

영등포구 청년을 위한

생성형 AI 활용 데이터 시각화 교육

파이썬의 이해



목 차

- 01 파이썬
- 02 개발 환경



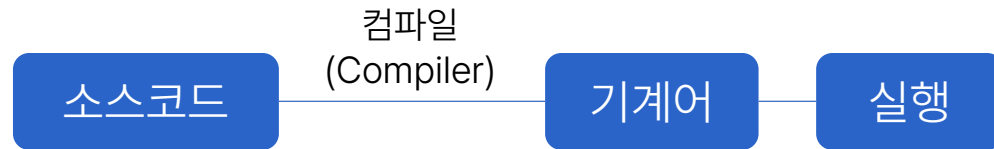
01 파이썬



파이썬의 이해

- 1990년 귀도 반 로섬(Guido Van Rossum)이 개발한 인터프리터 언어
 - Life is short, you need Python.

컴파일러(Compiler)



```

#include <stdio.h>

int main()
{
    printf("Hello, world!\n");

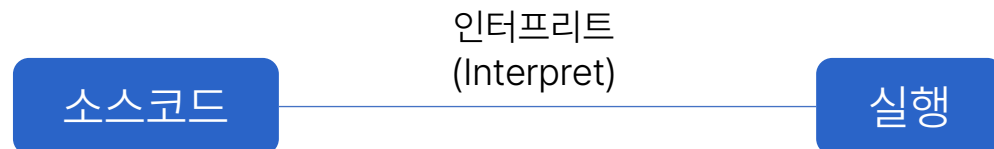
    return 0;
}
  
```

HelloWorld.c



HelloWorld.exe

인터프리터(Interpreter)



```

print("Hello, World! \n")
  
```

HelloWorld.py



파이썬의 이해

- 파이썬은 다른 언어보다 배우기 쉬움

C	JAVA	Python
 <pre>#include <stdio.h> int main() { printf("Hello World!"); return 0; }</pre>	 <pre>public class HelloWorld{ public static void main(String[] args){ System.out.println("Hello World!"); } }</pre>	 <pre>print("Hello World!")</pre>

파이썬의 이해

- 다양한 영역에서 사용 가능



02 개발 환경

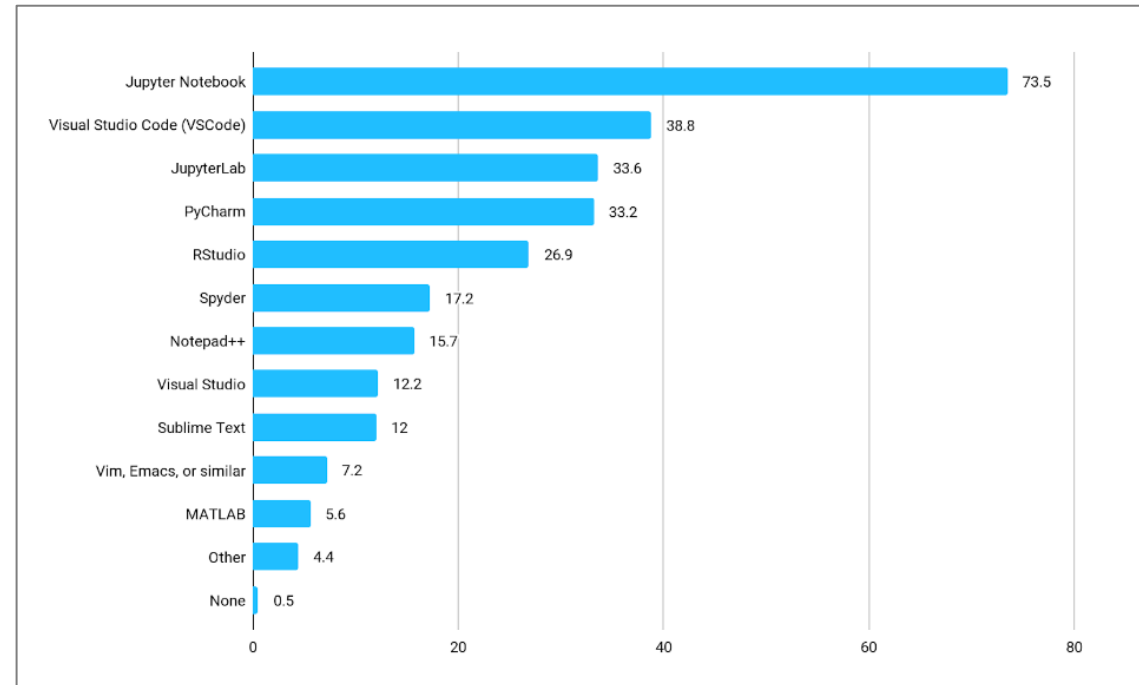


파이썬 개발 환경(Jupyter)

- 데이터 분석에 가장 많이 사용되는 통합 개발 환경(IDE, Integrated Development Environment)

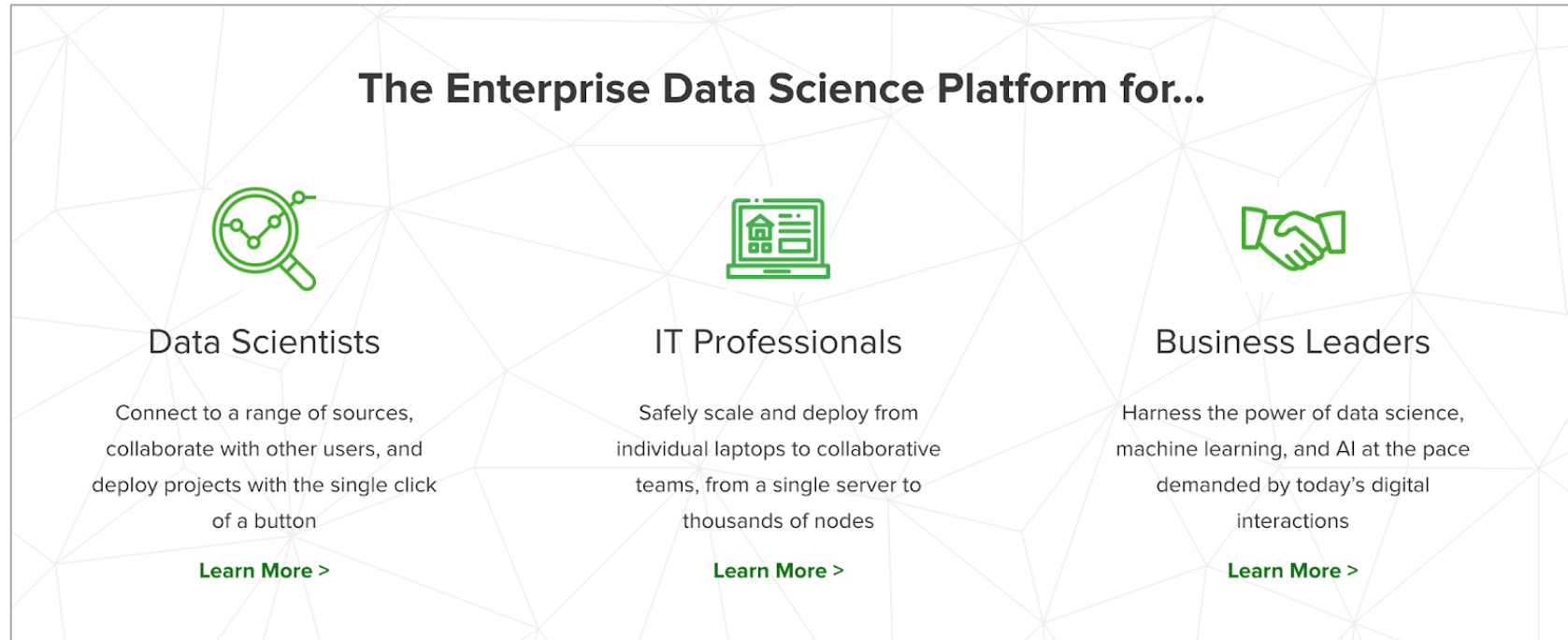


- 오픈소스 기반의 데이터분석 전용 IDE
- 로컬머신의 자원을 활용
- 웹 서비스로 열어서 외부에서도 활용가능



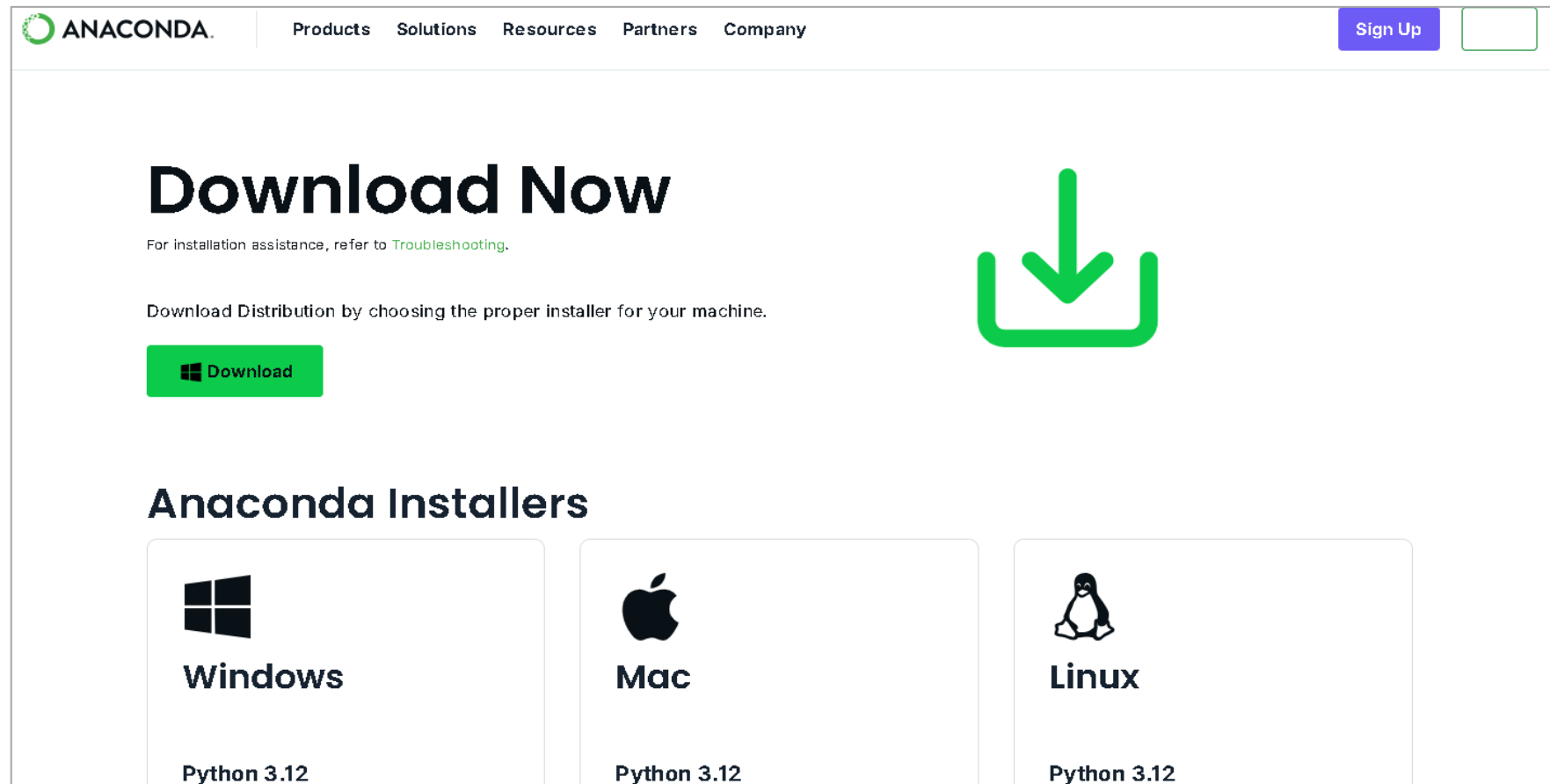
아나콘다(anaconda)

- 파이썬 기본패키지에 각종 수학/과학 라이브러리들을 같이 패키징해서 배포
- 데이터 시각화를 위한 다양한 도구들(pandas, numpy, matplotlib 등)이 존재

