

# Python 개발환경 구축

---

Wooju Lee (krosea@naver.com)

2021-03-30

@ IPSL

# Contents

---

- Anaconda
- Pycharm
- Python

# ANACONDA

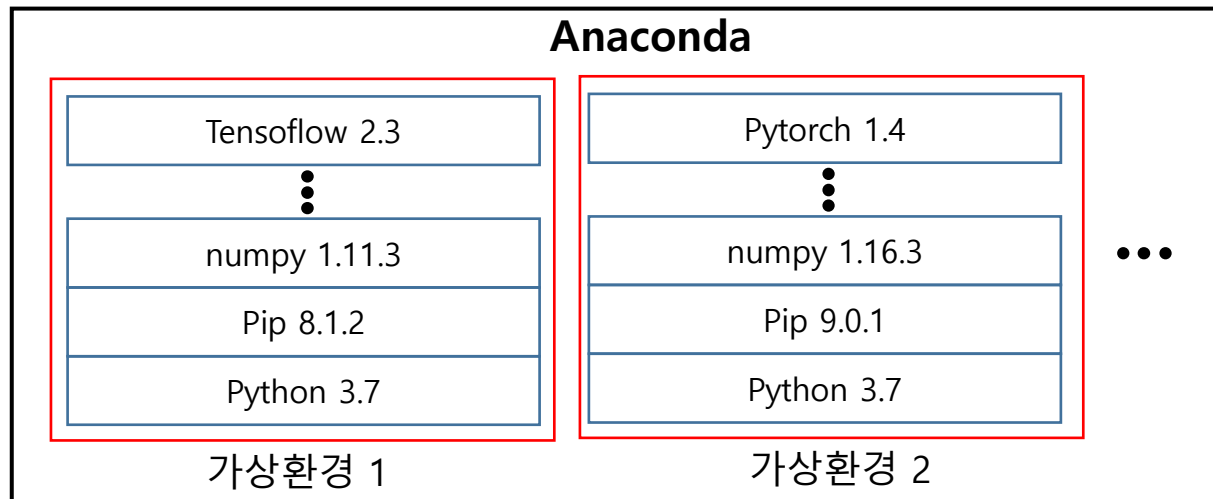
---

# Anaconda

## ■ Anaconda

- 수학과 과학 분야에서 사용되는 여러 패키지들을 묶어 놓은 패키지 및 환경 관리 소프트웨어
  - SciPy, Numpy, Matplotlib, Pandas, Tensorflow 등을 비롯한 많은 패키지들을 포함
- 가상환경을 제공하여 버전 및 패키지 관리 용이

```
File "/System/Library/Frameworks/Python.framework/Versions/2.7/lib/python2.7/os.remove(fullname)
OSError: [Errno 13] Permission denied: '/Library/Python/2.7/site-packages/pip-8.1.2/pip_links.txt'
You are using pip version 8.1.2, however version 9.0.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
```



# Anaconda 설치

## ■ 다운로드

– <https://www.anaconda.com/distribution/>

1. 2.



Individual Edition

## Your data science toolkit

With over 20 million users worldwide, the open-source Individual Edition (Distribution) is the easiest way to perform Python/R data science and machine learning on a single machine. Developed for solo practitioners, it is the toolkit that equips you to work with thousands of open-source packages and libraries.

Download

3. Anaconda3 2020.07 (64-bit) Setup



### Welcome to Anaconda3 2020.07 (64-bit) Setup

Setup will guide you through the installation of Anaconda3 2020.07 (64-bit).

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

Next >

Cancel



4. Anaconda3 2020.07 (64-bit) Setup



### Select Installation Type

Please select the type of installation you would like to perform for Anaconda3 2020.07 (64-bit).

Install for:

☐ Just Me (recommended)

☒ All Users (requires admin privileges)

Anaconda, Inc.

