처럼 나오는 것이 정상적으로 작동이 되는 것입니다.

```
choijin-young ~ -bash — 102x35
Last login: Mon May 28 23:58:18 on ttys000
choejin-yeong-ui-MacBook-Pro:~ choijin-young$ conda env
usage: conda-env [-h] {attach,create,export,list,remove,upload,update} ...

positional arguments:
  {attach,create,export,list,remove,upload,update}
    attach              WARNING: This command is deprecated in conda 4.4 and
                        scheduled for removal in conda 4.5. Embeds information
                        describing your conda environment into the notebook
                        metadata
    create              Create an environment based on an environment file
    export              Export a given environment
    list                List the Conda environments
    remove              Remove an environment
    upload              WARNING: This command is deprecated in conda 4.4 and
                        scheduled for removal in conda 4.5. Upload an
                        environment to anaconda.org
    update              Update the current environment based on environment
                        file

optional arguments:
  -h, --help            Show this help message and exit.

conda commands available from other packages:
  build
  convert
  develop
  env
  index
  inspect
  metapackage
  render
  server
  skeleton
  verify
```

## deactivate <환경명>

원하는 가상환경의 실행을 중지시키는 명령어입니다. 다른 가상환경을 실행시키고 싶다면 기존의 사용하던 가상환경을 위의 명령어로 중지시킨뒤 다른 가상환경을 실행시키면 됩니다. 굳이 사용할 필요는 없습니다.

## 4.3 python 및 pip 버전 확인하기

### python --version

위의 명령어를 입력하여 파이썬의 버전을 확인합니다.

### pip --version

위의 명령어를 입력하여 pip의 버전을 확인합니다.

```
명령 프롬프트
Microsoft Windows [Version 10.0.17134.112]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\₩₩₩ >pip --version
pip 9.0.1 from C:\ProgramData\Anaconda3\lib\site-packages (python 3.6)

C:\Users\₩₩₩ >python --version
Python 3.6.4 :: Anaconda, Inc.
```

명령어를 입력했을때 위의 사진처럼 나온다면 제대로 설치가 된 것입니다. 이 단계까지 왔다면 여러분은 이제 Anaconda를 사용하여 기계학습을 시작할 수 있는 환경이 준비된 것입니다.

여러분이 아나콘다를 설치하였다면, IPython/Jupyter는 이미 설치되었을 것입니다. Jupyter Notebook은 전통적인 IPython을 대체하는 새로운 IPython 개발환경이지만, IPython이라고 부르기도 합니다

그렇다면 Anaconda에서 제공하는 **Jupyter Notebook** 을 사용해보도록 하겠습니다.

## 5. Jupyter Notebook 사용하기

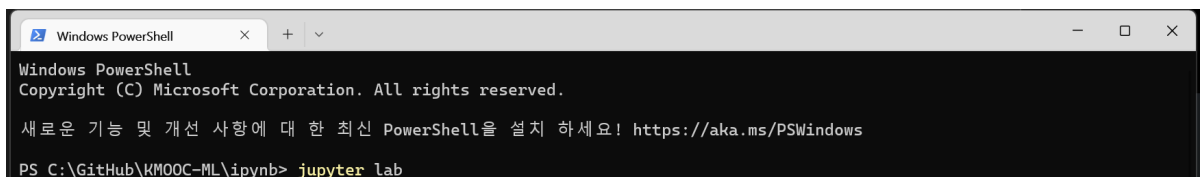
주피터 노트북은 웹 브라우저에서 파이썬, R 등등 40여 가지의 다양한 프로그램 언어들을 지원하며, 코드와 문서를 같이 작성할 수 있는 개발 환경이며 서버-클라이언트 애플리케이션입니다. 인터넷 접근없이 로컬에서 실행될 수 있을 뿐만 아니라 인터넷을 통해서 원격 서버에 설치될 수 있고 접근 가능합니다.

주피터 노트북에서 문서 작성은 대개 마크업(mark-up)이 아니라 마크다운(mark-down) 언어로 작성합니다.

처음에는 Ipython notebook 이라는 이름으로 파이썬 언어만 지원했지만, 다양한 언어를 지원함에 따라 이름을 주피터 노트북이란 이름을 갖게 되었습니다.

### 5.1 시작하기

먼저 명령 프롬프트 창 혹은 터미널을 실행시킵니다.