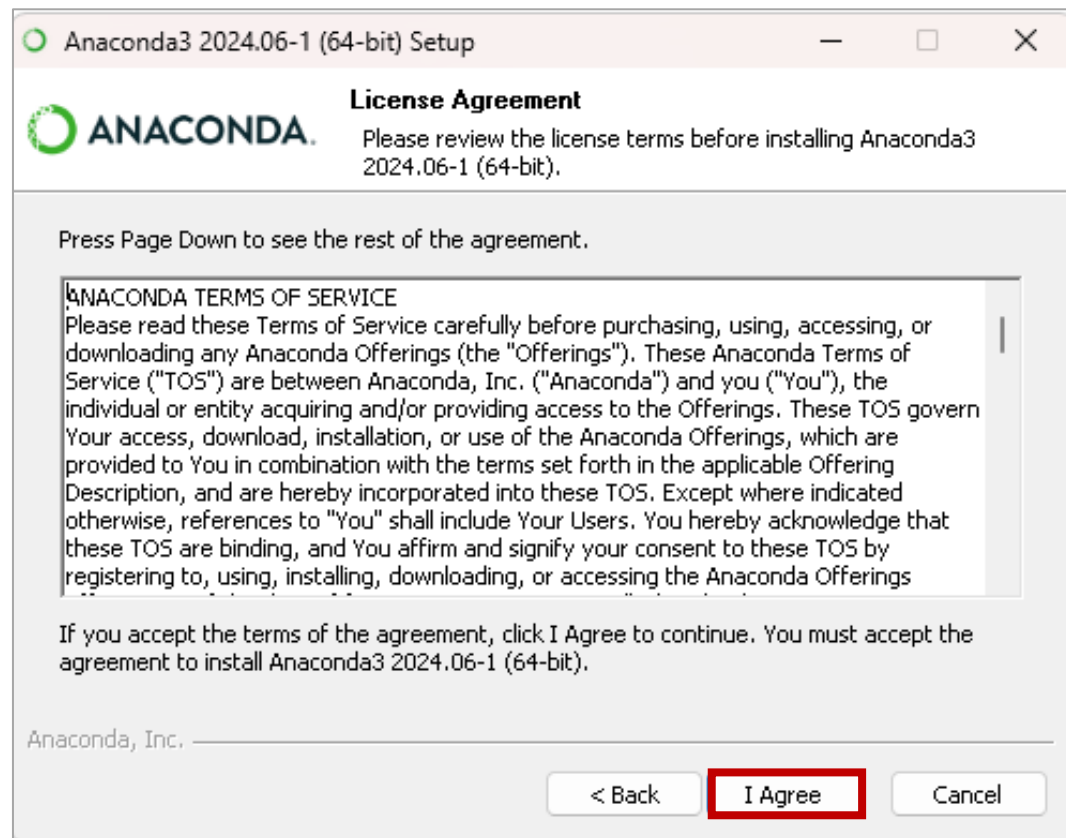
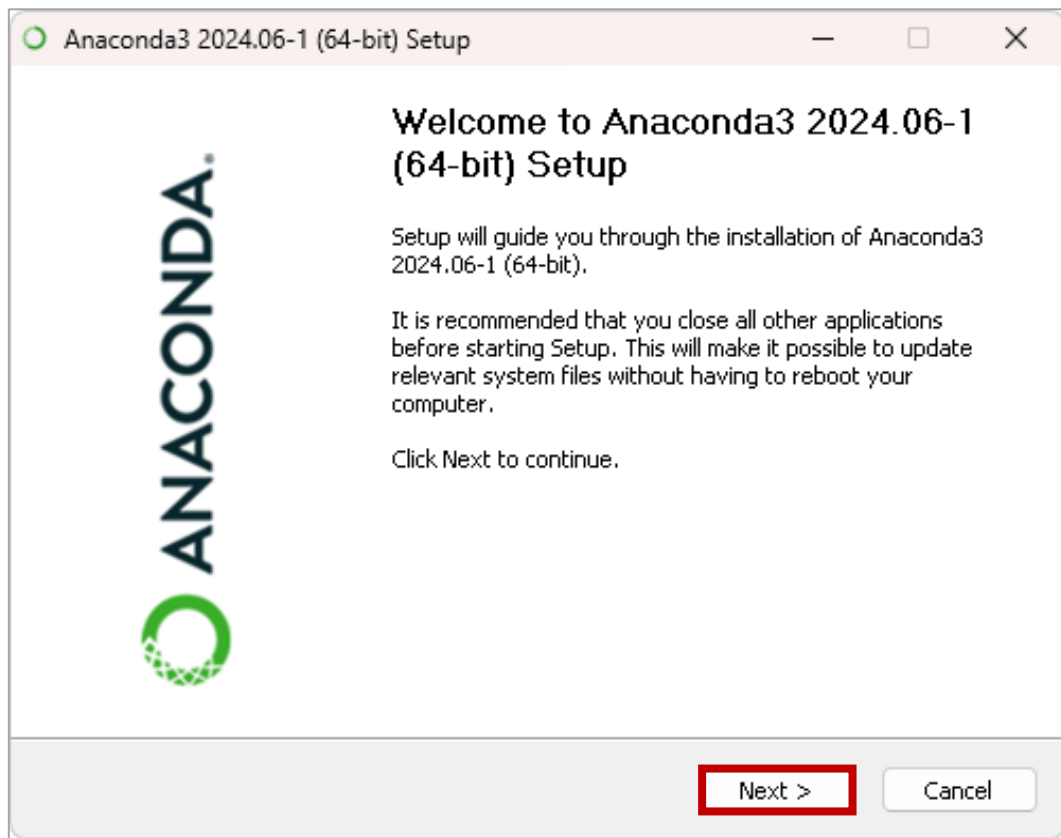


아나콘다(anaconda) 다운로드

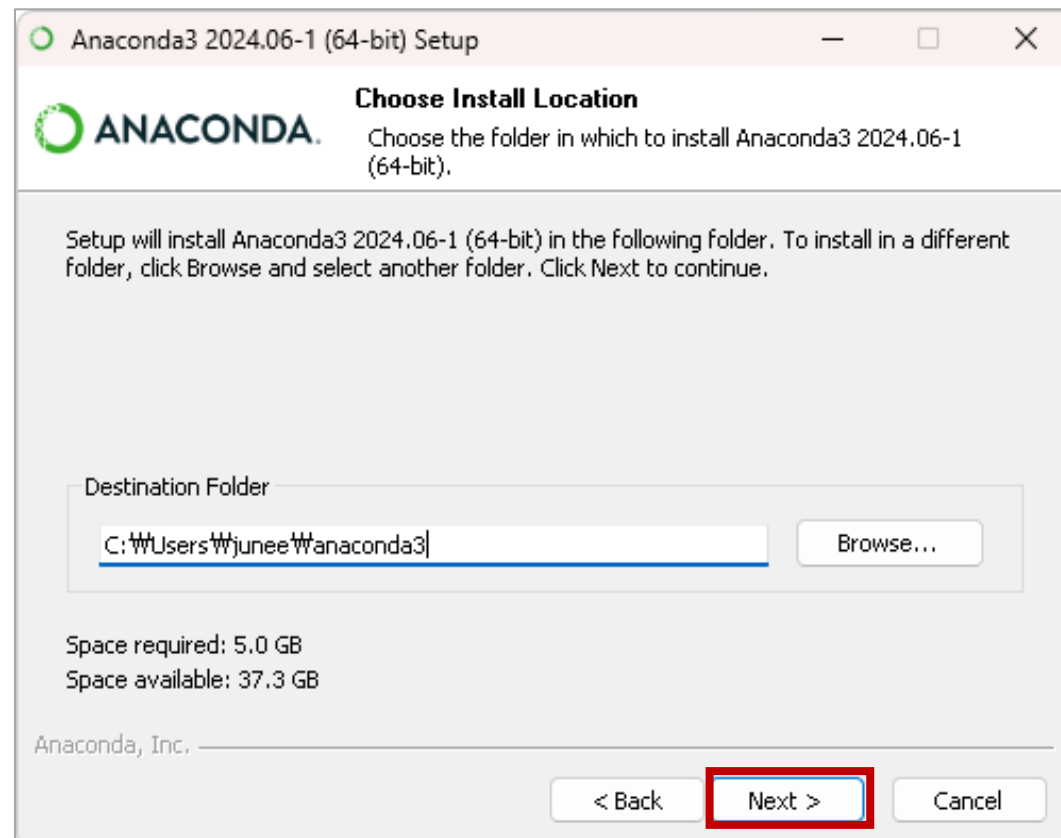
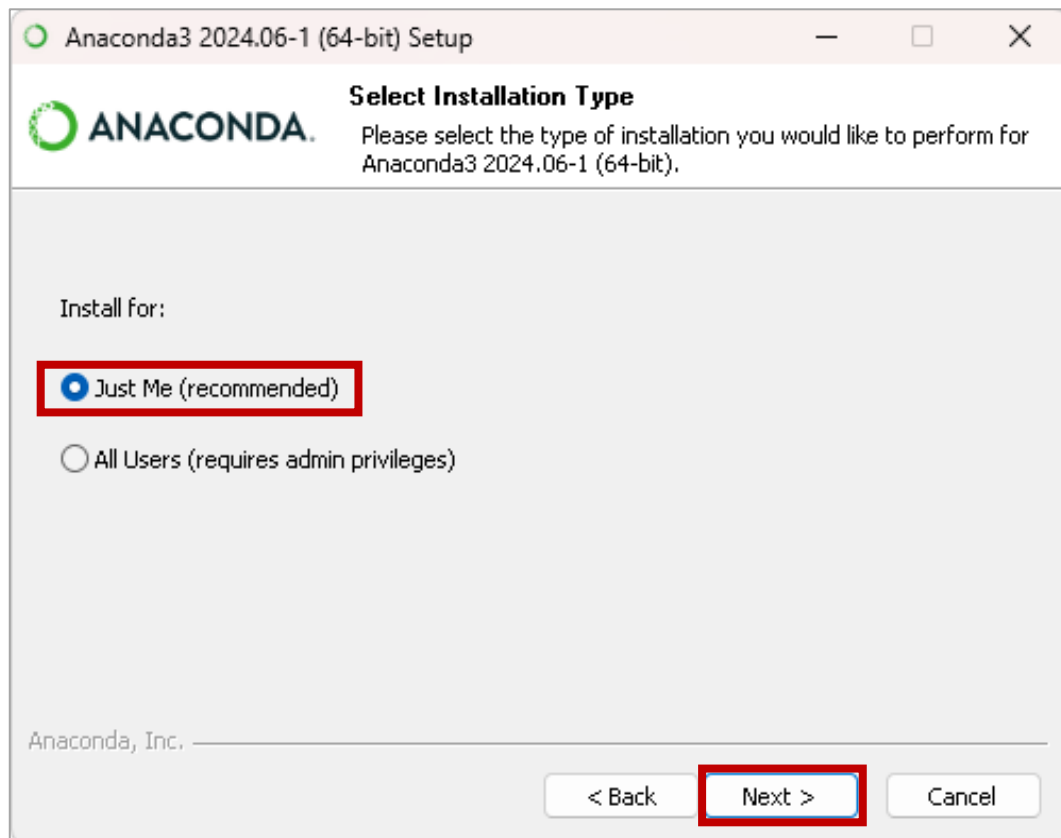
- <https://www.anaconda.com/download/success> 링크에서 아나콘다 다운로드