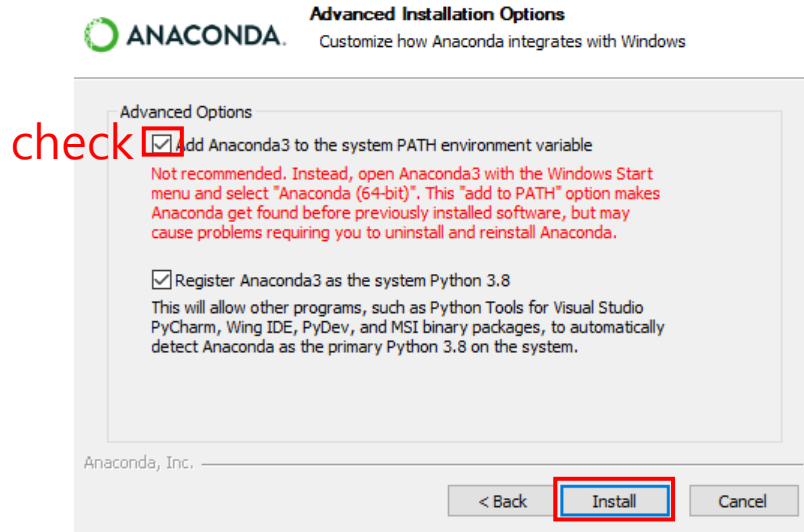
< Back

Next >

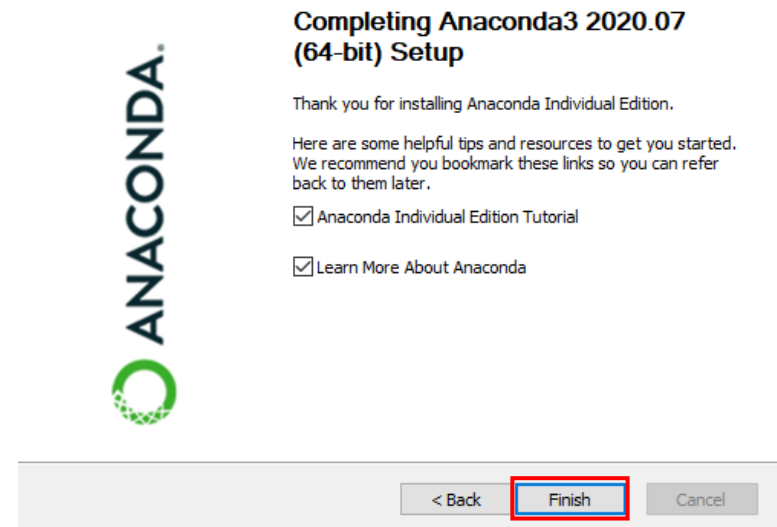
Cancel

# Anaconda 설치

6. Anaconda3 2020.07 (64-bit) Setup

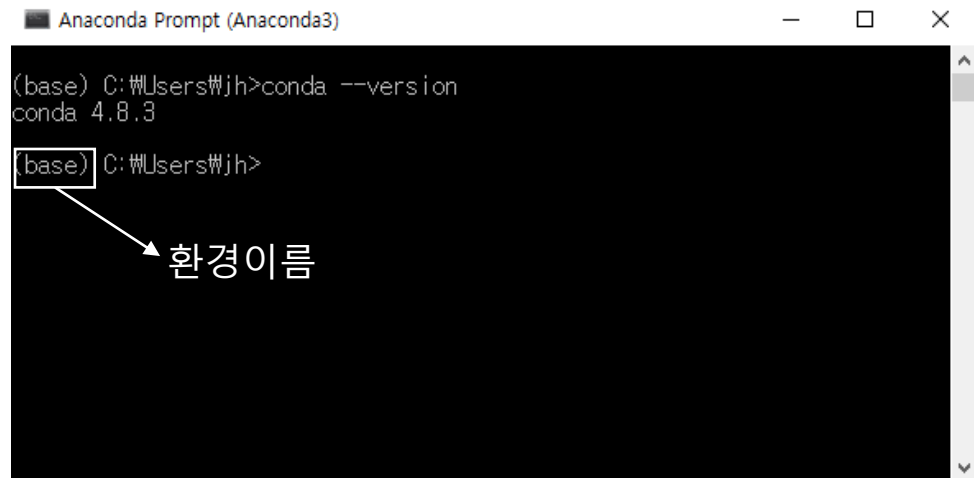
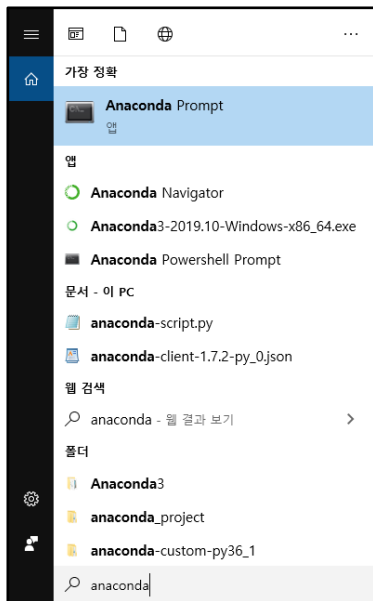


7. Anaconda3 2020.07 (64-bit) Setup



# Anaconda

- Anaconda
  - 윈도우 시작 메뉴에서 아나콘다 프롬프트 실행



# Anaconda Commands

## ■ 가상환경 생성

- conda create -n [생성할 가상환경 이름] python=버전명 [추가할 패키지 명]
  - e.g) conda create -name KW\_VIP python=3.7

```
(base) C:\Users\jh>conda create -n KW_VIP python=3.7
```

## ■ 생성된 가상환경 목록 확인

- conda env list

```
(base) C:\Users\jh>conda env list
# conda environments:
#
base
KW_VIP
* C:\ProgramData\Anaconda3
  C:\Users\jh\conda\envs\KW_VIP
```

현재 활성화 된 환경



# Anaconda Commands

- 생성한 가상환경으로 활성화 및 비활성화
  - activate [가상환경 이름]

```
(base) C:\Users\Wjh>activate KW_VIP
(KW_VIP) C:\Users\Wjh>
```

- conda deactivate
  - base 환경으로 돌아감

```
(KW_VIP) C:\Users\Wjh>conda deactivate
(base) C:\Users\Wjh>
```

- 가상환경 내에 설치된 패키지 목록확인
  - conda list

```
(KW_VIP) C:\Users\Wjh>conda list
# packages in environment at C:\Users\Wjh\conda\envs\KW_VIP:
#
# Name                    Version            Build    Channel
blas                      1.0                mkl
certifi                   2020.6.20          py36_0
intel-openmp              2020.2             254
mkl                       2020.2             256
mkl-service               2.3.0              py36hb782905_0
mkl_fft                   1.1.0              py36h45dec08_0
mkl_random                1.1.1              py36h47e9c7a_0
numpy                     1.19.1             py36h5510c5b_0
numpy-base                1.19.1             py36ha3acd2a_0
pip                       20.2.2             py36_0
python                    3.6.10             h9f7ef89_2
setuptools                49.6.0             py36_0
six                       1.15.0             py_0
sqlite                    3.33.0             h2a8f88b_0
vc                        14.1               h0510ff6_4
vs2015_runtime            14.16.27012        hf0eaf9b_3
wheel                     0.35.1             py_0
winertstore               0.2                py36h7fe50ca_0
zlib                      1.2.11             h62dcd97_4
```

# Anaconda Commands

## ■ 가상환경 지우기

- conda env remove -n [가상환경 이름]

```
(KW_VIP) C:\Users\wjh>conda env list
# conda environments:
#
base                  C:\ProgramData\Anaconda3
KW_VIP                * C:\Users\wjh\conda\envs\KW_VIP
KW_VIP2               C:\Users\wjh\conda\envs\KW_VIP2

(KW_VIP) C:\Users\wjh>conda env remove -n KW_VIP2

Remove all packages in environment C:\Users\wjh\conda\envs\KW_VIP2:

(KW_VIP) C:\Users\wjh>conda env list
# conda environments:
#
base                  C:\ProgramData\Anaconda3
KW_VIP                * C:\Users\wjh\conda\envs\KW_VIP

(KW_VIP) C:\Users\wjh>
```

## ■ 라이브러리 설치

- conda install [라이브러리이름]
- 라이브러리 버전을 명시하지 않는 경우 최신버전으로 설치됨
- 라이브러리 버전에 맞게 설치 가능
  - conda install [라이브러리이름]==버전

```
(KW_VIP) C:\Users\wjh>conda install jupyter
```

# Anaconda Commands

---

- 설치된 패키지 삭제
  - conda uninstall [삭제할 패키지 이름]
- 기존 가상환경 복제
  - conda create --name [새로 생성할 가상환경 이름] --clone [복제할 기존 가상환경 이름]
  - 기존 가상환경의 패키지를 복사하여 새로운 가상환경 생성

```
(KW_VIP) C:\Users\wjh>conda create -n KW_VIP2 --clone KW_VIP
```

# Anaconda Commands

## ■ 아나콘다 버전 확인

- conda info
- conda 버전에 따라서 다운가능한 패키지의 버전이 다름

```
Anaconda Prompt (Anaconda3) - deactivate - conda deactivate - conda install jupyter
(KW_VIP) C:\Users\Wjh>conda info

active environment : KW_VIP
active env location : C:\Users\Wjh\Anaconda3\envs\KW_VIP
shell level : 2
user config file : C:\Users\Wjh\Anaconda3\condarc
populated config files : C:\Users\Wjh\Anaconda3\condarc
conda version : 4.8.3
conda-build version : 3.18.11
python version : 3.8.3.final.0
virtual packages : _cuda=10.2
base environment : C:\ProgramData\Anaconda3 (read only)
```

## ■ 아나콘다 업데이트

- conda update [업데이트할 패키지명]
- 모든 패키지를 최신버전으로 업데이트
  - conda update --all

# Anaconda Commands

- 다운가능한 패키지 버전 확인
  - conda search 라이브러리명

```
(KW_VIP) C:\Users\Wjh>conda search pytorch
Loading channels: done
# Name                               Version          Build Channel
pytorch                             1.0.1            cpu_py36h39a92a0_0 pkgs/main
pytorch                             1.0.1            cpu_py37h39a92a0_0 pkgs/main
pytorch                             1.3.1            cpu_py36h9f948e0_0 pkgs/main
pytorch                             1.3.1            cpu_py37h9f948e0_0 pkgs/main
pytorch                             1.4.0            cpu_py36ha775e86_0 pkgs/main
pytorch                             1.4.0            cpu_py37ha775e86_0 pkgs/main
pytorch                             1.4.0            cpu_py38ha775e86_0 pkgs/main
pytorch                             1.5.0            cpu_py37h9f948e0_0 pkgs/main
pytorch                             1.6.0            cpu_py37h538a6d7_0 pkgs/main
pytorch                             1.6.0            cpu_py38h538a6d7_0 pkgs/main
```

# Anaconda Commands

- External library download
  - Pillow
  - Matplotlib
- Pytorch 설치
  - <https://pytorch.org/>

The screenshot shows the PyTorch Build configuration interface. The selected options are highlighted in red:

- PyTorch Build: Stable (1.6.0)
- Your OS: Windows
- Package: Conda
- Language: Python
- CUDA: 10.2

Run this Command:

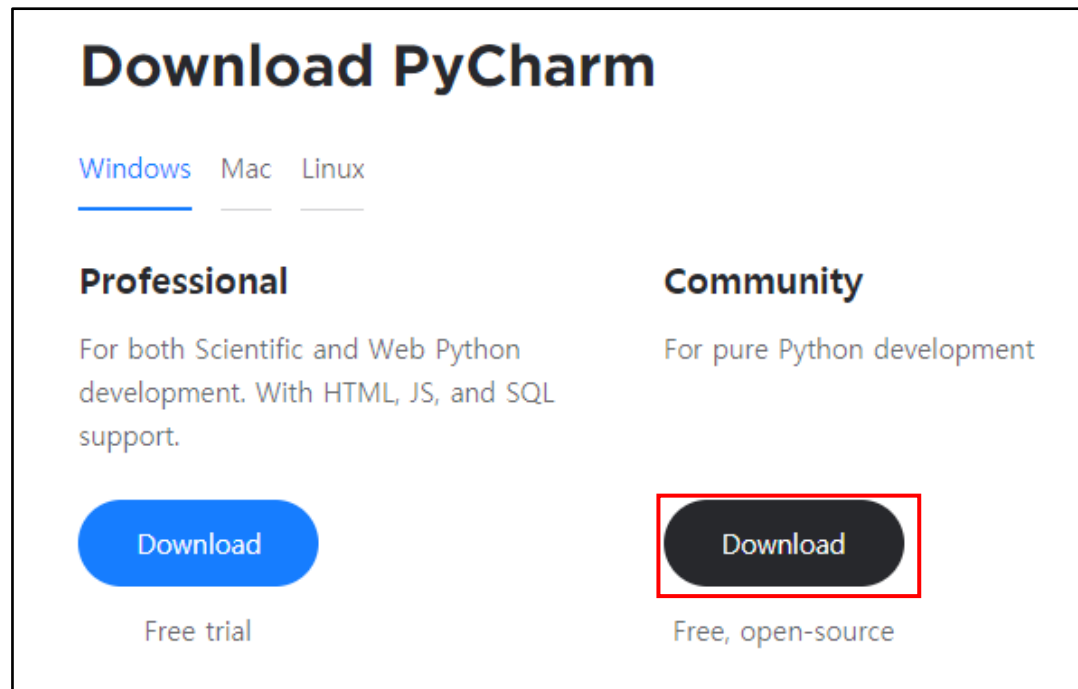
```
conda install pytorch torchvision cpuonly -c pytorch
```

# PYCHARM

---

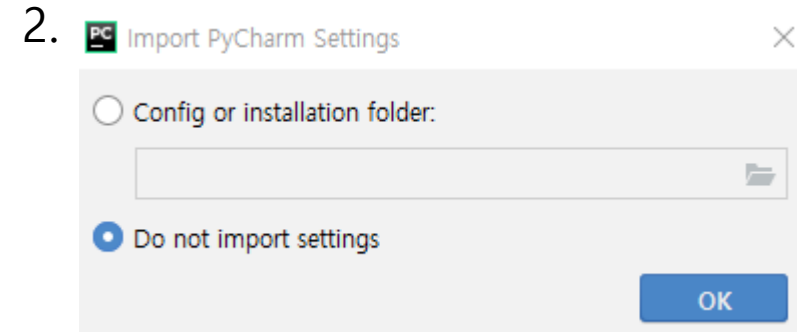
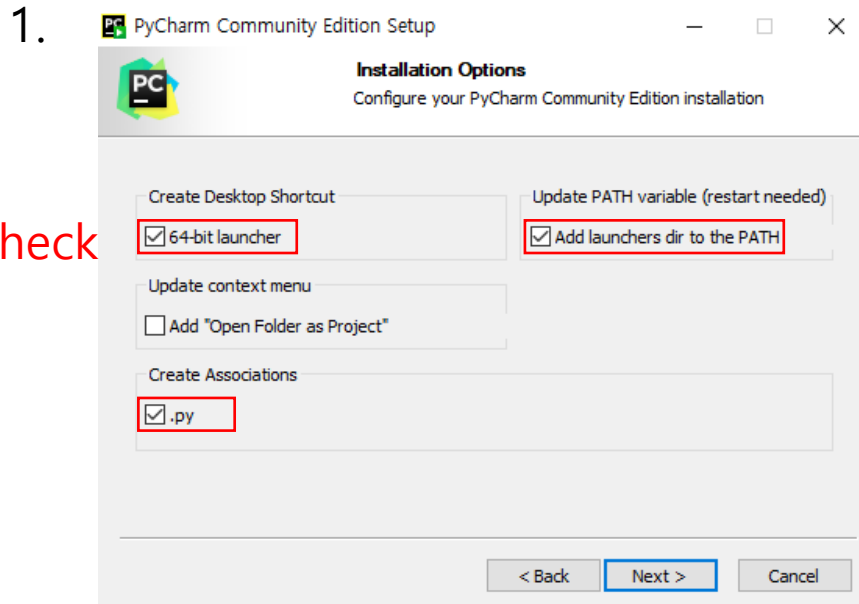
# Pycharm

- Python 프로그램을 쉽게 개발할 수 있도록 하는 통합개발환경 (IDE)
- Download
  - <https://www.jetbrains.com/pycharm/download/#section=windows>

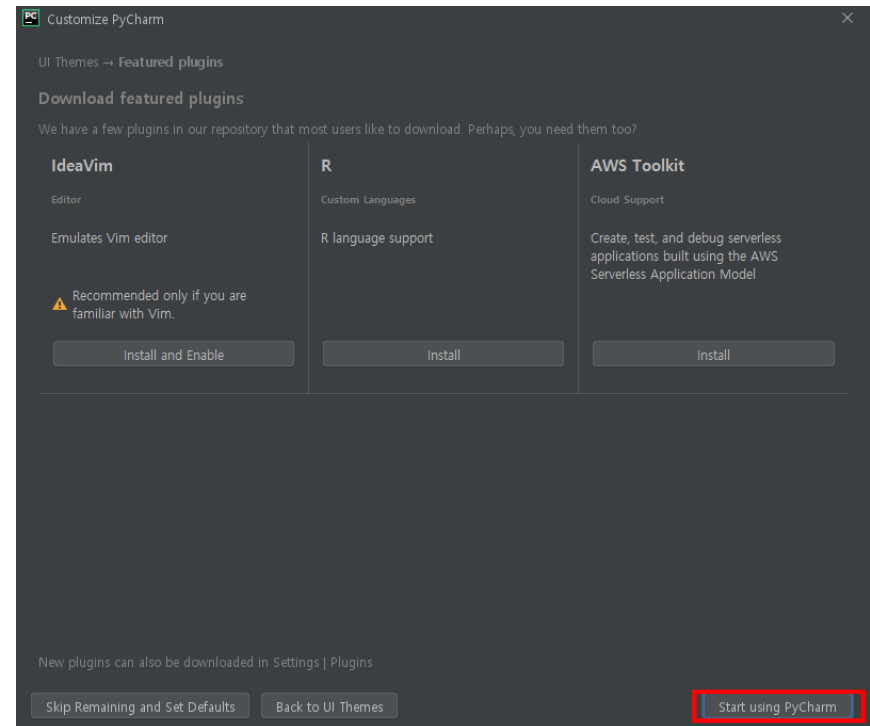
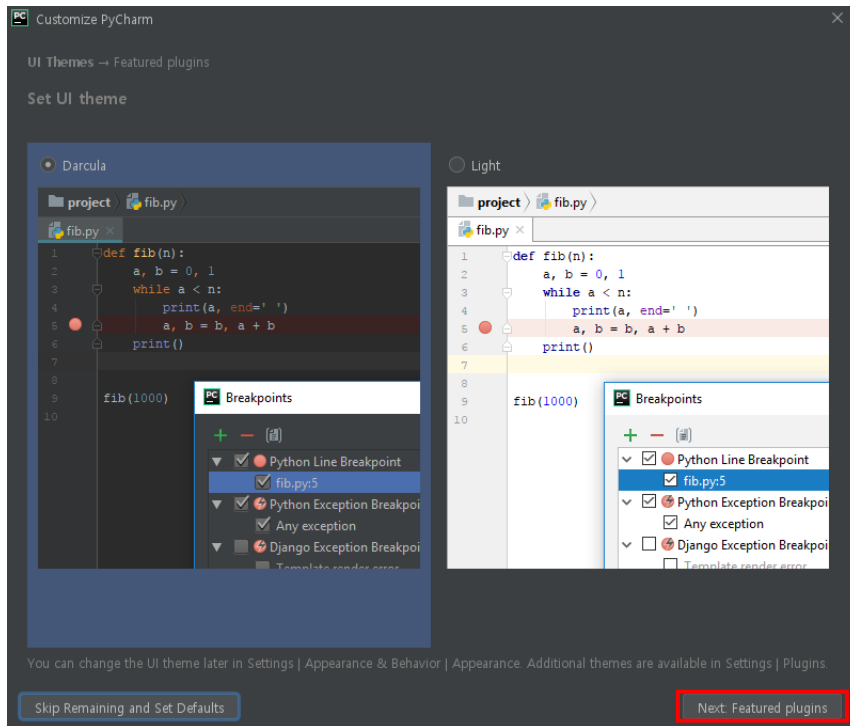