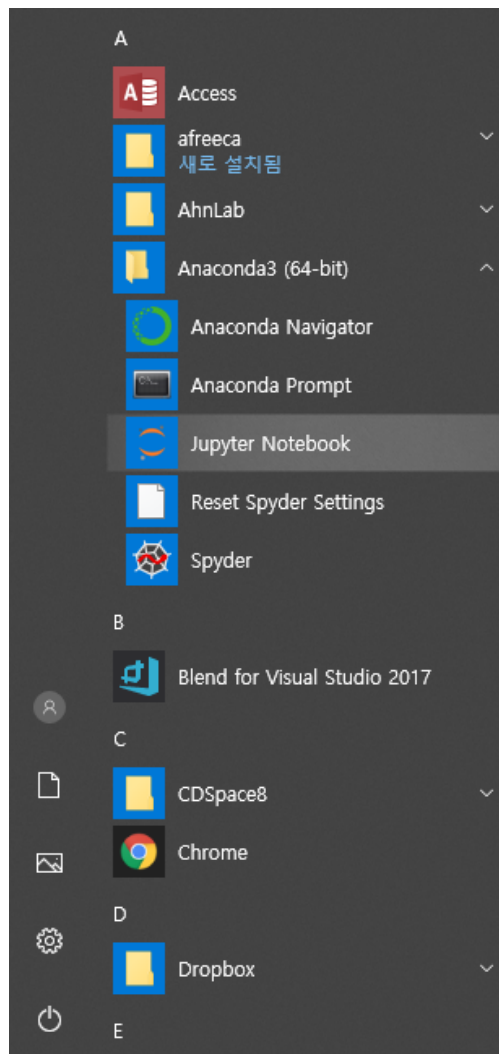
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