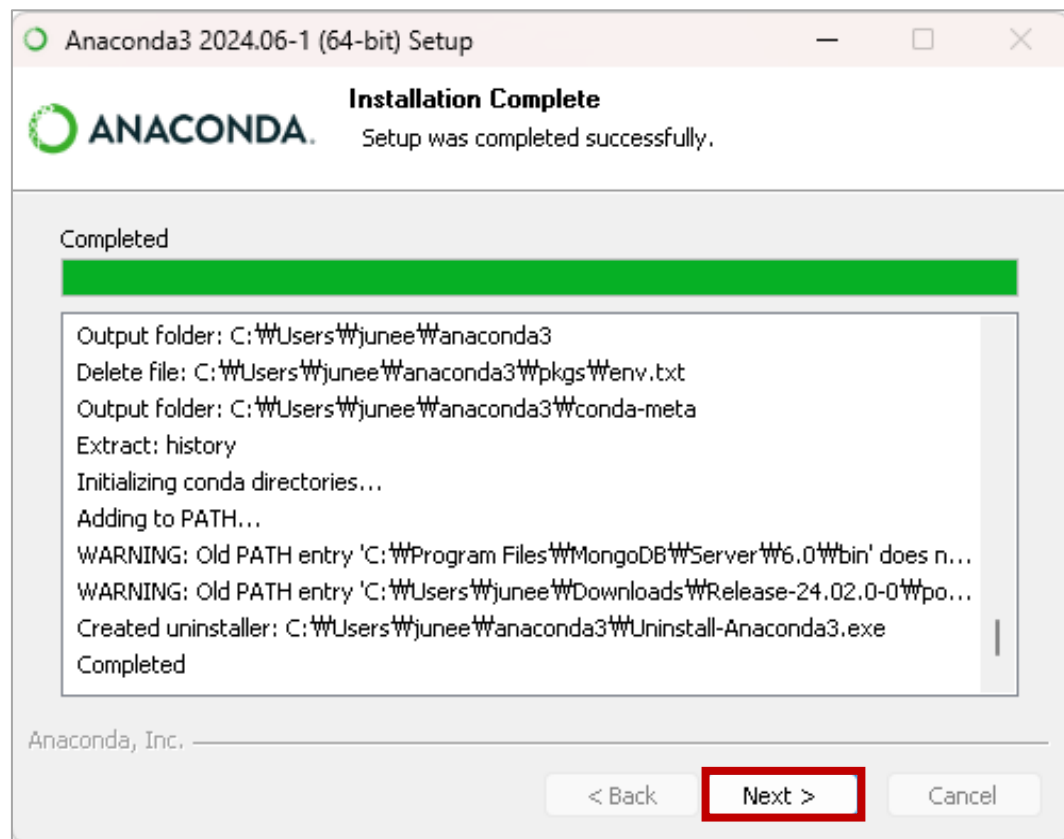
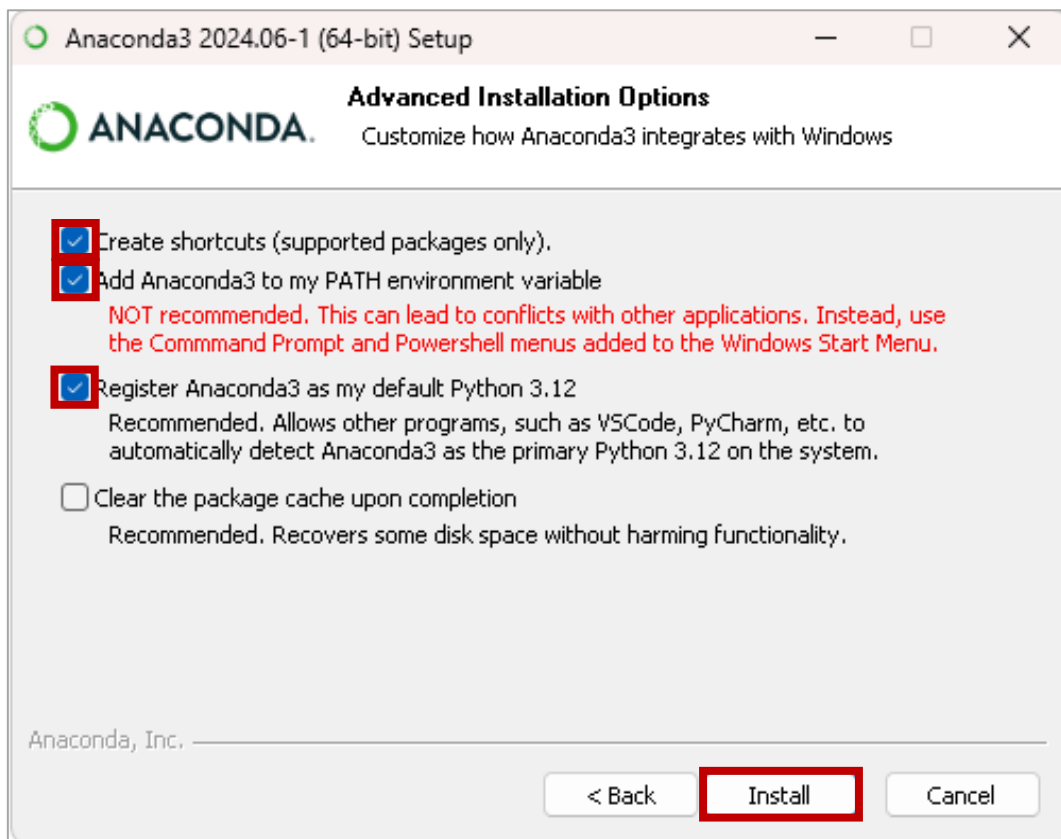
아나콘다(anaconda) 설치



아나콘다(anaconda) 설치



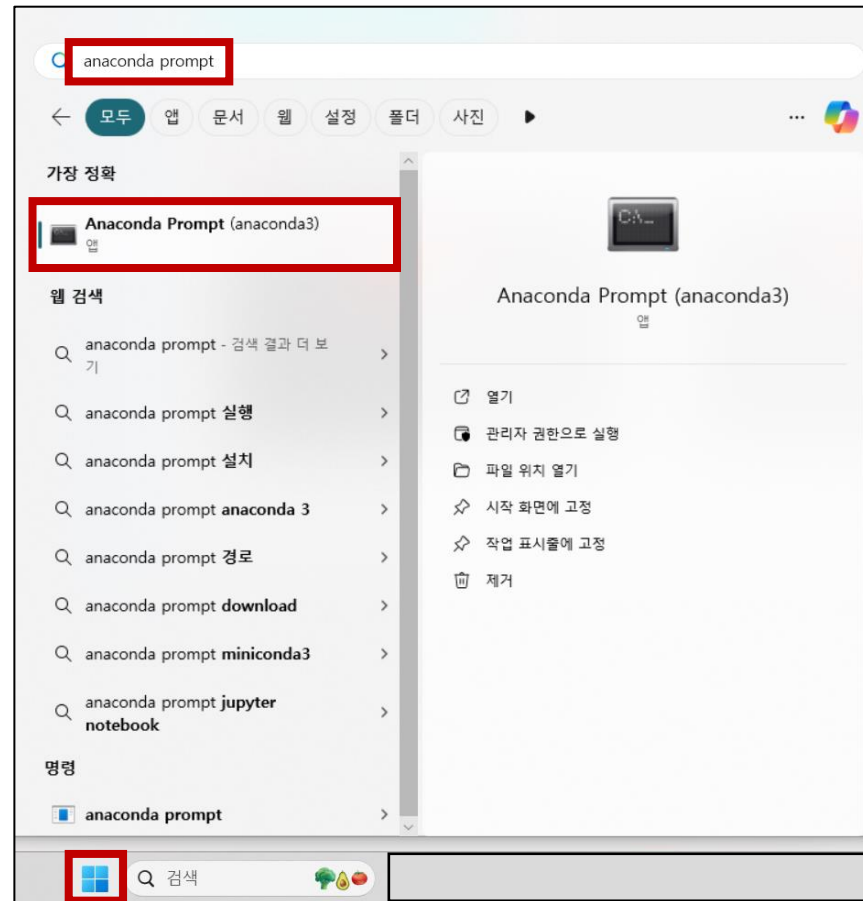
아나콘다(anaconda) 설치



아나콘다(Anaconda) 설치 확인

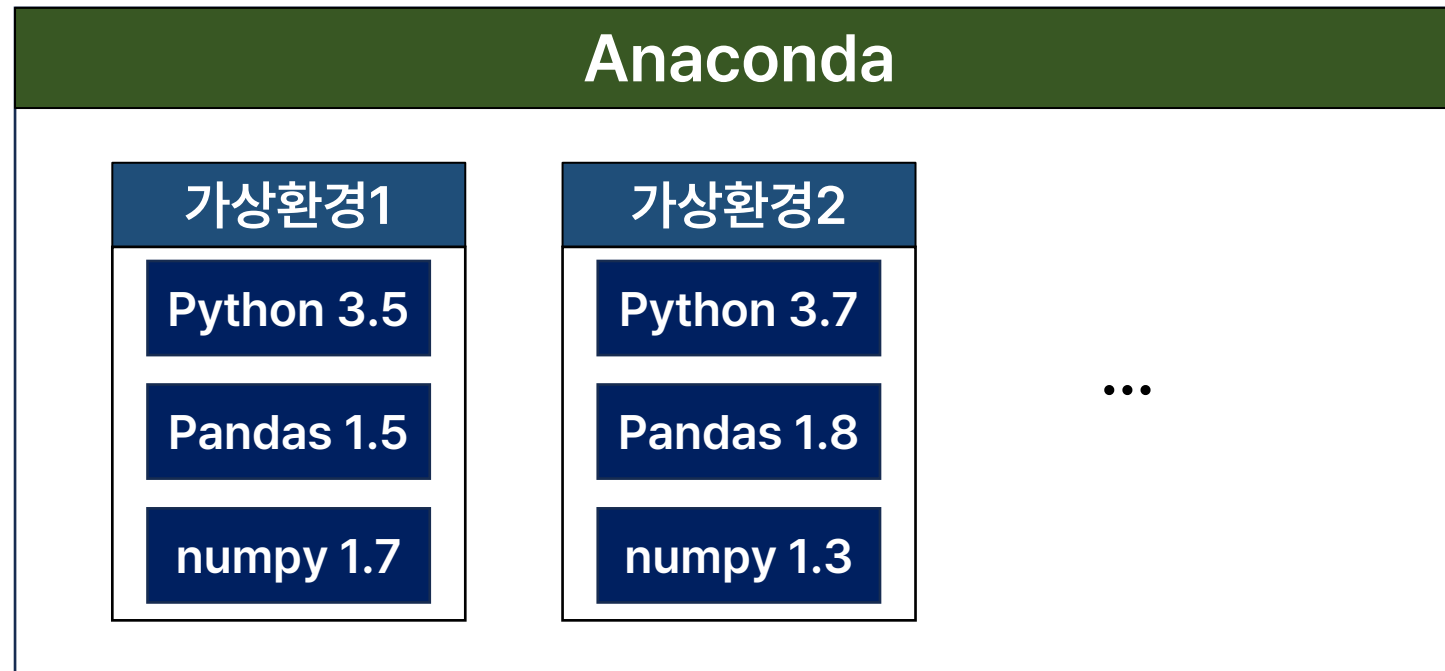
■ 아나콘다 실행하기

- 시작 메뉴 → Anaconda Prompt 실행



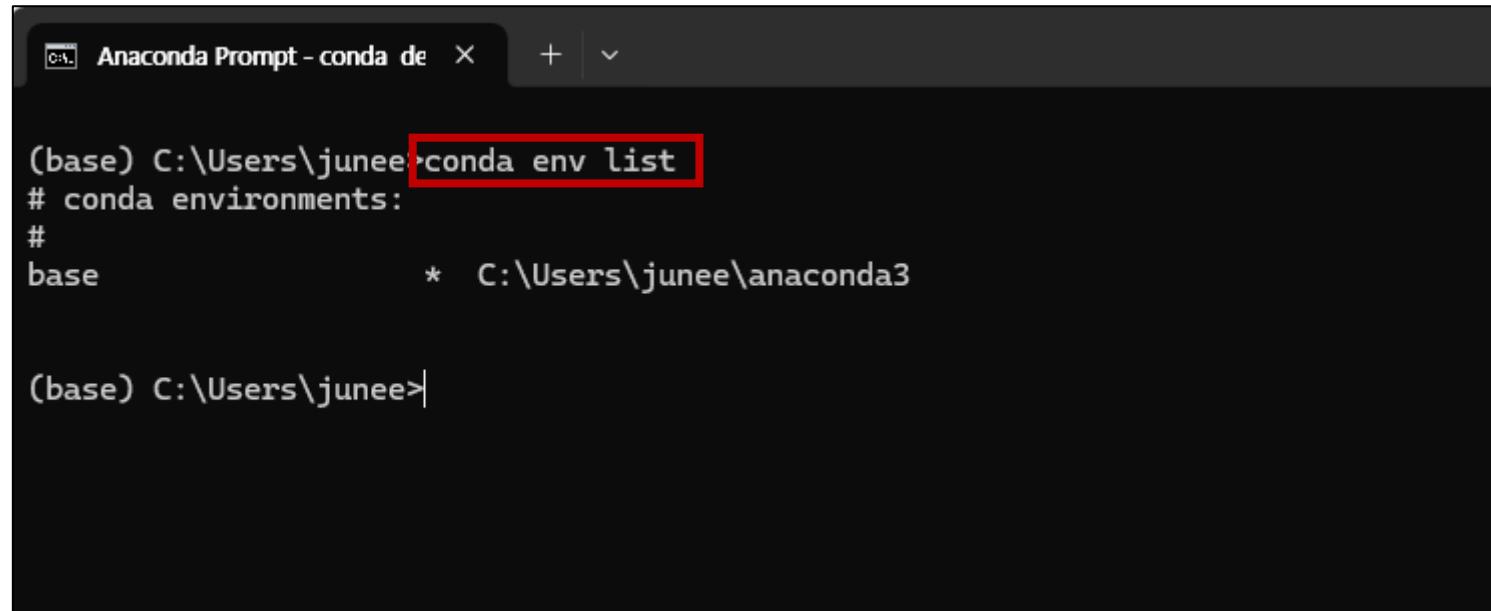
아나콘다(Anaconda) 가상환경

- 서로 다른 버전의 Python 및 패키지를 운영하고자 하는 경우 유용함
- 각 가상환경이 독립적으로 존재하여 다른 환경에 영향을 주지 않음
- 가상환경 생성 및 삭제가 용이함



아나콘다(Anaconda) 가상환경 구축 (1/6)

- 아나콘다 가상환경 목록 확인
 - conda env list



```
Anaconda Prompt - conda de X + v
(base) C:\Users\june>conda env list
# conda environments:
#
base                * C:\Users\june\anaconda3

(base) C:\Users\june>
```


아나콘다(Anaconda) 가상환경 구축 (2/6)

■ 아나콘다 가상환경 생성

- conda create -n [가상환경이름] [설치할패키지]

```
Anaconda Prompt - conda de X + v
(base) C:\Users\june: conda create -n test python=3.7
Channels:
- defaults
Platform: win-64
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

environment location: C:\Users\june\anaconda3\envs\test

added / updated specs:
- python=3.7

The following NEW packages will be INSTALLED:
```

```
Anaconda Prompt - conda de X + v
openssl pkgs/main/win-64::openssl-1.1.1w-h2
pip pkgs/main/win-64::pip-22.3.1-py37ha
python pkgs/main/win-64::python-3.7.16-h62
setuptools pkgs/main/win-64::setuptools-65.6.3
sqlite pkgs/main/win-64::sqlite-3.45.3-h2b
vc pkgs/main/win-64::vc-14.2-h2eaa2aa_
vs2015_runtime pkgs/main/win-64::vs2015_runtime-14
wheel pkgs/main/win-64::wheel-0.38.4-py37
wincertstore pkgs/main/win-64::wincertstore-0.2-

Proceed ([y]/n)? y

Downloading and Extracting Packages:

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
```