# Pycharm 설치



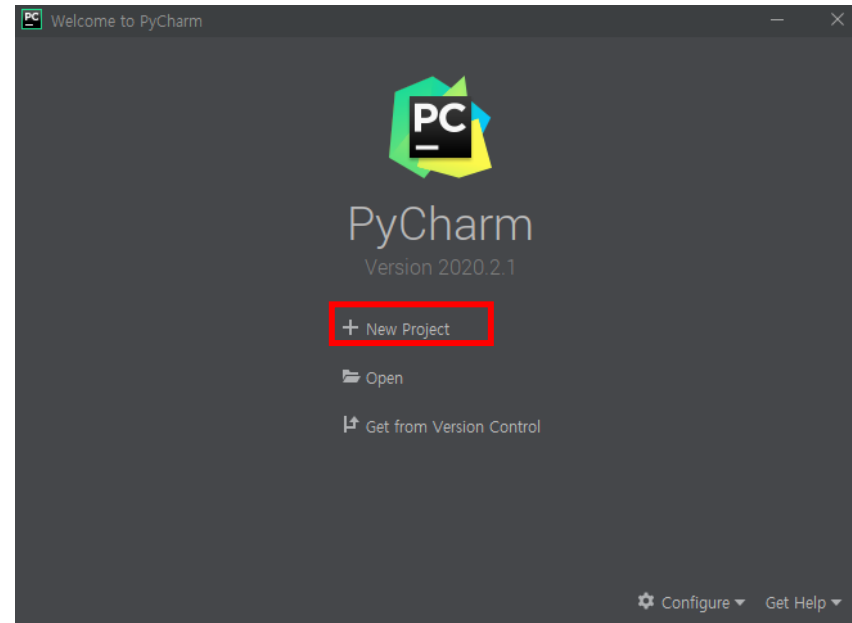
# Pycharm 설치



# Pycharm

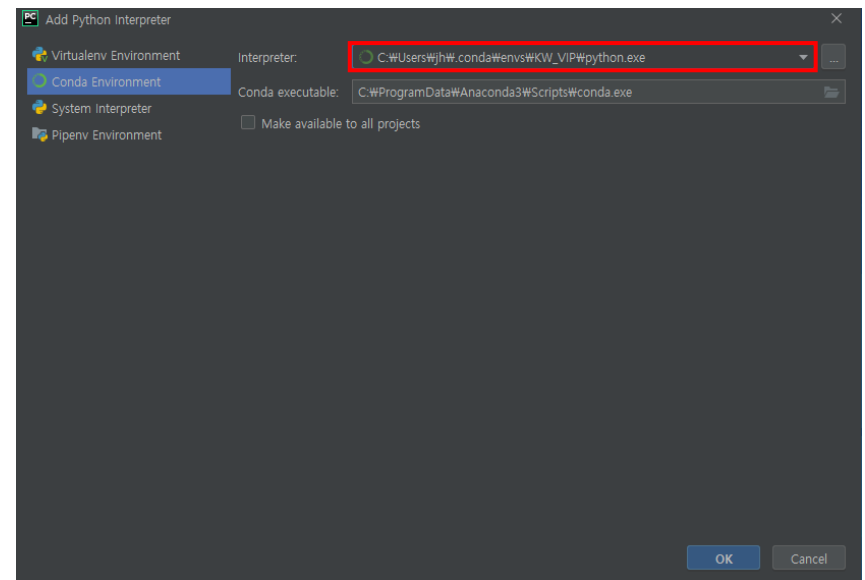
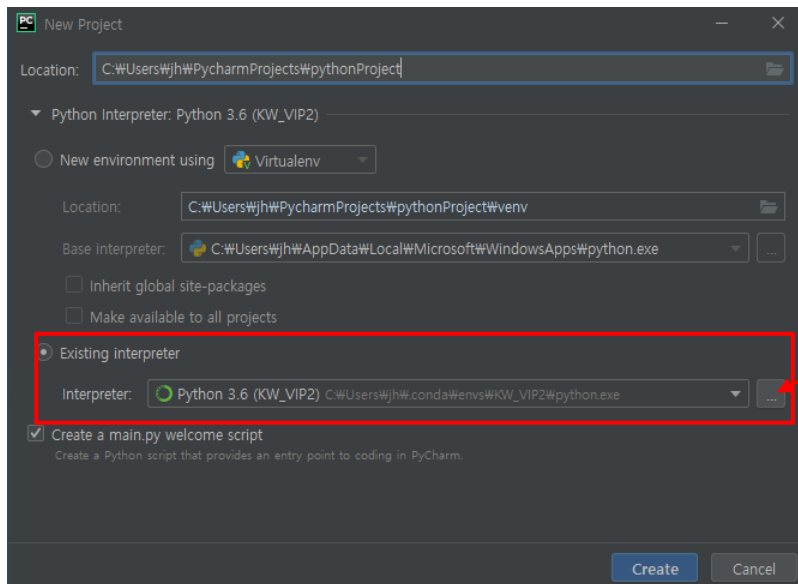
---