새로운 기능 및 개선 사항에 대한 최신 PowerShell을 설치하세요! https://aka.ms/PSWindows

PS C:\GitHub\KM00C-ML\ipynb> jupyter lab
```

위의 사진처럼 명령어를 입력합니다. 혹은 다음과 같이 컴퓨터에서 jupyter notebook 실행파일을 찾아서 직접 실행해주어도 됩니다.



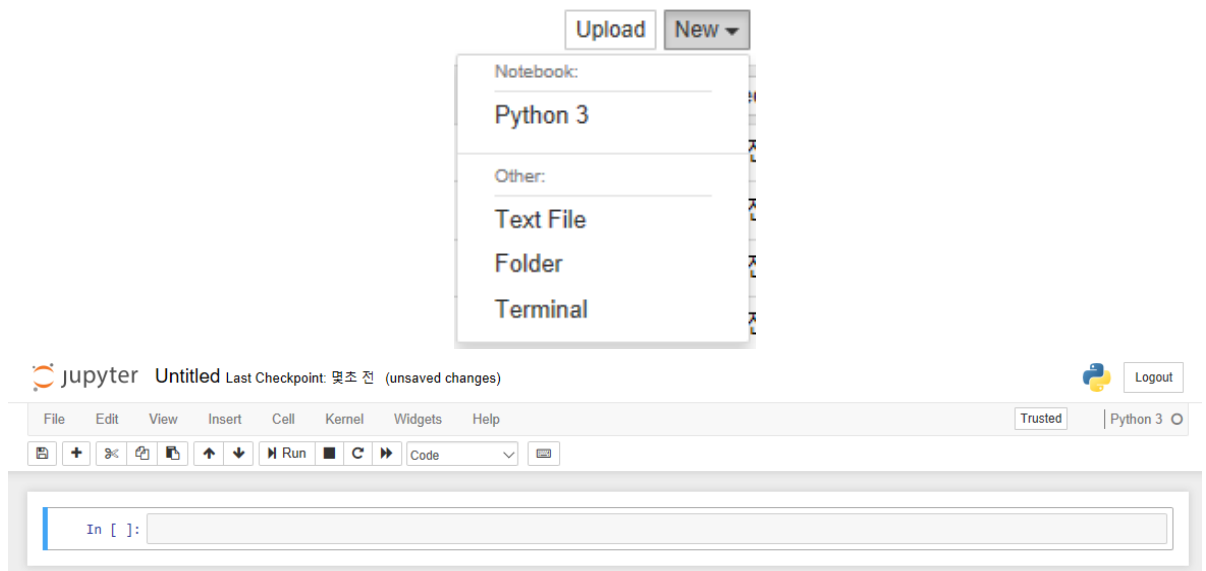


## 5.2 대쉬보드

jupyter notebook을 실행시키고 나면 기본적으로 아래의 그림처럼 사용자폴더에 있는 폴더와 파일들을 보여줍니다. 이 화면을 **대쉬보드** 라고 합니다. Window의 File Explorer라고 생각하면 됩니다. 그러므로, 현재의 폴더 위치에서 다른 폴더로 이동이 가능합니다.



새로운 파일을 만들고 싶다면 오른쪽 상단에 있는 *New*를 누르고 *python3*를 눌러 notebook파일을 만듭니다.



이 화면은 대화형 주피터 노트북입니다.

지금 주피터 노트북은 여러분의 명령을 기다리고 있는 상태입니다. 여러분이 명령을 입력하면, 노트북이 이를 수행해서 결과를 돌려줍니다. 그리고 다시 여러분의 명령을 기다립니다.

약간 복잡한 작업을 하고 싶다면 구역을 나누는 것이 좋습니다. 이렇게 하면 여러분의 생각을 쉽게 체계화 할 수 있고 큰 프로젝트의 경우 어느 부분이 잘못되었는지 쉽게 찾을 수 있습니다. Jupyter Notebook에서는 이 구역들을 셸이라고 부릅니다. 위의 Jupyter Notebook에는 시작할 때 빈 셸이 있는데, 커서가 깜박이면서 여러분의 명령을 입력할 때까지 기다리는 것을 볼 수 있습니다.

## 5.3 코딩 시작

코드 셸에서 다음과 같이 코딩을 하십시오.

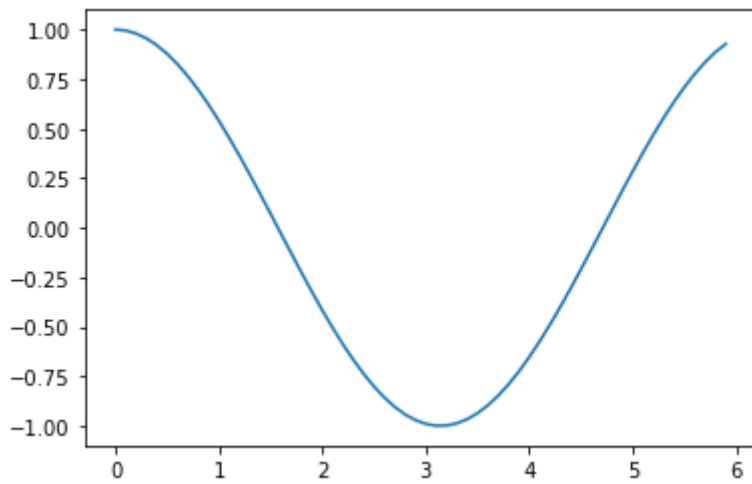
In [1]: `1 + 2 + 3`

Out[1]: 6

In [2]: 

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

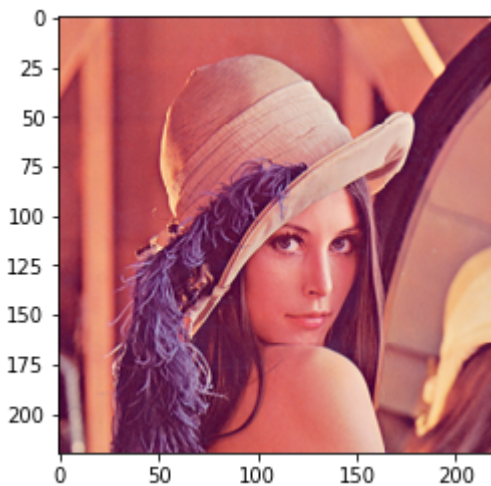
x = np.arange(0, 6, 0.1)
plt.plot(x, np.cos(x))
plt.show()
```



In [3]:

```
import matplotlib.pyplot as plt
from matplotlib.image import imread

img = imread('https://github.com/idebtor/KM00C-ML/blob/master/ipynb/images/lenna.png?r=1')
plt.imshow(img)
plt.show()
```