아나콘다(Anaconda) 가상환경 구축 (3/6)

■ 아나콘다 가상환경 삭제

- conda remove --name [가상환경이름] --all

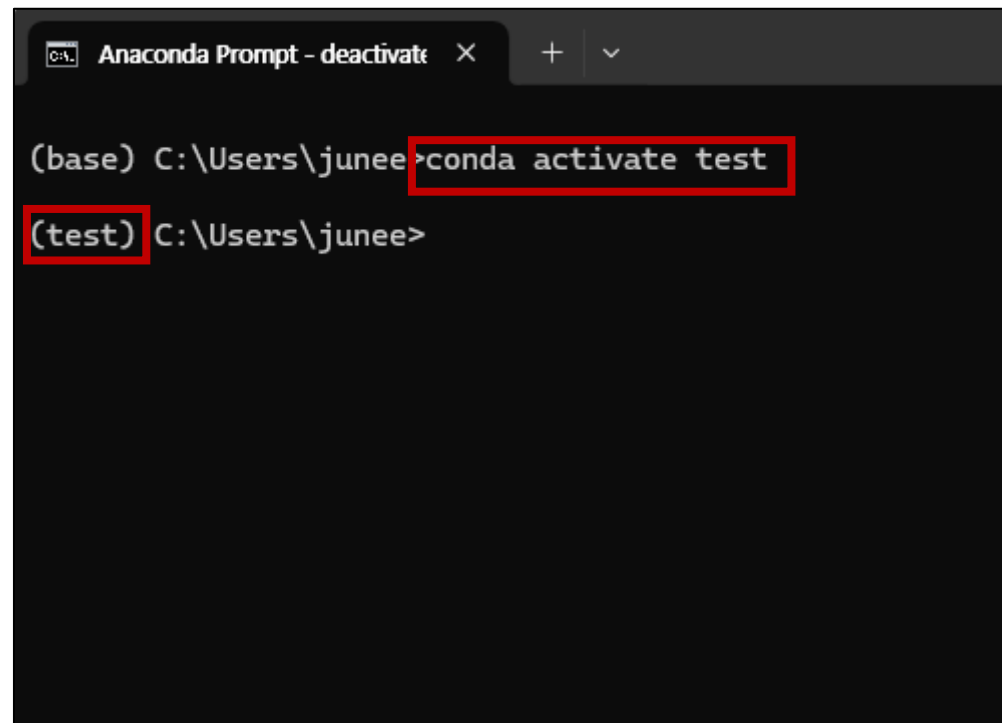
```
Anaconda Prompt - deactivate x + v
(base) C:\Users\june> conda remove --name test --all
Remove all packages in environment C:\Users\june\ana
## Package Plan ##
environment location: C:\Users\june\anaconda3\envs
The following packages will be REMOVED:
```

```
xz-5.4.6-h8cc25b3_1
zlib-1.2.13-h8cc25b3_1
zstd-1.5.5-hd43e919_2
Proceed ([y]/n)? y
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
Everything found within the environment (
ions and any non-conda files, will be del
(y/[n])? y
(base) C:\Users\june>
```

아나콘다(Anaconda) 가상환경 구축 (4/6)

■ 아나콘다 가상환경 활성화

- conda activate [가상환경이름]

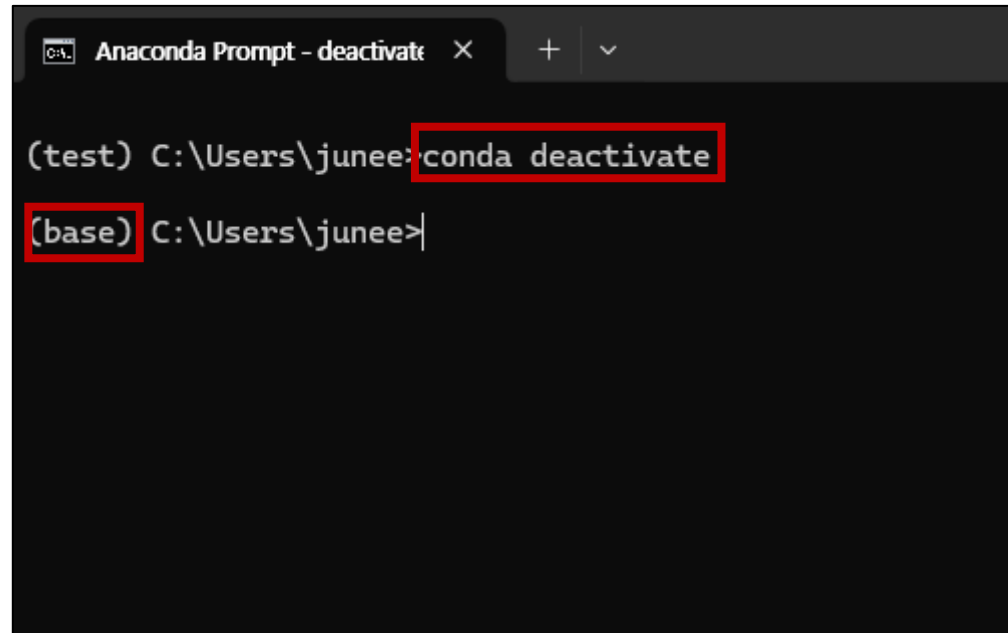


```
Anaconda Prompt - deactivate X + v
(base) C:\Users\june>conda activate test
(test) C:\Users\june>
```

The screenshot shows a terminal window titled "Anaconda Prompt - deactivate X". The prompt is "(base) C:\Users\june>". The command "conda activate test" is entered and highlighted with a red box. The prompt changes to "(test) C:\Users\june>", which is also highlighted with a red box.

아나콘다(Anaconda) 가상환경 구축 (5/6)

- 아나콘다 가상환경 비활성화
 - conda deactivate

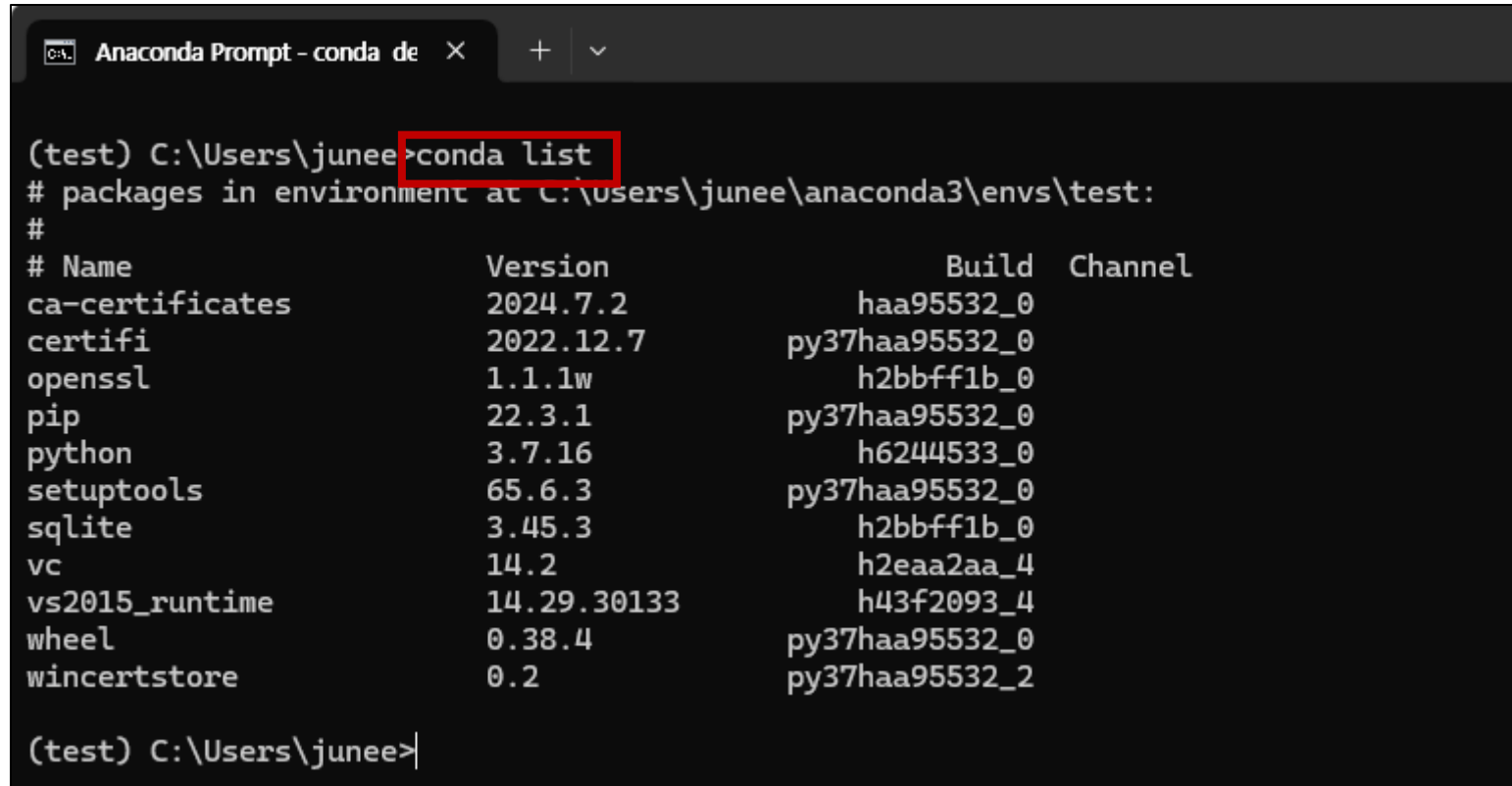


The screenshot shows a terminal window titled "Anaconda Prompt - deactivate". The prompt is "(test) C:\Users\junees>". The command "conda deactivate" is entered and highlighted with a red box. The prompt then changes to "(base) C:\Users\junees>".

```
(test) C:\Users\junees> conda deactivate
(base) C:\Users\junees>
```

아나콘다(Anaconda) 가상환경 구축 (6/6)

- 현재 활성화된 가상환경에 설치된 패키지 확인
 - conda list



```
(test) C:\Users\junees>conda list
# packages in environment at C:\Users\junees\anaconda3\envs\test:
#
# Name                      Version           Build    Channel
ca-certificates             2024.7.2          haa95532_0
certifi                     2022.12.7         py37haa95532_0
openssl                     1.1.1w            h2bbff1b_0
pip                         22.3.1            py37haa95532_0
python                      3.7.16            h6244533_0
setuptools                  65.6.3            py37haa95532_0
sqlite                      3.45.3            h2bbff1b_0
vc                          14.2              h2eaa2aa_4
vs2015_runtime              14.29.30133       h43f2093_4
wheel                      0.38.4            py37haa95532_0
wincertstore                0.2               py37haa95532_2

(test) C:\Users\junees>
```

JupyterLab 사용법

■ JupyterLab 실행

- jupyter lab
- jupyter lab 입력 후 프롬프트 창에 생기는 URL 중 하나를 Ctrl 키와 함께 클릭해도 접속 가능
- **아나콘다 프롬프트 창은 JupyterLab을 사용하는 동안에는 종료하면 안 되니 주의!**

```
Anaconda Prompt - deactivate x + v
(test) C:\Users\juneer>jupyter lab
[W 2024-08-05 10:22:58.453 ServerApp] A '_jupyter_server_extensions' function was found and
d in future releases of Jupyter Server.
[W 2024-08-05 10:22:58.489 ServerApp] A '_jupyter_server_extensions' function was found and
ted in future releases of Jupyter Server.
[I 2024-08-05 10:23:00.235 ServerApp] Extension package panel.
[I 2024-08-05 10:23:00.235 ServerApp] jupyter_lsp | extension
[I 2024-08-05 10:23:00.245 ServerApp] jupyter_server_terminals
[I 2024-08-05 10:23:00.253 ServerApp] jupyterlab | extension w
[I 2024-08-05 10:23:00.261 ServerApp] notebook | extension was
```