- Pycharm 프로젝트 생성
  - New Project



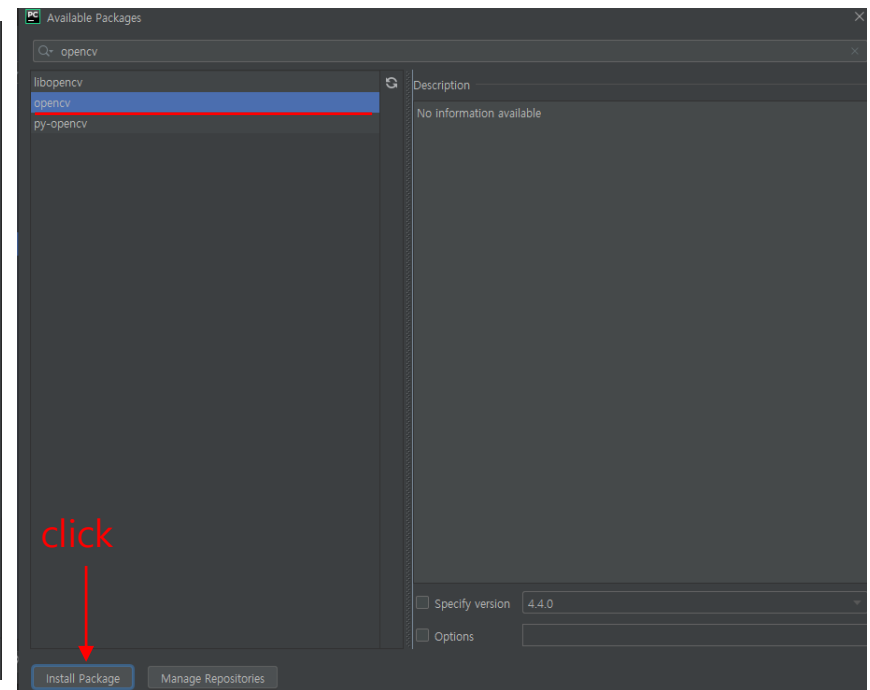
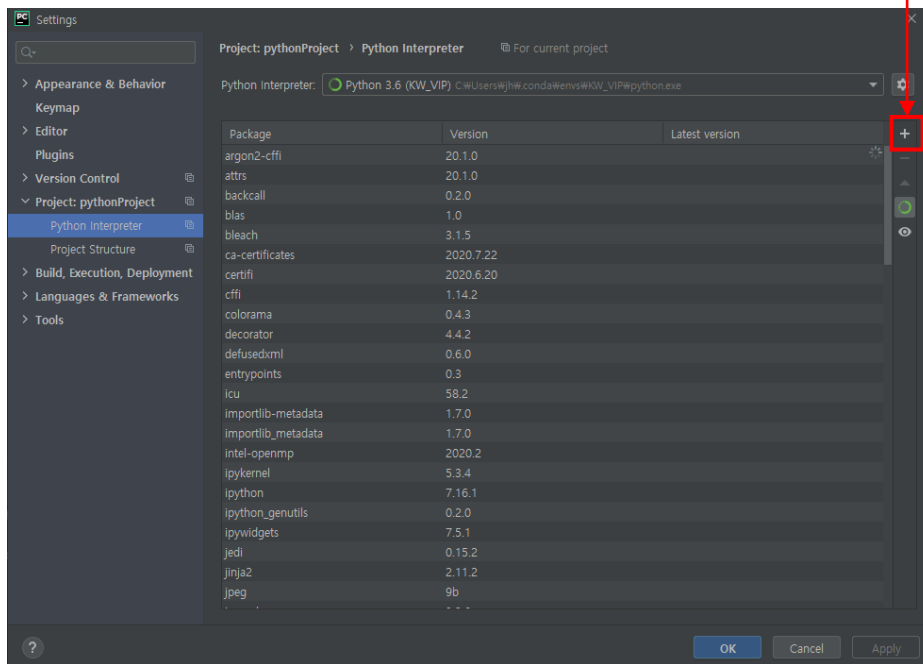
# Pycharm

- Python Interpreter 설정
  - Anaconda 가상환경에서 만든 Interpreter 등록



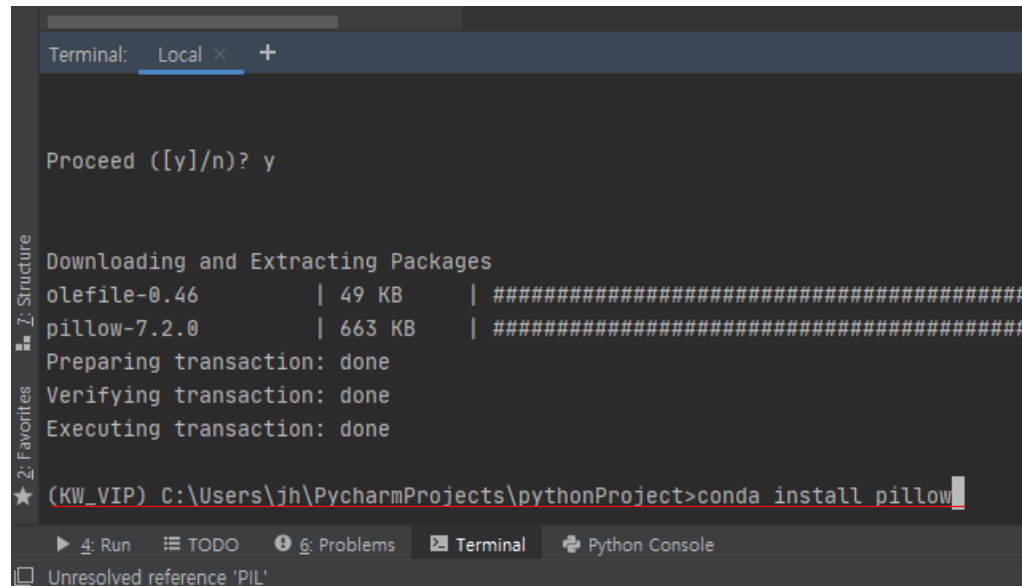
# Pycharm

- 패키지 설치 (GUI)
  - File -> Settings...



# Pycharm

- 패키지 설치 (Terminal)
  - Pycharm 좌측 하단 Terminal 창
  - conda install [패키지명]
  - 또는 pip install [패키지명]



The screenshot shows the PyCharm IDE interface with the Terminal window open. The terminal displays the output of the command `conda install pillow`. The output includes the prompt `Proceed ([y]/n)? y`, followed by a table of packages being downloaded and extracted, and then the transaction status.

```
Terminal: Local x +

Proceed ([y]/n)? y

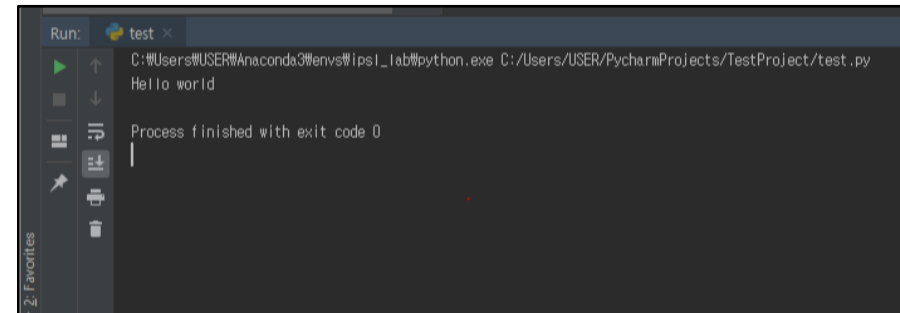
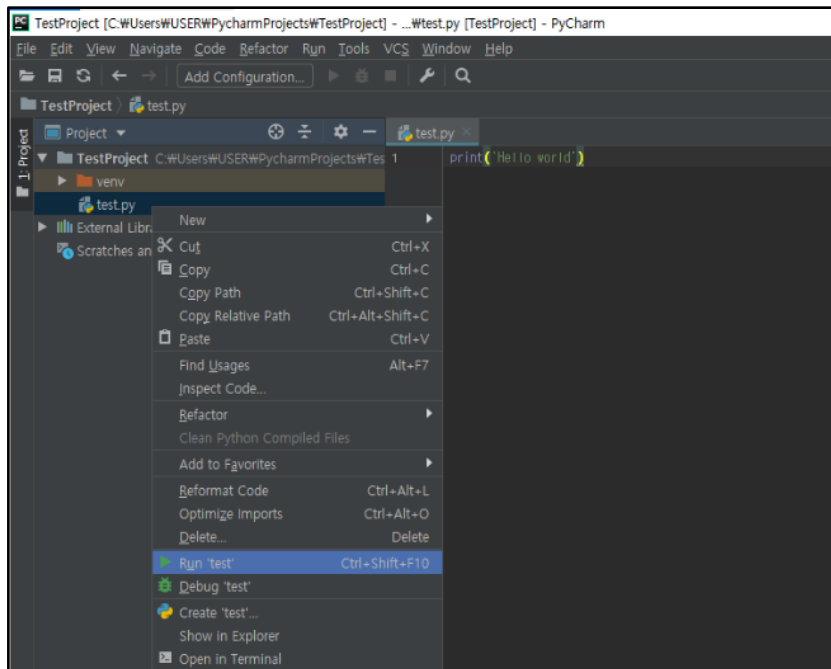
Downloading and Extracting Packages
olefile-0.46      | 49 KB      | #####
pillow-7.2.0     | 663 KB     | #####
Preparing transaction: done
Verifying transaction: done
Executing transaction: done

(KW_VIP) C:\Users\jh\PycharmProjects\pythonProject>conda install pillow
```

The bottom of the terminal shows the command `conda install pillow` being entered. The status bar at the bottom indicates "Unresolved reference 'PIL'".

# Pycharm

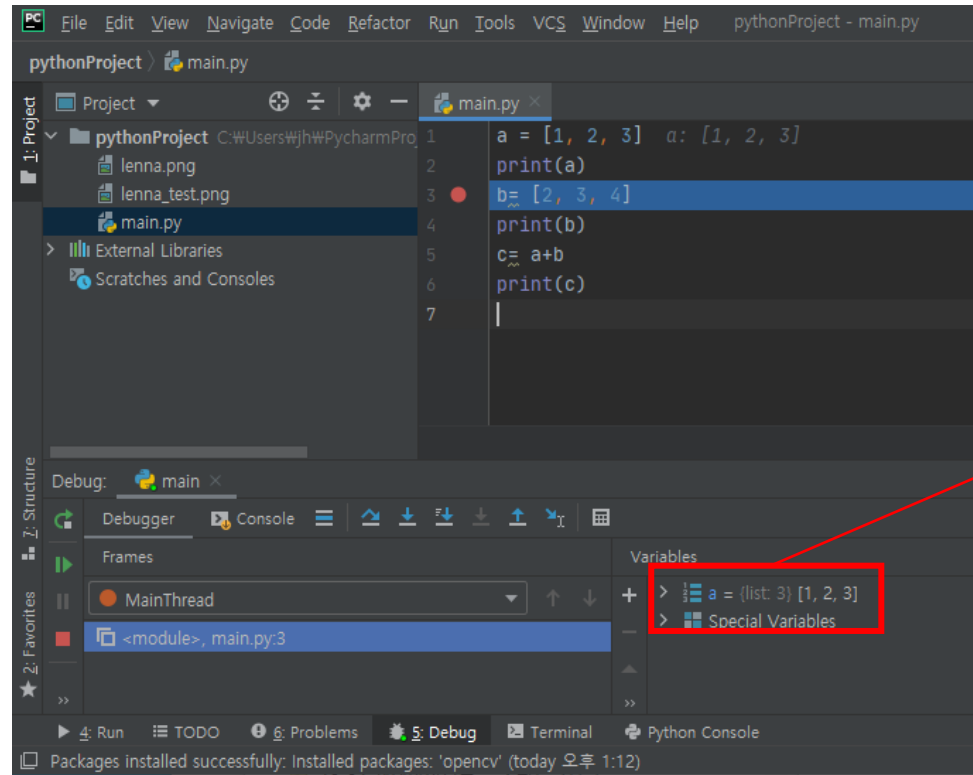
- Python Code 실행
  - Run -> Run ... (Shift + F10)



# Pycharm

## ■ Debug

- Run -> Debug... (Shift + F9)
- 중단점 설정 가능
- F9키를 통해 다음 중단점으로 이동
- F8키를 통해 한줄 씩 디버깅



변수확인



# PYTHON

---

# Python

---

- Python 소개
  - 플랫폼 독립적 언어
  - 인터프리터 언어
  - 객체 지향 언어
  
- Python 장점
  - 오픈 된 라이브러리 풍부
  - 쉬운 문법
  - 메모리 자동 관리

# Python

## ■ 자료형

- 숫자형
  - Integer
  - Float
- 불 자료형
- 문자형

```

>>> a = 0
>>> type(a)
<class 'int'>
>>>
>>> a = 0.0
>>> type(a)
<class 'float'>
>>>
>>> a = True
>>> type(a)
<class 'bool'>
>>>
>>> a = 'asdf'
>>> type(a)
<class 'str'>
>>> type(a)
<class 'str'>
>>> a = []
>>> type(a)
<class 'list'>
>>>
>>> a = {}
>>> type(a)
<class 'dict'>
>>>
>>> a = {0}
>>> type(a)
<class 'set'>

```

## ■ 숫자형 & 불 자료형

- 연산자
  - 덧셈(+), 뺄셈(-), 곱셈(\*), 나눗셈(/), 몫 반환(/), 나머지 반환(%), 제곱(\*\*)

Data type

Container

# Python

- if 문
  - 기본 구조
    - if
    - else
    - elif

```
if 조건문:  
    수행할 문장1  
    수행할 문장2  
    ...  
else:  
    수행할 문장A  
    수행할 문장B  
    ...
```

```
If <조건문>:  
    <수행할 문장1>  
    <수행할 문장2>  
    ...  
elif <조건문>:  
    <수행할 문장1>  
    <수행할 문장2>  
    ...  
elif <조건문>:  
    <수행할 문장1>  
    <수행할 문장2>  
    ...  
...  
else:  
    <수행할 문장1>  
    <수행할 문장2>  
    ...
```

# Python

## ▪ if 문

### – 비교 연산자

비교연산자	설명
<code>x &lt; y</code>	x가 y보다 작다
<code>x &gt; y</code>	x가 y보다 크다
<code>x == y</code>	x와 y가 같다
<code>x != y</code>	x와 y가 같지 않다
<code>x &gt;= y</code>	x가 y보다 크거나 같다
<code>x &lt;= y</code>	x가 y보다 작거나 같다

연산자	설명
<code>x or y</code>	x와 y 둘중에 하나만 참이어도 참이다
<code>x and y</code>	x와 y 모두 참이어야 참이다
<code>not x</code>	x가 거짓이면 참이다

```
>>> money = 2000
>>> card = True
>>> if money >= 3000 or card:
...     print("택시를 타고 가라")
... else:
...     print("걸어가라")
...
택시를 타고 가라
```

예시 코드

# Python

## ■ If 문

- 조건부 표현식
  - C 언어의 3항 연산자

조건 ? TRUE : FALSE;

조건문이 참인 경우 if 조건문 else 조건문이 거짓인 경우

# Python

- while 문
  - 기본 구조
    - break
    - continue

```
while <조건문>:
    <수행할 문장1>
    <수행할 문장2>
    <수행할 문장3>
    ...
```

```
>>> coffee = 10
>>> money = 300
>>> while money:
...     print("돈을 받았으니 커피를 줍니다.")
...     coffee = coffee - 1
...     print("남은 커피의 양은 %d개입니다." % coffee)
...     if coffee == 0:
...         print("커피가 다 떨어졌습니다. 판매를 중지합니다.")
...         break
```

예시 코드

```
>>> a = 0
>>> while a < 10:
...     a = a + 1
...     if a % 2 == 0: continue
...     print(a)
```

예시 코드

# Python

---

- for 문
  - 기본 구조
    - range(1, 10+1)

```
for 변수 in 리스트(또는 튜플, 문자열):  
    수행할 문장1  
    수행할 문장2  
    ...
```

```
>>> add = 0  
>>> for i in range(1, 11):  
...     add = add + i  
...  
>>> print(add)  
55
```



# Python

## ▪ Numpy

- Contains operations for vector and matrix
- From <https://www.lfd.uci.edu/~gohlke/pythonlibs/#numpy>

## ▪ array 생성

```
>>> import numpy as np
>>>
>>> array1 = np.array([[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12], [13, 14, 15, 16]])
>>>
>>> array1
array([[ 1,  2,  3,  4],
       [ 5,  6,  7,  8],
       [ 9, 10, 11, 12],
       [13, 14, 15, 16]])

>>>
>>> identity = np.array([[1, 0, 0, 0], [0, 1, 0, 0], [0, 0, 1, 0], [0, 0, 0, 1]])
>>> identity
array([[1, 0, 0, 0],
       [0, 1, 0, 0],
       [0, 0, 1, 0],
       [0, 0, 0, 1]])

>>> np.zeros(shape=(4, 4))
array([[ 0.,  0.,  0.,  0.],
       [ 0.,  0.,  0.,  0.],
       [ 0.,  0.,  0.,  0.],
       [ 0.,  0.,  0.,  0.]])

>>> np.ones(shape=(4, 4))
array([[ 1.,  1.,  1.,  1.],
       [ 1.,  1.,  1.,  1.],
       [ 1.,  1.,  1.,  1.],
       [ 1.,  1.,  1.,  1.]])
```

# Python

## ■ Numpy – 연산자

```
>>> # add
>>> array1 + identity
array([[ 2,  2,  3,  4],
       [ 5,  7,  7,  8],
       [ 9, 10, 12, 12],
       [13, 14, 15, 17]])

>>> # sub
>>> array1 - identity
array([[ 0,  2,  3,  4],
       [ 5,  5,  7,  8],
       [ 9, 10, 10, 12],
       [13, 14, 15, 15]])

>>>
>>> # matrix multiply
>>> np.matmul(array1, identity)
array([[ 1,  2,  3,  4],
       [ 5,  6,  7,  8],
       [ 9, 10, 11, 12],
       [13, 14, 15, 16]])

>>>
>>> # multiply each element
>>> array1 * identity
array([[ 1,  0,  0,  0],
       [ 0,  6,  0,  0],
       [ 0,  0, 11,  0],
       [ 0,  0,  0, 16]])

>>>
>>> 5 * identity
array([[5, 0, 0, 0],
       [0, 5, 0, 0],
       [0, 0, 5, 0],
       [0, 0, 0, 5]])
...
```

```
>>> # determinant
>>> np.linalg.det(array1)
-1.820448242817726e-31
>>> np.linalg.det(identity)
1.0

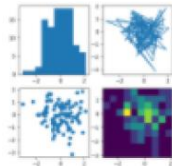
>>> # inverse matrix
>>> np.linalg.inv(array1)
array([[ 1.50119988e+15, -3.75299969e+14, -3.75299969e+15,
         2.62709978e+15],
       [-1.95155984e+16,  1.95155984e+16,  1.95155984e+16,
        -1.95155984e+16],
       [ 3.45275971e+16, -3.79052969e+16, -2.77721977e+16,
         3.11498974e+16],
       [-1.65131986e+16,  1.87649984e+16,  1.20095990e+16,
        -1.42613988e+16]])

>>> np.linalg.inv(identity)
array([[ 1.,  0.,  0.,  0.],
       [ 0.,  1.,  0.,  0.],
       [ 0.,  0.,  1.,  0.],
       [ 0.,  0.,  0.,  1.]])
```

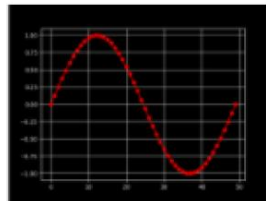
# Python

## ■ Matplotlib

- We can generate plot for some data including images
- Project site : <https://matplotlib.org/index.html>



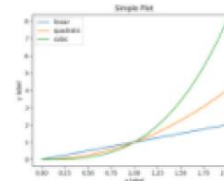
Sample plots in  
Matplotlib



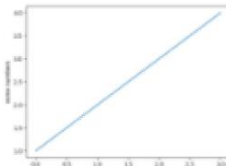
Customizing  
matplotlib



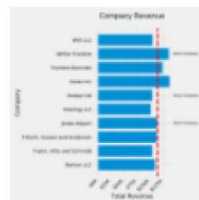
Image tutorial



Usage Guide



Pyplot tutorial



The Lifecycle of a  
Plot

# Python

- pillow(PIL)
  - Python image processing library
  - Example #1. Show image with **matplotlib** and print image data

```
import numpy as np
from PIL import Image
import matplotlib.pyplot as plt

# open image
image = Image.open("./Lenna.png")

# show image
plt.imshow(image)
plt.show()

# resize image
image2 = image.resize((32,32))

# show image
plt.imshow(image2)
plt.show()

print(image2)

print(type(image2))

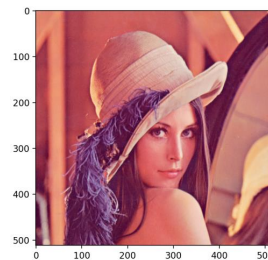
print(image2.size)

print(image2.mode)

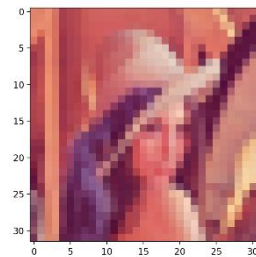
print(image2.getpixel((0,0)))

# rotate image
image_rotate = image.transpose(Image.ROTATE_180)

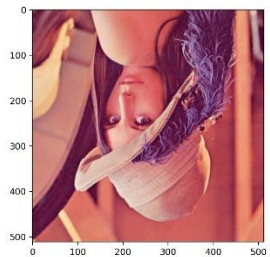
plt.imshow(image_rotate)
plt.show()
```