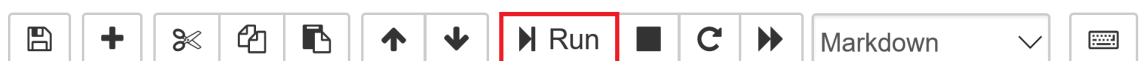


In [ ]:

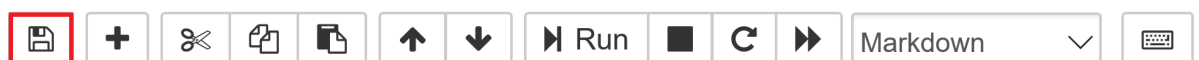
## 셀 실행

셀을 실행하는 세 가지 방법이 있습니다.

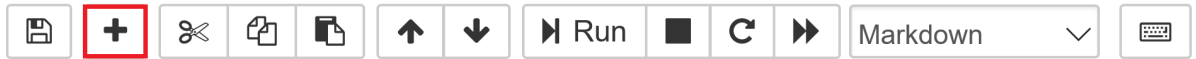
1. 셀 안에서 Ctrl + Enter를 누르면 해당 셀을 실행합니다.
2. Shift + Enter를 누르면 해당 셀을 실행하며 다음 셀로 이동합니다. 해당 셀이 마지막 셀이라면 새로운 셀을 생성합니다.
3. 다음 그림의 버튼을 누릅니다.



## 셀 저장

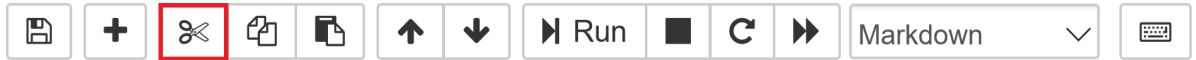


해당 버튼을 누르면 작성한 셀들을 저장할 수 있습니다. 항상 파일을 닫기 전 저장을 하시길 바랍니다. ##### 셀 추가



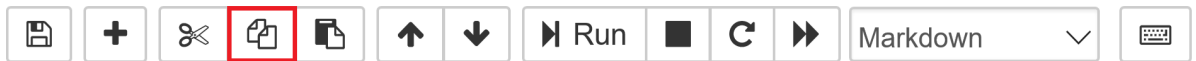
해당 버튼을 누르면 셀을 추가할 수 있습니다.

### 셀 잘라내기



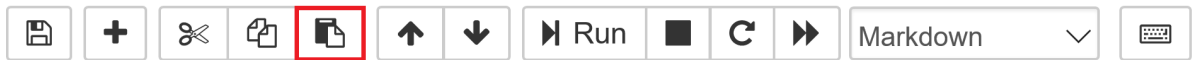
해당 버튼을 누르면 선택한 셀을 잘라낼 수 있습니다.

### 셀 복사하기



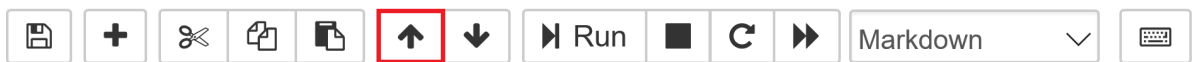
해당 버튼을 누르면 선택한 셀을 복사할 수 있습니다.

### 셀 붙여넣기



해당 버튼을 누르면 복사한 셀을 붙여넣을 수 있습니다.

### 셀 올리기



해당 버튼을 누르면 선택한 셀을 위로 올릴 수 있습니다.

### 셀 내리기



해당 버튼을 누르면 선택한 셀을 아래로 내릴 수 있습니다.

### 셀 실행중지 하기



해당 버튼을 누르면 실행하고 있는 셀을 중지시킬 수 있습니다.

### 셀 다시 실행하기

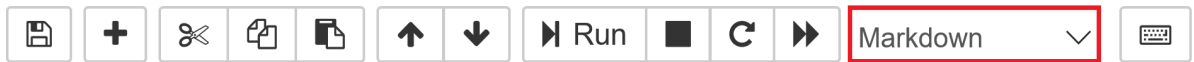


해당 버튼을 누르면 선택한 셀을 다시 실행시킵니다.