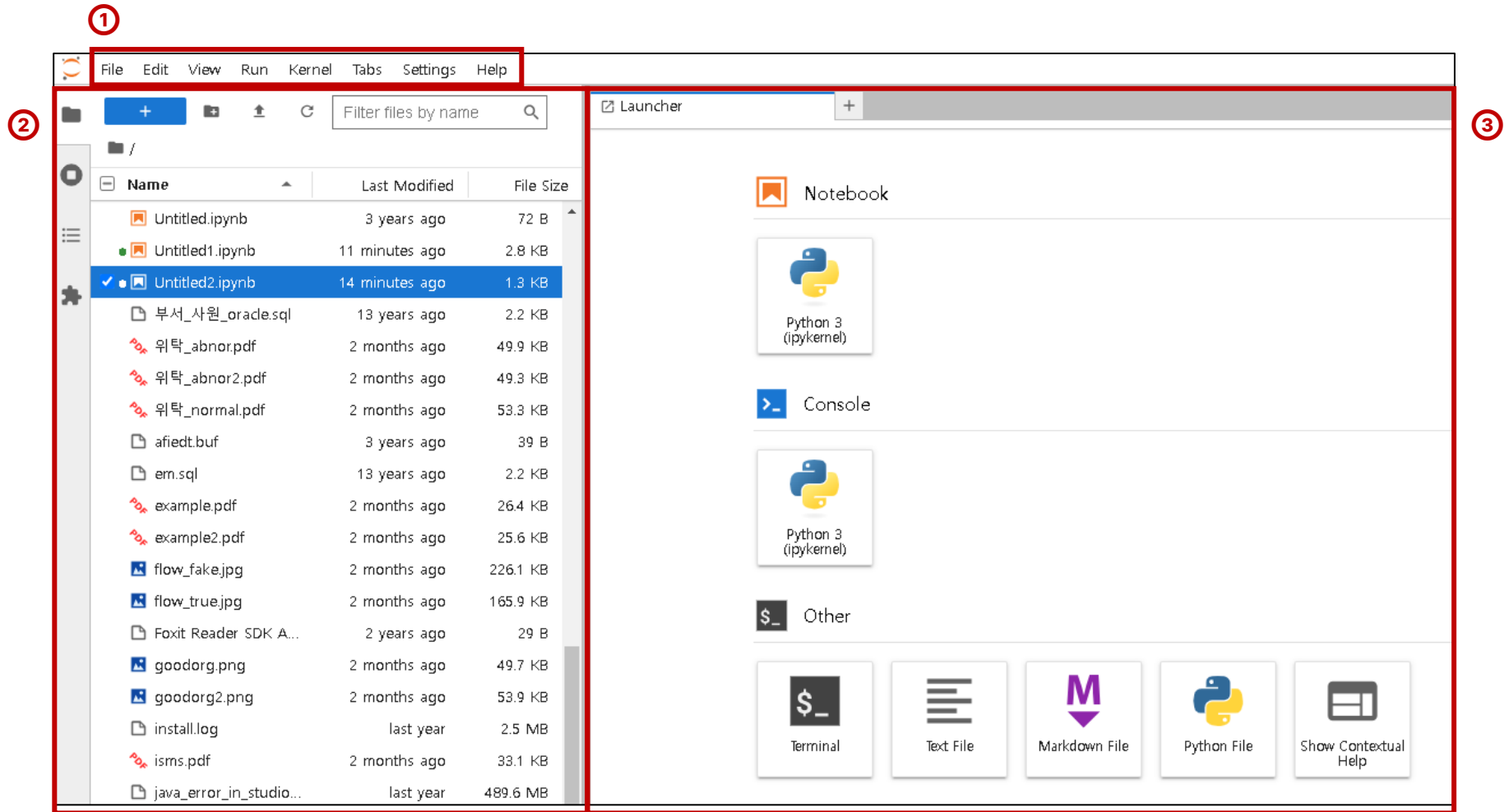
```
Anaconda Prompt - deactivate x + v
[I 2024-08-05 10:23:01.194 ServerApp] Serving notebooks from local directory: C:\Users\juneer
[I 2024-08-05 10:23:01.194 ServerApp] Jupyter Server 2.14.1 is running at:
[I 2024-08-05 10:23:01.196 ServerApp] http://localhost:8888/lab?token=e0d4552b705e3254914cb3f8
[I 2024-08-05 10:23:01.196 ServerApp] http://127.0.0.1:8888/lab?token=e0d4552b705e3254914c
fb
[I 2024-08-05 10:23:01.196 ServerApp] Use Control-C to stop this server and shut down all kern
ation).
[C 2024-08-05 10:23:01.278 ServerApp]

To access the server, open this file in a browser:
file:///C:/Users/juneer/AppData/Roaming/jupyter/runtime/jpserver-1152-open.html
Or copy and paste one of these URLs:
http://localhost:8888/lab?token=e0d4552b705e3254914cb3f8f1e47b2cfba73d1f5f273afb
http://127.0.0.1:8888/lab?token=e0d4552b705e3254914cb3f8f1e47b2cfba73d1f5f273afb
[I 2024-08-05 10:23:01.820 ServerApp] Skipped non-installed server(s): basn-language-server, c
nodejs, javascript-typescript-langserver, jedi-language-server, julia-language-server, pyright
r-languageserver, sql-language-server, texlab, typescript-language-server, unified-language-se
server-bin, vscode-html-languageserver-bin, vscode-json-languageserver-bin, yaml-language-serve
0.03s - Debugger warning: It seems that frozen modules are being used, which may
```

JupyterLab 사용법

■ 화면 구성

- ① 메인 메뉴
- ② 사이드 메뉴
- ③ 작업 영역



JupyterLab 사용법

■ 사이드 메뉴

① File Browser

- 현재 디렉터리의 파일들을 보여줌
- 상단의 버튼을 이용해 파일, 폴더를 추가하거나 업로드, 동기화 할 수 있음

② Running Terminal and Kernels

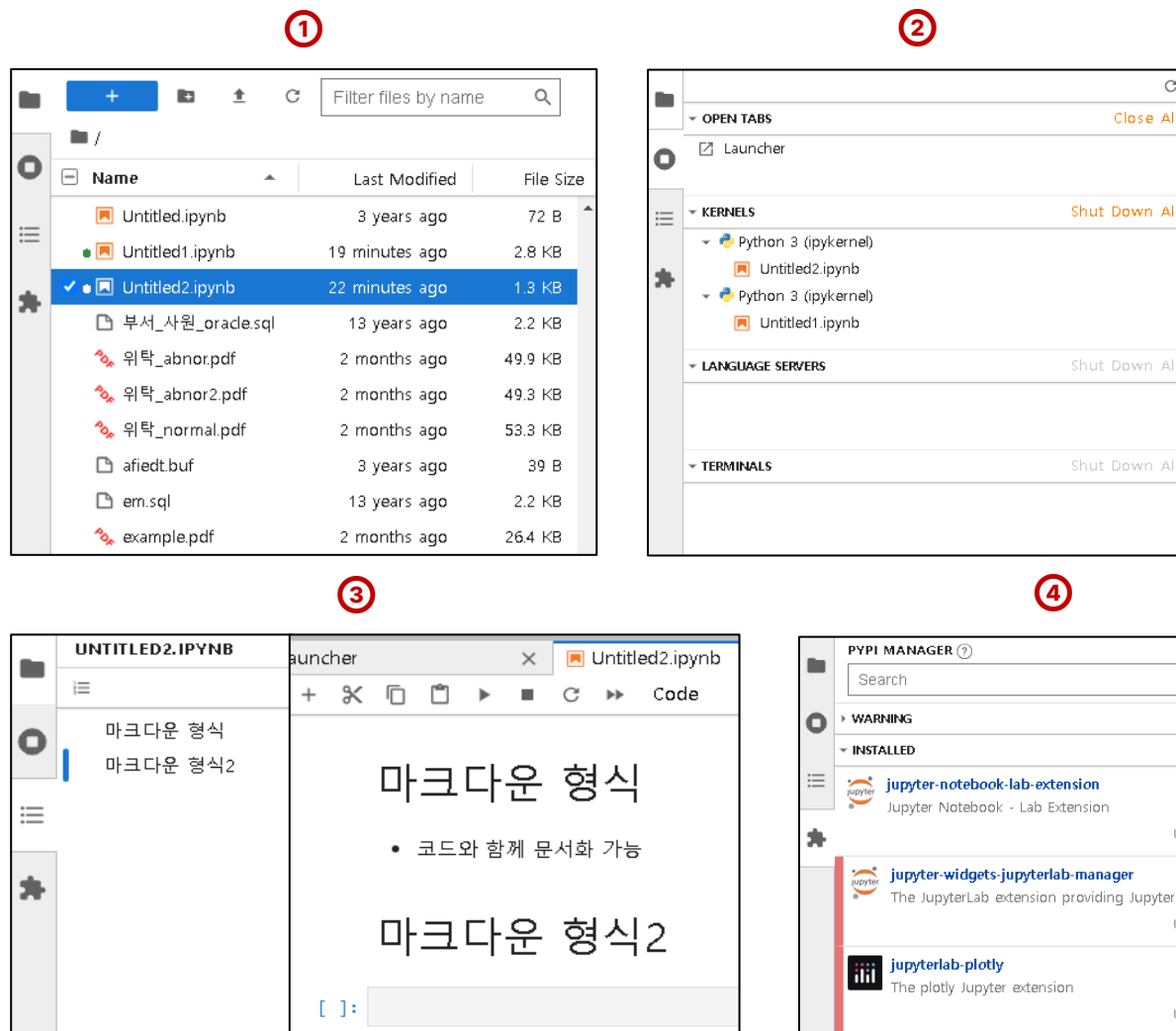
- 열려 있는 탭이나 커널을 보여주고 종료 시킬 수 있음

③ Table of Contents

- 코드, 마크다운 등의 콘텐츠를 요약해서 보여줌

④ Extension manager

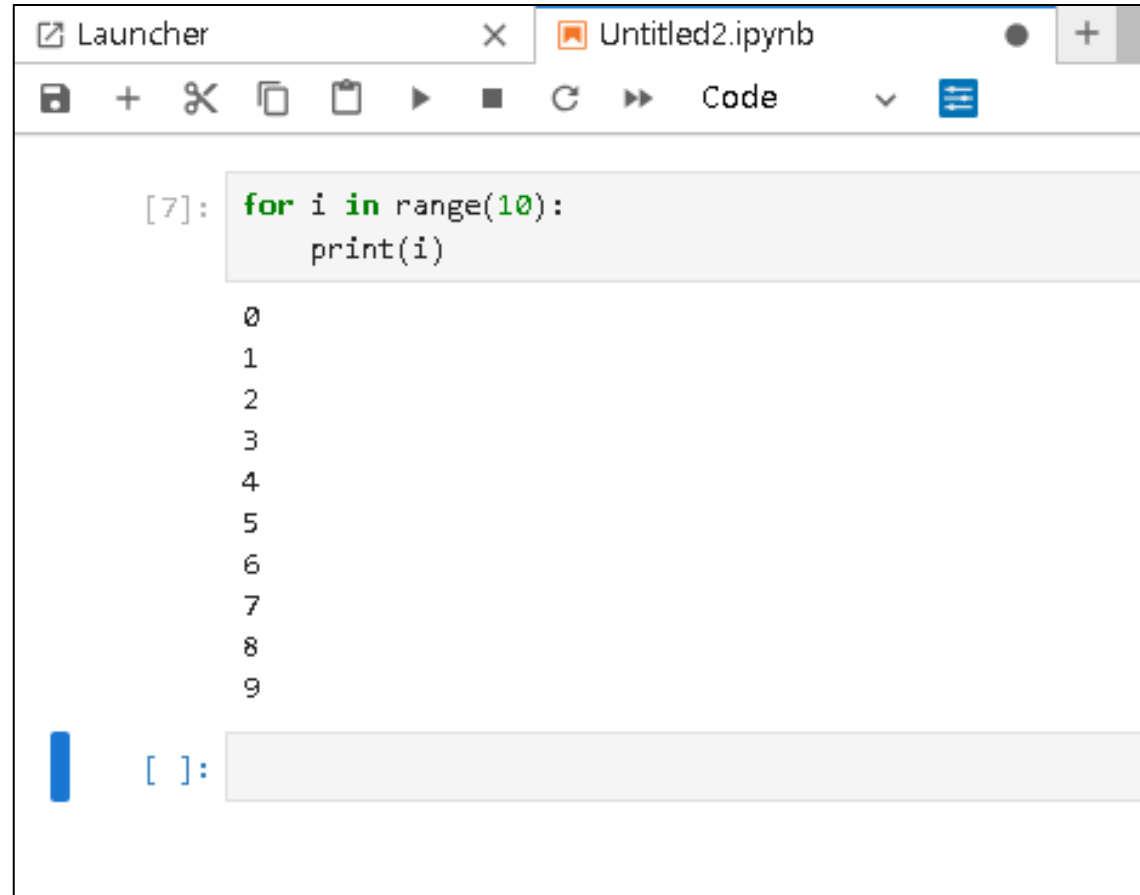
- 추가 Extension들을 설치하고 관리할 수 있음



JupyterLab 사용법

■ 작업영역

- 상단의 탭을 이용하여 여러 개의 파일을 이동
- 저장, 추가, 복사, 실행 등 버튼으로 사용가능
- ipynb이나 markdown등을 활용하여 다양한 작업이 가능한 영역
- 화면구성을 분할하여 사용도 가능
- 대부분의 작업을 하는 영역이기 때문에, 단축키를 숙지해 두는 것이 좋음



The screenshot displays the JupyterLab application window. At the top, there are two tabs: 'Launcher' and 'Untitled2.ipynb'. Below the tabs is a toolbar with icons for file operations (save, add, delete, copy, paste) and execution (run, step through, interrupt). The main area shows a code cell with the prompt '[7]:' and the following Python code:

```
for i in range(10):  
    print(i)
```

 The output of the code is displayed below the cell, showing the numbers 0 through 9 on separate lines. At the bottom, there is an empty code cell with the prompt '[]:' and a blue cursor icon to its left.

마크다운(Markdown)

- 마크다운(Markdown)은 일반 텍스트 기반의 경량 마크업 언어
 - 마크업 언어는 태그 등을 이용하여 문서나 데이터의 구조를 명기하는 언어의 한 가지
- 일반 텍스트로 서식이 있는 문서를 작성하는 데 사용되며, 일반 마크업 언어에 비해 문법이 쉽고 간단한 것이 특징

```

## 마크다운 장단점
- 문법이 쉬움
- 관리가 간편
- 지원 가능 플랫폼 및 프로그램이 다양
- 표준이 없어 작성 문법이 조금씩 상이
- 모든 HTML 마크업을 대체할 수 없음

## 그 외 사용법

이텔릭체는 *별* 혹은 _언더바_ 를 사용

두껍게는 **별** 혹은 __언더바__ 를 2번 사용

취소선은 ~~~물결거호~~~ 를 사용

<u>밑줄</u>은 지원하지 않기에 직접 "<u></u>" 태그를 사용
  
```



```

<h2>마크다운 장단점</h2>
<ul>
<li>• 문법이 쉬움</li>
<li>• 관리가 간편</li>
<li>• 지원 가능 플랫폼 및 프로그램이 다양</li>
<li>• 표준이 없어 작성 문법이 조금씩 상이</li>
<li>• 모든 HTML 마크업을 대체할 수 없음</li>
</ul>

<h2>그 외 사용법</h2>

이텔릭체는 *별* 혹은 _언더바_ 를 사용

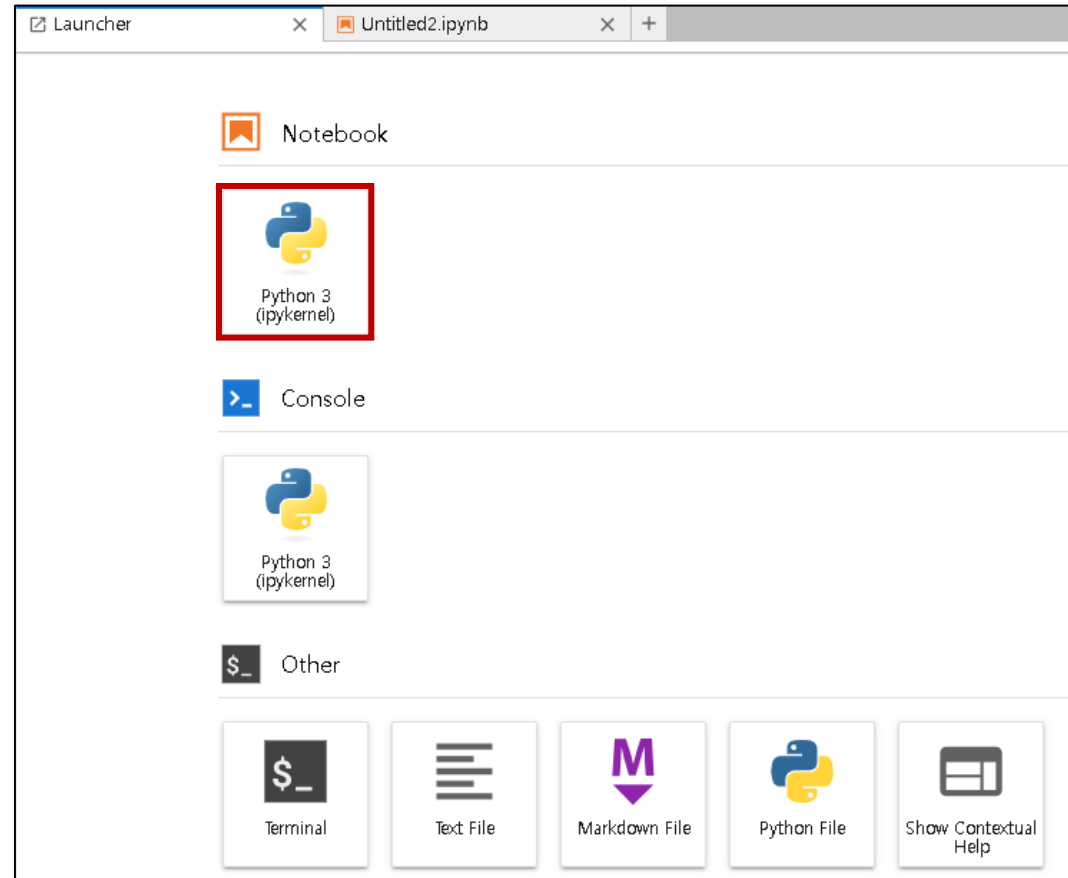
두껍게는 **별** 혹은 __언더바__ 를 2번 사용

취소선은 ~~~물결거호~~~ 를 사용

<u>밑줄</u>은 지원하지 않기에 직접 "<u></u>" 태그를 사용
  
```

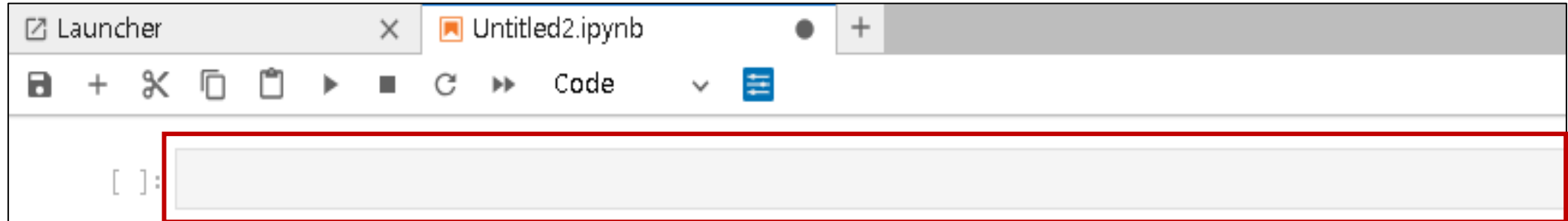
Notebook 사용법

- Launcher에서 Notebook 항목의 python3 클릭하여 notebook 생성



Notebook 사용법

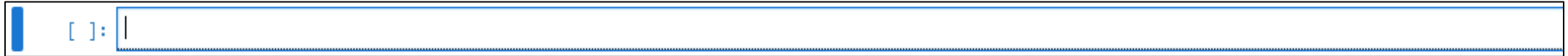
- 코드를 입력할 수 있는 부분을 '셀(cell)' 이라고 함



Notebook 사용법

▪ Edit mode

- 셀에 코드를 입력하거나 수정할 수 있는 상태



▪ Command mode

- 셀을 편집할 수 있는 상태
- 셀을 추가하거나 삭제할 수 있는 상태



Notebook 사용법

▪ command mode 단축키

a : above 위에 셀 생성

b : below 아래 셀 생성

c : copy 셀 복사

v : paste 셀 붙여넣기

d d : delete 삭제(두번)

z : undo 되돌리기

shift + z : 다시 실행

0 0 : restart kernel

enter : 셀 수정

shift + enter : 셀 실행

y : code로 변경

m : 마크다운으로 변경

03 파이썬 기초



변수(변하는 수) 이해하기

- 다양한 값을 지닌 하나의 속성 → 소득, 성별, 학점
- 변수 간의 관계를 파악하여 데이터 분석을 진행 ⇒ 데이터 분석의 대상
- 상수
 - 하나의 값으로만 되어 있는 속성
 - 변수와 달리 분석 대상이 될 수 없음

변수			상수
소득	성별	학점	국적
1,000만 원	남자	3.8	대한민국
2,000만 원	남자	4.2	대한민국
3,000만 원	여자	2.6	대한민국
4,000만 원	여자	4.5	대한민국

변수(변하는 수) 이해하기

- 변수를 생성한 후 변수를 이용하여 연산 가능
- 변수끼리는 연산할 수도 있고, 변수와 숫자를 조합해 연산도 가능

```
[5]: b = 2  
b
```

```
[5]: 2
```

```
[6]: a+b
```

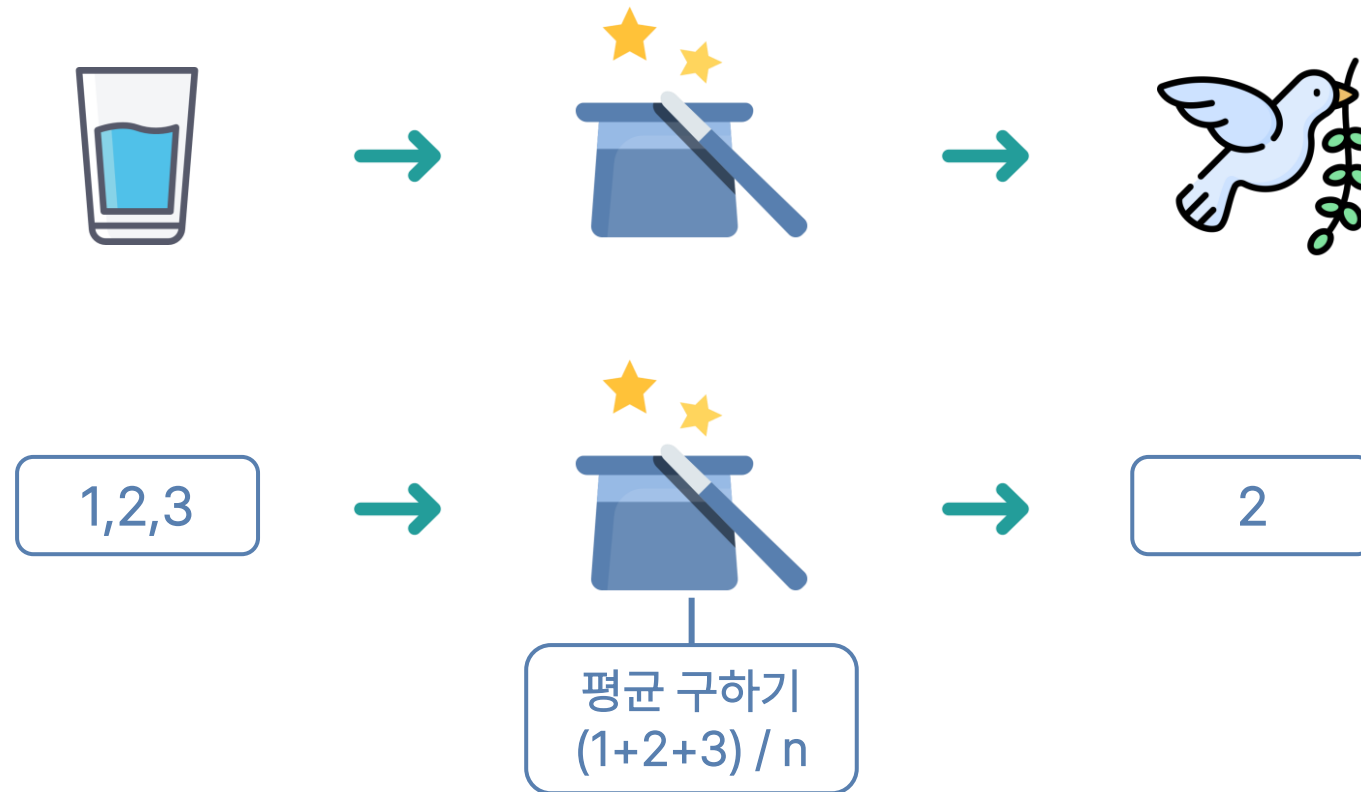
```
[6]: 3
```

```
[7]: 4/b
```

```
[7]: 2.0
```

함수 이해하기

- 함수에 값을 넣으면 특정한 기능을 수행해 처음과 다른 값을 결과값으로 확인 가능



함수 이해하기

- 함수에 값을 넣으면 특정한 기능을 수행해 처음과 다른 값을 결과값으로 확인 가능

```
[8]: x = [1,2,3]  
x
```

```
[8]: [1, 2, 3]
```

```
[9]: sum(x)
```

```
[9]: 6
```

```
[10]: x_sum = sum(x)  
x_sum
```

```
[10]: 6
```

패키지(함수 꾸러미) 이해하기

- 패키지에는 다양한 함수가 내장되어 있음
- 패키지 설치하는 한 번만 진행하면 되지만 로드하는 작업은 JupyterLab을 새로 시작할 때마다 반복

패키지 설치하기



패키지 로드하기



함수 사용하기

패키지(함수 꾸러미) 이해하기

- 아나콘다에는 데이터 분석에 필요한 주요 패키지가 포함
- 아나콘다에 포함된 패키지를 사용할 경우 설치 과정 생략 후 진행 가능

패키지 설치하기



패키지 로드하기

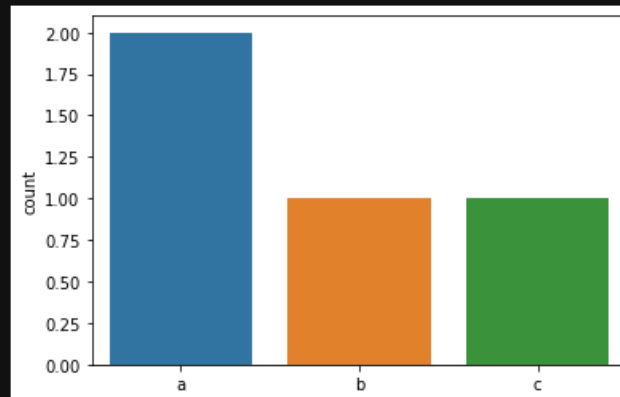


함수 사용하기

```
[11]: import seaborn
```

```
[14]: var = ['a', 'a', 'b', 'c']  
seaborn.countplot(x=var)
```

```
[14]: <AxesSubplot:ylabel='count'>
```



패키지(함수 꾸러미) 이해하기

- 패키지에는 함수의 기능을 테스트할 수 있는 예제 데이터가 내장되어 있음
- seaborn 패키지에 들어있는 titanic 데이터를 이용하여 그래프 만들기

데이터 불러오기

```
import seaborn as sns
df = sns.load_dataset('titanic')
df
```

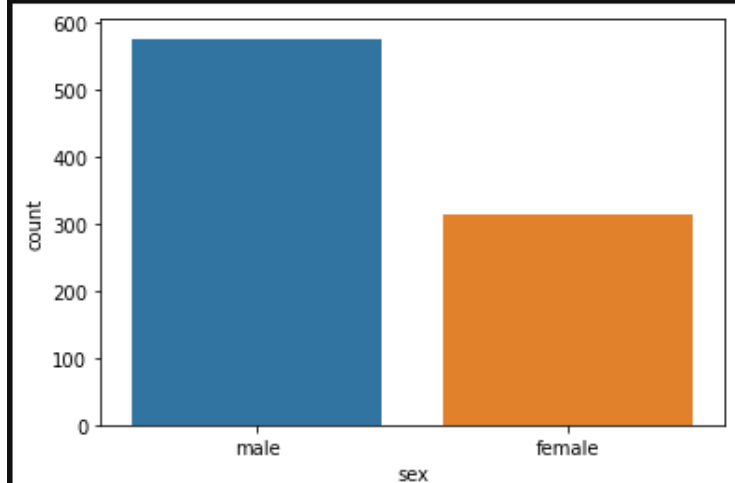
	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN
...
886	0	2	male	27.0	0	0	13.0000	S	Second	man	True	NaN
887	1	1	female	19.0	0	0	30.0000	S	First	woman	False	B
888	0	3	female	NaN	1	2	23.4500	S	Third	woman	False	NaN



그래프 그리기

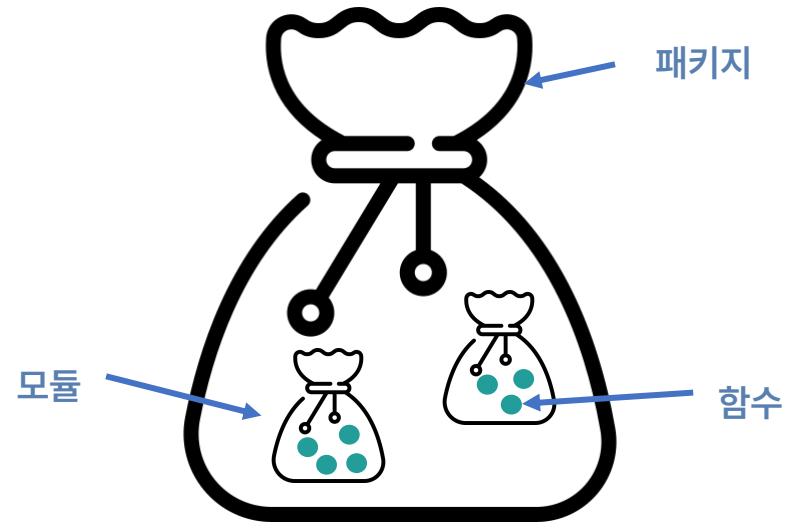
```
sns.countplot(data=df, x='sex')
```

<AxesSubplot:xlabel='sex', ylabel='count'>



패키지(함수 꾸러미) 이해하기

- 패키지라는 큰 꾸러미에 비슷한 함수들을 넣어둔 작은 꾸러미 → 비슷한 함수끼리 묶어 놓은 것



- 모듈 불러오기 → 패키지명.모듈명

```
import sklearn.metrics
```

패키지(함수 꾸러미) 이해하기

- 패키지명.모듈명.함수명()으로 함수 사용하기

```
sklearn.metrics.accuracy_score()
```

- 모듈명.함수명()으로 함수 사용하기

```
from sklearn import metrics  
metrics.accuracy_score()
```

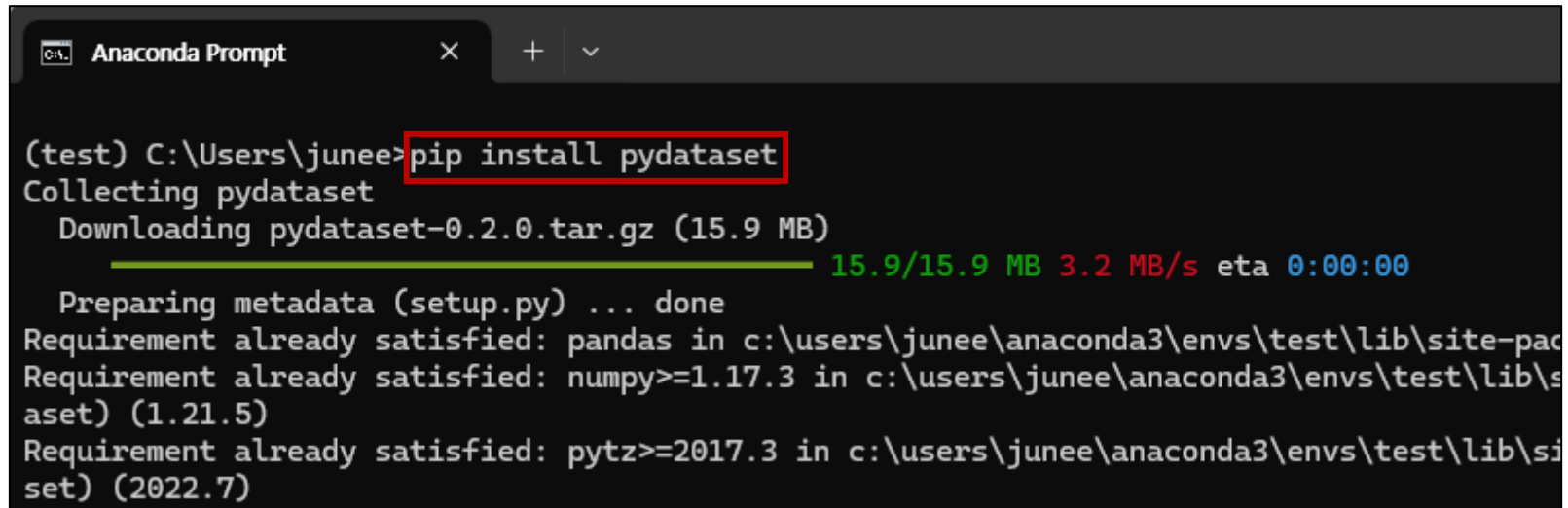
- 함수명()으로 함수 사용하기

```
from sklearn.metrics import accuracy_score  
accuracy_score()
```


패키지(함수 꾸러미) 이해하기

패키지 설치 방법

- pip install [원하는 패키지]

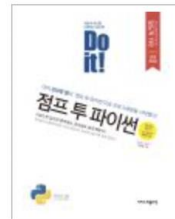


```
Anaconda Prompt
(test) C:\Users\june> pip install pydataset
Collecting pydataset
  Downloading pydataset-0.2.0.tar.gz (15.9 MB)
    15.9/15.9 MB 3.2 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: pandas in c:\users\june\anaconda3\envs\test\lib\site-packages (1.21.5)
Requirement already satisfied: numpy>=1.17.3 in c:\users\june\anaconda3\envs\test\lib\site-packages (1.21.5)
Requirement already satisfied: pytz>=2017.3 in c:\users\june\anaconda3\envs\test\lib\site-packages (2022.7)
```

패키지(함수 꾸러미) 이해하기

- 파이썬 기초 실습(<https://wikidocs.net/book/1>)

점프 투 파이썬



지은이 : 박응용

최종 편집일시 : 2022년 8월 6일 7:21 오후

저작권 : (CC) BY-NC-ND

e-book 판매가 : 5,000원 (구매하기)

👍 4,072 명이 추천

점프 투 파이썬 오프라인 책(개정판) 출간 !! (2019.06)

- 책 구입 안내

이 책은 파이썬이란 언어를 처음 접해보는 독자들과 프로그래밍을 한 번도 해 본적이 없는 사람들을 대상으로 한다. 프로그래밍을 할 때 사용되는 전문적인 용어들을 알기 쉽게 풀어서 쓰려고 노력하였으며, 파이썬이란 언어의 개별적인 특성만을 강조하지 않고 프로그래밍 전반에 관한 사항을 파이썬이란 언어를 통해 알 수 있도록 알기 쉽게 설명하였다.

파이썬에 대한 기본적인 지식을 알고 있는 사람이라도 이 책은 파이썬 프로그래밍에 대한 흥미를 가질 수 있는 좋은 안내서가 될 것이다. 이 책의 목표는 독자가 파이썬을 통해 프로그래밍에 대한 전반적인 이해를 갖게하는 것이며, 또 파이썬이라는 도구를 이용하여 원하는 프로그램을 쉽고 재미있게 만들 수 있게 하는 것이다.

"점프 투 파이썬" 이나 파이썬에 대한 질문은 최근 오픈한 파이썬 게시판 서비스인 파이보를 활용해 보자.

데이터 프레임

- 데이터를 다룰 때 가장 많이 사용하는 데이터 형태로 행과 열로 구성되어 있음
- 열
 - 세로로 나열되며 속성을 나타냄
 - 컬럼(Column) 또는 변수(Variable)이라고 불림
- 행
 - 가로로 나열되며 각 항목의 정보를 나타냄
 - 로(row) 또는 케이스(case)라고 불림



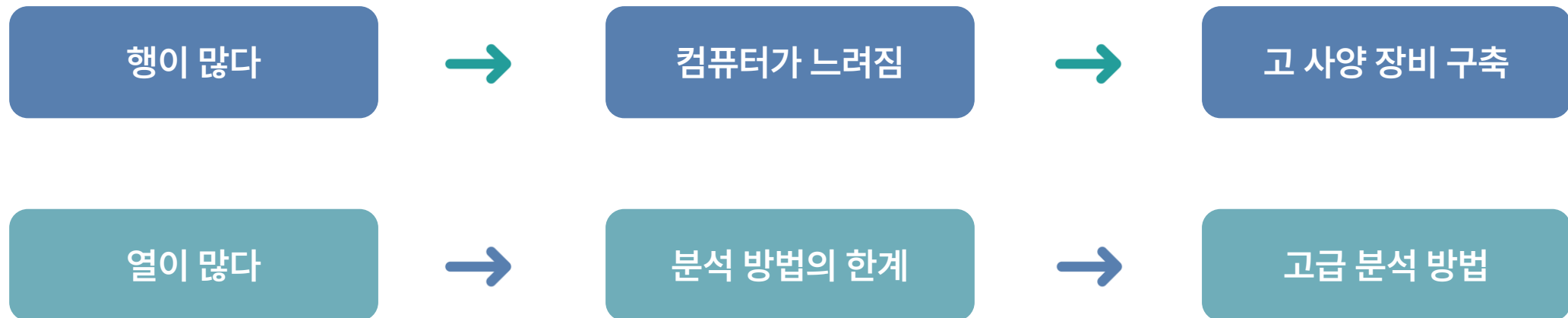
성별	연령	학점	연봉
남자	26	3.8	2,700만원
여자	42	4.2	4,000만원
남자	35	2.6	3,500만원

→ 4개의 열과 3개의 행으로 구성

데이터 프레임

- 행이 늘어나더라도 분석 기술 면에서는 별다른 차이가 생기지 않음
- 반면 열이 늘어난다면 변수를 조합할 수 있는 경우의 수가 늘어남

⇒ 데이터 분석에서는 데이터의 양을 의미하는 행보다 데이터의 다양성을 의미하는 열이 많은 것이 더 중요



데이터 프레임

- pandas를 이용하여 데이터 프레임 만들기
 - pandas : 데이터 가공 시 사용하는 패키지

패키지 로드하기



데이터 프레임 만들기

```
import pandas as pd
```

```
df = pd.DataFrame({'name' : ['김지훈', '이유진', '박동현', '김민지'],  
                  'english' : [90, 80, 60, 70],  
                  'math' : [50, 60, 100, 20]})
```

df

	name	english	math
0	김지훈	90	50
1	이유진	80	60
2	박동현	60	100
3	김민지	70	20

데이터 프레임

특정 변수의 값 추출



변수의 값을 이용하여
연산하기

```
df['english']
```

```
0    90  
1    80  
2    60  
3    70
```

```
Name: english, dtype: int64
```

```
sum(df['english'])
```

```
300
```

```
sum(df['english'])/4
```

```
75.0
```

데이터 프레임

- CSV(Comma-Separated Values)는 몇 가지 필드를 쉼표(,)로 구분한 텍스트 데이터 및 텍스트 파일

CSV 파일 확인



CSV 파일 불러오기

excel_exam.csv • 저장됨

검색(Alt+Q)

파일 홈 삽입 페이지 레이아웃 수식 데이터 검토 보기 도움말

잘라내기 붙여넣기 복사 서식 복사

맑은 고딕 11

가 자동 줄 바꿈 병합하고 가운데 맞춤

클립보드 글꼴 맞춤

E14

	A	B	C	D	E	F	G	H	I	J
1	id	nclass	math	english	science					
2	1	1	50	98	50					
3	2	1	60	97	60					
4	3	1	45	86	78					
5	4	1	30	98	58					
6	5	2	25	80	65					
7	6	2	50	89	98					
8	7	2	80	90	45					
9	8	2	90	78	25					
10	9	3	20	98	15					
11	10	3	50	98	45					
12	11	3	65	65	65					
13	12	3	45	85	32					

```
df_exam = pd.read_csv('excel_exam.csv')
df_exam
```

	id	nclass	math	english	science
0	1	1	50	98	50
1	2	1	60	97	60
2	3	1	45	86	78
3	4	1	30	98	58
4	5	2	25	80	65
5	6	2	50	89	98
6	7	2	80	90	45
7	8	2	90	78	25
8	9	3	20	98	15
9	10	3	50	98	45
10	11	3	65	65	65
11	12	3	45	85	32

데이터 프레임

CSV 파일 확인

첫 번째 행이 변수명이 아니라면?