원본 영상



축소 영상



회전 영상

# Python

– Example #2. Show image with **matplotlib** and print image data

```
import numpy as np
from PIL import Image
from PIL import ImageFilter
import matplotlib.pyplot as plt

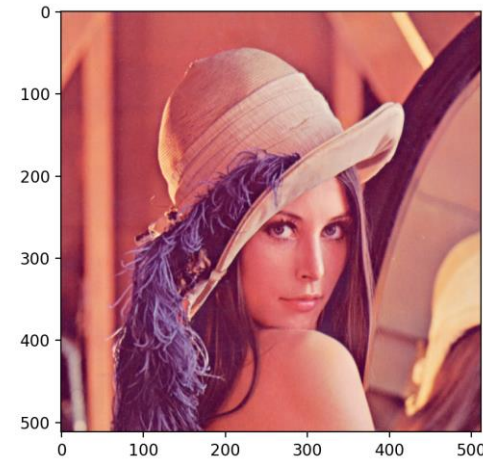
# open image
image = Image.open("./Lenna.png")

# show image
plt.imshow(image)
plt.show()

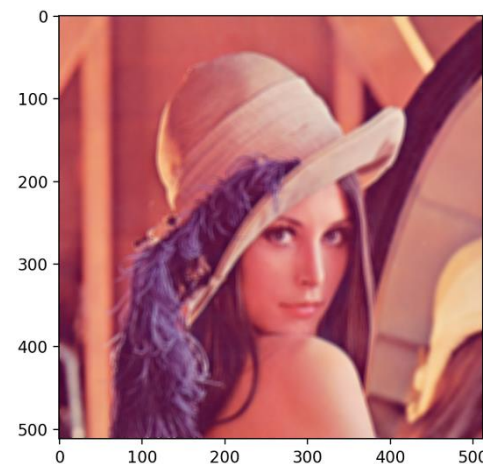
blurred_image = image.filter(ImageFilter.BLUR)

# show image
plt.imshow(blurred_image)
plt.show()

# save image
blurred_image.save("./blurred_Lenna.png")
```



원본 영상



블러 처리된 영상

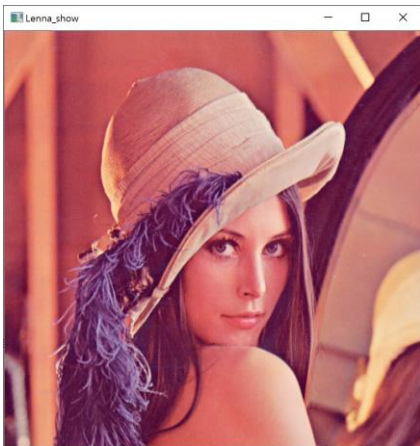
# ASSIGNMENT #1

---

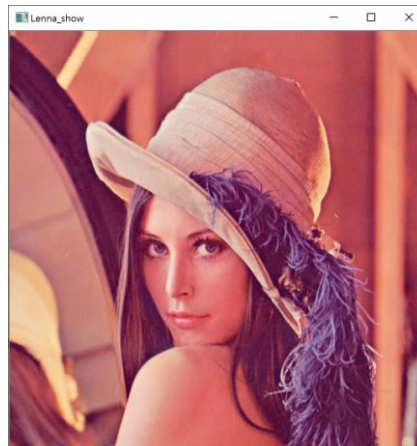
# Assignment #1

## ■ 과제 목표

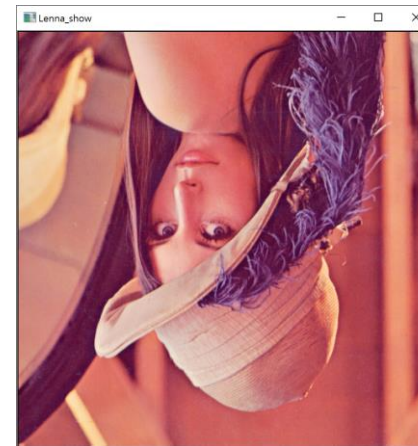
- 이미지 로드
- 좌우 반전된 이미지 출력
- 180도 회전된 이미지 출력
- 가로, 세로의 길이가 각각 2배 축소된 이미지 출력



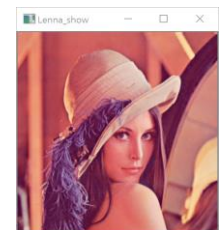
원본 이미지



좌우 반전된 이미지



180도 회전된 이미지



가로,세로  
2배 축소된 이미지

# Assignment #1

---

- 구현 시 고려 사항
  - 파이썬 영상처리 라이브러리(PIL)를 사용하여 구현
    - 라이브러리 내 다양한 영상처리 함수 사용
  - 구현/테스트에서 과제 공지 글에 첨부되어 있는 이미지 사용
- 과제 제출물
  - Python source 파일
  - 보고서(.pdf 형식)
    - 개요
      - 과제 목표를 설명
    - 구현 방법
      - 사용한 영상처리 라이브러리 함수의 기능에 대하여 설명
    - 결과 화면
      - 결과 화면 캡처 한 것을 첨부
- 과제 제출 기한
  - 4월 6일 23시 59분 까지
  - [krosea@naver.com](mailto:krosea@naver.com) 메일 제출



# 참고자료

## ▪ Object detection

### – YOLO

- 논문 : <https://arxiv.org/abs/1506.02640>
- 코드(c 언어) : <https://pjreddie.com/darknet/yolo/>
- 코드(pytorch) : <https://github.com/eriklindernoren/PyTorch-YOLOv3>

## ▪ GPU 가속화

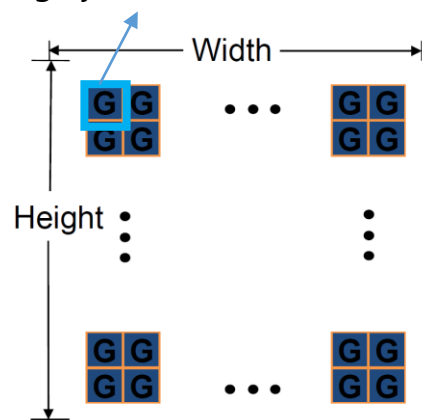
- CUDA, cuDNN 설치 필요
- 설치 방법 : <https://mickael-k.tistory.com/18>
- CUDA 버전에 맞는 pytorch 설치

PyTorch Build	Stable (1.6.0)		Preview (Nightly)	
Your OS	Linux	Mac	Windows	
Package	Conda	Pip	LibTorch	Source
Language	Python		C++ / Java	
CUDA	9.2	10.1	10.2	None
Run this Command:	conda install pytorch torchvision cudatoolkit=10.2 -c pytorch			

# 참고자료 – 이미지 포맷

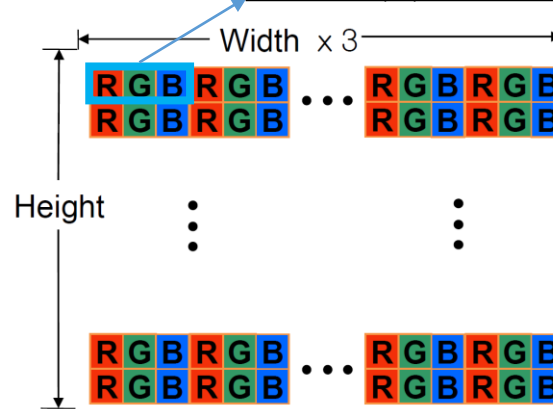
- 비트맵 방식의 이미지 포맷
  - 픽셀의 배열로 이미지를 구성
    - 이미지 해상도 : 가로 픽셀 개수 x 세로 픽셀 개수
  - BMP, JPG, GIF, PNG, TIFF 등
- 이미지 채널과 비트 덱스
  - 채널
    - 이미지가 지니고 있는 색상에 대한 정보
  - 비트 덱스
    - 픽셀의 값을 표현하는 정밀도
    - 8bit color 이미지의 경우 R, G, B 각각이 256가지의 값을 갖을 수 있음

픽셀 당 gray에 해당하는 단일 채널의 정보 존재



<1 채널 gray 이미지>

픽셀 당 R,G,B 세 채널의 정보 존재



<3 채널 color 이미지>

# END OF PRESENTATION

---

Q&A