### 전체 셀 다시 실행하기



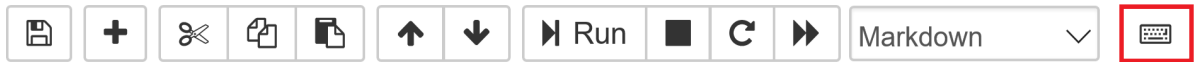
해당버튼을 누르면 전체 셀을 처음부터 다시 실행합니다. ##### 셀 타입 정하기



해당 버튼을 누르면 아래의 그림과 같이 셀의 타입을 정할 수 있습니다.

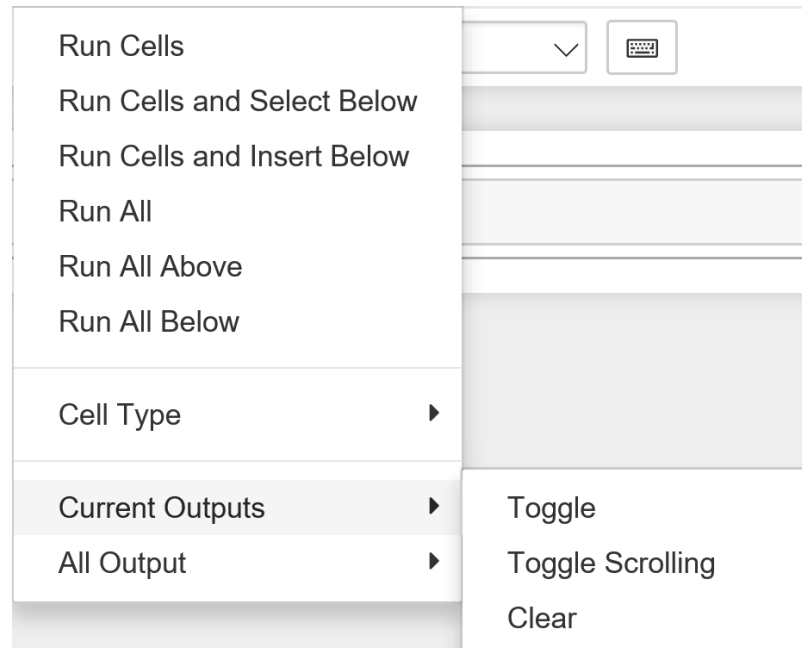


여러분들은 주로 `_code_`와 `_markdown_` 을 사용하게 될 것입니다. `####` 커맨드 파レット



해당 버튼을 누르면 셀을 어떻게 할 것인지 명령을 검색할 수 있습니다.

## 셀 결과 초기화하기



Cell → Current Output → Clear를 통해 셀의 결과를 초기화 할 수 있습니다.

## 간단한 코드 실행

기본적으로는 코드를 실행할 수 있는 셀로 되어있지만 다른 기능을 하는 셀로 변경할 수도 있습니다. 한 번 기본적인 코드를 실행시켜보도록 하겠습니다.

```
In [1]: num1 = 3
        num2 = 7
        print(num1 + num2)

10
```

여러분은 컴퓨터가 "10"이라는 올바른 답을 하는 것을 볼 수 있습니다. 우리는 컴퓨터에게 처음으로 명령을 했었고 그 명령에 대하여 올바르게 대답하는 것을 확인했습니다.

Jupyter Notebook에서 보면 "In[숫자]"와 "Out[숫자]"가 있을 것입니다. 이것은 크게 신경쓰지 않아도 됩니다. 이 기호는 여러분이 몇번 입력을 했고 그에 대한 실행을 몇 번하였는지 보여주는 방식 일 뿐입니다. 이 숫자들은 여러분의 입력과 실행에 대해 순서를 나타내는 숫자이며 여러분이 노트북의 해당 셀을 실행했는지 안했는지 확인 할수 있는 숫자입니다.

숫자대신 \*가 표시되어 있다면, 해당 셀을 현재 실행하고 있다는 표시입니다.

## 6. 마크다운 언어

마크다운 언어의 사용법에 대해 간략하게 알아보도록 하겠습니다. 먼저 마크다운이란 마크업 언어의 일종으로 존 그루버와 아론 스워츠가 만든 언어입니다. 여러분들이 흔히 볼 수 있는 예로는 위키피디아를 들 수 있습니다. 기본적으로는 텍스트파일 처럼 사용하면 됩니다. 하지만 텍스트파일 처럼만 사용할 것이라면 마크다운 언어가 존재할 필요가 없겠지요. 마크다운 언어의 유용한 문법들을 설명하겠습니다.