	A	B	C	D	E
1	1	1	50	98	50
2	2	1	60	97	60
3	3	1	45	86	78
4	4	1	30	98	58
5	5	2	25	80	65
6	6	2	50	89	98
7	7	2	80	90	45
8	8	2	90	78	25
9	9	3	20	98	15
10	10	3	50	98	45
11	11	3	65	65	65
12	12	3	45	85	32



CSV 파일 불러오기

'header = None' 사용

```
df_exam = pd.read_csv('excel_exam.csv', header = None)
df_exam
```

```

   0  1  2  3  4
0  1  1  50 98 50
1  2  1  60 97 60
2  3  1  45 86 78
3  4  1  30 98 58
4  5  2  25 80 65
5  6  2  50 89 98
6  7  2  80 90 45
7  8  2  90 78 25
8  9  3  20 98 15
9 10  3  50 98 45
10 11  3  65 65 65
11 12  3  45 85 32
```


데이터 프레임

데이터 프레임 만들기

```
df = pd.DataFrame({'english' : [90, 80, 60, 70],
                  'math' : [50, 60, 100, 20],
                  'nclass' : [1, 1, 2, 2]})
```

df

	english	math	nclass
0	90	50	1
1	80	60	1
2	60	100	2
3	70	20	2



CSV 파일로 저장

```
df.to_csv('output_data.csv')
```

자동 저장 ☒ output_data.csv ▾

파일 홈 삽입 페이지 레이아웃 수식 데이터 검토

A1 ▾ :

	A	B	C	D	E	F
1		english	math	nclass		
2	0	90	50	1		
3	1	80	60	1		
4	2	60	100	2		
5	3	70	20	2		

데이터 프레임

CSV 파일 확인

자동 저장 ☒ output_data.csv ▾

파일 홈 삽입 페이지 레이아웃 수식 데이터 검토

A1

	A	B	C	D	E	F
1		english	math	nclass		
2	0	90	50	1		
3	1	80	60	1		
4	2	60	100	2		
5	3	70	20	2		



인덱스 번호 제외

```
df.to_csv('output_data.csv', index=False)
```

자동 저장 ☒ output_data.csv ▾

파일 홈 삽입 페이지 레이아웃 수식 데이터 검토 보

G15

	A	B	C	D	E	F
1	english	math	nclass			
2	90	50	1			
3	80	60	1			
4	60	100	2			
5	70	20	2			

데이터 프레임 명령어

▪ head()

- 앞부분 출력하는 함수
- ()에 값을 입력하지 않은 경우 앞에서부터 다섯 번째 행까지 출력

```
df.head()
```

	id	nclass	english	science
0	1	1	80	50
1	2	1	90	60
2	3	1	95	78
3	4	2	45	58
4	5	2	10	65

```
df.head(3)
```

	id	nclass	english	science
0	1	1	80	50
1	2	1	90	60
2	3	1	95	78

데이터 프레임 명령어

▪ tail()

- 뒷부분 출력하는 함수
- ()에 값을 입력하지 않은 경우 뒤에서부터 다섯 번째 행까지 출력

```
df.tail()
```

	id	nclass	english	science
15	16	4	25	56
16	17	4	65	45
17	18	5	45	89
18	19	5	5	56
19	20	5	100	10

```
df.tail(3)
```

	id	nclass	english	science
17	18	5	45	89
18	19	5	5	56
19	20	5	100	10

데이터 프레임 명령어

▪ shape

- 행, 열 개수를 출력하는 속성
- 데이터 프레임의 크기를 알아볼 때 사용

```
df.shape
```

```
(20, 4)
```

데이터 프레임 명령어

▪ info()

- 데이터에 들어 있는 변수들의 속성 파악

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 20 entries, 0 to 19  
Data columns (total 4 columns):  
#   Column      Non-Null Count  Dtype  
---  ---  
0   id          20 non-null    int64  
1   nclass      20 non-null    int64  
2   english     20 non-null    int64  
3   science     20 non-null    int64  
dtypes: int64(4)  
memory usage: 768.0 bytes
```

데이터 프레임 명령어

▪ describe()

- '평균'처럼 변수의 값을 요약한 '요약 통계량'을 구하는 함수
- 변수의 특징을 파악하는데 도움

```
df.describe()
```

	id	nclass	english	science
count	20.00000	20.000000	20.000000	20.000000
mean	10.50000	3.050000	51.250000	57.400000
std	5.91608	1.316894	33.594917	24.839273
min	1.00000	1.000000	5.000000	10.000000
25%	5.75000	2.000000	23.750000	45.000000
50%	10.50000	3.000000	45.000000	57.000000
75%	15.25000	4.000000	82.500000	76.500000
max	20.00000	5.000000	100.000000	98.000000

함수

■ 내장 함수

- 가장 기본적인 함수로 함수 이름과 괄호를 입력하여 사용
- 파이썬에 내장되어 있기 때문에 별도의 과정 불필요

```
sum(var)  
max(var)
```

■ 패키지 함수

- 패키지 이름을 먼저 작성 후 점을 찍고 함수 이름과 괄호를 입력하여 사용
- 패키지 함수는 패키지 로드 시에만 사용 가능

```
import pandas as pd  
pd.read_csv('exam.csv')
```


메서드(method)

- 변수를 지니고 있는 함수
- 변수명 입력 후 점을 찍고 메소드 이름과 괄호를 입력하여 사용
- 변수의 자료 구조에 따라 사용할 수 있는 메서드가 다름 → `type()`을 이용하여 변수의 자료 구조 확인 가능

```
df.head()  
df.info()
```

내장 함수

|
sum()

패키지 함수

|
pd.read_csv()

메서드

|
df.head()

어트리뷰트(attribute)

- 변수가 지니고 있는 값
- 메서드와 마찬가지로 변수가 지니고 있으므로 변수명 뒤에 점을 찍고 입력
- 메서드와 달리 괄호는 입력하지 않아도 됨 ⇒ 괄호가 있으면 메서드, 없으면 어트리뷰트

메서드

`df.head()`

어트리뷰트

`df.shape`

파생변수

- 기존의 변수를 변형해 만든 변수

이름	영어 점수	수학 점수
김지훈	90	50
이유진	80	50
박동현	60	100
김민지	70	20



파생변수

이름	영어 점수	수학 점수	평균
김지훈	90	50	70
이유진	80	50	70
박동현	60	100	80
김민지	70	20	45

파생변수

데이터 프레임 생성



파생변수 생성

```
df = pd.DataFrame({'var1' : [4,3,8],  
                  'var2' : [2,6,1]})  
df
```

	var1	var2
0	4	2
1	3	6
2	8	1

```
df['var_sum'] = df['var1'] + df['var2']  
df
```

	var1	var2	var_sum
0	4	2	6
1	3	6	9
2	8	1	9

데이터 전처리

- 분석에 적합하게 데이터를 가공하는 작업
- 일부를 추출하거나, 종류별로 나누는 등 데이터를 자유롭게 가공할 수 있어야 목적에 맞게 분석 가능

파생변수

id	nclass	english	science
1	1	98	50
2	1	97	60
3	2	86	78
4	2	80	58
5	3	76	65
6	3	96	98
7	3	98	45



추출하기

nclass	english
1	98
1	97
2	86
2	80
3	76
3	96
3	98



요약하기

nclass	english
1	97.5
2	83
3	90

조건에 맞는 데이터만 추출하기

- 전체 데이터를 사용하기도 하지만 관심 있는 일부를 추출해 분석하기도 함
- 원하는 데이터만 추출하기 위해 pandas의 `df.query()` 사용

데이터프레임 확인

```
df = pd.read_csv('output.csv')  
df
```

	id	nclass	math	english	science
0	1	1	50	98	50
1	2	1	60	97	60
2	3	1	45	86	78
3	4	1	30	98	58
4	5	2	25	80	65



조건에 맞는 데이터 추출

```
df.query('nclass == 1')
```

	id	nclass	math	english	science
0	1	1	50	98	50
1	2	1	60	97	60
2	3	1	45	86	78
3	4	1	30	98	58

필요한 변수만 추출하기

- 데이터 분석 시 데이터에 들어 있는 모든 변수를 사용하기보다는 관심 있는 변수만 추출하거나 필요하지 않는 변수를 제거하여 사용하는 경우가 많음

변수 추출하기

```
df['math']
```

```
0    50  
1    60  
2    45  
3    30  
4    25
```

```
Name: math, dtype: int64
```

변수 제거하기

```
df.drop(columns = 'math')
```

	id	nclass	english	science
0	1	1	98	50
1	2	1	97	60
2	3	1	86	78
3	4	1	98	58
4	5	2	80	65

순서대로 정렬하기

- 데이터를 순서대로 정렬하면 값이 매우 크거나 매우 작아서 두드러지는 데이터 확인 가능

오름차순 정렬

```
df.sort_values('math')
```

	id	nclass	math	english	science
4	5	2	25	80	65
3	4	1	30	98	58
2	3	1	45	86	78
0	1	1	50	98	50
1	2	1	60	97	60

내림차순 정렬

```
df.sort_values('math', ascending = False)
```

	id	nclass	math	english	science
1	2	1	60	97	60
0	1	1	50	98	50
2	3	1	45	86	78
3	4	1	30	98	58
4	5	2	25	80	65

여러 정렬 기준 적용

```
df.sort_values(['nclass', 'math'])
```

	id	nclass	math	english	science
3	4	1	30	98	58
2	3	1	45	86	78
0	1	1	50	98	50
1	2	1	60	97	60
4	5	2	25	80	65

파생변수 추가하기

- 데이터에 들어 있는 변수만 이용해 분석할 수도 있지만, 변수를 조합하거나 함수를 이용해 새 변수를 만들어 분석하기도 함

데이터 프레임 확인

```
df
```

	id	nclass	math	english	science
0	1	1	50	98	50
1	2	1	60	97	60
2	3	1	45	86	78
3	4	1	30	98	58
4	5	2	25	80	65



파생변수 추가

```
df.assign(total = df['math'] + df['english'] + df['science'])
```

	id	nclass	math	english	science	total
0	1	1	50	98	50	198
1	2	1	60	97	60	217
2	3	1	45	86	78	209
3	4	1	30	98	58	186
4	5	2	25	80	65	170

집단별로 요약하기

- '집단별 평균'이나 '집단별 빈도'처럼 각 집단을 요약한 값을 구할 때는 `df.groupby()`와 `df.agg()` 사용
- 집단별로 요약할 경우 집단 간에 어떤 차이가 있는지 쉽게 파악 가능

`df.agg()`

```
df.agg(mean_math = ('math', 'mean'))
```

	math
mean_math	42.0

`df.groupby()`

```
df.groupby('nclass') \
    .agg(mean_math = ('math', 'mean'))
```

	mean_math
nclass	
1	46.25
2	25.00

데이터 합치기

■ 가로로 합치기

id	midterm
1	60
2	80
3	70

+

id	final
1	70
2	83
3	65

=

id	midterm	final
1	60	70
2	80	83
3	70	65

```
df1 = pd.DataFrame({'id' : [1,2,3],
                    'midterm' : [60,80,70]})
df1
```

	id	midterm
0	1	60
1	2	80
2	3	70

+

```
df2 = pd.DataFrame({'id' : [1,2,3],
                    'final' : [70,83,65]})
df2
```

	id	final
0	1	70
1	2	83
2	3	65

=

```
total = pd.merge(df1, df2, how = 'left', on = 'id')
total
```

	id	midterm	final
0	1	60	70
1	2	80	83
2	3	70	65

데이터 합치기

■ 세로로 합치기

id	midterm
1	60
2	80
3	70

+

id	midterm
4	50

=

id	midterm
1	60
2	80
3	70
4	50

```
df1 = pd.DataFrame({'id' : [1,2,3],
                    'midterm' : [60,80,70]})
df1
```

	id	midterm
0	1	60
1	2	80
2	3	70

+

```
df2 = pd.DataFrame({'id' : [4],
                    'midterm' : [50]})
df2
```

	id	midterm
0	4	50

=

```
total = pd.concat([df1, df2])
total
```

	id	midterm
0	1	60
1	2	80
2	3	70
0	4	50

감사합니다.