## 6.1 제목 달기

먼저 제목<sup>heading</sup>을 사용할 때는 #을 붙여주면 됩니다.

/# 이것은 제목입니다

이것은 제목입니다

의 갯수가 많아질 수록 제목의 크기도 줄어듭니다. #은 최대 6개까지 쓸 수 있습니다.

## 이것은 작은 제목입니다

이것은 작은 제목입니다

### 이것은 더 작은 제목입니다

이것은 더 작은 제목입니다

#### 이것은 가장 작은 제목입니다

이것은 가장 작은 제목입니다

##### 이것은 가장 작고 기울어진 제목입니다

이것은 가장 작고 기울어진 제목입니다

#####이것은 위의 것과 차이가 없는 제목입니다

이것은 위의 것과 차이가 없는 제목입니다

## 6.2 글씨체

- 볼드체를 쓰고 싶을 때는 글의 시작과 끝에 \_ (언더바 2개)를 앞뒤로 붙여주면 됩니다.

\_볼드체\_

**볼드체**

이텔릭체를 쓰고 싶을 때는 글의 시작과 끝에 \* 혹은 \_ 를 앞뒤로 붙여주면 됩니다.

\*이텔릭체\*

*이텔릭체*

이텔릭체  
*이텔릭체*

취소선을 긋고 싶을 때는 ~~ 를 앞뒤로 붙여주면 됩니다.  
~~이 문장을 취소합니다~~

## 6.3 사진 첨부하기

사진을 첨부하고 싶을 때는 에서 \만 빼주면 됩니다.

## 6.4 줄 바꾸기 및 문단 나누기

문단을 나누고 싶을 때에는 엔터를 두번 누르면 됩니다.

줄을 바꾸고 싶을 때에는 <br />에서 \만 빼주면 됩니다.  
이미 줄은  
바뀌어 있습니다.

## 6.5 블럭 만들기

블럭을 만들고 싶을 때는 블럭 인용문자 > 을 사용하면 됩니다

여기가 블럭입니다

## 6.6 목록 만들기

### 6.6.1 순서있는 목록

순서 있는 목록은 숫자와 점을 사용합니다.

1. 검은돌
2. 흰돌
3. 이세돌

1. 검은돌
2. 흰돌
3. 이세돌

숫자의 순서는 상관없습니다.

1. 검은돌
4. 흰돌
2. 이세돌

1. 검은돌
2. 흰돌
3. 이세돌

### 6.6.2 순서없는 목록

순서없는 목록은 +, \*, -를 사용합니다  
+ 즐거운

- + 기계
  - + 학습

- 즐거운
  - 기계
    - 학습

- \* 즐거운
  - \* 기계
    - \* 학습

- 즐거운
  - 기계
    - 학습

- 즐거운
  - 기계
    - 학습

- 즐거운
  - 기계
    - 학습

## 6.7 수평선 만들기

\* \* \*

\*\*\*

위의 것들은 모두 수평선을 만들어 냅니다.

---

## 7. 유용한 셀 명령어들

셀 편집상태에서 **Esc** 또는 **Ctrl + M** 을 누르면 커맨드 모드로 진입을 하게 됩니다.

- '**m**' 을 누르면 셀 타입을 마크다운으로 바꿀 수 있습니다.
- '**y**' 을 누르면 셀 타입을 코드로 바꿀 수 있습니다.
- '**dd**' 을 누르면 해당 셀을 지울 수 있습니다.
- '**x**' 을 누르면 해당 셀을 잘라낼 수 있습니다.
- '**c**' 을 누르면 해당 셀을 복사할 수 있습니다.
- '**s**' 을 누르면 저장합니다.
- '**Shift + m**' 을 누르면 아래의 셀과 합쳐집니다.
- '**Enter**' 을 누르면 편집 모드로 진입을 합니다.
- '**I**' 로 셀의 모든 라인에 번호매김을 하거나 숨길 수 있습니다.

셀의 타입이 코드일때,

- '**!**' 뒤에는 명령 프롬프트 창이나 콘솔에서 쓸 수 있는 명령어들을 사용할 수 있습니다.

## 학습 정리



- 기계학습 개발환경 이해
  - 기계학습 개발환경 구축
  - 파이썬과 주피터 노트북 사용법
- 

In [ ]: